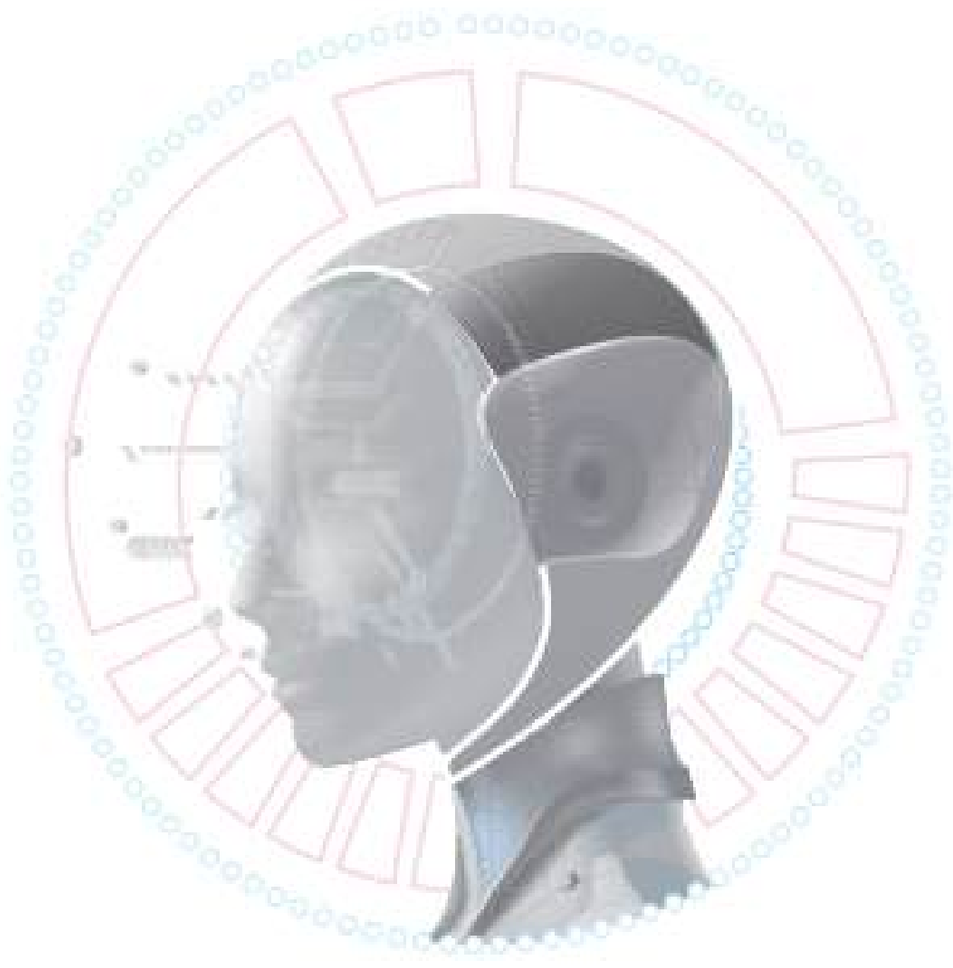
ІНФОРМАТИКА

Підручник для 8 класу
закладів загальної середньої освіти

Рекомендовано
Міністерством освіти і науки України



Київ
«Генеза»
2025

УДК 004(075.3)
І-74

Авторський колектив:
Йосиф Ривкінд, Тетяна Лисенко, Людмила Чернікова, Віктор Шакотько

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 21.02.2025 р. № 347)

Відповідно до модельної навчальної програми
«Інформатика. 7–9 класи» для закладів загальної середньої освіти
(автори: *Ривкінд Й.Я., Лисенко Т.І., Чернікова Л.А., Шакотько В.В.*)

Переглянути електронний додаток до підручника можна за посиланням *https://sites.google.com/view/informatika-8class* або QR-кодом

Додатки до підручника можна знайти за адресою
https://cutt.ly/6enwo0jJ або QR-кодом

І-74 **Інформатика :** підруч. для 8-го кл. закл. заг. серед. освіти / Й. Ривкінд та ін. — Київ : Генеза, 2025. — 304 с. : іл.
ISBN 978-___-____-__-_.

УДК 004(075.3)

ISBN 978-___-____-__-_

© Ривкінд Й.Я., Лисенко Т.І.,
Чернікова Л.А., Шакотько В.В., 2025
© «Генеза», оригінал-макет, 2025

Дорогі восьмикласниці та восьмикласники!
Шановні вчительки та вчителі!

Учні та учениці восьмого класу продовжуватимуть вивчення найсучаснішого шкільного предмета – ***інформатики***.

Інформатика – це наука про інформаційні процеси та інформаційні ресурси, про комп'ютери та їх використання. У сучасному світі інформаційно-комунікаційні технології є важливим інструментом для досягнення успіху. Знання інформатики відкривають можливості для інновацій, для створення нових продуктів і послуг, що змінюють світ. Уміння доцільно й ефективно використовувати цифрові технології в навчальній діяльності та в повсякденному житті – одна з основних вимог до учнів та учениць Нової української школи.

Підручник, який підготував наш авторський колектив, сприятиме успішному вивченню інформатики. Він допоможе навчитися логічно мислити, обґрунтовувати власні дії та свій вибір цифрових засобів для розв'язування задач з різних галузей з використанням інформаційних технологій. А також бути креативними, розвивати навички розв'язування проблем, генерувати власні ідеї та доброчесно використовувати чужі для створення нових інформаційних об'єктів.

На уроках інформатики у 8 класі ви дізнаєтеся про особливості кодування даних, одиниці вимірювання довжини двійкового коду, апаратне забезпечення персонального комп'ютера. Ознайомитеся з поняттями «Smart-технології», «Інтернет речей», особливостями доцільного використання штучного інтелекту. Навчитеся виконувати операції з архівами даних, створювати та публікувати власні вебресурси. Продовжите створювати проєкти однією з сучасних мов програмування – **Python** або **Object Pascal**, опрацьовувати дані, подані в електронних таблицях. Набуті компетентності ви застосовуватимете під час виконання навчальних проєктів на уроках інформатики та інших предметів.

Матеріал підручника поділено на вісім розділів. Розділи складаються з пунктів, а кожний пункт – з підпунктів. Корисними та цікавими для вас будуть такі рубрики:


Пригадайте


Працюємо з комп'ютером


Інтеграція (поєднання) наук


Для тих, хто працює з Android


Поміркуйте


Практична робота


Працюємо у групах / в парах


Цікаві факти з історії


Обговоріть і зробіть висновки


Найважливіше в цьому пункті


Дослідіть


Виконайте завдання


Для тих, хто хоче знати більше


Дайте відповіді на запитання


Готуємось до вивчення нового матеріалу


Оцініть свої знання та вміння

Значна увага в підручнику приділяється практичній діяльності, у процесі якої й формуються відповідні інформатичні компетентності. Учням та ученицям буде запропоновано обдумати та обговорити в парах або в невеликих групах відповіді на деякі запитання, виконати завдання зі створення чи удосконалення певного інформаційного продукту.

Детальніше зрозуміти, наскільки справдились сподівання учнів та учениць щодо опанування навчального матеріалу, яких результатів досягнуто, вони зможуть, відповівши на запитання рубрики **Оцініть свої знання та вміння**, яку розміщено в кінці кожного розділу підручника. Вона містить запитання для самооцінювання досягнення обов'язкових результатів навчання відповідно до вимог Державного стандарту базової середньої освіти та модельної програми.

У підручнику описуються дії з програмним забезпеченням на базі операційної системи **Windows 10**, якщо не зазначено іншу операційну систему.

Матеріали для виконання завдань, вправ, у тому числі файли-заготовки, розміщено на сайті **Інформатика для всіх** за адресою *http://sites.google.com/pu.org.ua/allinf* або за QR-кодом.

Зичимо плідного та захопливого навчання!

Авторський колектив

КОДУВАННЯ ДАНИХ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- кодування та декодування повідомлень;
- кодування текстових, графічних і мультимедійних даних;
- таблиці кодів символів;
- колірні моделі;
- двійкове кодування;
- одиниці вимірювання довжини двійкового коду.

1.1. КОДУВАННЯ ТА ДЕКОДУВАННЯ ПОВІДОМЛЕНЬ

У цьому пункті йтиметься про:

- сутність процесу кодування;
- правила для здійснення кодування;
- сутність процесу декодування.

КОДУВАННЯ ПОВІДОМЛЕНЬ

Пригадайте

• Які інформаційні процеси ви знаєте? • Які існують способи подання повідомлень? • Що ви знаєте про шифрування повідомлень? Хто і для чого його використовує? У чому воно полягає?

Під час опрацювання повідомлень, поданих словами, числами, графічними зображеннями, звуками тощо, часто змінюють спосіб подання з метою зберігання, передавання, опрацювання або захисту повідомлень. Наведемо кілька прикладів:

- Усні повідомлення записують на папері, замінюючи звуки людської мови літерами алфавіту. При цьому виконується збереження повідомлень.
- Під час розмови мобільним телефоном звукові сигнали перетворюються на електромагнітні хвилі. Це робить можливим передавання повідомлень на великі відстані.
- Розв'язуючи задачу на уроці математики, числівники записують цифрами, а математичні операції – спеціальними знаками. Це спрощує опрацювання числових даних.
- Для захисту повідомлень від сторонніх осіб здійснюють шифрування, замінюючи літери в тексті повідомлення іншими літерами, числами або іншими умовними позначеннями.

Процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю сигналів називають **кодуванням повідомлення.**

Код (лат. *codex* – книга) – система умовних сигналів для передавання, опрацювання та зберігання повідомлень.

У наш час кодування широко використовують для опрацювання повідомлень не лише людиною, а й цифровими пристроями. Для цього потрібно подати дані у вигляді, придатному для їх опрацювання пристроями.

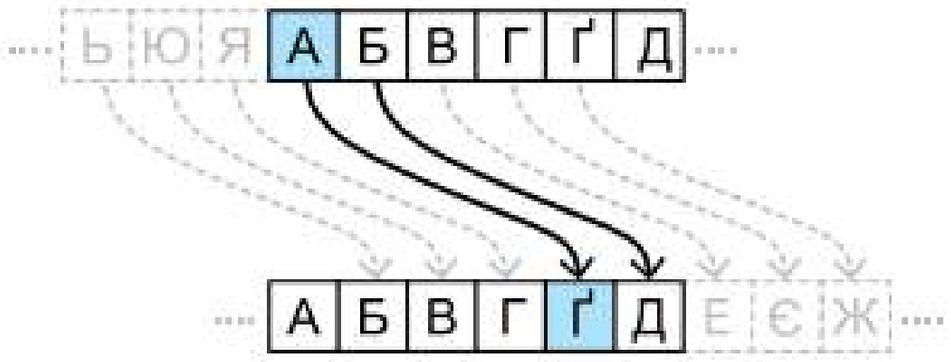


Мал. 1.1. Штрихкод товару

Наприклад, відомості про товари кодують з використанням штрихкодів (мал. 1.1). Групи цифр на штрихкоді є кодами країни та підприємства, що є виробником певного товару, та самого товару. Людина може проаналізувати цей числовий код, коли купує товар у магазині. Але ті ж самі відомості кодуються також товщиною ліній штрихкоду та відстанню між ними. Цей графічний код сприймається спеціальними сканерами та опрацьовується з використанням комп'ютера, коли ви оплачуєте товар на касі.

Усі повідомлення – текстові, числові, графічні, звукові та інші, які ми створюємо, зберігаємо та опрацьовуємо з використанням комп'ютера, кодуються з використанням двійкового коду, з яким ви ознайомитеся на наступних уроках.

Для кодування повідомлень визначають не лише набір сигналів, які будуть використані для заміни сигналів заданого повідомлення, а й правила, за якими здійснюється ця заміна.



Мал. 1.2. Правила заміни символів за шифром Цезаря зі зсувом на 4 літери праворуч

Наприклад, з історії добре відомо шифр, яким користувався для секретного листування зі своїми генералами римський імператор Гай Юлій Цезар (100 р. до н. е. – 44 р. до н. е.). Правило кодування полягає в тому, що кожна літера в тексті повідомлення замінюється іншою, що міститься в алфавіті на відстані кількох позицій від заданої літери (мал. 1.2).

Так, закодувавши повідомлення «*привіт*» з використанням шифру Цезаря зі зсувом на 4 літери праворуч, отримуємо повідомлення «*уфкєлц*».

На малюнку 1.3 наведено вигляд диска, який було виготовлено для спрощення процесу кодування повідомлень за шифром Цезаря.



Мал. 1.3. Диск для кодування повідомлень за шифром Цезаря

У наш час шифрування широко використовується для захисту повідомлень від несанкціонованого доступу під час їх передавання комп'ютерними мережами.

ДЕКОДУВАННЯ ПОВІДОМЛЕНЬ



Поміркуйте

● Чи може шифрування та розшифрування виконуватися за одними й тими самими правилами? ● Що відбудеться, якщо спробувати відкрити графічний файл у програмі **Блокнот**? Із чим це пов'язано?

На основі правила, за яким кодується повідомлення, утворюється правило для відновлення початкового повідомлення. Процес отримання початкового повідомлення із закодованого називають **декодуванням** повідомлення.

Декодування повідомлень відбувається, коли ми читаємо вголос надрукований текст, виконуємо музичний твір по нотах, розшифровуємо повідомлення, визначаємо за штрихкодом країну, у якій виготовлено товар, тощо.

Так, для декодування повідомлення, закодованого шифром Цезаря зі зсувом на 4 літери праворуч, потрібно кожну літеру закодованого повідомлення замінити іншою, що розміщена в алфавіті на 4 позиції ліворуч від заданої. Таким чином, декодувавши повідомлення *«зтдфкн зисб»*, отримуємо *«добрий день»*.

У комп'ютерних системах файли, у яких зберігаються дані різних типів, є прикладами закодованих повідомлень. Для кодування даних кожного типу (текстових, графічних, звукових тощо) використовують різні алгоритми. Відповідно до використаних алгоритмів кодування утворюються файли різних форматів (**DOCX**, **TXT**, **BMP**, **JPG**, **MP3** та інші).

Коли файл відкривають для опрацювання у відповідній програмі, відбувається процес декодування. Якщо спробувати відкрити файл у невідповідній програмі, то початкове повідомлення не буде відтворено, оскільки така програма не містить потрібного алгоритму декодування.

Кодування та декодування повідомлень є прикладами інформаційних процесів опрацювання даних.



Для тих, хто хоче знати більше

Зрозуміло, що не для кожного закодованого повідомлення передбачається його вільне декодування будь-яким користувачем.

Шифрування – це процес кодування даних, який використовується для захисту вмісту повідомлення від сторонніх осіб і лише визначені користувачі можуть виконати **дешифрування**. Для цього вони повинні знати правило декодування. Таким чином забезпечується конфіденційність даних.

Шифрування передбачає використання криптографічного ключа в поєднанні з різними математичними алгоритмами. **Криптографічний ключ** – це набір даних, як правило, математичних величин, узгоджених між відправником і отримувачем повідомлення.

Шифрування може виконуватися з використанням симетричних або асиметричних алгоритмів.

Криптографія (грец. *κρυπτός* – прихований, *γράφω* – писати) – наука про математичні методи забезпечення захисту інформації.

Симетричний алгоритм шифрування використовує один криптографічний ключ і симетричні правила для шифрування та дешифрування даних. Відомий вам шифр Цезаря є симетричним алгоритмом шифрування. У ньому число, що визначає величину зсуву, є криптографічним ключем. Якщо для шифрування вибрано, наприклад, зсув на 4 літери праворуч, то для дешифрування – зсув на 4 літери ліворуч. Симетричність полягає в тому, що для шифрування та дешифрування величина зсуву однакова, а напрямки протилежні.

Асиметричний алгоритм шифрування, на відміну від симетричного, передбачає використання різних ключів для шифрування та дешифрування даних. Один з цих ключів є **відкритим**, а інший – **закритим**. Використовуючи відкритий ключ, будь-хто може зашифрувати повідомлення, але дешифрувати його може лише той, хто має закритий ключ. У цьому полягає асиметричність алгоритму.

Відкритий і закритий ключі математично пов'язані один з одним. Але, знаючи відкритий ключ, для пошуку закритого ключа, навіть з використанням найпотужніших комп'ютерів, потрібно часу більше, ніж тривалість людського життя.

Одним з популярних алгоритмів асиметричного шифрування є алгоритм **RSA**, який запропонували в 1977 році вчені Массачусетського університету Р. Рівест, А. Шамір і Л. Адлеман. Алгоритм отримав назву від перших літер їх прізвищ (**R**ivest, **S**hamir, **A**dleman). Відповідно до цього алгоритму обирають два різних простих числа, у записі яких понад сто цифр, та знаходять їх добуток. Він є складовою відкритого ключа. Закритий ключ обчислюють на основі одного з обраних простих чисел.

Алгоритм RSA використовують для створення електронних підписів, шифрування каналів мобільного зв'язку, в системах електронної комерції тощо.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Je2cOcCZ* або QR-кодом.



Найважливіше в цьому пункті

Кодування повідомлення – це процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю сигналів.

Кодування повідомлень виконується з метою їх зберігання, передавання, подальшого опрацювання, захисту. Для кодування повідомлень визначають набір сигналів, які будуть використані для заміни сигналів заданого повідомлення, та правила, за якими здійснюється ця заміна.

Декодування повідомлення – це процес отримання початкового повідомлення із закодованого.

Кодування використовується для опрацювання повідомлень не лише людиною, а й цифровими пристроями. Для цього потрібно подати дані у вигляді, придатному для опрацювання пристроями. У комп'ютерних системах файли, у яких зберігаються дані різних типів, є прикладами закодованих повідомлень. Для кодування даних кожного типу використовують різні алгоритми. Відповідно до використаних алгоритмів кодування утворюються файли різних форматів.



Дайте відповіді на запитання

1. У чому полягає процес кодування повідомлень?
2. З якою метою виконують кодування повідомлень?
3. Що потрібно попередньо визначити для кодування повідомлення?
4. У чому полягає процес декодування повідомлень?
5. Що означає формат файлу, який обирають під час збереження даних у файлі?

Обговоріть і зробіть висновки

1. У яких життєвих ситуаціях вам доводилося змінювати спосіб подання повідомлень? Наведіть приклади кодування, які траплялися вам у повсякденному житті.

2. З якою метою звуки музики кодують нотами; повідомлення про особливості руху по дорозі кодують дорожніми знаками?

3. Які повідомлення можна закодувати з використанням кольорів; графічних позначень? Наведіть приклади.

4. За якими правилами у грі «Морський бій» кодують положення кораблів? Як це кодування використовують під час гри?

Виконайте завдання

1. Наведіть приклади правил кодування повідомлень, які використовують у музиці, фізиці, хімії, географії та в інших науках.

2. Закодуйте з використанням шифру Цезаря зі зсувом на 4 літери праворуч повідомлення:

а) *інформатика*;

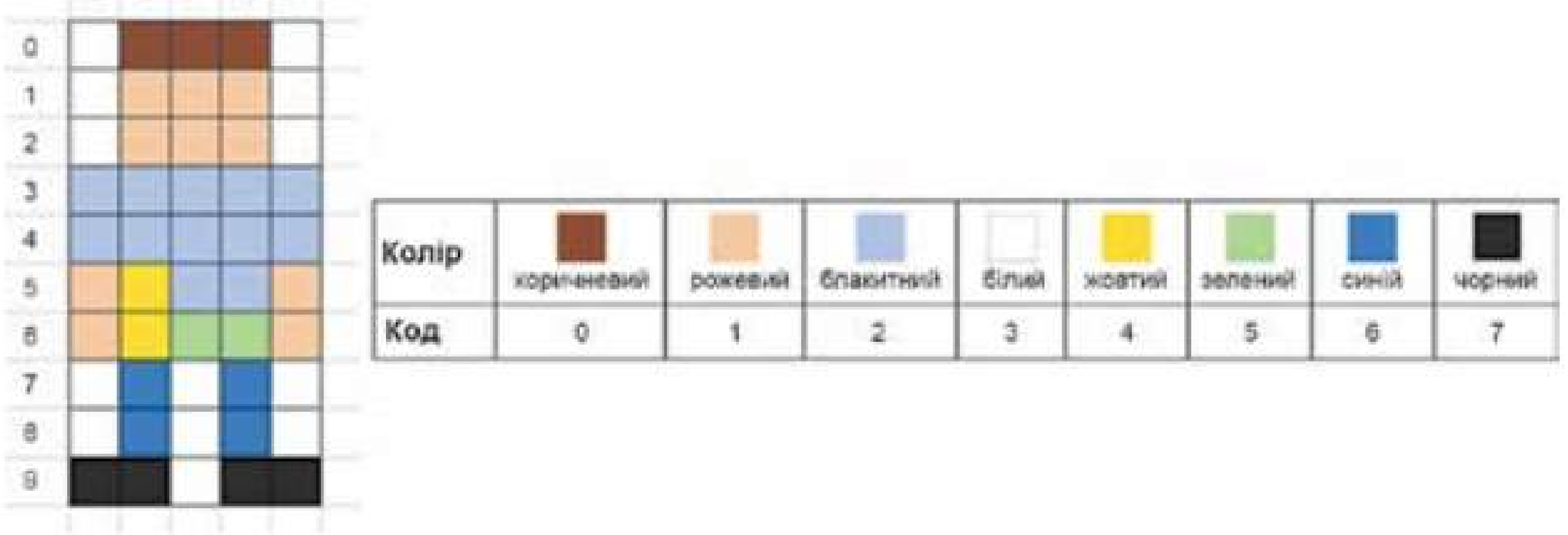
б) *моя школа*.

3. Повідомлення закодовано шифром Цезаря зі зсувом на 3 літери праворуч. Декодуйте повідомлення:

а) *дзузфзра*;

б) *егург тсесжг*.

4. Зображення складається з квадратів, які розміщені в 5 стовпцях і 10 рядках. Кожен квадрат пофарбовано в один з 8 кольорів (мал. 1.4).



Колір	коричневий	рожевий	блакитний	білий	жовтий	зелений	синій	чорний
Код	0	1	2	3	4	5	6	7

Мал. 1.4. Зображення та таблиця кодів кольорів

Для кодування зображення визначається код кожного квадрата – трицифрове число. Перша цифра означає номер стовпця, у якому розташований квадрат на зображенні (від 0 до 4), друга цифра – номер рядка (від 0 до 9), третя цифра – код кольору (від 0 до 7). Наприклад, лівий верхній квадрат на зображенні має код *003*, а правий нижній – код *497*. Закодуйте рядок з номером 6 даного зображення.

5. Один з найдавніших шифрів був розроблений Полібієм (III ст. до н. е.) – грецьким істориком, полководцем, державним діячем – та отримав назву «квадрат Полібія». Кожна літера алфавіту цього шифру міститься в таблиці. Під час кодування повідомлення кожна літера замінюється парою цифр – номером стовпця та номером рядка таблиці, на

перетині яких вона розміщена. Для кодування повідомлень українською мовою можна використати таблицю з 6 стовпців і 6 рядків (мал. 1.5).

Декодуйте та закодуйте з використанням наведеної таблиці повідомлення:

а) **24 14 43 13 21 13 62;**
б) **33 14 61 64 31 11 63 63 36;**
в) *інформатика*;
г) *повідомлення*.

	1	2	3	4	5	6
1	А	Б	В	Г	Ґ	Д
2	Е	Є	Ж	З	И	Й
3	І	Ї	К	Л	М	Н
4	О	П	Р	С	Т	У
5	Ф	Х	Ц	Ч	Ш	Щ
6	Ь	Ю	Я			

Мал. 1.5. Квадрат Полібія

6. Знайдіть відомості та підготуйте повідомлення про способи кодування:

а) азбука Морзе;
б) шрифт Брайля;
в) QR-коди;
г) коди у вишивці.

7. Об'єднайтесь у групи по 4 особи та реалізуйте проєкт «Шифрувальник»:

- кожна пара учасників / учасниць групи розробляє власні правила кодування символів тексту та кодує з їх використанням фразу *сьогодні гарний день для футболу* та ще одне власне повідомлення;
- закодовані повідомлення передають другій парі учасників / учасниць;
- на основі початкового та закодованого вигляду відомої фрази потрібно розгадати алгоритм кодування та декодувати друге повідомлення.

Проаналізуйте час, потрібний для розгадування алгоритму кодування, та зробіть висновок про надійність цього алгоритму для захисту повідомлень.

Готуємось до вивчення нового матеріалу

1. Наведіть приклади алгоритмів кодування текстових повідомлень.
2. Які ви знаєте види графіки? Схарактеризуйте їх.

1.2. КОДУВАННЯ ТЕКСТОВИХ, ГРАФІЧНИХ І МУЛЬТИМЕДІЙНИХ ДАНИХ

У цьому пункті йтиметься про:

- кодування текстових даних;
- таблиці кодів символів;
- кодування графічних даних;
- колірні моделі;
- кодування мультимедійних даних.

КОДУВАННЯ ТЕКСТОВИХ ДАНИХ

Поміркуйте

• Чому в текстовому процесорі **Word** список, елементи якого позначено літерами, також уважається нумерованим? • Чи можуть бути однаковими коди великих і малих літер під час кодування текстових повідомлень? На що це може впливати? • Чи можуть бути однаковими коди літер з однаковим написанням англійської та української абетки, наприклад літер «а», «о», «е» тощо? На що це може впливати?

Усі дані, що призначені для опрацювання автоматичними пристроями, повинні бути певним чином закодовані. Для опрацювання, передавання та зберігання текстових повідомлень з використанням комп'ютера символи кодують числами.

Для кодування текстів використовують **таблиці кодів символів**, у яких кожному символу, що може бути використаний у текстовому повідомленні, поставлено у відповідність деяке число. У 1963 році у США була розроблена одна з перших таблиць кодів символів, призначена для передавання повідомлень телетайпом. Пізніше вона стала стандартом для використання в комп'ютерній техніці та отримала назву **таблиця кодів символів ASCII** (англ. *American Standard Code for Information Interchange* – американський стандартний код для обміну інформацією).

Телетайп (лат. *tele* – віддалений, англ. *type* – друкування) – електромеханічна друкувальна машина, яку використовують для передавання дротами текстових повідомлень.

У таблиці **ASCII** літерам англійського алфавіту, цифрам, розділовим знакам, символам редагування та форматування тексту поставлено у відповідність числа від 0 до 127 (табл. 1.1).

Таблиця 1.1

Приклади груп кодів символів таблиці ASCII

Діапазон кодів	*Група символів*	*Приклад коду*	*Символ*
Від 48 до 57	Цифри	48	*0*
		57	*9*
Від 65 до 90	Великі літери англійського алфавіту	65	*A*
		66	*B*
		90	*Z*
Від 97 до 122	Малі літери англійського алфавіту	97	*a*
		98	*b*
		122	*z*

Інші коди відповідають управляючим, недрукованим, неалфавітним символам, знакам математичних операцій, дужкам, розділовим знакам тощо.

Текстові символи, що кодуються в таблиці **ASCII** числами від 32 до 127, наведено в ***Додатку 1*** підручника.

Таблиця кодів символів **ASCII** містить коди літер лише англійського алфавіту. Для кодування літер інших алфавітів були розроблені інші таблиці кодів символів. Наприклад, таблиці **KOI8-U** (KOI – код обміну інформацією, англ. *Ukrainian* – українська) та **Windows-1251** уключають без змін усі коди таблиці **ASCII**, а також містять коди літер української абетки. Кодами цих літер є натуральні числа в діапазоні від 128 до 255.

Таблиця кодів символів **Windows-1251** була розроблена для кодування в операційній системі **Windows** літер абеток групи слов'янських мов, до яких належать українська, болгарська, сербська, боснійська та інші. У ній, наприклад, літера «*а*» української абетки має код 224, літера «*і*» – код 179, літера «ґ» – код 180 та інші.

Цілих чисел від 0 до 255 вистачає, щоб закодувати символи деяких двох абеток, наприклад англійської та української, та деякі інші символи. Але для кодування символів інших систем письма (грецької та арабської абетки, ієрогліфів, клинопису тощо) потрібно значно більше значень кодів. Для кодування символів усіх алфавітів світу розроблено таблицю кодів символів **Юнікод** (англ. ***Universal Code, Unicode*** – універсальний код).

Таблиця **Юнікод** складається з 17 наборів символів по 65 536 значень кодів у кожному, тобто містить 1 114 112 символів та їх коди. Як і в інших таблицях кодів символів, у **Юнікоді** перші 128 кодів відповідають таблиці **ASCII**. Окремий розділ у таблиці **Юнікод** містить коди літер української абетки. Наприклад, літера «*а*» української абетки має код 1072, літера «*і*» – код 1110, літера «ґ» – код 1169 та інші.

Використання таблиці кодів символів **Юнікод** робить можливим уставлення в текстові повідомлення не лише літер, а й смайлів, музичних символів, символів валюти, знаків зодіаку, зображення кісток доміно, гральних карт і багатьох інших неалфавітних символів (мал. 1.6).



Мал. 1.6. Текстове повідомлення зі вставленими неалфавітними символами

У текстовому процесорі **Word** для вставлення символів у текст можна використати їх коди з таблиці **Юнікод**. Для цього потрібно натиснути та утримувати клавішу **Alt** і набрати код символа на додатковій цифровій клавіатурі. Приклади символів наведено в таблиці 1.2.

Таблиця 1.2

Приклади символів таблиці Юнікод

Символ	*Назва символа*	*Код символа*
₴	Грошова одиниця гривня	8372
☃	Сніговик	9731
♘	Шахова фігура «кінь»	9822
♪	Нота	9834
❥	Серце у вигляді квітки	10 087
웃	Чоловік	65 664
웃	Жінка	65 665

Знайти символ і відповідний йому код можна на сайті **JavaScript Довідка** (*яваскрипт.укр/таблиця-unicode*).

У таблиці 1.3 наведено порівняння структури різних таблиць кодів символів.

Таблиця 1.3

Порівняння структури таблиць кодів символів

Таблиця кодів символів	*Кількість кодів символів*	*Символи з кодами від 0 до 127*	*Символи з кодами від 128 до 255*	*Символи з кодами більше за 255*
ASCII	128	Літери англійського алфавіту, цифри, розділові знаки, спеціальні символи	Немає	Немає
KOI8-U	256	Як у таблиці **ASCII**	Символи української абетки та деякі інші символи	Немає
Windows-1251	256	Як у таблиці **ASCII**	Символи української абетки та деякі інші символи	Немає
Юнікод	1 114 112	Як у таблиці **ASCII**	Символи алфавітів різних мов світу та інші символи	

Для тих, хто хоче знати більше

У табличному процесорі **Excel** є функції, призначені для визначення коду заданого символа, а також за заданим кодом – відповідного символа в кодових таблицях **Windows-1251** та **Юнікод**:

- *CODE* – визначення коду заданого символа в кодовій таблиці **Windows-1251**;
- *UNICODE* – визначення коду заданого символа в кодовій таблиці **Юнікод**;
- *CHAR* – визначення символа за його числовим кодом у таблиці **Windows-1251**;
- *UNICHAR* – визначення символа за його числовим кодом у таблиці **Юнікод**.

На малюнку 1.7 можна побачити приклади використання наведених функцій у текстовому процесорі **Excel** і результат їх виконання.

	A	B	C
1	Код літери "Ф"		
2	Кодова таблиця	Формула	Результат
3	Windows-1251	=CODE("Ф")	212
4	Юнікод	=UNICODE("Ф")	1060
5			
6	Символи, що відповідають кодам		
7	Кодова таблиця	Формула	Результат
8	Windows-1251	=CHAR(169)	©
9	Юнікод	=UNICHAR(9999)	▬
10			

Мал. 1.7. Приклади використання функцій у **Excel**

КОДУВАННЯ ГРАФІЧНИХ ДАНИХ



Пригадайте

• Які види графіки ви знаєте? • Що є найменшим елементом растрового графічного зображення? Які властивості він має? • Що є найменшими елементами векторного графічного зображення? Які властивості вони мають?

Кодування графічних даних виконується по-різному, залежно від виду графіки.

Як ви вже знаєте, растрове зображення складається з окремих пікселів. Для кодування растрового зображення потрібно для кожного пікселя закодувати його колір.

Векторне зображення складається з окремих геометричних фігур (графічних примітивів): відрізків, прямокутників, овалів, кривих тощо. Кодування векторного зображення полягає в описі значень властивостей графічних примітивів, з яких складається це зображення. Наприклад, для побудови круга достатньо закодувати місце розміщення його центра, радіус круга, товщину, стиль і колір лінії кола, колір і стиль заливки.

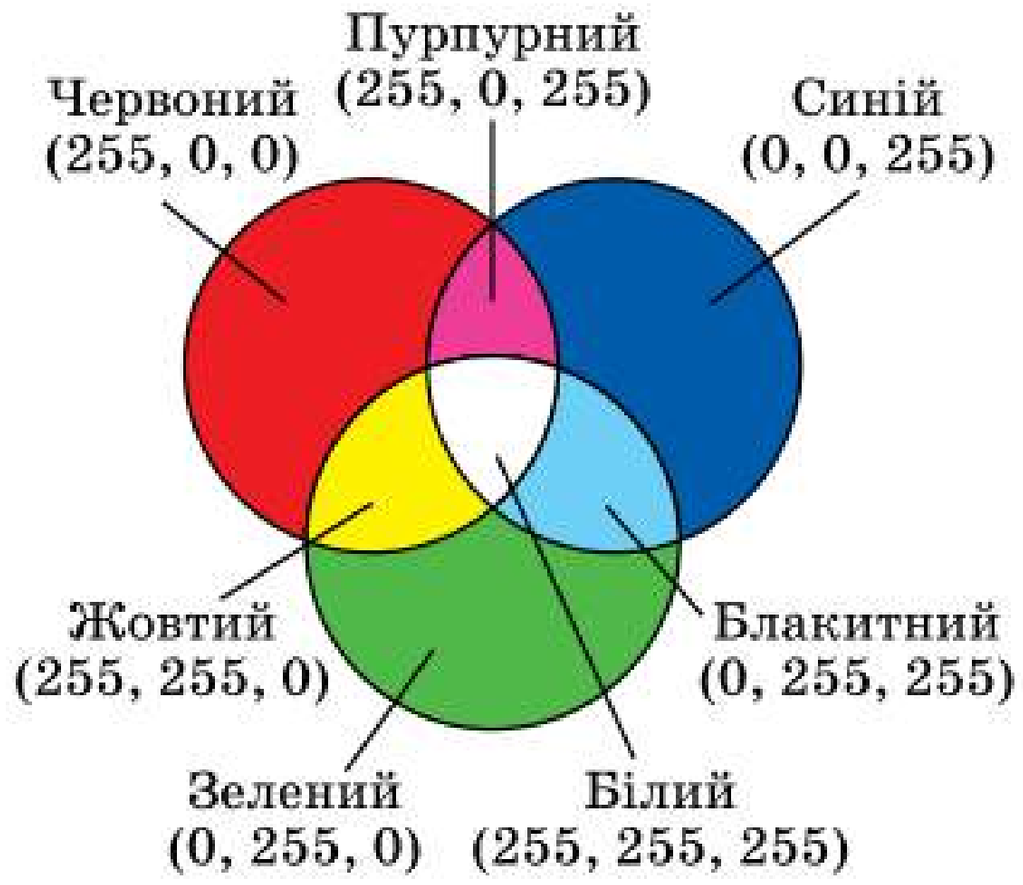
Спектр (лат. *spectrum* – привид) – багатоколірна смуга, яка утворюється під час проходження білого світла через призму або якесь інше середовище, що заломлює світло.

Існують різні правила кодування кольору – **колірні моделі**.

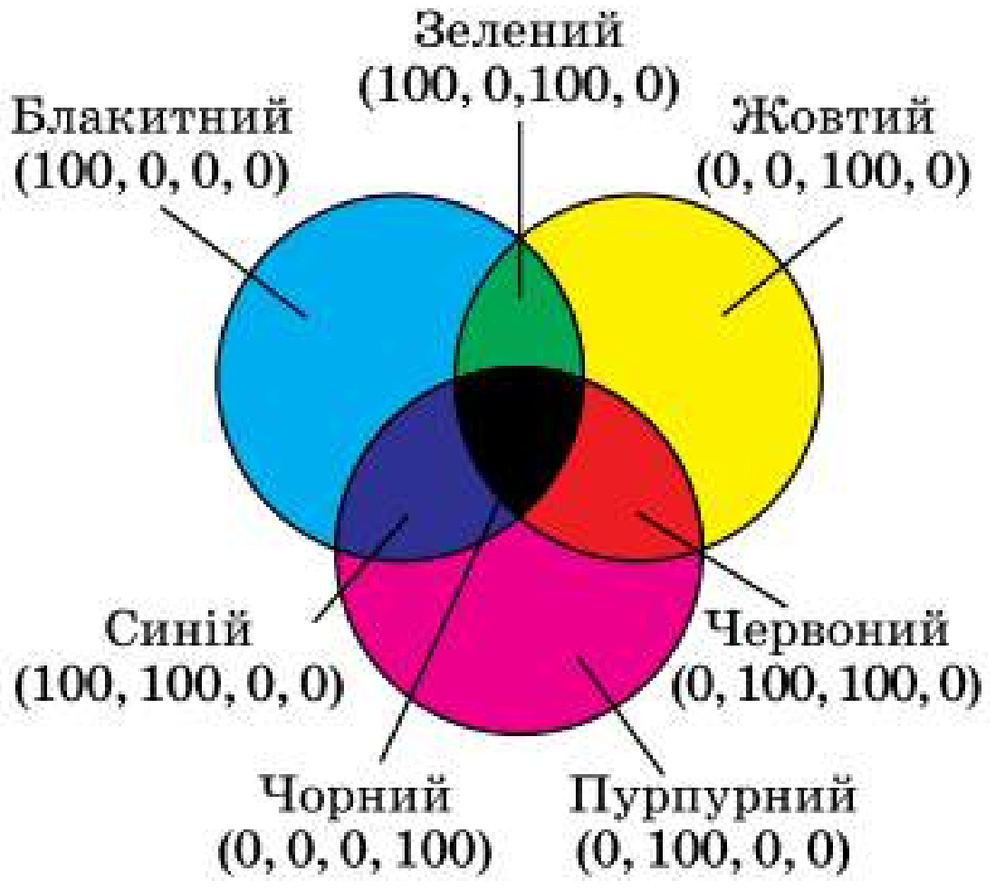
Колірна модель – це спосіб кодування різних кольорів спектра впорядкованим набором числових значень певних **базових компонентів**.

Найчастіше в комп'ютерній графіці використовують колірну модель **RGB**, якщо зображення буде відтворюватись на екрані, модель **CMYK** – для друку зображення та модель **HSV (HSB)** – для коригування кольорів у комп'ютерній графіці.

У колірній моделі **RGB** (мал. 1.8) базовими компонентами є три кольори спектра – *червоний* (англ. ***Red***), *зелений* (англ. ***Green***) і *синій* (англ. ***Blue***). Значення інтенсивності кожного компонента задається цілим числом від 0 до 255. Використовуючи модель **RGB**, можна закодувати $256^3 \approx 16{,}7$ млн кольорів.



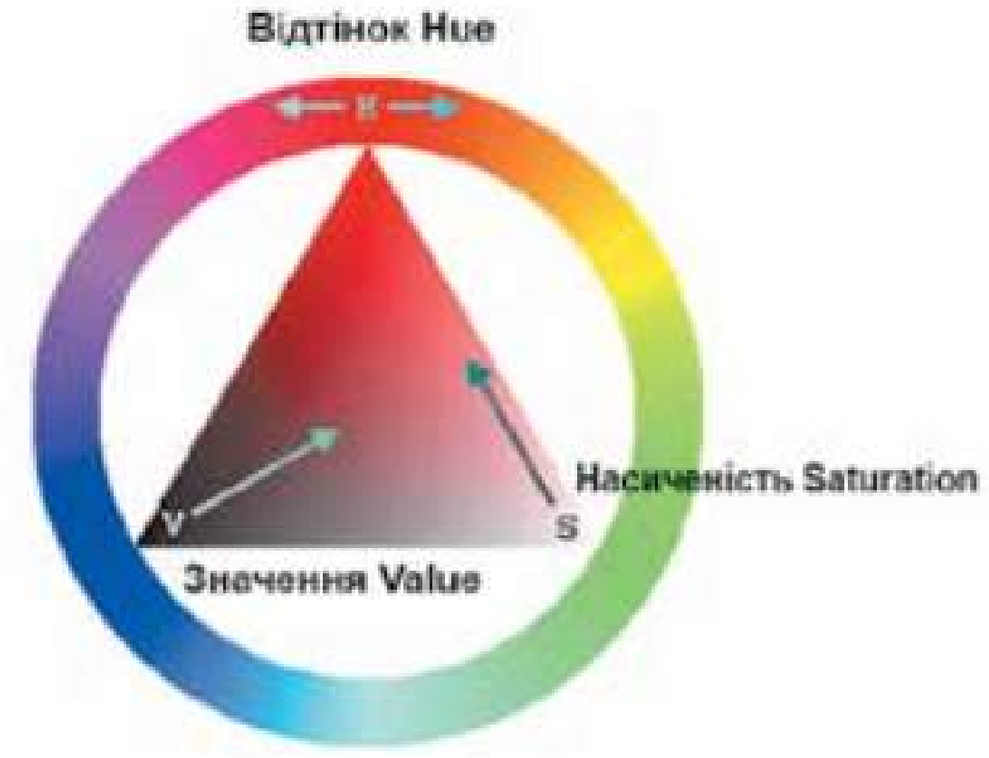
Мал. 1.8. Колірна модель **RGB**



Мал. 1.9. Колірна модель **CMYK**

У колірній моделі **CMYK** використовують чотири базові компоненти: *блакитний* (англ. ***Cyan***), *пурпурний* (англ. ***Magenta***), *жовтий* (англ. ***Yellow***), *чорний* (англ. *blac**K***) кольори (мал. 1.9). Частка кожного з базових компонентів задається у відсотках (цілим числом від 0 до 100). Теоретично модель **CMYK** дає можливість закодувати $101^4 \approx 100$ млн кольорів, але на практиці кількість кольорів залежить від якості чорнил та обладнання, на якому відбувається друк.

Колірна модель **HSV (HSB)** має три базові компоненти: *відтінок* (англ. ***Hue***), *насиченість* (англ. ***Saturation***) і *значення (яскравість)* (англ. ***Value*** або ***Brightness***) (мал. 1.10). *Відтінок* (положення кольору на кольоровому кільці) задається у градусах у діапазоні від 0° до 360°, *насиченість* (чистота кольору, відсутність домішок білого та чорного кольорів) і *яскравість* – у відсотках від 0 до 100. Значення базових компонентів можуть подаватися дробовими числами, тому в цій моделі теоретично можна закодувати необмежену кількість кольорів. На практиці ж існують обмеження, що пов'язані з можливостями пристроїв, на яких відтворюються кольори.



Мал. 1.10. Колірна модель **HSV**

Подібною до **HSV** є модель **HSL**, у якій третій базовий компонент – *світлість* (англ. ***Lightness***) виражає близькість до білого кольору.

У таблиці 1.4 наведено приклади кодування кольорів у колірних моделях **RGB, CMYK** та **HSV**.

Таблиця 1.4

Приклади кодування кольорів у колірних моделях RGB, CMYK та HSV

Колір	Модель RGB			Модель CMYK				Модель HSV		
	R	G	B	C	M	Y	K	H	S	V
Білий	255	255	255	0	0	0	0	0	0	100
Сірий	122	122	122	0	0	0	50	0	0	48
Чорний	0	0	0	0	0	0	100	0	0	0
Смарагдовий	80	200	120	60	0	40	22	140	60	78
Рожевий	255	128	192	0	25	20	0	330	50	100
Жовтий	255	255	0	0	0	100	0	60	100	100

Побачити код вибраного вами кольору можна під час роботи в різних програмах. Наприклад, у графічному редакторі **Krita** під час вибору кольору об'єктів код кольору подається в моделі **RGB** або **HSV** (мал. 1.11, а), а в текстовому процесорі **Word** – у моделях **RGB** та **HSL** (мал. 1.11, б).



а) б)

Мал. 1.11. Визначення коду кольору в графічному редакторі **Krita** (а) та текстовому процесорі **Word** (б)

КОДУВАННЯ МУЛЬТИМЕДІЙНИХ ДАНИХ

Пригадайте

- Які дані називають *мультимедійними*?
- Як утворюються рухомі зображення?
- Що ви знаєте про звук і його властивості?

Закодувати для опрацювання цифровими пристроями можна не лише текстові та графічні, а й мультимедійні дані.

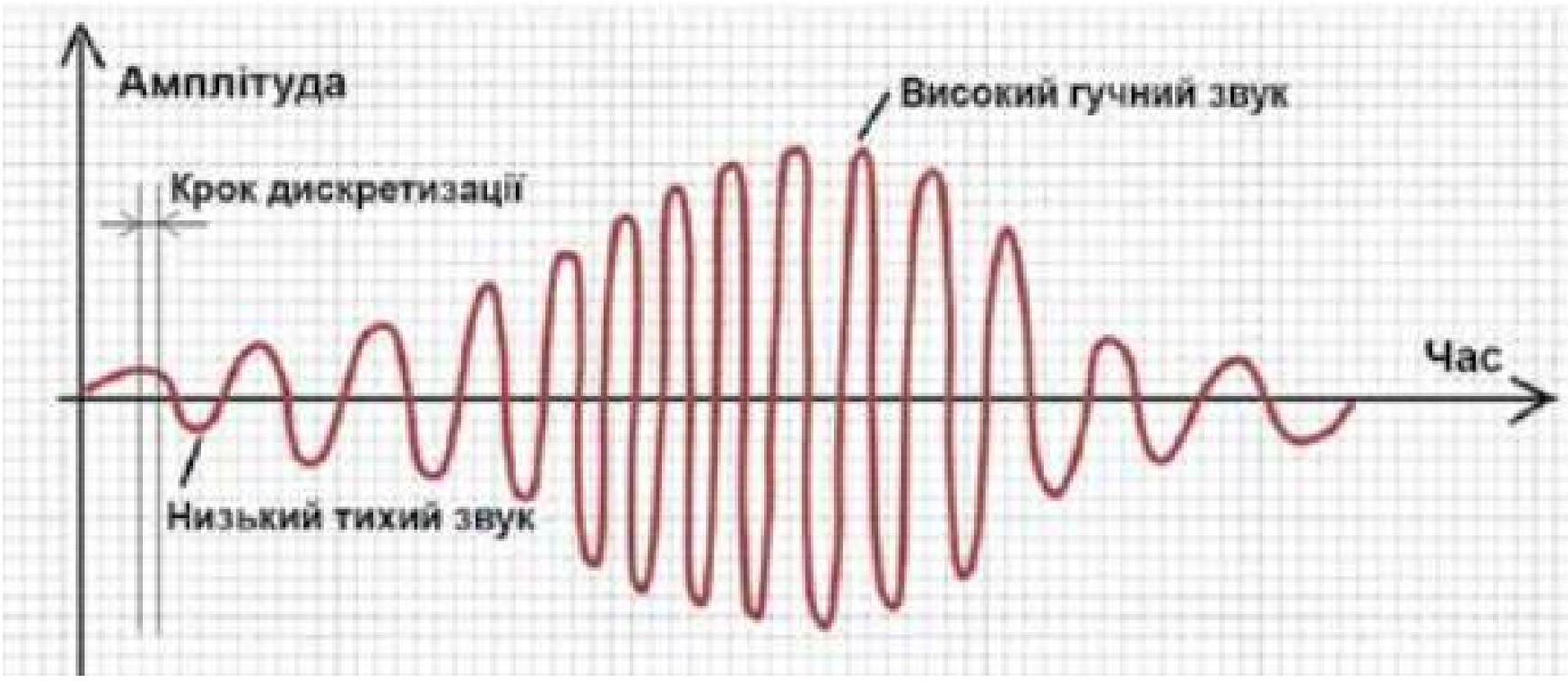
Рухомі зображення – відео та анімація – фактично є набором окремих графічних кадрів, які демонструються почергово з певною швидкістю. Тому для кодування відео та анімації потрібно закодувати зображення кожного кадру. Для отримання плавних рухів у відео швидкість відтворення зазвичай складає 24 або більше кадрів за секунду. Усі закодовані кадри рухомого зображення зберігаються в одному файлі.

Ви вже знаєте, що звук – це хвилі, що розповсюджуються в деякому пружному середовищі (газі, рідині, твердому тілі). Основними властивостями звуку, від яких залежить його сприймання людиною, є:

- *висота тону*, що визначається частотою коливань звукової хвилі. Чим більша частота хвилі, тим звук вище, а чим менша частота, тим звук нижче;
- *гучність* – визначається амплітудою хвилі. Чим більша амплітуда, тим гучніший звук, а чим менша амплітуда, тим звук тихіший.

Кодування звукових повідомлень полягає в кодуванні числами значень цих властивостей.

Значення властивостей звукового повідомлення змінюються під час його передавання, тому для кодування звуку їх вимірюють і кодують багаторазово (кілька тисяч разів) упродовж кожної секунди, поки повідомлення лунає. Кількість вимірювань значень властивостей звукового сигналу за одну секунду називають *частотою дискретизації* звукових даних, а інтервал часу між послідовними вимірюваннями – *кроком дискретизації* (мал. 1.12).



Мал. 1.12. Графік зміни амплітуди та частоти звукової хвилі з часом

Для кодування звуку людського голосу достатньо проводити вимірювання і кодування 8 тис. разів за секунду, а для якісного запису музичних творів частота дискретизації повинна бути не нижче від 44 тис. вимірювань за секунду.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/xe2cOFDv* або QR-кодом.



Найважливіше в цьому пункті

Для опрацювання, передавання та зберігання текстових повідомлень з використанням комп'ютера символи повідомлення кодують числами. Для цього можуть використовуватися різні **таблиці кодів символів**, наприклад **ASCII**, **KOI8-U**, **Windows-1251**, **Юнікод**.

Таблиця кодів символів **ASCII** містить коди 128 символів, **KOI8-U**, **Windows-1251** – по 256 символів, **Юнікод** – 1 114 112 символів. У всіх таблицях кодів символів незмінними залишаються перші 128 кодів, що відповідають таблиці **ASCII**.

Для кодування растрового зображення потрібно для кожного пікселя закодувати його колір. Кодування векторного зображення полягає в кодуванні значень властивостей графічних примітивів, з яких складається це зображення.

Колірна модель – це спосіб кодування різних кольорів спектра впорядкованим набором числових значень певних **базових компонентів**.

Найчастіше в комп'ютерній графіці використовують колірну модель **RGB**, якщо зображення буде відтворюватися на екрані, модель **CMYK** – для друку зображення та модель **HSV** (**HSB**) – для коригування кольорів у комп'ютерній графіці.

Для кодування відео та анімації потрібно закодувати зображення кожного кадру, з яких складається рухоме зображення. Кодування звукових повідомлень полягає в кодуванні числами значень властивостей *висота тону*, *гучність* та інших.



Дайте відповіді на запитання

1. Як кодують символи текстових повідомлень під час опрацювання їх з використанням комп'ютера?

2. Які символи можуть бути закодовані з використанням таблиць кодів символів **ASCII, KOI-8U, Windows-1251, Юнікод**?

3. У чому полягає кодування растрового, векторного та рухомого графічного зображення?

4. Які властивості звуку використовують для його кодування?



Обговоріть і зробіть висновки

1. Що спільного та чим відрізняються таблиці кодів символів **ASCII, KOI-8U, Windows-1251, Юнікод?**

2. Під час упорядкування слів за алфавітом у комп'ютерних програмах порівняння символів відбувається за їх кодами в кодових таблицях. Чи важливо, записано слова великими чи маленькими літерами? Відповідь поясніть.

3. У растровому та векторному графічних редакторах створено однакові зображення світлофора. Довжина коду якого з цих зображень більша? Поясніть чому.

4. Як залежить довжина коду звукового повідомлення від значення частоти дискретизації?



Виконайте завдання

1. Повідомлення закодовано з використанням таблиці кодів символів **ASCII**. Декодуйте його (скористайтеся ***Додатком 1*** до підручника):

а) *83 117 110*;

б) *104 111 109 101*.

2. Закодуйте повідомлення кодами з таблиці кодів символів **Юнікод**, використовуючи текстовий процесор:

а) *Інформатика*;

б) ваше прізвище та ім'я.



3. Окремі слова в повідомленні закодовано з використанням таблиці кодів символів **Юнікод**. Знайдіть символи за їх кодами в таблиці. Запишіть отриманий текст. Для пошуку символів можна використовувати сайт **SYMBL** (*symbl.cc*):

Сьогодні зранку 9925. Синоптики обіцяють 9928. Збираючись у 127979, не забудьте взяти із собою 9730.

4. Визначте, які символи будуть вставлені до текстового документа шляхом уведення їх коду за натиснутої клавіші **Alt** для кодів *1234*, *3754*, *8986*, *9708*, *9415*, *10001*, *10239*.

5. Відкрийте файл **Розділ 1\Пункт 1.2\завдання 1.2.5.xlsx**. У клітинках електронної таблиці записано коди кольорів колірної моделі **RGB**. Зафарбуйте клітинки в кольори відповідно до наведених кодів. Збережіть електронну таблицю у вашій папці у файлі з тим самим іменем.

6. Створіть на аркуші **Аркуш 1** електронної таблиці зображення українського орнаменту, розфарбувавши клітинки в різні кольори. Закодуйте зображення, розмістивши у відповідних клітинках на **Аркуші 2** коди кольорів зафарбованих клітинок отриманого зображення відповідно до колірної моделі **RGB**. Збережіть електронну таблицю у вашій папці у файлі з іменем **завдання 1.2.6.**



7. Відкрийте у вікні браузера сторінку **Chrome Music Lab** (*musiclab.chromeexperiments.com*). Виберіть експеримент **Oscillators** (англ. *oscillators* – генератори коливань). Вибирайте довільну точку в робочому полі сторінки та слідкуйте за висотою звуку та відповідним числовим кодом у рядку *oscillator.frequency.value* (англ. *frequency* – частота, *value* – значення). Зробіть висновок, як частота звукової хвилі пов'язана з висотою звуку, який ми чуємо.

Готуємось до вивчення нового матеріалу

1. Що означають префікси *кіло-*, *мега-*, *гіга-* та інші в записі одиниць вимірювання маси та довжини?
2. Що вам відомо про запис чисел римськими цифрами? Які числа записано: III, VII, IX, XXXIV?
3. Що означає цифра *1* у записі чисел *1000* та *0,01*?

1.3. ДВІЙКОВЕ КОДУВАННЯ

У цьому пункті йтиметься про:

- сутність двійкового кодування;
- одиниці вимірювання довжини двійкового коду – біт і байт;
- двійкове кодування текстових, числових і мультимедійних даних;
- двійкову систему числення;
- кратні одиниці вимірювання довжини двійкового коду;
- перехід від одних одиниць вимірювання довжини двійкового коду до інших.

ДВІЙКОВЕ КОДУВАННЯ. БІТ І БАЙТ. ДОВЖИНА ДВІЙКОВОГО КОДУ ПОВІДОМЛЕНЬ

Поміркуйте

● Для чого може бути потрібним вимірювання обсягів даних? Як можна виміряти ці обсяги? ● На здійснення яких інформаційних процесів можуть впливати обсяги даних? Ефективна робота яких пристроїв може залежати від обсягів даних?

Як ви вже знаєте, для кодування даних використовують різні набори сигналів. Для кодування звуків мови використовують літери. В українській абетці 33 літери, в англійській – 26. Для кодування чисел можна використовувати 10 цифр, а для кодування команд, що регулюють порядок руху на пішохідних переходах, – 3 кольори світлофора.

Найменшу кількість сигналів для кодування текстових повідомлень має азбука Морзе – лише 2 сигнали: *крапка* та *тире*.

З 1844 року азбуку, запропоновану американським художником Семюелем Морзе (1791–1872) та названу на його честь, використовували для передавання повідомлень телеграфом (мал. 1.13).



Мал. 1.13. Семюель Морзе та телеграф

В азбуці Морзе літери та цифри кодуються послідовностями крапок і тире. Приклади кодування літер української абетки наведено в таблиці 1.5.

Таблиця 1.5

Приклади кодування літер у азбуці Морзе

Літера	*Значення коду в азбуці Морзе*
а	·–
о	–––
п	·––·
р	·–·
т	–

Наприклад, слово *парта* кодується з використанням азбуки Морзе такою послідовністю сигналів «·––· ·– ·–· – ·–».

Кодування повідомлень з використанням сигналів лише двох видів називають **двійковим кодуванням.** Повідомлення, отримане в результаті двійкового кодування, називають **двійковим кодом.**

Двійкове кодування використовується й у сучасних комп'ютерах. У них усі повідомлення кодуються послідовностями електричних сигналів двох видів.

Сигнал одного виду умовно позначається цифрою **0**, а іншого виду – **1**.

Цифра 0 або 1 у двійковому коді повідомлення називається **біт** (англ. ***b**inary digit* – двійкова цифра).

Слово **біт** у записі поряд із числом позначається маленькою літерою **б**. Наприклад, 20 біт буде записано 20 б.

З таблиці 1.5 видно, що в азбуці Морзе для кодування літери «т» використовується один сигнал – тире, для кодування літери «а» – два, а для літери «п» – чотири сигнали. Під час передавання повідомлень телеграфом одну літеру від іншої відокремлюють проміжками в подачі сигналів. Для опрацювання даних з використанням комп'ютера це не зручно. Простіше відокремити одну літеру від іншої, коли для їх кодування буде використано однакову кількість сигналів.

Якщо для кодування повідомлення використовується лише один двійковий сигнал (біт) 0 або 1, то можна закодувати, наприклад:

- висновок про правильність твердження: *хибне – 0* або *істинне – 1*;
- стать людини: *жіноча – 0* або *чоловіча – 1*;
- стан вимикача: *вимкнено – 0* або *увімкнено – 1*

тощо.

З двох бітів можна скласти 4 (2^2) різних коди (00, 01, 10 і 11). Ними можна закодувати, наприклад, чотири сторони горизонту:

- 00 – *північ*;
- 01 – *схід*;
- 10 – *південь*;
- 11 – *захід*.

З трьох бітів можна скласти вже 8 (2^3) різних кодів (000, 001, 010, 011, 100, 101, 110, 111). Ними можна закодувати, наприклад, номери рядків або стовпців шахівниці. Також трьох бітів достатньо, щоб закодувати 8 кольорів.

Із чотирьох бітів можна скласти $2^4 = 16$ кодів, з п'яти – $2^5 = 32$ коди і т. д.

З восьми бітів можна скласти $2^8 = 256$ кодів, і цієї кількості кодів достатньо, щоб закодувати всі літери англійського та українського (або якогось іншого) алфавіту, арабські цифри, розділові знаки, знаки арифметичних дій, а також деякі інші символи. Саме така кількість кодів міститься, наприклад, у таблицях кодів символів **KOI-8U** та **Windows-1251**.

Послідовність з восьми бітів називають **байтом** і позначають **Б**.

1 Б = 8 б

Обсяг даних, або **довжина двійкового коду повідомлення**, – це кількість бітів у двійковому коді цього повідомлення. Також для вимірювання довжини двійкового коду використовують похідні одиниці – байти та інші.

Знаючи довжину двійкового коду, можна розрахувати, скільки повідомлень різних типів можна розмістити на носії даних, який обсяг даних потрібно передати мережею, скільки часу знадобиться для опрацювання даних тощо.

ДВІЙКОВЕ КОДУВАННЯ ТЕКСТОВИХ, ЧИСЛОВИХ ТА МУЛЬТИМЕДІЙНИХ ДАНИХ



Пригадайте

- Які ви знаєте таблиці кодів символів для кодування текстових повідомлень?
- Як можна закодувати графічне зображення?
- Як можна закодувати кольори?

Як ви знаєте, у таблиці кодів символів **Windows-1251** кожний символ має код – ціле число від 0 до 255, яке може бути закодоване одним байтом.

Приклади такого кодування наведено в таблиці 1.6.

Таблиця 1.6

Приклади двійкового кодування символів

Символ	*Код символа в таблиці Windows-1251*	*Відповідний байт*
!	33	00100001
@	64	01000000
W	103	01100111
ю	254	11111110
я	255	11111111

Таким чином, кожен символ у таблиці кодів символів Windows-1251 має двійковий код завдовжки 1 байт.

Якщо текстове повідомлення складається з кількох символів і кожний символ має двійковий код завдовжки 1 байт, то можна обчислити довжину двійкового коду всього повідомлення. Наприклад, повідомлення *Інформатика – цікавий предмет!* містить 30 символів (включаючи символи пропусків, тире, знак оклику). Якщо кожний символ кодувати двійковим кодом завдовжки 1 байт, то довжина двійкового коду такого повідомлення дорівнюватиме 30 байтам.

Щоб виконати двійкове кодування текстового повідомлення, потрібно код кожного символа цього повідомлення, отриманий з таблиці **Windows-1251**, замінити його двійковим кодом. Саме так кодуються текстові документи, створені в текстовому редакторі **Блокнот**, якщо під час їх зберігання на носії даних використовувати кодування **ANSI** (англ. *American National Standards Institute* – Американський національний інститут стандартів). Довжина двійкового коду повідомлення, що міститься в такому документі, буде дорівнювати в байтах кількості символів у текстовому документі. Але треба врахувати, що кожне натискання клавіші **Enter** кодується двома спеціальними символами – символом переходу на новий рядок і символом переходу на початок рядка.

Під час кодування текстів з використанням кодової таблиці **Юнікод** довжина двійкового коду різних символів є різною – від 1 Б до 6 Б.

Для двійкового кодування чисел їх переводять у **двійкову систему числення**. У звичній для нас **десятковій** системі числення для запису числа використовують цифри від 0 до 9, а у двійковій системі – лише цифри 0 та 1. Число 10 є **основою** десяткової системи числення, а число 2 – **основою** двійкової системи. Щоб зазначити основу системи числення в записі числа, використовують нижній індекс.

Як ви знаєте, цифри в десятковому записі числа (від останньої до першої) вказують, скільки одиниць, десятків, сотень, тисяч та інших степенів числа 10 міститься в заданому числі. Наприклад:

$$5328_{10} = 5 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 8 = 5 \cdot 1000 + 3 \cdot 100 + 2 \cdot 10 + 8$$

У двійковому записі числа цифри вказують, скільки одиниць, двійок, четвірок, вісімок та інших степенів числа 2 міститься в заданому числі. Наприклад:

$$1011_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1$$

Якщо в останньому виразі виконати обчислення, то запис числа з двійкового можна перевести у десятковий:

$$1011_2 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 = 11_{10}$$

Відповідно можна перевести десятковий запис числа у двійковий, подавши число сумою степенів двійки:

$$13_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_2$$

Якщо двійковий запис числа доповнити зліва нулями до отримання кількості цифр, кратної восьми, то отримаємо набір байтів, якими кодується число. Перші 16 цілих невід'ємних чисел, записані в десятковій і двійковій системах числення, та відповідні їм байти наведено у таблиці 1.7.

Таблиця 1.7

Запис чисел у десятковій і двійковій системах числення та відповідні байти

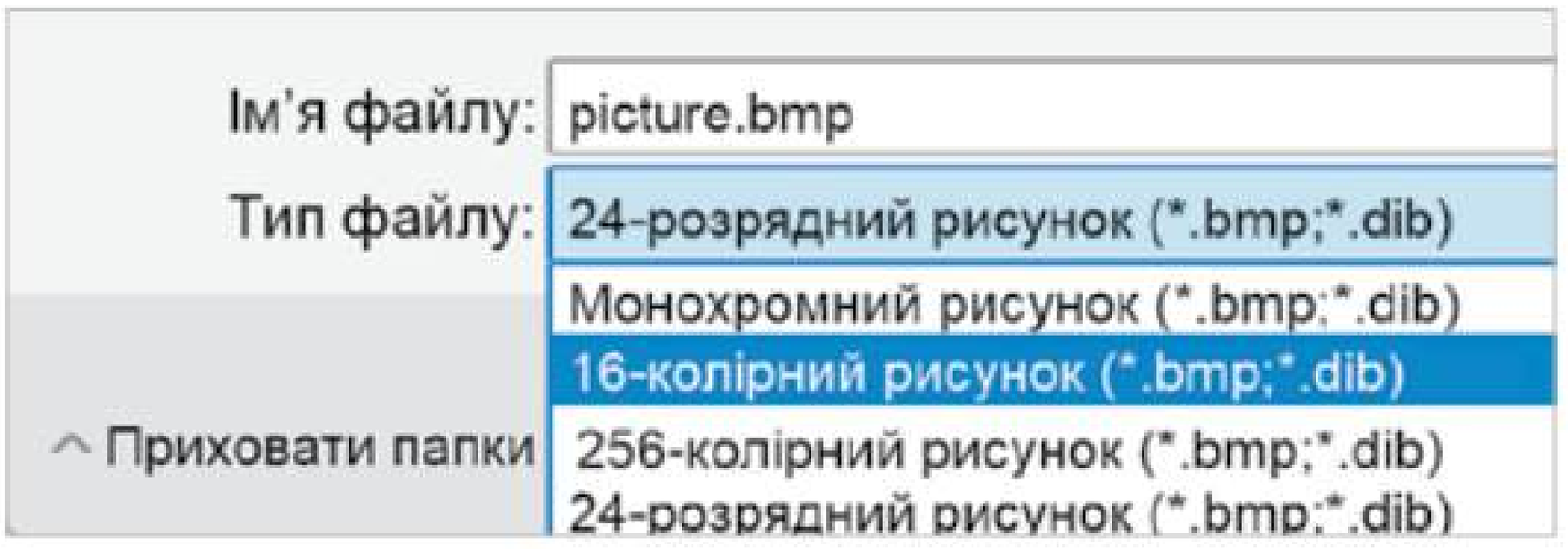
Десяткова система	*Двійкова система*	*Байт*	*Десяткова система*	*Двійкова система*	*Байт*
0	0	00000000	8	1000	00001000
1	1	00000001	9	1001	00001001
2	10	00000010	10	1010	00001010
3	11	00000011	11	1011	00001011
4	100	00000100	12	1100	00001100
5	101	00000101	13	1101	00001101
6	110	00000110	14	1110	00001110
7	111	00000111	15	1111	00001111

Графічні, звукові, відеоповідомлення для їх опрацювання комп'ютером також кодуються двійковими кодами.

Алгоритм двійкового кодування графічного зображення залежить від типу цього зображення та кількості бітів, вибраних для кодування кольорів, – **глибини кольору**.

Наприклад, у графічному редакторі **Paint** для зображення у форматі **BMP** можна вибрати під час збереження одне з чотирьох значень глибини кольору (мал. 1.14): *монохромний рисунок* – 1 біт для кодування кольору кожного пікселя, *16-колірний рисунок* – 4 біти, *256-колірний рисунок* – 8 бітів, *24-розрядний рисунок* – 3 байти, по 1 байту для кодування складових кольору з колірної моделі **RGB**. Двійковий код графічного зображення у форматі **BMP** з глибиною кольору 3 байти має таку структуру: дані про алгоритм кодування, розміри зображення, глибину кольору тощо, які утворюють заголовок розміром 54 байти, та по 3 байти кодів кольорів на кожен піксель зображення.

Для двійкового кодування звукових даних використовують алгоритми, які враховують числові значення властивостей звуку.



Мал. 1.14. Вибір глибини кольору під час збереження зображення у форматі **BMP**

КРАТНІ ОДИНИЦІ ВИМІРЮВАННЯ ДОВЖИНИ ДВІЙКОВОГО КОДУ



Пригадайте

- Що означають префікси *кіло-*, *мега-*, *гіга-*, *тера-*? З якою метою їх використовують?
- На яких уроках ви використовували іменовані величини з префіксами *кіло-*, *мега-*, *гіга-*, *тера-*?

Якщо довжина двійкового коду повідомлення велика, то для зручності використовують кратні одиниці вимірювання, які утворюються з використанням префіксів *кіло-*, *мега-*, *гіга-*, *тера-* та інших. Перелік цих префіксів для позначення кратних одиниць вимірювання визначено в Міжнародній системі одиниць (СІ). Там також наведено відповідні множники – 10^3, 10^6, 10^9, 10^{12} тощо.

Але дані в комп'ютерних системах кодуються двійковими кодами, тому для їх опрацювання зручно оперувати степенями числа 2, а не степенями числа 10. Оскільки $2^{10} = 1024$, що приблизно дорівнює $10^3 = 1000$, то історично склалося так, що саме 2^{10} Б = 1024 Б, а не 1000 Б, стали називати **кілобайт.**

У 1999 році Міжнародна електротехнічна комісія (МЕК) увела стандарт іменування двійкових префіксів. За цим стандартом 1 кілобайт вважається рівним 1000 байтів, а величина 1024 байтів отримала назву **кібібайт** (**кіло** **бі**нарний **байт**), скорочено **КіБ** (табл. 1.8).

Таблиця 1.8

Кратні одиниці вимірювання довжини двійкового коду та їх назви

Система СІ			*Стандарт МЕК*		
Назва	***Скорочення***	***Кількість байтів***	***Назва***	***Скорочення***	***Кількість байтів***
кілобайт	кБ	1000	кібібайт	КіБ	1024
мегабайт	МБ	1000^2	мебібайт	МіБ	1024^2
гігабайт	ГБ	1000^3	гібібайт	ГіБ	1024^3
терабайт	ТБ	1000^4	тебібайт	ТіБ	1024^4

Але до цього часу виробники комп'ютерної техніки та програмного забезпечення майже не вживають позначення стандарту МЕК, а позначення кБ, МБ, ГБ, ТБ та інші використовують відповідно до історичних традицій (так звана **звичайна конвенція**):

1 кБ (кілобайт) = 2^{10} Б = 1024 Б

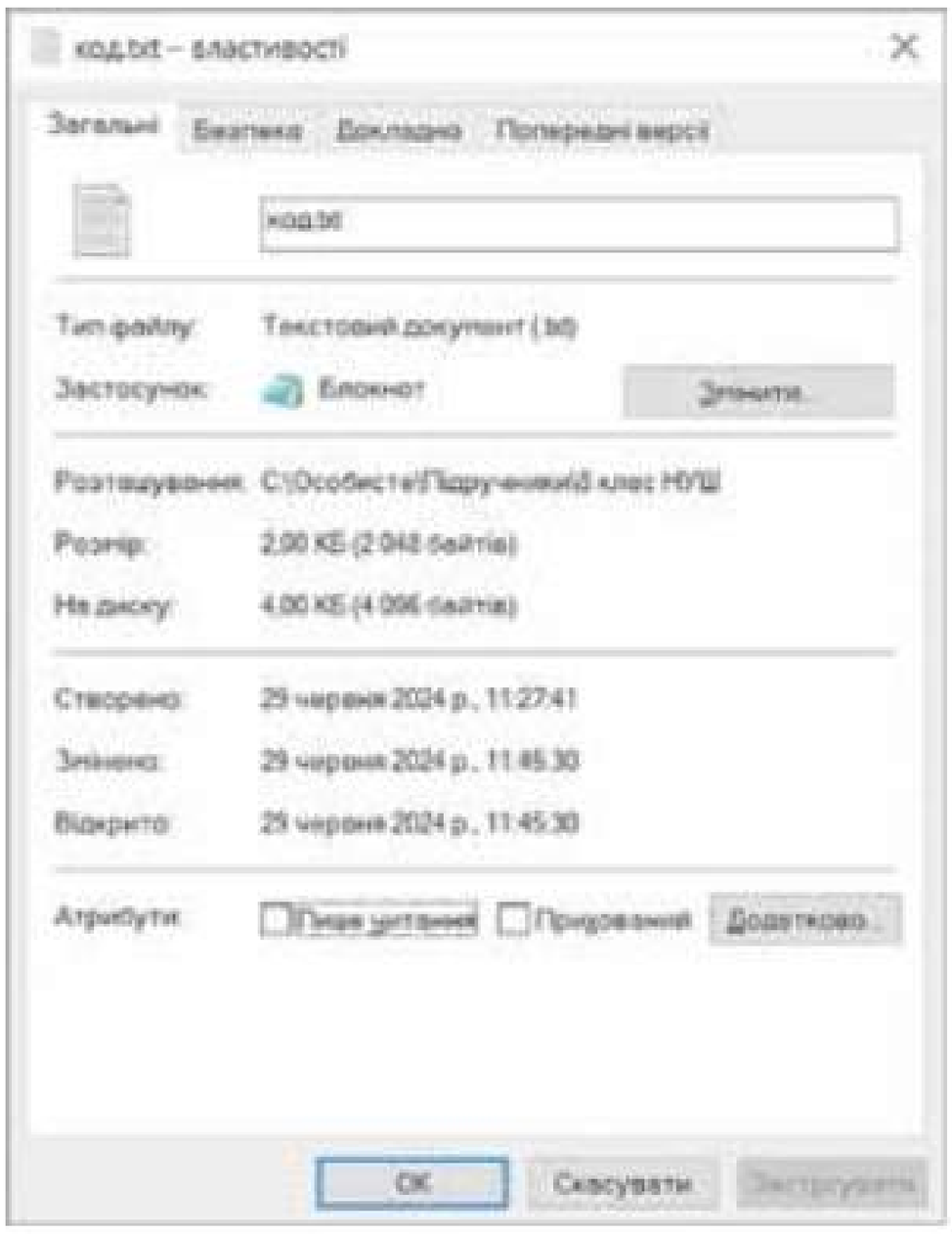
1 МБ (мегабайт) = 2^{10} кБ = 2^{20} Б = 1 048 576 Б

1 ГБ (гігабайт) = 2^{10} МБ = 2^{20} кБ = 2^{30} Б

1 ТБ (терабайт) = 2^{10} ГБ = 2^{20} МБ = 2^{30} кБ = 2^{40} Б

Такі позначення одиниць вимірювання довжини двійкового коду використовують і в операційній системі **Windows.** Наприклад, якщо створити та зберегти текстовий файл, що містить 2048 символів, закодованих з використанням таблиці кодів символів **ASCII**, то у вікні

його властивостей буде зазначено розмір *2,00 кБ (2048 байтів)* (мал. 1.15). Це відповідає звичайній конвенції. Тому й у цьому підручнику будемо вважати, що 1 кілобайт = 1024 байти.



Мал. 1.15. Властивості файлу

Кратні одиниці вимірювання довжини двійкового коду використовують для позначення розмірів файлів, ємності носіїв даних, обсягу даних, що передаються мережами, тощо.

ПЕРЕХІД ВІД ОДНИХ ОДИНИЦЬ ВИМІРЮВАННЯ ДОВЖИНИ ДВІЙКОВОГО КОДУ ДО ІНШИХ



Пригадайте

- Як перевести кілограми у грами, мегатонни в тонни, міліметри в метри?
- З використанням якої математичної операції можна перейти від крупніших одиниць вимірювання до дрібніших? Від дрібніших до крупніших?

Довжину двійкового коду повідомлень можна вказувати в бітах, байтах і в кратних їм одиницях. Розглянемо кілька задач на обчислення довжини двійкового коду повідомлень і на перехід при визначенні довжини двійкового коду повідомлення від одних одиниць вимірювання до інших.

Задача 1. Довжина двійкового коду текстового повідомлення складає 4,5 МБ. Скільки символів у кодуванні **Windows-1251** було закодовано?

Розв'язання. Оскільки в таблиці **Windows-1251** один символ кодується одним байтом, то для отримання відповіді на запитання потрібно виразити 4,5 МБ в байтах.

Для переходу від мегабайтів до байтів спочатку виразимо довжину двійкового коду цього повідомлення в кілобайтах. За домовленістю вважаємо 1 МБ = 1024 кБ.

4,5 МБ = 4,5 · 1024 кБ = 4608 кБ.
Переведемо кілобайти в байти, враховуючи, що 1 кБ = 1024 Б.
4608 кБ = 4608 · 1024 Б = 4 718 592 Б.
Відповідь: було закодовано 4 718 592 символи.

Задача 2. Обчислити наближено довжину двійкового коду тексту підручника з інформатики (не враховуючи форматування та малюнки).

Розв'язання. Уважатимемо, що текст закодовано з використанням таблиці кодів символів **Windows-1251**, тобто довжина двійкового коду кожного символа складає 1 Б.

У кожному рядку тексту міститься приблизно 60 символів, тобто довжина двійкового коду рядка складає приблизно 60 Б. На кожній сторінці розміщено в середньому 50 рядків. Таким чином, довжина двійкового коду сторінки складає приблизно 60 · 50 = 3000 (Б).

У підручнику приблизно 250 сторінок, тобто довжина двійкового коду підручника складає приблизно 3000 · 250 = 750 000 (Б).

Перейдемо до крупнішої одиниці вимірювання довжини двійкового коду:

$$750\,000 \text{ Б} : 1024 \approx 732{,}4 \text{ кБ}.$$

Відповідь: довжина двійкового коду тексту підручника з інформатики складає приблизно 732,4 кБ.

Задача 3. Обчислити розмір файлу зображення, створеного у графічному редакторі **Paint**. Файл формату **BMP** збережено як 24-розрядний рисунок. Зображення має ширину 25 та висоту 20 пікселів.

Розв'язання. Для розрахунку довжини двійкового коду зображення з указаними значеннями властивостей потрібно знайти суму довжин коду заголовка (54 байти) та кодів кольорів усіх пікселів зображення.

Обчислимо довжину двійкового коду кольорів одного рядка зображення, враховуючи, що в рядку 25 пікселів, а у 24-розрядному рисунку колір кожного пікселя кодується трьома байтами:

$$25 \cdot 3 \text{ Б} = 75 \text{ Б}.$$

Відповідно до алгоритму кодування довжина коду рядка повинна бути кратна числу 4, тому до отриманої довжини може бути додано від 1 до 3 байтів так званого **вирівнювання рядків**. У даному випадку вирівнювання складає 1 Б, а довжина коду рядка вважатиметься рівною 76 Б.

Обчислимо довжину двійкового коду кольорів усіх пікселів зображення, враховуючи, що воно складається з 20 рядків:
20 · 76 Б = 1520 Б.
Загальна довжина двійкового коду заголовка та кольорів пікселів:
54 Б + 1520 Б = 1574 Б.
Відповідь: розмір файлу зображення 1574 Б.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/3e2cO4IR* або QR-кодом.



Найважливіше в цьому пункті

Кодування повідомлень з використанням сигналів лише двох видів називається **двійковим кодуванням**. Повідомлення, отримане в результаті двійкового кодування початкового повідомлення, називають **двійковим кодом**.

Цифру 0 або 1 у двійковому коді повідомлення називають **біт** і позначають маленькою літерою **б**.

Послідовність з восьми бітів називають **байт** і позначають **Б**.

1 байт = 8 бітів

Обсяг даних, або **довжина двійкового коду повідомлення**, – це кількість бітів у двійковому коді цього повідомлення. Також для вимірювання довжини двійкового коду використовують похідні одиниці – байти та інші.

Кожному символу в таблиці кодів символів **Windows-1251** відповідає двійковий код завдовжки 1 Б. Щоб виконати двійкове кодування текстового повідомлення, потрібно код кожного символа цього повідомлення, отриманий з таблиці **Windows-1251**, замінити його двійковим кодом.

Для двійкового кодування чисел їх переводять у **двійкову систему числення**. У двійковій системі числення для запису числа використовують цифри 0 та 1. У двійковому записі числа цифри вказують, скільки одиниць, двійок, четвірок, вісімок та інших степенів числа 2 міститься в заданому числі.

Алгоритм двійкового кодування графічного зображення залежить від типу цього зображення та кількості бітів, вибраних для кодування кольорів, – **глибини кольору**.

Для двійкового кодування звукових даних використовують алгоритми, які враховують числові значення властивостей звуку.

Дані в комп'ютерних системах кодуються двійковими кодами, тому для їх опрацювання зручно оперувати степенями числа 2. На цьому базується **звичайна конвенція** утворення кратних одиниць вимірювання довжини двійкового коду:

1 кБ (кілобайт) = 2^{10} Б = 1024 Б
1 МБ (мегабайт) = 2^{10} кБ = 2^{20} Б = 1 048 576 Б
1 ГБ (гігабайт) = 2^{10} МБ = 2^{20} кБ = 2^{30} Б
1 ТБ (терабайт) = 2^{10} ГБ = 2^{20} МБ = 2^{30} кБ = 2^{40} Б



Дайте відповіді на запитання

1. У чому полягає двійкове кодування повідомлень?

2. Що таке *1 біт*? Чому дорівнює *1 байт*? Що таке *довжина двійкового коду повідомлення*?

3. Якою є довжина двійкового коду кожного символа в таблиці кодів символів **Windows-1251**? Як можна обчислити довжину двійкового коду повідомлення, символи якого закодовано з використанням цієї таблиці кодів символів?

4. Як виконується двійкове кодування числових даних? Як виконується двійкове кодування графічних зображень?

5. Чому дорівнює 1 кБ, 1 МБ, 1 ГБ, 1 ТБ у звичайній конвенції?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, в комп'ютерах використовується двійкова система числення, а в побуті – десяткова?

2. Крім десяткової та двійкової, існують інші системи числення. Чи може існувати система числення з основою, меншою за 2? Чи можуть існувати системи числення з основами, більшими за 10? Відповідь поясніть.

3. Як залежить якість графічного зображення від глибини кольору?

Виконайте завдання

1. Наведіть приклади, де використовується двійкове кодування повідомлень.

2. Закодуйте з використанням азбуки Морзе слова *порт*, *рота*, *торт*. Порівняйте кількість сигналів у двійковому коді цих слів.

3. Визначте та запишіть у зошиті довжину двійкового коду текстових повідомлень за умови, що повідомлення закодовано з використанням таблиці кодів символів **Windows-1251**:

а) *Слава Україні!*

б) *В алфавіті азбуки Морзе два символи – крапка та тире.*

4. Об'єднайтесь у пари. Кожен з учасників / учасниць пари кодує власне повідомлення з використанням азбуки Морзе, не залишаючи проміжку між літерами, та з використанням таблиці кодів символів **Windows-1251**, а потім передає код напарнику / напарниці. Другий / -га учасник / учасниця декодує обидва коди. Обговоріть, яка система кодування зручніша для декодування.

5. Заповніть пропуски на основі звичайної конвенції:

а) 5 Б = _____ б;

б) 4096 Б = _____ кБ;

в) 10 кБ = _____ б;

г) 3 МБ = _____ Б;

д) 2 кБ = _____ Б;

е) 100 МБ = _____ кБ.

6. Фірма-виробник зазначила ємність флешнакопичувача 16 МБ. Зробіть розрахунки, на скільки байтів відрізняється ємність накопичувача, якщо вважати, що 1 кБ = 10^3 Б або 1 кБ = 2^{10} Б. У якому випадку користувач отримує більшу ємність носія даних?

7. В одному рядку деякого тексту розміщується в середньому 60 символів, а на одній сторінці – 40 рядків. На носії даних залишилося 8 МБ вільного місця. Скільки закодованих сторінок цього тексту може вміститися на носії, якщо для кодування використати таблицю **Windows-1251**?

8. Два графічних зображення збережено у файлах формату **BMP** як 24-розрядні рисунки. Обчисліть і порівняйте розміри цих файлів, якщо в одного ширина 75 пікселів, а висота 50, в іншого ширина 50 пікселів, а висота 75. Запишіть висновки, чи залежить розмір графічного файлу формату **BMP** від орієнтації зображення.

Готуємось до вивчення нового матеріалу

1. Яку установу називають архівом? Які функції вона виконує?

2. Під час виконання яких інформаційних процесів суттєвою є довжина двійкового коду повідомлень?

3. Як впливає на довжину двійкового коду повідомлень вибраний користувачем спосіб кодування даних?

Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу «Кодування» (початковий, середній, достатній, високий рівень).

- Я можу пояснити сутність процесу кодування.
- Я розумію сутність двійкового кодування даних.
- Я вмію виконувати кодування даних різних типів відповідно до окреслених правил.

- Я можу пояснити принципи подання та опрацювання даних різних типів (числа, текст, звуки, зображення) у двійковому та інших видах кодування.
- Я знаю одиниці вимірювання довжини двійкового коду повідомлень.
- Я вмію переводити довжину двійкового коду повідомлень з одних одиниць вимірювання в інші.

Повторіть той матеріал, який ви знаєте недостатньо.



Практична робота № 1. «Розв'язування задач на визначення довжини двійкового коду повідомлень»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Обчисліть довжину двійкового коду деякого тексту за умови, що символи тексту закодовано з використанням таблиці кодів символів **Windows-1251**. У тексті 32 сторінки, кожна сторінка тексту містить 48 рядків, у кожному рядку в середньому 56 символів. Виразіть довжину двійкового коду цього тексту цілим числом у найбільших можливих одиницях.
2. Визначте, використовуючи текстовий редактор **Блокнот**, та запишіть у зошит довжину двійкового коду повідомлення *Практична робота № 1* за умови, що під час збереження файлів використовується кодування:
   а) **ANSI** (таблиця кодів символів **Windows-1251**);
   б) **UTF-8** (таблиця кодів символів **Юнікод**).
   Збережіть текстові документи з повідомленням у файлах з іменами **практична 1 ANSI.txt** і **практична 1 UTF-8.txt**.
3. Заповніть у табличному процесорі таблицю на основі звичайної конвенції.

Бітів	*Байтів*	*Кілобайтів*	*Мегабайтів*
614 400			
	524 288		
		256	
			4

Збережіть результат у файлі з іменем **практична 1**.

РОЗДІЛ 2

АРХІВИ ДАНИХ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- архівування та стиснення даних;
- види стиснення даних;
- програми для створення архівів (архіватори);
- типи архівних файлів;
- створення архівів;
- операції над архівами;
- резервне копіювання даних.

2.1. СТИСНЕННЯ ДАНИХ. АРХІВУВАННЯ ДАНИХ. РЕЗЕРВНЕ КОПІЮВАННЯ

У цьому пункті йтиметься про:

- способи стиснення даних;
- архіви та способи їх створення у сфері інформаційних технологій;
- створення копій даних (архівів) в операційній системі з використанням **Резервного копіювання**;
- відновлення даних з резервних копій.

СТИСНЕННЯ ДАНИХ

Пригадайте

- Як здійснюється кодування текстових і числових даних?
- Як здійснюється кодування мультимедійних даних?

Незважаючи на те, що ємності носіїв даних постійно збільшуються, залишається потреба в зменшенні розмірів файлів, особливо у випадках їх передавання комп'ютерними мережами, використання в мобільному зв'язку та на мобільних пристроях. Для зменшення розмірів файлів використовують спеціальні алгоритми (методи) стиснення даних. Стиснення даних застосовують під час створення файлів певних типів, наприклад графічних типу **TIFF**, **JPEG**, **PNG**, звукових типу **MPEG3**, **WMA**, відео типу **MPEG4**, **WMV**, **MOV**, **WebM**, **Matroska** тощо.

Стиснення даних – це процес кодування даних, який здійснюється з метою зменшення розмірів файлів. Одним з перших застосувань стиснення даних для їх подальшого передавання є використання **азбуки Морзе**, про яку йшла мова в попередньому пункті. Основною особливістю цієї азбуки, яка і впливала на довжину коду, що передавався електричними мережами (телеграф), було ***врахування частотності використання літер*** у певній мові. Що частіше використовувалася певна літера, то коротший був її код. Наприклад, для найбільш уживаної в англійських текстах літери «e» використовувався код «.» (одна крапка), а для літери «t» – «–» (одне тире). Найменш уживані літери «b», «h», «j», «p», «q», «v», «x», «y», «z» кодуються комбінаціями з чотирьох символів.

Для тих, хто хоче знати більше

На жаль, таблиця кодування української абетки не в повній мірі реалізує систему, придуману С. Морзе. Річ у тому, що кодування в українському варіанті азбуки Морзе здійснено на основі спорідненості звуків, які позначають літери цих абеток, а не частотності літер в українській мові. Наприклад, найбільш уживаною в українських текстах є літера «о» (позначається в азбуці Морзе – – –), а літера «е» – лише дев'ята за частотністю.

Тому азбука Морзе в українському варіанті не забезпечує в повній мірі стиснення даних.

Стиснення текстів з використанням азбуки Морзе можна відслідкувати на прикладі порівняння двійкового кодування з використанням цієї азбуки з кодуванням ASCII фрагмента тексту **HELLO WORLD** (табл. 2.1). Навіть при застосуванні основної таблиці ASCII ми бачимо, що кодуванням одного і того самого тексту за системою, що запропонував С. Морзе, ми зменшимо кількість символів для передавання більше ніж удвічі. Для комп'ютерних систем, як пояснено в попередньому пункті, це не зручно.

Таблиця 2.1

Порівняння кодування тексту

	Система кодування	
	Код ASCII	***Азбука Морзе***
H	0100 1000	• • • •
E	0100 0101	•
L	0100 1100	• – • •
L	0100 1100	• – • •
O	0100 1111	– – –
W	0101 0111	• – –
O	0100 1111	– – –
R	0101 0010	• – •
L	0100 1100	• – • •
D	0100 0100	– • •
Кількість символів	80	32

Розрізняють алгоритми стиснення, що забезпечують стиснення ***без втрати даних***, і алгоритми, що передбачають ***часткову втрату даних***. Алгоритми із частковою втратою даних використовують, коли цілісність даних не є дуже суттєвою. Наприклад, під час стиснення графічних, відео-, звукових файлів, оскільки органи чуття людини не завжди здатні помітити незначну різницю у відтінках кольорів на фотографії, у відтворенні звукових або відеоданих. А також якщо втрата якості не настільки важлива, як потреба зменшити розміри файлів для запису на зовнішній носій малої ємності або для використання в інтернеті. Однак алгоритми з втратою даних не можна застосовувати під час стиснення текстових і числових даних, оскільки це призводить до повного спотворення закодованих даних.

Для тих, хто хоче знати більше

Методи стиснення даних

Алгоритми Д. Хаффмана та **Шеннона – Фано** використовують кодування зі змінною кількістю символів коду на символ повідомлення. Символи, що частіше зустрічаються в повідомленні, кодуються меншою кількістю бітів, а які рідше – більшою. Так, наприклад, для кодування з використанням таблиці кодів символів **Windows-1251** попереднього речення, яке містить 106 символів, включаючи пропуски, слід використати код завдовжки 1 байт на кожний символ або 106 байтів на

все речення. Якщо ж порахувати, скільки разів той чи інший символ трапляється в реченні, то виявиться, що всього різних символів, які є в реченні, – 28, частіше трапляється символ «пропуск» – 14 разів, символ «і» – 11, «о» – 7, «с» – 6, «в» – 4 і т. д. Найменше, по одному разу, зустрічаються символи «щ», «з», «–», «.». Позначимо символи, які частіше трапляються, кодом завдовжки 1 біт: символ «пропуск» – 0, а символ «і» – 1; наступні за частотою повторюваності – кодом завдовжки 2 біти: символ «о» – 00, символ «с» – 01 і т. д. Для символів, що трапляються рідше, буде використано код завдовжки 6 біт. У результаті загальна довжина коду закодованого у такий спосіб речення буде складати 384 біти або приблизно 48 байтів. Як бачимо, загальний обсяг коду речення зменшився більш ніж удвічі. Під час збільшення довжини тексту стиснення буде набагато ефективніше.

Звертаємо вашу увагу, що у стиснутому таким способом файлі повинна додатково міститися сама таблиця кодування та додаткові службові відомості. Тому під час кодування дуже малих файлів може відбутися не зменшення, а, навпаки, – збільшення розмірів файлів. А для файлів з великим обсягом текстових чи числових даних цей алгоритм дає високий ступінь стиснення.

Алгоритм RLE (англ. ***Run-length encoding*** – кодування повторів або довжин серій) використовується здебільшого для кодування графічних даних. Цей метод передбачає заміну послідовності однакових даних її скороченим описом. Наприклад, у малюнку логотипа корпорації **Apple** (мал. 2.1) використано тільки два кольори – чорний і білий. Якщо позначити чорний колір кодом «1», а білий – «0», то код першого рядка зображення повинен складатися із 48 кодів «0» (48 бітів). Але їх можна замінити описом, указавши код кольору і кількість повторів – 0 48. На код опису буде відведено 9 бітів: 1 біт на код «0» і по чотири біти на дві цифри кількості повторів.

Другий ряд пікселів буде закодовано так само, а третій – 0 29 1 4 0 15, що складає 23 біти замість 48. Для кольорових зображень з великою кількістю одноколірних фрагментів даний алгоритм забезпечить високу ефективність стиснення.



Мал. 2.1. Логотип **Apple**

АРХІВУВАННЯ ДАНИХ І РЕЗЕРВНЕ КОПІЮВАННЯ

Поміркуйте

- Розгляньте малюнок (мал. 2.2), на якому зображено працівників архіву. Які операції можуть здійснювати працівники архіву? З якою метою здійснюють ці операції? Як можна удосконалити їх роботу?



Мал. 2.2. Працівники архіву

Незважаючи на підвищення надійності комп'ютерів і комп'ютерних носіїв даних, усе ж повної гарантії збереження даних вони не дають. Втрата даних може призвести до дуже серйозних наслідків. Видалення або пошкодження даних про вклади та перерахування

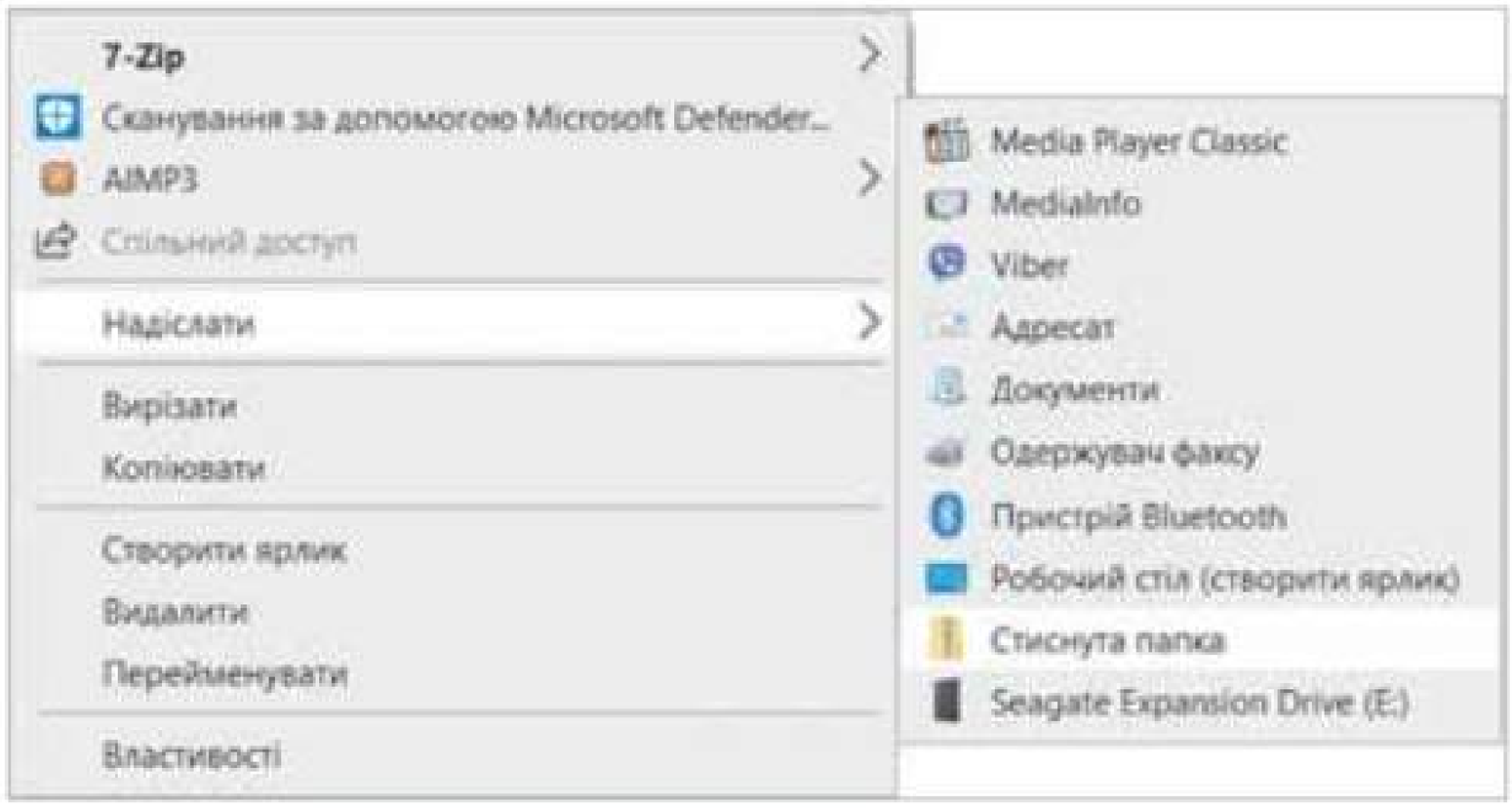
коштів клієнтів призведе до краху банку, втрата даних про продаж квитків спричинить перебої в перевезенні пасажирів, знищення результатів дослідів може звести нанівець багаторічні наукові дослідження. Навіть втрата списку контактів у мобільному телефоні принесе вам значні незручності. Тому виникає потреба у створенні копій даних.

Найважливіші дані дублюють, записуючи на інші носії даних, на віртуальні диски в комп'ютерних мережах тощо. Зазвичай для зручності збереження та зменшення обсягів даних файли та папки під час створення резервних копій упаковують в один файл або в одну папку. Такі копії даних називаються **архівами**, а файли, у які вони упаковуються, – **архівними файлами**.

Створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, називається **архівуванням**.

Для створення архівів файлів і папок з використанням засобів операційної системи слід:

1. Зробити поточною папку, у якій розміщено файли і папки, які планується включити до архіву.
2. Виділити файли і папки, які планується включити до архіву.
3. Вибрати в контекстному меню команди ***Надіслати*** ⇒ ***Стиснута папка*** (мал. 2.3).
4. Увести замість запропонованого ім'я архівного файлу або погодитись із запропонованим.
5. Натиснути клавішу **Enter**.



Мал. 2.3. Контекстне меню виділених об'єктів ОС

У поточній папці буде створено архівний файл з розширенням іменні **zip** і піктограмою або подібною. Архівний файл цього типу опрацьовується операційною системою як звичайна папка. Над цим об'єктом як над папкою та об'єктами, розміщеними в ній, можна виконувати стандартні операції операційної системи: копіювання, переміщення, видалення, відкриття тощо. В іменах виділених для архівування файлів і папок не повинно бути літер українського алфавіту **і, ї, є**.

Більшість операційних систем мають у своєму складі засоби створення резервних копій даних. Ці програми зазвичай використовують для архівування даних з метою відновлення операційної системи на іншому диску або в мережі, а також створюють архіви файлів користувача.

Для створення резервних копій файлів слід:

1. Відкрити вікно налаштувань резервного копіювання (***Пуск*** ⇒ ***Налаштування*** ⇒ ***Оновлення та захист*** ⇒ ***Резервне копіювання файлів***).
2. Вибрати у групі **Резервне копіювання з використанням банку файлів** кнопку **Додати диск** для першого запуску резервного копіювання або посилання **Інші параметри** для наступних.
3. Вибрати диск, на який будуть записуватися резервні копії файлів, у вікні **Виберіть диск.**
4. Вибрати посилання **Інші параметри** у групі **Резервне копіювання з використанням банку файлів.**
5. Установити у вікні **Параметри резервного копіювання** значення властивостей резервного копіювання (мал. 2.4):



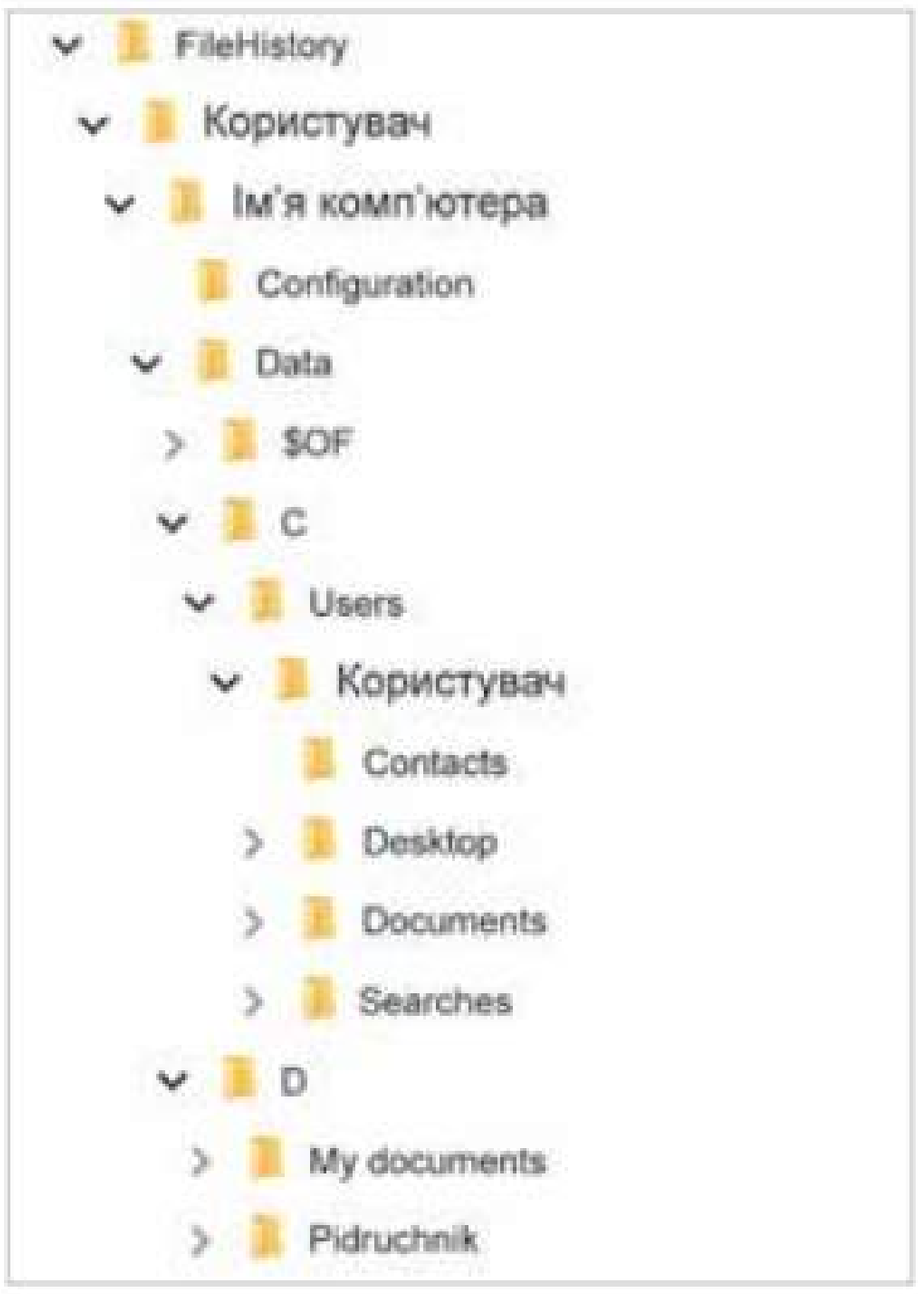
Мал. 2.4. Вікно **Параметри резервного копіювання**

- *періодичність операцій оновлення резервних файлів* – у списку, що відкривається, **Резервне копіювання файлів** (від *10 хвилин* до *Щодня*);
- *тривалість періоду зберігання резервних копій* – у списку, що відкривається, **Зберегти мої резервні копії** (від *одного місяця* до *Завжди* або *Доки не знадобиться простір*);
- *перелік папок*, файли з яких будуть включені до резервної копії, – у списку **Створити резервні копії цих папок.** За замовчуванням до цього переліку включено папки облікового запису відповідного користувача (**Документи, Зображення, Робочий стіл** та інші). Для додавання іншої папки слід вибрати кнопку **Додати папку** та здійснити пошук потрібної папки. Для видалення папки зі списку слід вибрати потрібну папку в списку та вибрати кнопку **Видалити.**

6. Вибрати кнопку **Створити резервну копію зараз** у групі **Огляд.**

Після завершення створення резервної копії файлів на диску, який вибрано для зберігання резервної копії, буде створено папку **FileHistory** (англ. *file history* – історія файлів), яка має структуру, подібну до наведеної на малюнку 2.5, де:

- **Користувач** – папка з іменем облікового запису користувача, у якій створено резервні копії файлів;
- **Ім'я комп'ютера** – папка, що має таке саме ім'я, як і комп'ютер;
- **Configuration** – папка з файлами налаштування резервної копії;
- **Data** – папка з папками і файлами користувача, які він уключив до резервної копії;
- **C, D** – папки резервних копій файлів і папок, що зберігаються на дисках C: та D: відповідно;
- **Users** – папка з копіями папок, що автоматично створюються для кожного облікового запису ОС **Windows**;
- **Contacts, Desktop, Documents, Searches** – копії папок **Контакти, Робочий стіл, Документи, Пошук,** що автоматично створюються для кожного облікового запису ОС **Windows** і які користувач уключив до резервної копії;
- **My documents, Pidruchnik** – копії папок, які користувач додатково включив до резервної копії.

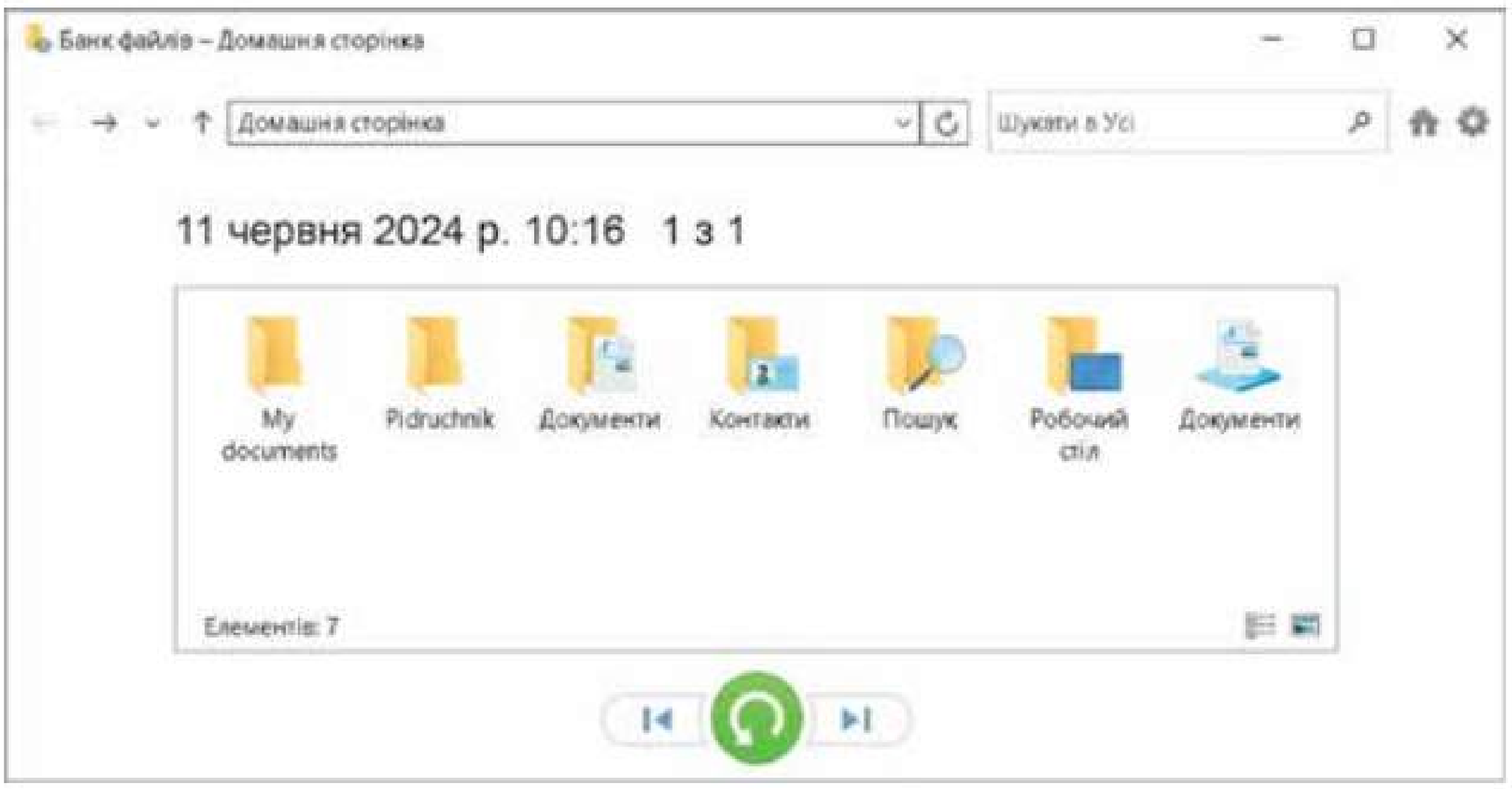


Мал. 2.5. Структура папки **FileHistory**

Слід зазначити, що резервні копії даних користувача, створені в описаний спосіб, не передбачають стиснення даних. Ця операція призначена для автоматизації створення резервних копій даних.

У разі збоїв у роботі операційної системи та/або втрати файлів з даними користувача їх можна відновити на ті самі диски і в ті самі папки з резервної копії (з папки **FileHistory**). Для відновлення файлів з резервної копії даних слід:

1. Виконати ***Пуск*** ⇒ ***Налаштування*** ⇒ ***Оновлення і захист*** ⇒ ***Резервне копіювання*** ⇒ ***Інші параметри*** ⇒ ***Відновити файли з поточної резервної копії.***
2. Вибрати у вікні **Банк файлів** (мал. 2.6) папки і файли, уміст яких потрібно відновити.
3. Вибрати кнопку **Відновити в початковому розташуванні** .



Мал. 2.6. Вікно **Банк файлів**

У разі наявності кількох версій резервних копій файлів для обрання потрібної версії слід скористатися кнопками **Попередня версія** або **Наступна версія** . Дата і час створення версії резервної копії відображаються над полем з іменами збережених папок.

Засіб резервного копіювання надає користувачу можливість відновлювати не лише всі дані, але й окремі папки та файли, які вибрані у вікні **Банк файлів**. Якщо в певній папці є вже такий самий файл або папка, то програма відкриє діалогове вікно із запитом на заміну чинних даних.

ЗБЕРЕЖЕННЯ АРХІВУ ДАНИХ «У ХМАРІ»

Пригадайте

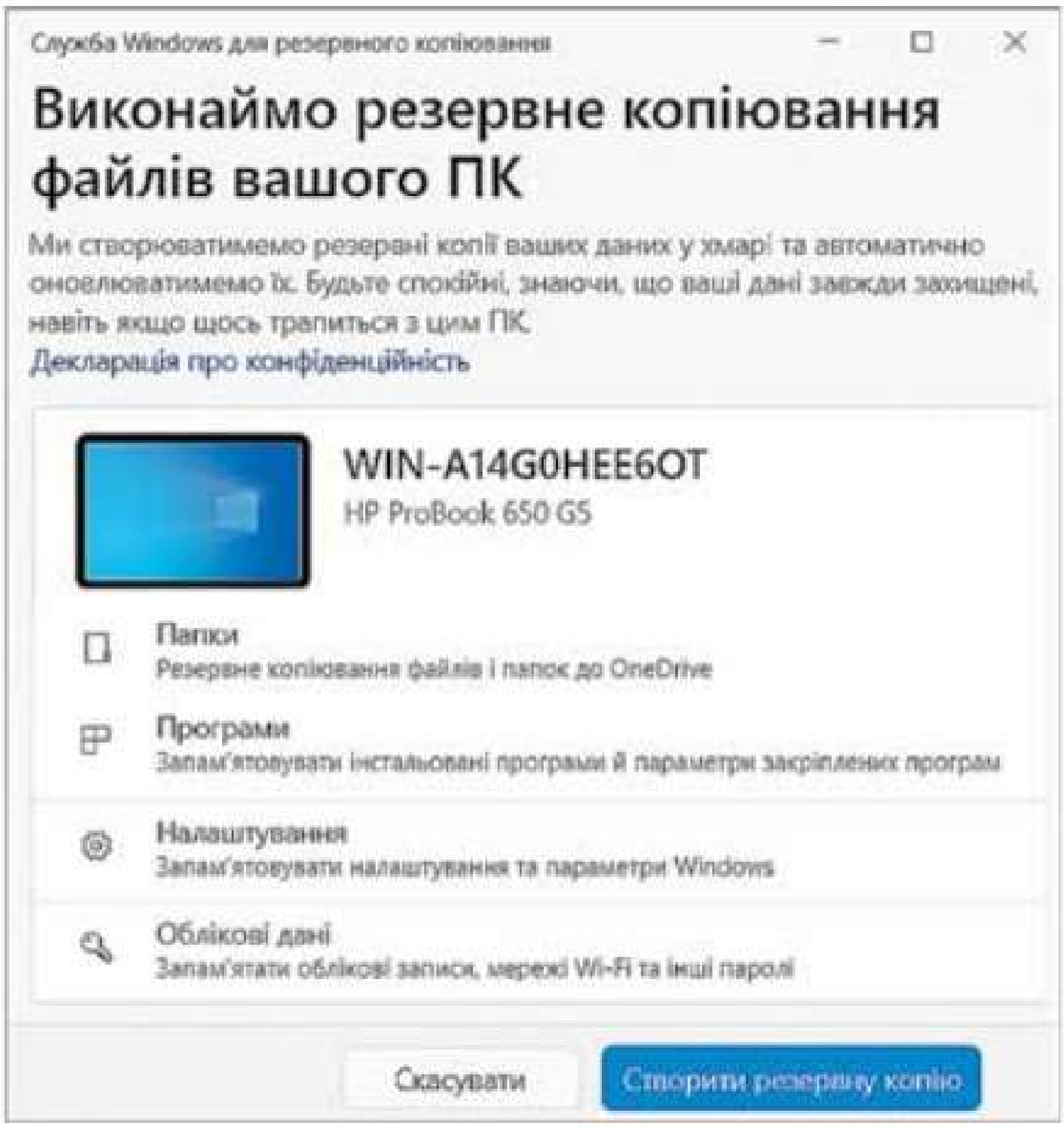
- Які сервіси інтернету ви знаєте? Які із цих сервісів призначені для зберігання даних користувача?

Різноманітні сервіси інтернету для зберігання даних зазвичай надають можливість зберігати файли і папки користувача, за певних умов синхронізують їх з файлами на інших комп'ютерних пристроях.

В операційній системі **Windows**, починаючи з десятої версії, з'явилася програма, у чомусь схожа з програмою резервного копіювання, але зі збереженням даних користувача в інтернеті – «у хмарі».

Для створення резервної копії даних користувача з певного комп'ютера слід:

1. Виконати ***Пуск ⇒ Служба Windows для резервного копіювання.***
2. У вікні **Служба Windows для резервного копіювання** (мал. 2.7) вибрати кнопку **Створити резервну копію.**
3. Увести дані свого акаунта **Microsoft.**
4. Дочекатися завершення резервного копіювання.



Мал. 2.7. Вікно **Служба Windows для резервного копіювання**

За потреби можна змінити налаштування резервного копіювання, вибравши відповідні посилання **(Папки, Програми, Налаштування, Облікові дані).**

КОНТРОЛЬНІ ТОЧКИ ВІДНОВЛЕННЯ СИСТЕМИ

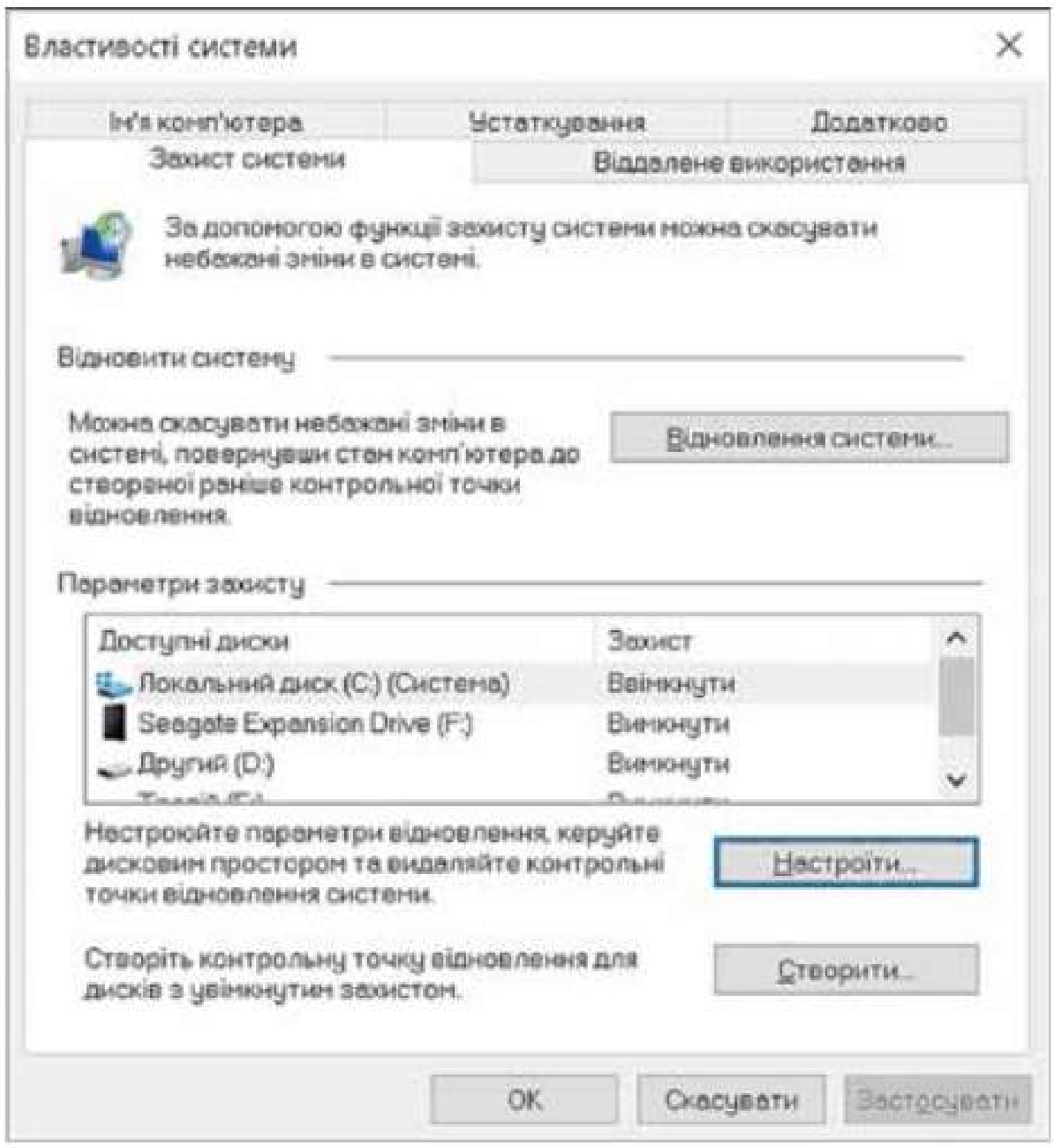
Поміркуйте

- Операційна система на вашому комп'ютері почала працювати значно повільніше після її оновлення, виникли проблеми з роботою окремих програм. Як повернути систему до попереднього стану?

Операційні системи зазвичай автоматично створюють копії (архіви) системних налаштувань у вигляді **Контрольних точок відновлення.** Ці архіви системних налаштувань користувач може використати в разі збоїв у роботі операційної системи або сповільнення її роботи.

В операційній системі **Windows** програма створення контрольних точок відновлення вмикається за замовчуванням і активізується системою під час виконання певних критичних дій, які можуть призвести до нестабільної роботи операційної системи: інсталяція певних драйверів пристроїв, оновлення операційної системи тощо. Користувач також може створювати контрольні точки відновлення перед діями, що можуть призвести до дестабілізації операційної системи, наприклад перед інсталяцією сумнівних програм, наданням дозволу на керування комп'ютером ззовні, коригуванням певних налаштувань операційної системи.

Для налаштування значень параметрів створення контрольних точок користувачем або відновлення системи з використанням контрольних точок потрібно виконати ***Пуск*** ⇒ ***Налаштування*** ⇒ ***Додаткові налаштування системи*** та відкрити вкладку **Захист системи** у вікні **Властивості системи** (мал. 2.8).



Мал. 2.8. Вкладка **Захист системи** вікна **Властивості системи**

Слід зважати на те, що контрольна точка відновлення системи зберігає відомості про основні налаштування операційної системи на момент створення і вона не містить файлів користувача.

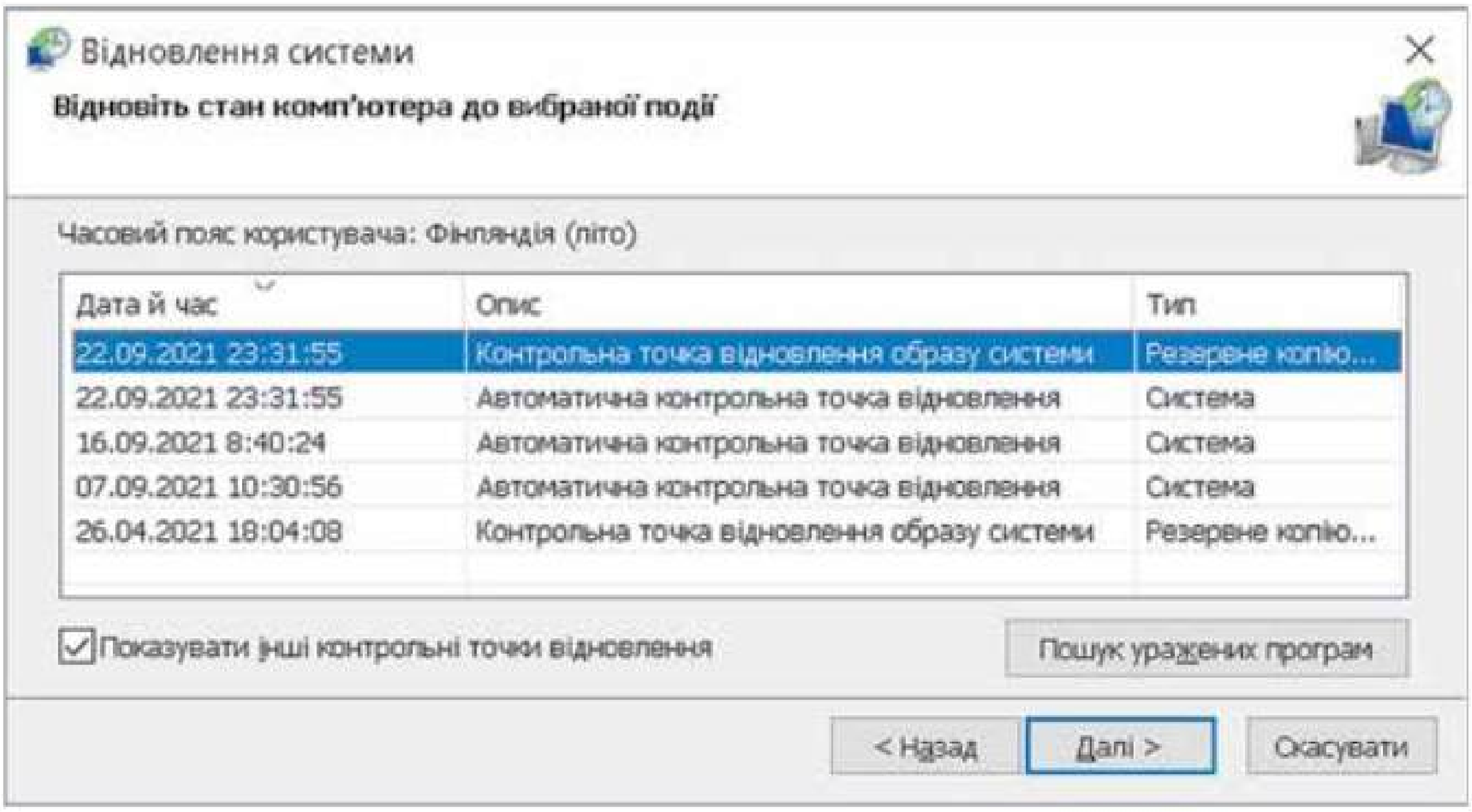
Вибравши кнопку **Настроїти**, користувач у вікні **Захист системи для...** має можливість:

- увімкнути або вимкнути захист системи (у разі вимкнення видаляються всі раніше створені контрольні точки);

- установити обсяг максимального дискового простору для запису всіх контрольних точок (у разі перевищення встановленого значення старі точки відновлення будуть видалятися для звільнення місця для запису нових);
- видалити всі попередні точки відновлення.

Для створення нової точки відновлення потрібно на вкладці **Захист системи** у вікні **Властивості системи** вибрати кнопку **Створити** та в діалоговому вікні ввести опис точки відновлення.

У разі збоїв у роботі операційної системи для її відновлення з образу системи слід у вікні **Властивості системи** на вкладці **Захист системи** вибрати кнопку **Відновлення системи**. Після цього підтвердити намір відновлення параметрів системи вибором кнопки **Далі**. У наступному вікні в таблиці (мал. 2.9) обрати потрібну точку відновлення та вибрати кнопку **Далі**.



Мал. 2.9. Вікно **Відновіть стан комп'ютера до вибраної події**

Систему буде відновлено до стану, що відповідає стану системи в обраній контрольній точці відновлення. При цьому файли користувача, що були створені до і після створення контрольної точки, зберігаються.

Для тих, хто працює з Android

Операційна система **Android** також має систему резервного копіювання даних для зберігання копій даних користувача, таких як sms-повідомлення, контакти, веб-закладки, електронні листи та інше. Зазвичай резервні копії даних можуть зберігатися на хмарних сервісах відповідно до тих облікових записів, які ви зареєстрували на вашому смартфоні або планшетному комп'ютері. Зазвичай це облікові записи **Google** та виробника вашого пристрою. Для ввімкнення системи резервного копіювання найчастіше слід виконати ***Налаштування*** ⇒ ***Системне керування*** ⇒ ***Резервне копіювання та скидання*** ⇒ ***Резервне копіювання даних*** і вибрати той сервіс, який ви будете використовувати для резервного копіювання даних.

Слід зауважити, що для резервування значної частини даних зі смартфона може використовуватися функція синхронізації певних сервісів, наприклад сервісів **Google**. Тоді дані, що синхронізуються, будуть зберігатись у хмарі – тобто на сервері власника відповідних сервісів.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ce2cPgcd* або QR-кодом.



Найважливіше в цьому пункті

Для зменшення обсягів даних використовують стиснення даних за спеціальними алгоритмами (методами) стиснення. **Стиснення даних** – це процес перекодування даних, який здійснюється з метою зменшення розмірів файлів.

Розрізняють алгоритми стиснення, що забезпечують стиснення *без втрати даних*, і алгоритми, що передбачають *часткову втрату даних*.

Створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, називається **архівуванням**. Копії даних називаються **архівами**, а файли, у які вони упаковуються, – **архівними файлами**.

Для створення резервних копій використовують засоби операційної системи або відповідні службові програми. З використанням останніх здійснюють копіювання даних для відновлення операційної системи. Ці дані копіюються на інший носій даних на цьому комп'ютері або у хмарному сховищі. При цьому можуть створюватись архіви файлів користувача.

Операційна система **Windows** здійснює операції над файлами архівів формату **ZIP**, як над звичайними папками, тобто відкриває їх вміст, копіює файли з архівного файлу в іншу папку, видаляє файли з архіву тощо. У контекстному меню виділених файлів і папок можна виконати ***Надіслати*** ⇒ ***Стиснута папка*** й отримати файл архіву формату **ZIP**. В іменах виділених для архівування файлів і папок не повинно бути літер українського алфавіту **і**, **ї**, **є**.



Дайте відповіді на запитання

1. Для чого використовують стиснення даних?

2. У яких випадках можливе використання стиснення із частковою втратою даних?

3. Для чого використовують архівування даних?

4. Які формати архівних файлів можуть опрацьовуватися в операційній системі **Windows** як звичайні папки? Як можна створити файл архіву з використанням контекстного меню файлів і папок засобами операційної системи?

5. Які засоби створення архівів даних користувача та резервного копіювання даних має операційна система **Windows**?

6. Як уключити до переліку папок для резервного копіювання одну з папок, наприклад папку **Foto** з диска **D:**?

7. Як виключити з переліку папок для резервного копіювання одну з папок, наприклад папку **Різне** з папки **Документи**?



Обговоріть і зробіть висновки

1. Що таке *архівування* і що таке *стиснення файлів*? Який між ними зв'язок і в чому відмінність?

2. З якою метою користувач створює резервні копії власних файлів? Чи є у вас потреба створювати резервні копії ваших папок і файлів?

3. Як перевірити, чи увімкнено захист системи для диска **C:** вашого комп'ютера?



Виконайте завдання

1. Є файли типів: **7z, webp, docx, pptx, gif, pdf, psd, mov, png.** Використовуючи дані з відкритих джерел, дослідіть:



а) файли яких з наведених типів створено з використанням стиснення даних;

б) файли яких з наведених типів створено з використанням стиснення без втрат даних, а які – із частковою втратою даних;

в) укажіть для відповідних типів файлів назву алгоритму, що використовується для стиснення даних.

Результати досліджень оформте у вигляді таблиці в текстовому документі та збережіть у вашій папці у файлі з іменем **завдання 2.1.1.**



2. Створіть в одному з відомих вам растрових графічних редакторів копію картини відомого художника **Казимира Малевича «Червоний квадрат».** Збережіть створений малюнок у чотирьох різних форматах: **bmp, jpg, png, gif.** Дослідіть:

а) файл якого із цих форматів має найбільший розмір? Найменший?

б) чи відрізняється якість зображення у цих форматах під час перегляду їх з використанням убудованого переглядача зображень **Windows** (наприклад, **Фотографії**)? Якщо відрізняється, то, на вашу думку, чому?

Результати досліджень оформте у текстовому документі та збережіть у вашій папці у файлі з іменем **завдання 2.1.2.**



3. Проведіть дослідження на тему «Використання стиснення даних без втрат у комп'ютерних системах». Результати досліджень оформте у вигляді презентації (обсяг – 5–10 слайдів) і збережіть у вашій папці у файлі з іменем **завдання 2.1.3.** Ознайомте з результатами вашого дослідження однокласників / однокласниць на уроці інформатики.

4. Створіть резервну копію ваших файлів за наведеним у пункті алгоритмом створення резервних копій файлів. У цю копію включіть копію папки **Робочий стіл** і копію папки **Документи.** Ще одну папку додайте на ваш розсуд. Створіть у текстовому документі звіт про ваші дії під час створення резервної копії файлів з додаванням скриншотів відповідних вікон і збережіть у файлі з іменем **завдання 2.1.4** у папці **Документи.** Надішліть цей файл учителю / учительці на електронну пошту або в **Google Клас.**

5. Створіть з використанням засобів **Провідника** архівний файл (стиснуту папку) перших 10 файлів з папки **Документи** (або з папки **Розділ 2\Пункт 2.1\Природа**). Перемістіть цей файл у вашу папку. Перемістіть з архівного файлу у вашу папку перші два файли.



Готуємось до вивчення нового матеріалу

1. Які операції над об'єктами операційної системи можна виконати?

2. Що таке *архів*?

3. У яких випадках варто робити копії своїх даних?

2.2. АРХІВАТОРИ. ОПЕРАЦІЇ НАД АРХІВАМИ ФАЙЛІВ

У цьому пункті йтиметься про:

- програми для створення архівів файлів і папок;
- типи файлових архівів;
- операції над файлами архівів: створення, додавання, видобування, переміщення;
- засоби операційної системи для роботи з архівами.

АРХІВАТОРИ ТА ТИПИ АРХІВІВ

Пригадайте

- Чи здійснювали ви завантаження групи файлів з **Google Диска**? Який результат завантаження групи файлів? • Для яких файлів не бажано застосовувати стиснення з частковою втратою даних?

Крім засобів операційної системи, існують спеціальні службові програми для архівування даних. Програми, які використовують для виконання операцій над архівами, називаються **архіваторами**. У цих програмах використовується стиснення без втрати даних.

Прикладами архіваторів є програми **WinZIP**, **7-Zip**, **PowerArchiver**, **PeaZip**, **IZArc**, **Bandizip**, **TUGZip**, **HaoZip** та інші. У результаті архівування з використанням цих програм створюється архівний файл, який містить у стисненому або нестисненому стані файли та папки. У процесі архівування можуть бути використані додаткові заходи щодо захисту даних від несанкціонованого доступу, наприклад установлення пароля на доступ до даних в архіві.

Залежно від алгоритмів, за якими здійснюється стиснення та архівування даних, розрізняють такі формати архівних файлів: **ZIP**, **7z**, **ACE**, **ARJ**, **CAB**, **LZH**, **GZIP**, **RAR** та інші (мал. 2.10). Найчастіше, особливо в мережі «Інтернет», використовують архівні файли формату **ZIP**. Наприклад, завантажена група файлів або папка з **Google Диска** будуть поміщені в архівний файл формату **ZIP**.



Мал. 2.10. Значки архівних файлів

ОПЕРАЦІЇ НАД АРХІВАМИ

Пригадайте

- Які операції над файлами можна виконати в операційній системі?
- Які команди містить контекстне меню об'єкта операційної системи?

До основних операцій над архівами належать:

- **архівування** – створення архівів файлів і папок з можливим стисненням даних;
- додавання файлів і папок до вже наявних архівів;
- перегляд умісту архівів;
- заміщення та оновлення файлів і папок в архівах;

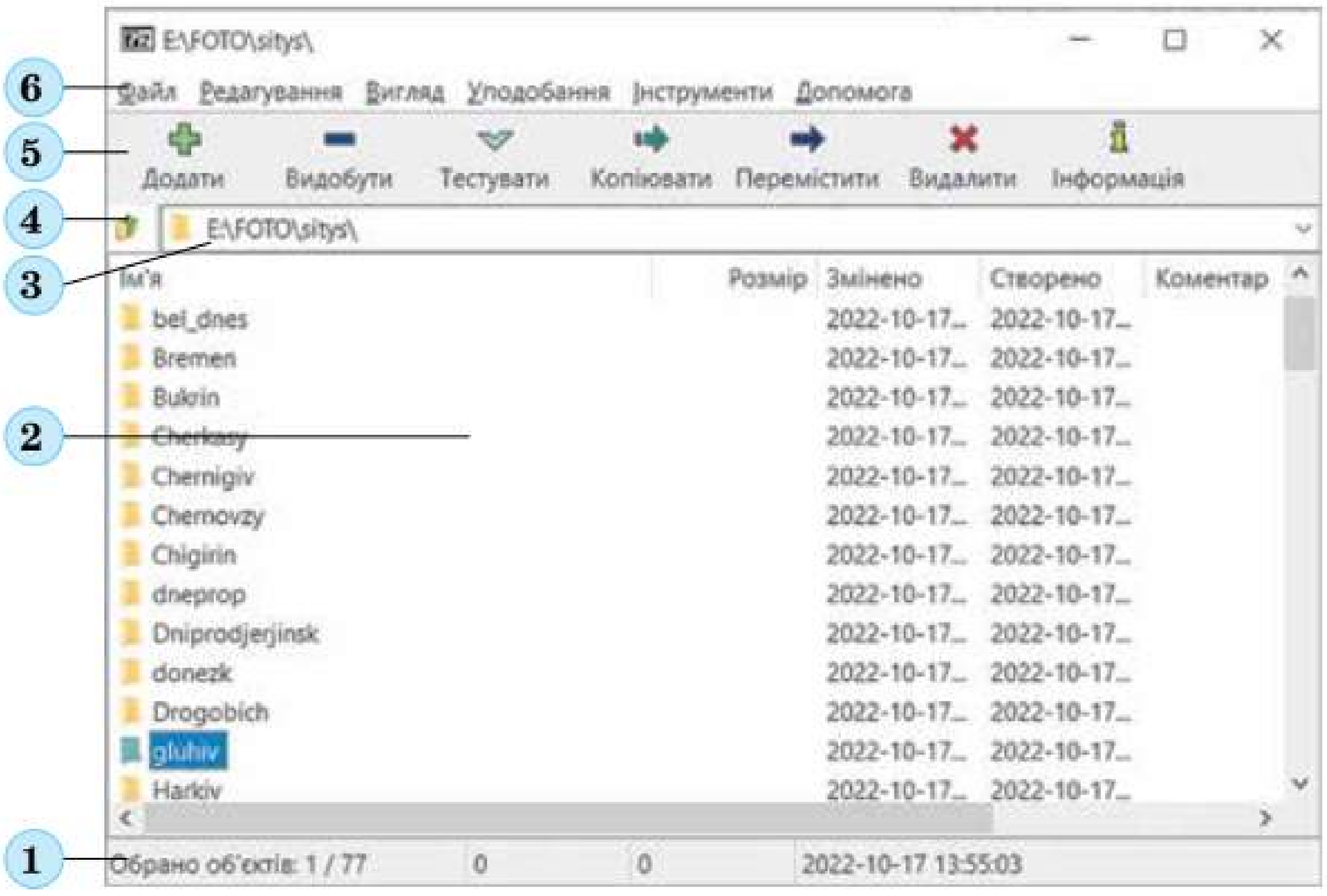
- **розархівування** – видобування з архіву всіх або тільки обраних файлів і папок;
- створення багатотомних архівів (архів розбивається на кілька окремих файлів – томів); розмір томів установлює користувач;
- створення архівів, які містять модуль видобування файлів і папок й не потребують для видобування наявності архіватора, – так званих **SFX**-архівів (англ. ***SelF eXtracting*** – самовидобування);
- перевірка цілісності даних в архівах;
- шифрування даних та імен файлів в архівах

та інші.

Розглянемо послідовність здійснення операцій над архівами з використанням програми **7-zip**. Ця програма належить до програм з ліцензією **free software** (англ. *free software* – вільне програмне забезпечення) і може бути вільно завантажена із сайту розробників програми (*https://7-zip.org*). З її використанням можна:

- створювати архівні файли та видобувати файли форматів **7z**, **TAR**, **ZIP**, **WIM**;
- переглядати та видобувати файли, крім уже зазначених, понад 30 форматів, наприклад **ARJ**, **CAB**, **CHM**, **EXT**, **FAT**, **ISO**, **LZH**, **MSI**, **NTFS**, **RAR**, **XAR**, **Z**;
- створювати у форматі **7z** архівні файли з функцією саморозпакування (самовидобування) з розширенням імені файлу **exe**;
- створювати багатотомні архіви.

Існують версії цієї програми для ОС сімейства **Windows**, **Linux**, **Unix**, а також для **MacOS (7zX)** та **Android**. Вікно програми має вигляд, наведений на малюнку 2.11.

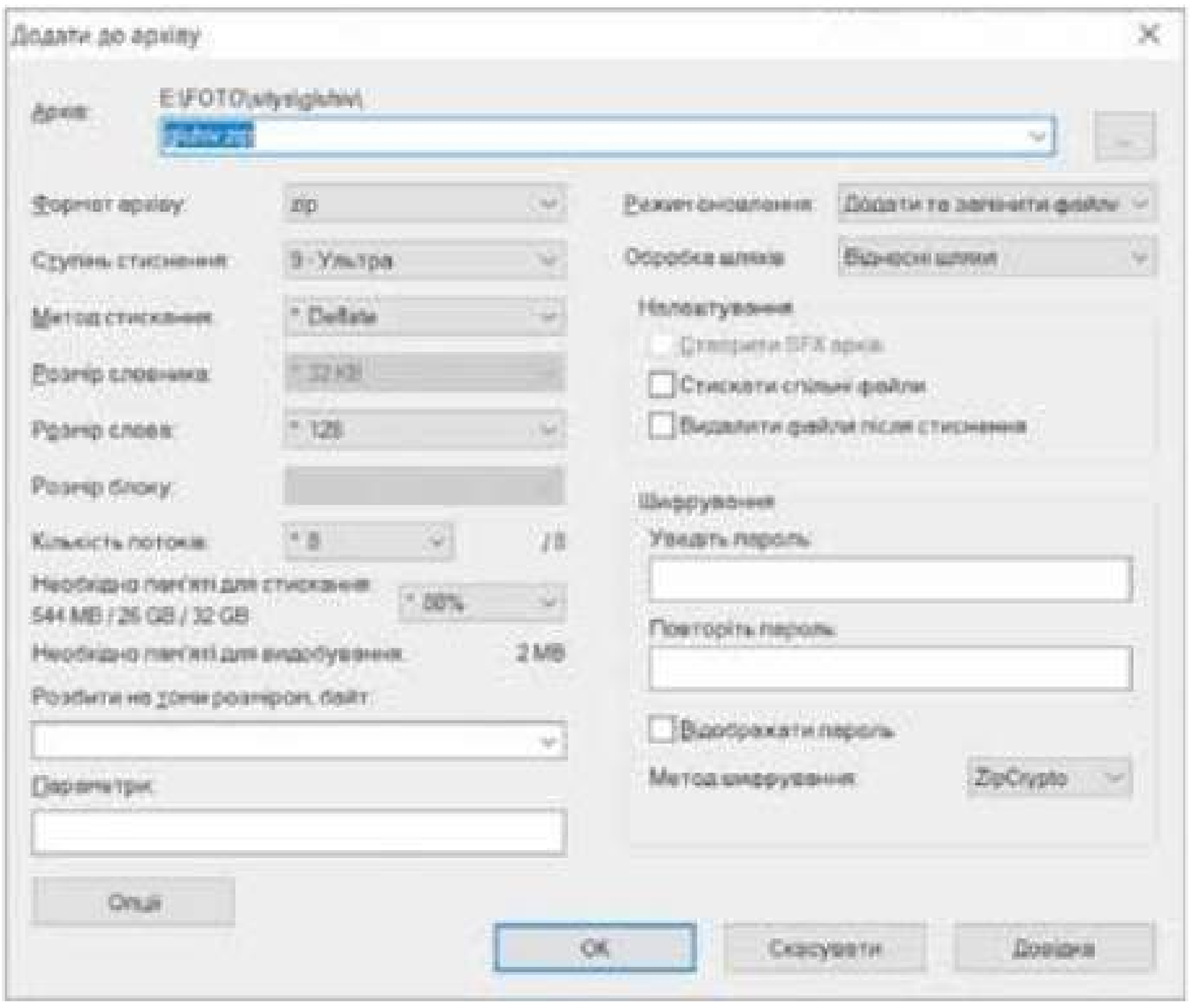


Мал. 2.11. Вікно програми **7-zip**:
1 – **Рядок стану**; *2* – **Робоче поле**; *3* – **Рядок адреси**; *4* – кнопка переходу до папки вищого рівня; *5* – **Панель інструментів**; *6* – **Рядок меню**

Для створення архівного файлу слід виконати такий алгоритм:

1. Запустити програму **7-zip** (***Пуск*** ⇒ ***7-zip*** ⇒ ***7-zip File Manager***).
2. Зробити поточною папку, у якій розміщено файли та папки, що планується включити до архіву. Для змінення поточної папки використати **Рядок адреси** та кнопку переходу до папки вищого рівня  .
3. Виділити файли та папки, які планується включити до архіву, та вибрати кнопку **Додати** 

.
4. Установити у вікні **Додати до архіву** значення властивостей файлу архіву або залишити без змін установлені за замовчуванням (мал. 2.12):



Мал. 2.12. Вікно **Додати до архіву**

- **Ім'я** – у полі імені файлу;
- **Місце збереження** – використовуючи кнопку **Переглянути** 

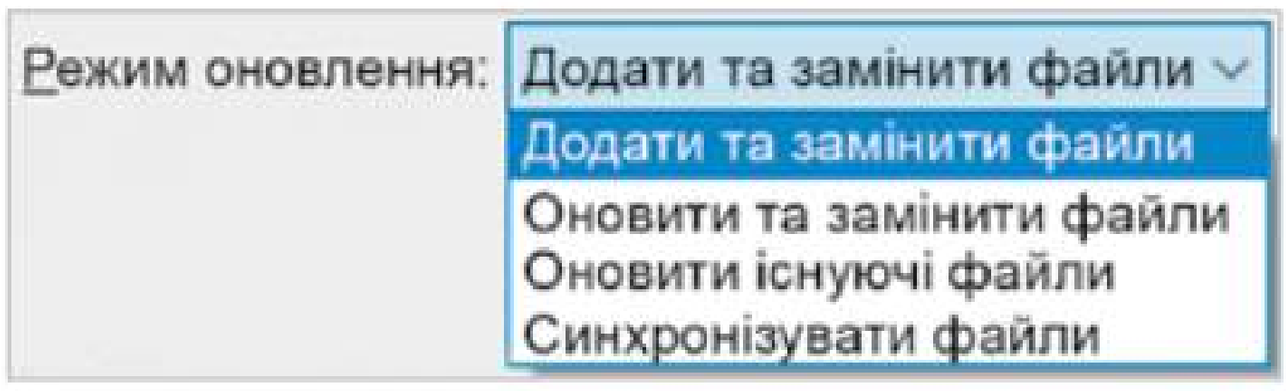
;
- **Формат архіву** – *7z*, *tar*, *wim* або *zip*;
- **Ступінь стиснення** – *без стиснення*, *найшвидше*, *швидке*, *нормальне*, *максимальне*, *ультра* (ступінь стиснення збільшується від *без стиснення* до *ультра*, при цьому зменшується швидкість створення архіву);
- **Розбити на томи розміром, байт** – за потреби записати архівний файл на кілька носіїв, що мають порівняно невеликий розмір;
- **Створити SFX-архів** – для випадків, коли потрібно буде здійснити видобування файлів з архіву без програми-архіватора;
- **Уведіть пароль** і **Повторіть пароль** – для захисту вмісту архіву від доступу сторонніх осіб

та інші.

5. Вибрати кнопку **ОК**.

Для ***додавання файлів до вже наявного архіву*** потрібно під час установлення значень властивостей архіву вказати ім'я та шлях до вже наявного файлу-архіву, використовуючи кнопку **Переглянути** … (мал. 2.12). Під час додавання об'єктів до архіву, що може містити файли з тим самим іменем, але в новій редакції, варто встановити потрібний варіант оновлення файлів у списку **Режим оновлення** (мал. 2.13):

- **Додати та замінити файли** – для додавання нових файлів і заміни існуючих на ті, що додаються;
- **Оновити та замінити файли** – для оновлення файлів у архіві, які створювалися (редагувалися) раніше, ніж ті, що додаються, та додавання нових файлів;
- **Оновити існуючі файли** – для оновлення тільки тих файлів, які вже є в архіві, і лише на новіші;
- **Синхронізувати файли** – для додавання тих файлів, яких немає в архіві, та видалення з архіву тих файлів, яких немає серед обраних для додавання.



Мал. 2.13. Список **Режим оновлення**

Для перегляду вмісту архівного файлу слід відкрити його у вікні **Провідника** або програми-архіватора. Для цього двічі клацнути на імені файлу.

Для ***видобування всіх об'єктів*** з архіву слід:

1. Запустити програму **7-zip**.
2. Зробити поточною папку, у якій розміщено файл архіву, та вибрати його.
3. Вибрати кнопку **Видобути** 

.
4. Указати шлях до папки, у яку буде здійснено видобування, та встановити інші значення властивостей у вікні **Видобути**.
5. Вибрати кнопку **ОК**.

Для ***видобування окремих файлів і папок*** з архіву слід:

1. Запустити програму **7-zip**.
2. Зробити поточною папку, у якій розміщено файл архіву, та вибрати його.
3. Виконати ***Файл*** ⇒ ***Відкрити*** або двічі клацнути на імені файлу.
4. Виділити файли та папки, які потрібно видобути.
5. Вибрати кнопку **Видобути** 

.

6. Указати шлях до папки, у яку буде здійснено видобування, у вікні **Копіювати**.
7. Вибрати кнопку **ОК**.

Для видалення файлів з архіву слід відкрити файл архіву в програмі **7-zip**, виділити потрібні файли та вибрати кнопку **Видалити**



.

Після інсталяції програми-архіватора до контекстного меню файлів і папок додаються команди роботи з архівами. Так, якщо виділеним є файл (файли) архівів (мал. 2.14), то, використовуючи контекстне меню цього файлу (цих файлів), можна:



Мал. 2.14. Контекстне меню архівного файлу

- **Відкрити архів** – відкрити архівний файл у програмі **7-zip** для виконання операцій над архівом;
- **Видобути файли** – відкрити діалогове вікно **Видобути**, у якому потрібно вказати папку для видобування файлів і значення певних властивостей видобування (пароль, особливості перезапису файлів з однаковими іменами тощо);
- **Видобути до поточної папки** та **Видобути до «ім'я файлу архіву»** – видобути всі файли архіву до поточної папки або до папки з іменем, що відповідає імені файлу архіву (ця папка буде створена в поточній папці).

Для інших виділених файлів і папок, використовуючи список команд **7-zip** контекстного меню, можна:

- **Додати до архіву** – відкрити діалогове вікно для додавання виділених файлів і папок до архівного файлу;
- **Додати до «ім'я файлу.7z»** і **Додати до «ім'я файлу.zip»** – створити архіви вказаних форматів та імен зі значеннями властивостей за замовчуванням.

Для перегляду вмісту архівного файлу з модулем самовидобування (**SFX**-архів з розширенням імені файлу **exe**) та здійснення операцій з окремими файлами в ньому слід відкрити цей файл у програмі **7-zip** одним із способів:

- виконати команду **Відкрити архів** зі списку команд **7-zip** контекстного меню файлу;
- у вікні програми **7-zip** виконати ***Файл*** ⇒ ***Відкрити*** та вибрати потрібний файл.

Для тих, хто працює з Android

В операційній системі **Android** опрацювання архівних файлів типу **ZIP** можна здійснювати з використанням стандартної програми **Диспетчер файлів** (інструмент **Zip-файли**). Також користувачі ОС **Android** доволі активно (понад 100 млн завантажень) використовують програму, яку можна вважати певним аналогом програми **7-zip**. Це безкоштовна програма **ZArchiver**, призначена для роботи з архівними файлами різних типів, у тому числі **zip** та **7z**.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/oe2cPUdU* або QR-кодом.

Найважливіше в цьому пункті

Програми, які використовують для виконання операцій над архівами, називаються **архіваторами**. У результаті архівування з використанням цих програм створюється ***архівний файл***, який містить у стисненому або нестисненому стані вибрані файли та папки.

Над архівами виконуються такі операції: *створення* (звичайних, багатотомних і **SFX**-архівів), *перегляд* вмісту, *додавання* нових і *заміна* раніше включених об'єктів, *видалення* об'єктів, *видобування* всіх або окремих об'єктів тощо.

Однією з розповсюджених програм для опрацювання архівів папок і файлів є програма, що вільно поширюється, – **7-zip**.

Дайте відповіді на запитання

1. Як називаються програми, що виконують архівування даних? Які операції над архівами може виконати користувач, використовуючи ці програми?

2. Які формати архівних файлів ви знаєте?

3. Як видобути групу файлів з архівів у певну папку?

4. На що впливає ступінь стиснення файлів у архівах?

5. Для чого використовують багатотомні, саморозпаковувальні архіви?

Обговоріть і зробіть висновки

1. Яка послідовність дій під час створення архіву з кількох файлів, розміщених у різних папках?

2. Яка послідовність дій під час створення архіву, що самороз-паковується? Для яких випадків створюються подібні архіви?

3. Яка послідовність дій під час створення архіву кількох файлів з використанням контекстного меню цих файлів за умови інсталювання архіватора, наприклад **7-zip**?

4. Чому, на вашу думку, архіватори, у тому числі і **7-zip**, не використовують стиснення даних із частковою втратою даних?

Виконайте завдання

1. Запустіть на виконання архіватор. Створіть архів з перших п'яти файлів з розширенням імені **doc**, які розміщені в папці **Архів** папки **Розділ 2\Пункт 2.2**. Помістіть цей файл з іменем **завдання 2.2.1** у вашу папку. Порівняйте розмір початкових п'яти файлів **doc** до архівації та архівного файлу.

2. Видобудьте всі файли з файлу **Архів1.7z** з папки **Розділ 2\Пункт 2.2\Архів** у папку **Мій архів** вашої папки. Порівняйте розмір архіву та видобутих файлів.

3. Додайте до архівного файлу **Архів2.zip**, який розміщено в папці **Розділ 2\Пункт 2.2\Архів**, два останніх файли цієї самої папки. Збережіть змінений архів з тим самим іменем у вашій папці.

4. Запустіть архіватор і:

а) створіть архів з перших п'ятнадцяти файлів з розширенням імені **docx**, що містяться в папці **Документи** (або з п'ятнадцяти останніх файлів з розширенням імені **docx**, що розміщені в папці **Архів** папки **Розділ 2\Пункт 2.2**);

б) установіть під час архівування такі значення параметрів: ім'я архіву – *документи*; папка – *Робочий стіл*; формат архіву – *ZIP*; метод стиснення – *без стиснення*;

в) використовуючи контекстне меню, видобудьте всі файли зі створеного архіву в папку **DOC**, яку створіть у вашій папці.

5. Проведіть дослідження щодо ефективності стиснення одних і тих самих файлів при різних ступенях стиснення у форматі **7z**. Для дослідження використайте всі файли з розширенням імені **bmp**, що розміщені в папці **Розділ 2\Пункт 2.2\Малюнки**. Заповніть таблицю.

№	Ім'я файлу архіву	Ступінь стиснення	Розмір файлів		Коефіцієнт стиснення
			до стиснення	після стиснення	
1	Малюнки1.7z	Без стиснення			
2	Малюнки2.7z	Найшвидше			
3	Малюнки3.7z	Швидке			
4	Малюнки4.7z	Нормальне			
5	Малюнки5.7z	Максимальне			
6	Малюнки6.7z	Ультра			

Визначте, наскільки відрізняються за ефективністю стиснення різні режими. (*Підказка*. Коефіцієнт стиснення визначається як відношення розміру файлу (файлів) до стиснення до розміру файлу (файлів) після стиснення.)

6. Проведіть дослідження щодо коефіцієнта стиснення одних і тих самих файлів при різних алгоритмах (методах) стиснення ступенем стиснення *ультра* у форматі архівів **7z**. Для дослідження ефективності архівування використайте всі файли з розширенням імені **bmp**, які розміщені в папці **Розділ 2\Пункт 2.2\Малюнки**. Заповніть таблицю.

№	Ім'я файлу архіву	Метод стиснення	Розмір файлів		Коефіцієнт стиснення
			до стиснення	після стиснення	
1	Малюнки01.7z	LZMA2			
2	Малюнки02.7z	LZMA			
3	Малюнки03.7z	PPMd			
4	Малюнки04.7z	BZip2			

Визначте, наскільки відрізняються за ефективністю стиснення різні алгоритми стиснення графічних даних. (*Підказка*. Коефіцієнт стиснення визначається як відношення розміру файлу (файлів) до стиснення до розміру файлу (файлів) після стиснення.)

7. Запустіть архіватор і:

а) створіть у вашій папці архів з перших двох файлів, які розміщені в папці **Зразки музики (Розділ 2\Пункт 2.2\Зразки музики)**, установивши при цьому такі значення параметрів архівації:
- ім'я архіву – *Archiv06*;
- формат архіву – *7z*;
- ***SFX-архів***;

б) видаліть зі створеного архівного файлу **Archiv06.exe** останній файл;

в) додайте до цього архіву третій файл з папки **Зразки зображень (Розділ 2\Пункт 2.2\Зразки зображень)**;

г) видобудьте всі файли зі створеного архівного файлу в папку **Музика та малюнки**, яку створіть у вашій папці.

Готуємось до вивчення нового матеріалу

1. Які види комп'ютерів ви знаєте?
2. Назвіть складові частини персонального комп'ютера.
3. Які додаткові пристрої можна під'єднати до персонального комп'ютера?

Оцініть свої знання та вміння

*Оцініть свої навчальні досягнення з розділу «**Архіви даних**» (початковий, середній, достатній, високий рівень).*

- Я розумію, з якою метою використовують архівування даних.

- Я вмію добирати засоби для архівування даних.
- Я розумію, з якою метою в комп'ютерних системах використовують стиснення даних.
- Я маю уявлення про способи стиснення даних під час здійснення інформаційних процесів.
- Я можу пояснити, як кодування використовується для стиснення даних.
- Я вмію добирати формати архівних файлів для стиснення та архівації даних.
- Я виконую операції над архівними файлами з використанням засобів операційної системи та програм-архіваторів.
- Я застосовую захист особистих даних під час їх архівації.

Повторіть той матеріал, який ви знаєте недостатньо.



Практична робота № 2. «Архівування та розархівування даних»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Запустіть на виконання архіватор **7-zip**.
2. Створіть у вашій папці архів з перших шістнадцяти файлів, які розміщені в папці **Розділ 2\Пункт 2.2\Малюнки**, установивши при цьому такі значення параметрів архівації:
   - ім'я архіву – *фотографії*;
   - формат архіву – *7z*;
   - ступінь стиснення – *швидкий*;
   - створити *SFX-архів*;
   - папка для архівного файлу – *ваша папка*.
3. Визначте та запишіть у зошит:
   - загальний розмір файлів до архівації;
   - загальний розмір файлів у архіві після архівації;
   - коефіцієнт стиснення.
4. Виконайте такі дії: створіть у вашій папці папку **Копія архіву**, скопіюйте в неї архів і видаліть з нього останні п'ять файлів.
5. Використовуючи контекстне меню архівного файлу у вікні **Провідника**, видобудьте всі файли з архіву, розміщеного в папці **Копія архіву**, у папку **Мої малюнки**, яку створіть у вашій папці.
6. Використовуючи архіватор, видобудьте перші десять файлів з архіву у вашій папці в папку **1_10**, яку створіть у вашій папці.
7. Закрийте всі відкриті вікна.

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ПЕРСОНАЛЬНОГО КОМП'ЮТЕРА

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- сучасний персональний комп'ютер, його основні складові, перспективні напрями вдосконалення;
- процесор, його основні властивості;
- пристрої пам'яті, їх види та властивості;
- пристрої введення даних, їх види та властивості;
- пристрої виведення даних, їх види та властивості;
- діагностика апаратної та програмної складової комп'ютера, усунення збоїв у їх роботі;
- історія обчислювальних і комп'ютерних пристроїв.

3.1. ПЕРСОНАЛЬНИЙ КОМП'ЮТЕР І ЙОГО ОСНОВНІ СКЛАДОВІ

У цьому пункті йтиметься про:

- види персональних комп'ютерів;
- складові персональних комп'ютерів;
- процесор як основний пристрій опрацювання даних у комп'ютері, його властивості;
- материнську (системну) плату;
- носії даних і їх різновиди;
- пристрої пам'яті комп'ютера, їх основні властивості.

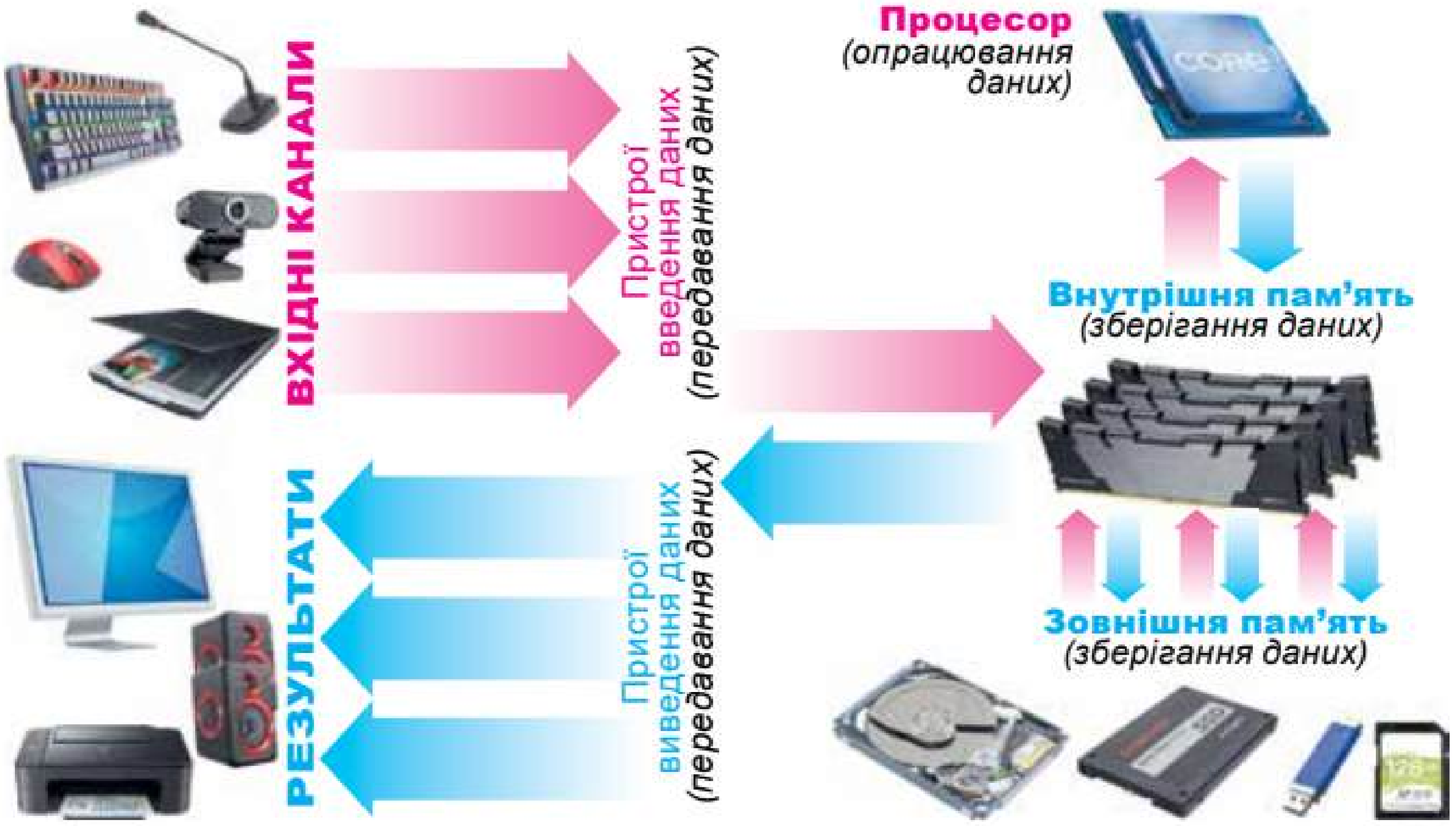
КОМП'ЮТЕР, ЙОГО СКЛАДОВІ ТА ІНФОРМАЦІЙНІ ПРОЦЕСИ

Пригадайте

- Які пристрої входять до складу комп'ютера? Для чого вони призначені?
- Які інформаційні процеси ви знаєте?

У п'ятому класі на уроках інформатики ви вже ознайомилися зі складовими персонального комп'ютера. Ці пристрої реалізують основні інформаційні процеси: передавання, зберігання та опрацювання даних. Розглянемо детальніше схему реалізації інформаційних процесів у комп'ютері з використанням його пристроїв (мал. 3.1). Дані потрапляють до внутрішніх пристроїв пам'яті (*внутрішньої пам'яті*) комп'ютера через пристрої введення даних (клавіатура, графічний планшет, мікрофон, миша, сканер тощо) або із зовнішніх пристроїв пам'яті (*зовнішньої пам'яті*) – пристроїв для роботи з жорсткими магнітними та оптичними дисками, флешнакопичувачів, флешкарт тощо.

З внутрішньої пам'яті дані потрапляють до процесора, у якому відбувається їх опрацювання. Результати опрацювання даних знову передаються до внутрішньої пам'яті, а звідти – до пристроїв виведення



Мал. 3.1. Схема реалізації інформаційних процесів у персональному комп'ютері

даних (монітор, принтер, планшет, звукові колонки тощо) або до пристроїв зовнішньої пам'яті.

Усі операції в комп'ютері, пов'язані з реалізацією інформаційних процесів, можливі тільки під керуванням відповідних програм.

ПРОЦЕСОР, ЙОГО ПРИЗНАЧЕННЯ, ОСНОВНІ ВЛАСТИВОСТІ



Пригадайте

• Який пристрій комп'ютера виконує опрацювання даних? Де зазвичай він розміщується? • Які властивості пристрою опрацювання даних ви знаєте?

Ви вже знаєте, що опрацювання даних у комп'ютері виконується процесором (мал. 3.2). Процесор комп'ютера є його основною складовою. Він є пристроєм, який забезпечує виконання комп'ютерних програм.

Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерних програм, та **арифметично-логічний пристрій**, який здійснює операції над даними. До складу сучасних процесорів входить ще й складова внутрішньої пам'яті – так звана **кеш-пам'ять**. Її призначення розглянемо дещо пізніше.

Властивостями процесора є: **тактова частота** роботи процесора, **кількість ядер, розрядність, обсяг кеш-пам'яті** тощо (табл. 3.1). Значення цих та інших властивостей процесора визначають швидкість опрацювання даних процесором.



Мал. 3.2. Процесор **AMD Ryzen 7**

Таблиця 3.1

Приклади значень властивостей процесора для настільного і планшетного комп'ютерів

Властивість	***Що характеризує***	***Значення властивостей процесора***	
		для настільного комп'ютера	***для планшетного комп'ютера***
Тактова частота	Частота керуючих сигналів, які узгоджують роботу пристроїв процесора	4,3 ГГц	4,3 ГГц
Кількість ядер	Кількість однакових за структурою процесорів, які об'єднано в один пристрій	64	24
Розрядність	Кількість двійкових розрядів у коді даних, які може одночасно опрацьовувати процесор	64 біти	64 біти

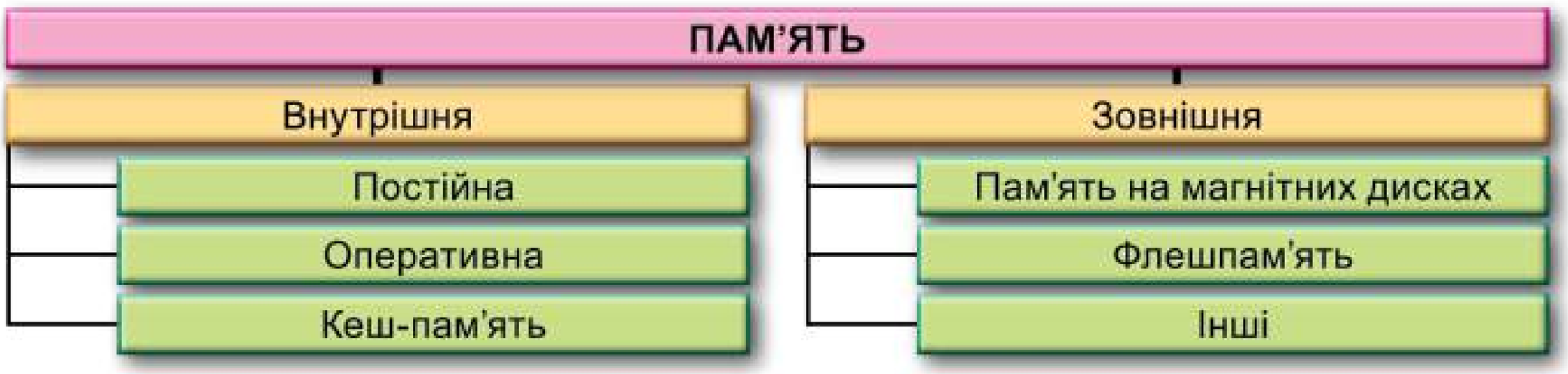
ПАМ'ЯТЬ КОМП'ЮТЕРА. ВНУТРІШНЯ ПАМ'ЯТЬ



Пригадайте

● Які пристрої комп'ютера використовують для зберігання даних? Де зазвичай вони розміщуються? ● Які одиниці вимірювання використовують для визначення обсягу пристрою збереження даних?

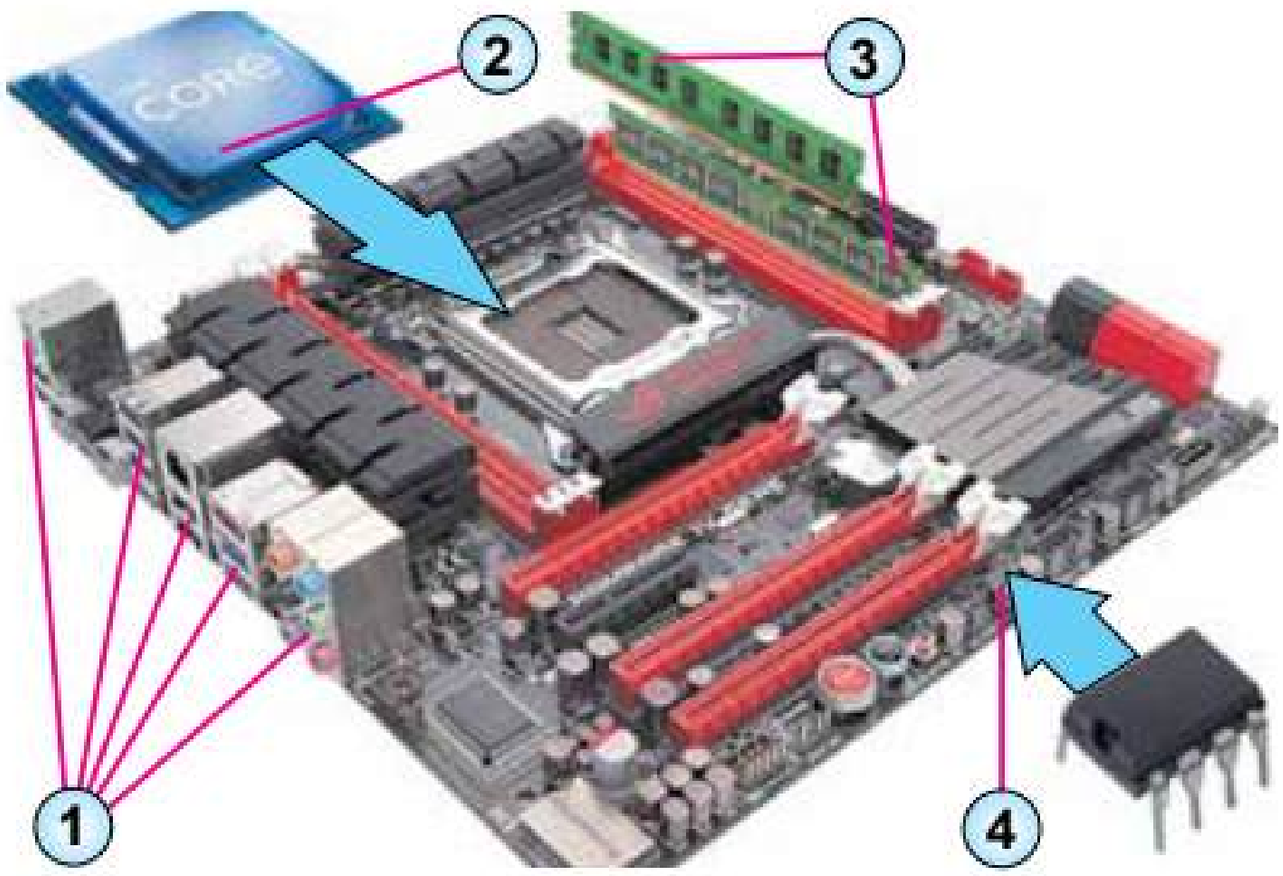
Пам'ять комп'ютера призначена для зберігання даних. Її поділяють на **внутрішню** та **зовнішню** (мал. 3.3). Основною властивістю всіх видів пам'яті є їх ємність, яка вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.



Мал. 3.3. Види пам'яті комп'ютера

Внутрішня пам'ять призначена для забезпечення роботи процесора. Її складові (*оперативна пам'ять*, *постійна пам'ять*, *кеш-пам'ять*), як і процесор, розміщуються на **материнській платі** комп'ютера (мал. 3.4). **Материнська** (англ. *motherboard*), або **системна**, плата – основна складова сучасних персональних комп'ютерів, яка забезпечує передавання даних між пристроями комп'ютера.

Оперативна пам'ять є основною в комп'ютері, у ній розміщуються програми і дані, які в подальшому опрацьовуються процесором. Щоб



Мал. 3.4. Материнська плата комп'ютера:
1 – роз'єми для підключення пристроїв уведення та виведення даних;
2 – процесор; *3* – блоки мікросхем оперативної пам'яті;
4 – мікросхема постійної пам'яті

дані були опрацьовані процесором, вони попередньо повинні бути передані («завантажені») з пристроїв уведення даних або пристроїв зовнішньої пам'яті до оперативної пам'яті.

Після вимкнення живлення комп'ютера всі дані з оперативної пам'яті зникають.

Оперативна пам'ять виготовляється у вигляді плати блоків мікросхем, яка встановлюється в спеціальні роз'єми – **слоти** (англ. *slot* – щілина, паз, отвір) на материнській платі (мал. 3.4, 3). Зазвичай на материнській платі можна встановити кілька плат з мікросхемами оперативної пам'яті (мал. 3.5).



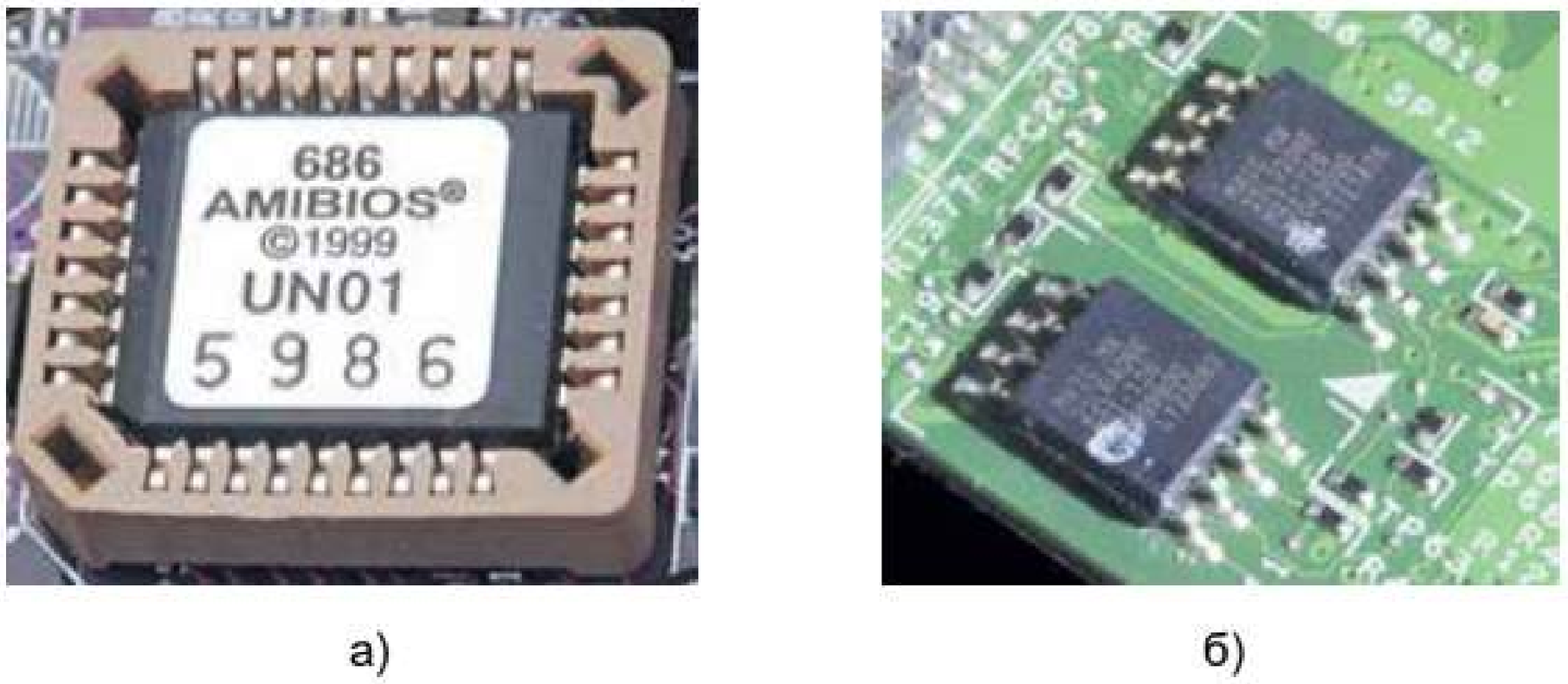
Мал. 3.5. Блоки мікросхем оперативної пам'яті

Як уже зазначалося, до складу сучасних процесорів включають кеш-пам'ять. Вона призначена для прискорення обміну даними між оперативною пам'яттю та процесором. Розрізняють кеш-пам'ять першого, другого і третього рівнів. Ємність кеш-пам'яті першого рівня в більшості процесорів – 128 кБ. Ємності кеш-пам'яті другого і третього рівнів суттєво відрізняються в різних процесорах. На кінець 2023 року кеш-пам'ять другого рівня у процесорах для настільних персональних комп'ютерів становила від 1 до 32 МБ, а третього – від 2 до 256 МБ. Збільшення ємності кеш-пам'яті зазвичай збільшує швидкодію процесора.

Програми та дані, які потрібні для виконання дій, пов'язаних з початком роботи комп'ютера, зберігаються в **постійній пам'яті.** Після ввімкнення комп'ютера дані з постійної пам'яті копіюються («завантажуються») в оперативну пам'ять, і запускаються програми підготовки комп'ютера до роботи.

Постійна пам'ять виготовляється у вигляді однієї або двох мікросхем, що вставляються у спеціальні гнізда – сокети (англ. *socket* – за-

глиблення, гніздо, розетка) (мал. 3.6, а) або впаюються на материнській платі (мал. 3.6, б).



а) б)

Мал. 3.6. Мікросхеми постійної пам'яті на материнській платі

ЗОВНІШНЯ ПАМ'ЯТЬ

Пригадайте

● Які пристрої пам'яті використовують для довготривалого зберігання даних у комп'ютері? ● Які пристрої пам'яті використовують для перенесення даних на інший комп'ютер, який не підключений до комп'ютерних мереж?

Зовнішня пам'ять призначена для довготривалого зберігання даних. Від внутрішньої пам'яті її відрізняє не тільки призначення, але й відмінності у значеннях окремих властивостей:

- збільшений обсяг даних (у персональних комп'ютерах – до 25 ТБ);
- збереження даних після вимкнення комп'ютера (енергонезалежність);
- можливість приєднання окремих пристроїв зовнішньої пам'яті без вимкнення комп'ютера

тощо.

Основними носіями даних зовнішньої пам'яті є електронні схеми та жорсткі магнітні диски. Раніше у пристроях зовнішньої пам'яті як носії даних активно використовували гнучкі магнітні диски, оптичні диски та магнітні стрічки. Носії даних розміщуються у відповідних пристроях.

Зовнішню пам'ять на електронних мікросхемах ще називають *флешпам'яттю*. Вона реалізується у пристроях пам'яті трьох основних типів:

- **твердотілий**, або **SSD**-накопичувач (англ. ***Solid-State Drive*** – твердотілий пристрій);
- **USB**-флешнакопичувач («флешка»);
- **флешкарта** (карта пам'яті).

Твердотілі, або SSD-накопичувачі (мал. 3.7), мають таке саме призначення, як і накопичувачі на жорстких магнітних дисках (НЖМД). Здебільшого їх використовують як основні пристрої зовнішньої пам'яті для мобільних комп'ютерів –



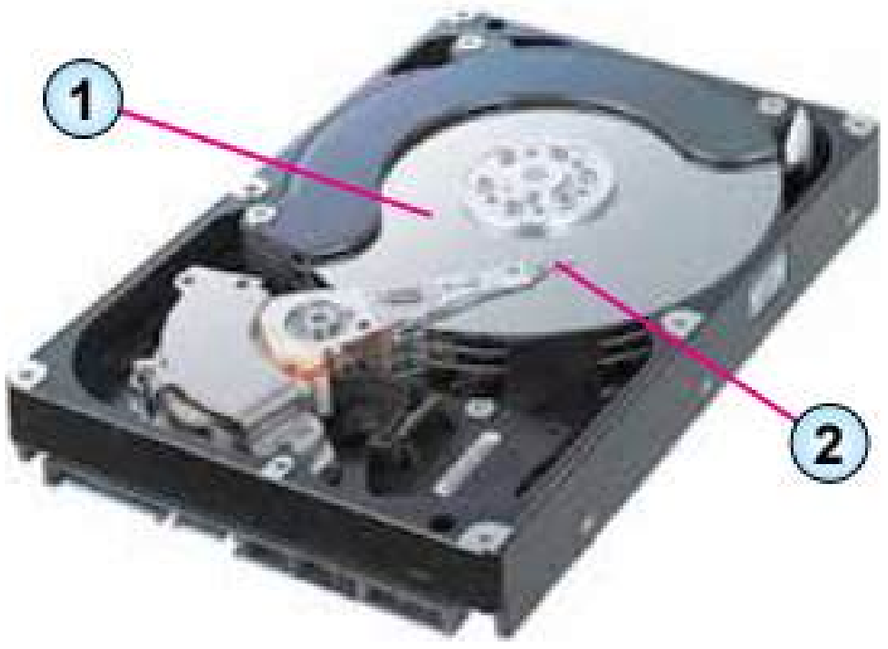
Мал. 3.7. SSD-накопичувач

ноутбуків, планшетних комп'ютерів. В останні роки SSD-накопичувачі почали встановлювати й на стаціонарні комп'ютери як системні – для пришвидшення роботи операційної системи під час здійснення операцій з дисками.

Порівняно з НЖМД SSD-накопичувачі мають більшу швидкість обміну даних, але й більшу вартість за однакової ємності. За максимальною ємністю SSD-накопичувачі практично зрівнялися з накопичувачами на жорстких магнітних дисках – 15 ТБ.

Жорсткі магнітні диски є носіями даних у пристроях, які називають **накопичувачі на жорстких магнітних дисках.** Ці пристрої зовнішньої пам'яті можуть мати один або кілька металевих дисків, покритих шаром магнітної речовини (мал. 3.8). Запис даних виконується шляхом намагнічування ділянок поверхні диска з використанням електромагнітної головки, яка виконує і зчитування даних.

Властивості накопичувачів на жорстких магнітних дисках і приклади їх значень подано в таблиці 3.2.



Мал. 3.8. Будова накопичувача на жорстких магнітних дисках: *1* – поверхня одного з жорстких дисків; *2* – електромагнітна головка

Таблиця 3.2

Значення властивостей сучасних НЖМД

Властивість	*Значення властивості НЖМД для*	
	настільних комп'ютерів	**ноутбуків**
Ємність	2–28 ТБ	0,5–5 ТБ
Діаметр жорстких дисків	3,5 дюйма	2,5 дюйма
Швидкість обертання жорстких дисків	5400–15 000 об/хв	5400 –7200 об/хв

Для підвищення швидкодії комп'ютера за рахунок збільшення швидкості обміну даними значна частина персональних комп'ютерів оснащується двома пристроями зовнішньої пам'яті – більш «швидким» **SSD-накопичувачем** (для розміщення програм, які дуже часто використовуються комп'ютером, наприклад операційної системи) і значно «повільнішим» накопичувачем на жорстких магнітних дисках для забезпечення потрібної ємності зовнішньої пам'яті.

Для створення резервних копій даних, банку конфіденційних даних значного обсягу, переміщення великих обсягів даних без використання комп'ютерних мереж та інших цілей можуть використовуватися так звані зовнішні накопичувачі на жорстких магнітних дисках або на **SSD**-накопичувачах. На відміну від пристроїв, які підключаються до материнської плати через відповідні роз'єми і розміщуються в системному блоці або корпусі мобільного комп'ютера, зовнішні при-

строї підключаються до комп'ютера з використанням USB-роз'ємів (мал. 3.9).



Мал. 3.9. Підключення зовнішнього накопичувача



Мал. 3.10. Флешкарти різних типів

USB-флешнакопичувачі використовуються в основному для обміну даними між різними комп'ютерами. Для їх підключення до комп'ютера використовують USB-роз'єми різних модифікацій (USB Type-A, USB Type-C, Micro-USB). Більшість USB-флешнакопичувачів мають ємність від 4 до 128 ГБ, хоча є пристрої і значно більшої ємності – до 2 ТБ.

Різноманітні флешкарти (мал. 3.10) здебільшого слугують як пристрої пам'яті для смартфонів, фото- та відеокамер тощо. Для їх підключення використовують залежно від типу флешкарти відповідні слоти. Найпопулярнішими на сьогодні у зв'язку з їх широким використанням у смартфонах є флешкарти типу **microSD**.

Ємність флешкарт коливається в межах від 4 до 512 ГБ.

Цікаві факти з історії

Один з перших накопичувачів на жорстких магнітних дисках розробила корпорація IBM, і в 1956 році його було представлено для продажу. Це був один з перших пристроїв з довільним доступом до даних. Пристрій отримав назву **IBM 350 Disk File**, мав обсяг «цілих» 5 МБ і складався з 52-х металевих дисків, укритих шаром магнітної речовини. Запис здійснювався на 100 поверхнях для запису, на кожній поверхні розмічалося 100 доріжок. Диски оберталися зі швидкістю 1200 об/хв. Діаметр дисків – 24 дюйми, або 60,96 см, а маса всього пристрою перевищувала тонну (мал. 3.11).



Мал. 3.11. Завантаження **IBM 350 Disk File** у багажний відсік літака

Цікаві факти з історії

Визначним ученим, який зробив суттєвий внесок у розробку технологій створення сучасних накопичувачів на магнітних дисках, є **Любомир Романків** (нар. 17.04.1931, м. Жовква, Львівська обл. – помер 27.06.2024, США), американський учений українського походження, у минулому – провідний співробітник корпорації IBM. Він розробив технологію створення головок записування / зчитування даних з поверхні магнітних дисків, що значно зменшило витрати на виготовлення НЖМД. Ця технологія уможливила виготовлення пристроїв, що могли розміститися в корпусах персональних комп'ютерів. Л. Романків є співавтором понад 60 патентів.



Мал. 3.12. Любомир Романків

Для тих, хто хоче знати більше

Тривалий час як носії даних використовували оптичні диски, які відрізнялися ємністю та можливістю здійснювати запис і перезапис даних. Основні типи оптичних дисків, їх властивості та значення властивостей подано в таблиці 3.3.

Таблиця 3.3

Основні типи оптичних дисків

Позначення дисків, призначених для			*Ємність дисків*
тільки зчитування даних	*зчитування і запису даних*	*зчитування, запису та перезапису даних*	
CD-ROM	CD-R	CD-RW	640–800 МБ
DVD-ROM	DVD-R, DVD+R	DVD-RW DVD+RW	9,4 ГБ
BD-ROM	BD-R	BD-RE	100 ГБ

У пристроях для роботи з оптичними дисками використовують промінь лазера для зчитування та запису даних.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/se2cP4NZ* або QR-кодом.

Найважливіше в цьому пункті

Обов'язковими компонентами персонального комп'ютера є пристрої, які використовуються для введення, виведення, опрацювання та зберігання даних.

Процесор – це основний пристрій, який здійснює опрацювання даних. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметично-логічний пристрій**, який здійснює операції над даними. Властивостями процесора є: ***тактова частота*** роботи процесора, ***кількість ядер***, ***розрядність***, ***обсяг кеш-пам'яті*** тощо. Значення цих та інших властивостей визначають швидкість опрацювання даних процесором.

Пам'ять комп'ютера призначена для зберігання даних, її поділяють на **внутрішню** та **зовнішню**.

Пристрої внутрішньої пам'яті (оперативна, постійна, кеш-пам'ять) розміщуються на материнській платі. Дані з оперативної пам'яті та кеш-пам'яті після вимкнення живлення зникають, а з постійної – ні.

Пристрої зовнішньої пам'яті (накопичувачі на жорстких магнітних дисках, SSD-накопичувачі та інші пристрої, що використовують флешпам'ять) призначені для довготривалого зберігання даних. Після вимкнення живлення дані з носіїв зовнішньої пам'яті не зникають.

Основною властивістю носіїв усіх видів пам'яті є їх ємність, яка вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.



Дайте відповіді на запитання

1. Як реалізуються інформаційні процеси у комп'ютері за малюнком 3.1?

2. Для чого призначено процесор? Назвіть пристрої, які входять до складу процесора.

3. Який пристрій пам'яті включено до складу сучасних процесорів?

4. Які основні властивості процесора ви знаєте? Наведіть приклади їх значень.

5. На які складові можна поділити пам'ять? Яка основна властивість пристроїв пам'яті? Які одиниці вимірювання значень цієї властивості?

6. Для чого призначена оперативна пам'ять? Для чого призначена постійна пам'ять комп'ютера? Де розміщуються ці види пам'яті?

7. Які пристрої належать до пристроїв зовнішньої пам'яті?



Обговоріть і зробіть висновки

1. У чому відмінність постійної пам'яті від оперативної? Чи є можливість залишити в комп'ютері тільки оперативну пам'ять або тільки постійну?

2. Для чого призначена кеш-пам'ять? Наведіть аналогію роботи кеш-пам'яті до відомих вам навчальних або виробничих процесів.

3. Що спільного і в чому відмінності між пристроями зовнішньої пам'яті? Які з них, на вашу думку, стануть у найближчі 5 років основними у стаціонарних ПК? У мобільних пристроях?



Виконайте завдання

1. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад *https://ek.ua/ua/*, запишіть значення властивостей трьох різних процесорів, які надійшли останнім часом у продаж у магазини України. Створіть у текстовому процесорі таблицю та заповніть її:

Властивість	*Значення властивості процесора*		
Модель			
Тактова частота			
Кількість ядер			
Розрядність			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.1.1.**

2. Дослідіть, використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад *https://ek.ua/ua/*, значення властивостей оперативної пам'яті з максимальною ємністю, що пропонуються для ноутбуків. Створіть у текстовому процесорі таблицю та заповніть її:

Властивість	***Значення властивості оперативної пам'яті***		
Модель			
Виробник			
Ємність			
Вартість			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.1.2.**

3. Визначте значення властивостей процесора та пристроїв пам'яті вашого домашнього комп'ютера. Створіть у текстовому процесорі таблиці результатів аналогічно до таблиць з рубрики **Працюємо з комп'ютером** і заповніть їх. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.1.3.**

4. За відомостями з інтернету або каталогами комп'ютерних магазинів визначте значення властивостей процесорів, які доступні на ринку комп'ютерів вашого регіону. Створіть у текстовому процесорі таблицю, аналогічну до таблиці 3.1, та заповніть її. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.1.4.**

5. За відомостями з інтернету або каталогами комп'ютерних магазинів порівняйте ємність кеш-пам'яті другого рівня процесорів з однаковою тактовою частотою та кількістю ядер від різних виробників. Результати досліджень унесіть до текстового документа, якій збережіть у вашій папці у файлі з іменем **завдання 3.1.5.**

6. Дослідіть, використовуючи матеріали інтернету, чи можливо змінити дані в постійній пам'яті комп'ютера. Якщо так, то опишіть способи, як це зробити для домашнього комп'ютера. Результати досліджень унесіть до текстового документа, якій збережіть у вашій папці у файлі з іменем **завдання 3.1.6.**

Готуємось до вивчення нового матеріалу

1. Які пристрої ви використовували для введення даних у комп'ютер?
2. Які операції можна виконувати з використанням миші?

3.2. ПРИСТРОЇ ВВЕДЕННЯ ДАНИХ

У цьому пункті йтиметься про:

- основні види пристроїв уведення даних, що використовуються в персональних комп'ютерах;
- клавіатури, їх види та основні властивості;
- пристрої для керування роботою програм, їх види та основні властивості;
- пристрої для введення графічних даних, їх види та основні властивості;
- пристрої для введення мультимедійних даних, їх види та основні властивості.

ВИДИ ПРИСТРОЇВ УВЕДЕННЯ ДАНИХ

Пригадайте

● Дані яких видів вам доводилося вводити до комп'ютера? Які пристрої ви для цього використовували? ● За значеннями яких властивостей ви обираєте той чи інший пристрій уведення даних?

Як ви вже знаєте, до складу комп'ютерів уходять пристрої введення даних.

Пристрої введення даних можна розподілити за типом даних, які вони вводять. Так можна виділити пристрої введення даних:

- текстових (**клавіатура**);
- графічних (**сканер, цифрова фотокамера, графічний планшет**);
- звукових (**мікрофон, цифровий диктофон**);
- відеоданих (**відеокамера, вебкамера, ТВ-тюнер**).

Ще однією групою пристроїв уведення даних є пристрої для керування роботою різноманітних програм. До них належать **миша, клавіатура, тачпед, мультимедійна (електронна) дошка, сенсорний екран, джойстик, геймпед** (англ. *gamepad* – ігровий майданчик) (мал. 3.13), **кейпед** (англ. *keypad* – майданчик для клавіш) (мал. 3.14), **кермо, педалі, денспед** (англ. *dancepad* – майданчик для танців) (мал. 3.15) тощо.



Мал. 3.13. Геймпед



Мал. 3.14. Кейпед



Мал. 3.15. Денспед

Окремі пристрої можуть уводити дані різних типів, наприклад вебкамери можуть передавати аудіо- та відеодані.

У комп'ютерах, які використовують для навчальних і наукових цілей, можуть застосовуватися інші види пристроїв для введення даних. Наприклад, датчики для визначення значення температури, вологості повітря, наявності певних домішок у речовинах, швидкості руху різноманітних об'єктів тощо.

Розглянемо властивості, їх значення та класифікацію окремих пристроїв уведення даних.

КЛАВІАТУРА

Поміркуйте

● Значення яких властивостей користувач враховує під час вибору клавіатури для комп'ютера? ● Чим відрізняється клавіатура стаціонарного комп'ютера від клавіатури ноутбука?

Клавіатура призначена для введення текстових даних і команд. Залежно від значень певних властивостей клавіатури можна поділити на кілька груп:

- за призначенням: *стандартні* (для використання у стаціонарних комп'ютерах для введення текстових даних), *компактні* (для використання з мобільними пристроями малих розмірів), *мультимедійні* (для керування діями з мультимедійним контентом безпосередньо з клавіатури), *ігрові* (для керування діями в комп'ютерних іграх) тощо;
- за типом підключення до системного блока: *дротові* (**USB, PS/2**) або *бездротові* (**радіо, Bluetooth, Wi-Fi** тощо).

Проєкційна клавіатура (мал. 3.16) є ефективним пристроєм уведення даних для мобільних пристроїв; вона поєднує зручність уведення даних з клавіатури стандартних розмірів і мобільність – її зображення можна спроєктувати на будь-яку плоску поверхню. В іншому робота з такою клавіатурою нічим не відрізняється від роботи зі стандартною клавіатурою.



Мал. 3.16. Проєкційна клавіатура

Цікаві факти з історії

У місті Норвіч (Велика Британія) у бруківці на вулиці Принцес можна побачити вирізьблену на камені клавіатуру (мал. 3.17). Місцеві жителі та гості міста сушать собі голову над тим, яким чином клавіатура з'явилася на бруківці, якій кілька сотень років. Чи це знак від позаземних цивілізацій, чи жарт студентів місцевого університету мистецтв? У травні 2020 року Пітер Лі розкрив правду про місцеву легенду. Відбиток клавіатури був створений у 90-х роках XX ст. тодішньою студенткою Норвічського університету мистецтв на ім'я Моллі Соул.



Мал. 3.17. Клавіатура на вулиці міста Норвіч

ПРИСТРОЇ КЕРУВАННЯ РОБОТОЮ ПРОГРАМ

Пригадайте

• Яким чином подаються команди в операційній системі з графічним інтерфейсом? • Які пристрої ви використовуєте під час малювання у графічному редакторі?

Маніпулятор **миша** використовують для передавання команд від користувача комп'ютерним програмам.

Розробники комп'ютерних програм передбачають, які дії у програмі відбудуться після натиснення тієї чи іншої кнопки миші, прокручування її коліщатка. Може виконуватися певна команда меню, обиратися об'єкт, збільшуватись або зменшуватись масштаб відображення, відкриватися контекстне меню, вставлятися об'єкт тощо.

Маніпулятори **миша** поділяються залежно від значень таких властивостей:

- за призначенням: *для настільних ПК*, для роботи з *графічними планшетами*, *ігрові* тощо;

- за типом підключення до системного блока: *дротові* (**USB, PS/2**) або *бездротові* (**радіо, Bluetooth, Wi-Fi** тощо);
- за типом датчиків руху: *оптичні, лазерні, гіроскопічні* тощо;
- за кількістю кнопок: *2, 3–5, 6–9, 10 і більше* тощо;
- за типом корпусу: *симетрична стандартна, симетрична для шульги, ергономічна* тощо.

На малюнку 3.18 зображено ігрову дротову лазерну ергономічну мишу, яка має 10 кнопок та одне коліщатко.



Мал. 3.18. Миша

Тачпеди (англ. *touch* – дотик, *pad* – площадка, майданчик, подушечка) є складовими практично всіх типів ноутбуків і нетбуків, проте можуть використовуватись і під час роботи з настільними ПК у вигляді додатково підключеного пристрою (мал. 3.19). Переміщення пальця по поверхні тачпеда приводить до переміщення вказівника на екрані монітора. Одноразовий короткочасний дотик до поверхні тачпеда аналогічний до одноразового клацання лівою кнопкою миші, а подвійний дотик – подвійного клацання лівою кнопкою миші. Ліва і права кнопки тачпеда виконують ті самі функції, що й відповідні кнопки миші.

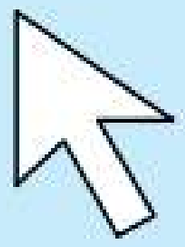


Мал. 3.19. Зовнішній тачпед

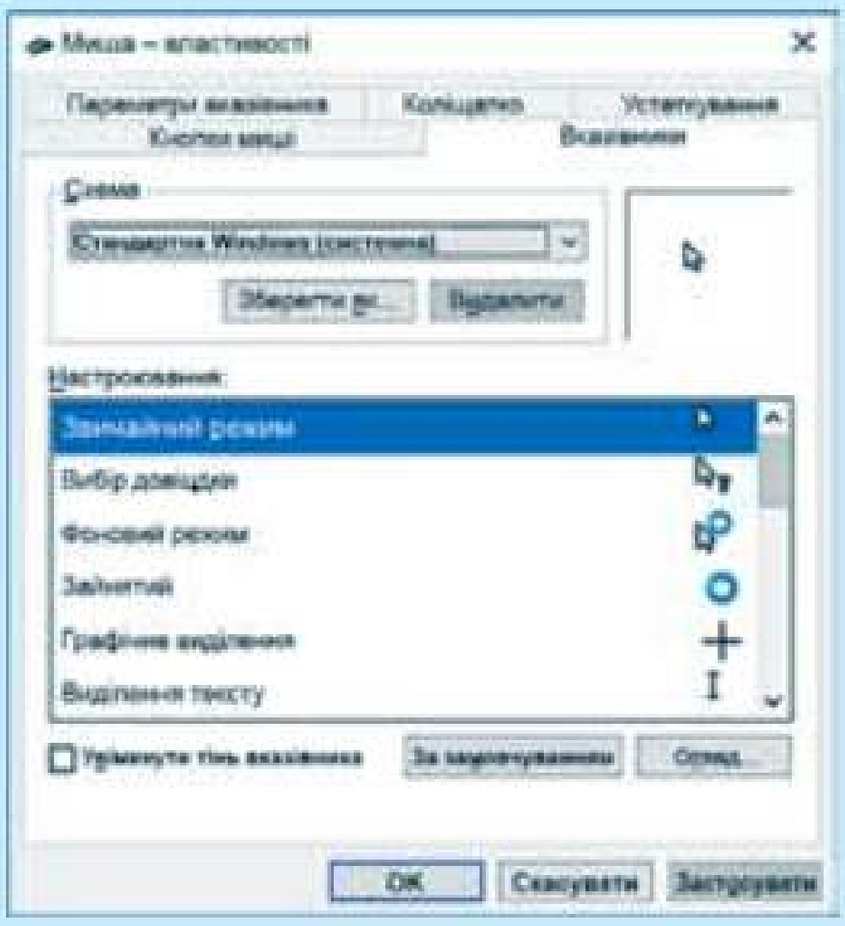
Для тих, хто хоче знати більше

Як змінити вигляд вказівника

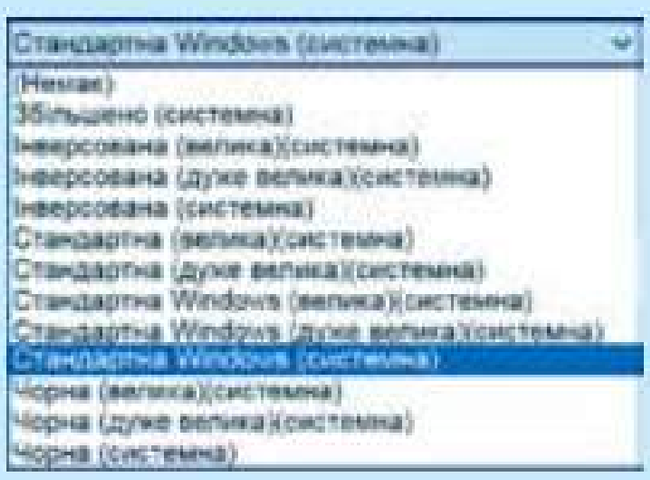
Вказівник на екрані монітора комп'ютера за різних ситуацій має різне зображення. У більшості випадків він стандартно має вигляд стрілки (мал. 3.20). Однак зовнішній вигляд вказівника можна змінити.



Мал. 3.20. Вказівник



Мал. 3.21. Вікно **Миша – властивості**



Мал. 3.22. Список **Схема**

Для цього слід виконати ***Пуск*** ⇒ ***Налаштування*** ⇒ ***Пристрої*** ⇒ ***Миша*** ⇒ ***Додаткові параметри миші***. І у вікні **Миша – властивості** відкрити вкладку **Вказівники** (мал. 3.21). У подальшому є кілька варіантів змінення зовнішнього вигляду вказівника:

- змінити схему оформлення вказівників. Для цього слід:
  1. Відкрити список **Схема** (мал. 3.22).
  2. Вибрати потрібну схему оформлення.
  3. Вибрати кнопку **Застосувати**.
  4. Закрити вікно.
- вибрати для кожного з режимів у списку **Настроювання** інше зображення. Перелік файлів із зображеннями вказівників, що під час інсталяції операційної системи розміщуються в папці **Cursor**, доволі обмежений. Однак користувач зможе знайти в інтернеті великі колекції зображень, наприклад, за ключовими словами «курсори миші». Колекції зображень вказівників – це набори файлів з розширенням імені **cur** чи **ani**. Ці файли бажано скопіювати в папку **Cursor** і для їх використання виконати вищенаведену послідовність дій.

Сенсорні екрани використовують здебільшого у планшетних комп'ютерах, смартфонах, а також у різноманітних довідкових і презентаційних системах, платіжних терміналах. Уведення даних здійснюється дотиком пальця або **стилуса** до певних ділянок поверхні екрана. Екран «відчуває», у якому місці відбувся дотик, і передає відповідний сигнал комп'ютеру. Під час роботи з сенсорним екраном зазвичай використовується змінена, порівняно з тачпедом або мишею, система команд для виклику меню, переміщення об'єкта тощо. Основні властивості – розмір екрана (довжина діагоналі може бути від *3 до 70 дюймів* і більше) і принцип відстеження місця дотику.

Стилус (лат. *stylus* – загострена паличка) – інструмент у вигляді стержня для введення команд з використанням сенсорного екрана.

Мультимедійні (електронні) дошки використовують здебільшого в закладах освіти, а також під час проведення різноманітних презентацій. Розрізняють дошки залежно від їх розмірів (довжина діагоналі *70–79*, *80–89* чи *понад 90 дюймів*) і принципу визначення місця дотику на дошці.

ПРИСТРОЇ ДЛЯ ВВЕДЕННЯ ГРАФІЧНИХ ДАНИХ



Поміркуйте

• Який процес, на вашу думку, називають скануванням? • Як використовують сканування в торгівлі? Для яких цілей?

Сканери (англ. *scanner* – той, що відстежує) використовують для введення графічних даних у комп'ютер. На об'єкт, комп'ютерне зображення якого потрібно отримати, від спеціального пристрою направляється потік світла. Датчики аналізують інтенсивність і колір відбитого світла та перетворюють ці дані в електронну форму подання зображення.

Сканувати можна й текстовий документ, але в результаті буде отримано графічне зображення тексту. Це зображення можна перетворити на текст, використавши спеціальні програми – системи оптичного розпізнавання тексту (англ. **OCR** – ***Optical Character Recognition*** – оптичне розпізнавання символів), наприклад **Freemore OCR, Online OCR, Free Online OCR**. Подібні програми є у складі програмного забезпечення більшості смартфонів.

Сканери поділяють на *ручні*, *настільні* та *проєкційні*.

Ручні сканери (мал. 3.23) мають малі розміри і зручні для введення невеликих за розмірами зображень. Під час сканування об'єкт залишається нерухомим, а переміщують сам сканер. Такі типи сканерів широко використовують у магазинах для сканування штрихкодів, нанесених на товари, у поїздах для сканування QR-кодів на проїзних документах, у бібліотеках тощо.

Останнім часом набули розповсюдження ручні сканери з протяжним механізмом, що використовуються в комплекті з різними мобільними комп'ютерами для отримання якісних відсканованих зображень. Такі сканери мають невеликі розміри та живляться з використанням USB-підключення або від автономної батареї. Зазвичай вони можуть бути підключені до мережі з використанням Wi-Fi-з'єднання. Один з таких сканерів подано на малюнку 3.24.



Мал. 3.23. Ручний сканер



Мал. 3.24. Ручний мобільний протяжний сканер



Мал. 3.25. Планшетний сканер

Серед **настільних** сканерів найпоширеніші **планшетні** (мал. 3.25 і табл. 3.4). У них об'єкт, що сканується, нерухомо розміщується на склі. Сканування відбувається під час автоматичного переміщення механізму зчитування. Для сканування великої кількості сторінок із зображеннями чи текстами використовують сканери з автоматичною подачею сторінок і вбудованою пам'яттю для зберігання файлів відсканованих зображень.

Таблиця 3.4

Основні властивості планшетних сканерів

Властивість	***Що характеризує***	***Одиниці вимірювання***	***Значення властивостей у сучасних сканерах, наприклад у моделі Epson V600 Photo***
Формат	Розмір робочої області для сканування об'єктів	мм (формат)	216 мм на 297 мм (формат А4)
Роздільність	Здатність розпізнавати певну кількість точок на одному дюймі зображення	Точок на один дюйм – **dpi** (англ. ***dot per inch*** – точок на дюйм)	6400 на 9600
Кількість кольорів	Кількість кольорів, які може розпізнати сканер	Одиниці	$2{,}8*10^{14}$ (48 бітів на піксель)

Крім планшетних, є настільні сканери, у яких об'єкт сканування протягується відносно нерухомого механізму зчитування (мал. 3.26).

Проєкційні сканери ще називають безконтактними або документ-сканерами. Їх особливістю є те, що зображення вони отримують, як у фотоапараті, відразу всієї сторінки.

Об'єкт сканування розміщується під модулем сканування (мал. 3.27). Змінюючи відстань від об'єкта до модуля сканування, можна отримувати зображення об'єктів різних розмірів, а не тільки обмежуватись, наприклад, форматом А4. Значна частина таких сканерів має пристрій для підсвічування об'єкта сканування. Такі сканери активно використовують у бібліотеках для створення електронних копій книжок, журналів, текстових документів, фотографій, інших зображень.



Мал. 3.26. Настільний сканер з протяжним механізмом



Мал. 3.27. Проєкційний сканер



Мал. 3.28. Графічний планшет

Близький за принципом дії до сенсорного екрана **графічний планшет.** Він використовується для створення малюнків і введення інших графічних даних. Користувач створює малюнок, дотикаючися стилусом до поверхні графічного планшета. Залежно від сили натискання, наприклад, змінюється товщина лінії. В окремих видах графічних планшетів робоча поверхня не відображає зображення, воно з'являється на екрані комп'ютера, до якого приєднано графічний планшет. В інших видах створене зображення відображається на робочій поверхні самого графічного планшета (мал. 3.28).

Розрізняють графічні планшети за розміром робочої поверхні (від А7 до А3), роздільністю (від 2000 до понад 4000 пікселів на дюйм), а також кількістю варіантів реагування на силу натискання стилусом на робочу поверхню (від 512 до 2048).

ІНШІ ПРИСТРОЇ ВВЕДЕННЯ ДАНИХ



Поміркуйте

- Перегляньте зображення пристроїв комп'ютера (мал. 3.29). Для яких цілей, на вашу думку, використовують ці пристрої?







Мал. 3.29

Як уже було зазначено, для введення відео- та звукових даних використовують **відеокамери** та **вебкамери, мікрофони** та інші пристрої введення мультимедійних даних.

Зазначимо, що основною властивістю відеокамер і вебкамер є їхня роздільність, яка визначає якість зображення. Для відеокамер якість зображення визначається стандартами, наприклад, **SD** (англ. ***Standard Definition*** – стандартна чіткість, 720 × 576 точок), **HD** (англ. ***High Definition*** – висока чіткість, 1280 × 720 точок), **Ultra HD** (англ. ***Ultra High Definition*** – надвисока чіткість, 3840 × 2160 точок).

Для **вебкамер** роздільність визначається такими самими, як і в інших цифрових камерах, значеннями кількості точок по вертикалі та горизонталі, додатковою характеристикою є максимальна кількість кадрів, які може записувати камера за одну секунду. Зазвичай значення цієї властивості перебуває в межах від 15 до 60 кадрів за секунду.

Мікрофони призначено для введення звукових даних до комп'ютера. Їх можна поділити на кілька груп залежно від:

- принципу дії (*динамічні, п'єзоелектричні, магнітоелектричні, електростатичні* тощо);
- системи під'єднання: *вмонтовані* (у корпуси ноутбуків, вебкамер тощо) і *відокремлені* (під'єднуються додатково до пристроїв комп'ютера, зазвичай з використанням USB-з'єднання (мал. 3.30)), *гарнітури* (конструкції, що складаються з навушників і мікрофона)

тощо.



Мал. 3.30. Відокремлений мікрофон

Мікрофони також розрізняють за значенням основних властивостей: ***чутливість*** (наприклад, -60 децибелів), ***смуга пропускання частот*** (наприклад, від 100 Гц до 16 000 Гц), ***спрямованість на джерело звуку*** (*всеспрямований, односпрямований, двонаправлений* та інші).

Використання мікрофонів у комп'ютерній техніці набуло особливого розповсюдження з розвитком систем комп'ютерного зв'язку, систем голосового введення текстів, проведенням аудіо- та відеоконференцій тощо.

Значення властивостей пристроїв уведення даних постійно змінюються, вони вдосконалюються за дизайном та якістю введення даних, з'являються нові види пристроїв. Варто перед придбанням пристроїв ознайомитися з думками експертів і користувачів щодо їх використання. Такі матеріали регулярно розміщують в інтернеті.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/oe2cAkxM* або QR-кодом.

Найважливіше в цьому пункті

До **пристроїв уведення даних** належать пристрої введення текстових (**клавіатура**), графічних (**сканер**, **фотокамера**, **графічний планшет**), звукових (**мікрофон**), відеоданих (**відеокамера**, **вебкамера**, **ТВ-тюнер**), а також пристрої для забезпечення керування об'єктами в різноманітних програмах – **миша**, **тачпед**, **мультимедійна** (**електронна**) **дошка**, **сенсорний екран**, **джойстик**, **геймпед**, **кейпед**, **кермо**, **педалі**, **денспед** тощо.

Під час добору пристроїв варто зважати на їх призначення, значення основних властивостей і ціну. Для цього можна використовувати інтернет-сервіси, які надають покупцеві дані для знаходження потрібних моделей, порівняння за значенням різних властивостей, що допомагає визначитись з вибором товару, магазину.

Дайте відповіді на запитання

1. Які пристрої комп'ютера належать до пристроїв уведення даних?

2. За значенням яких властивостей класифікують клавіатури? Наведіть приклади.

3. Які види маніпуляторів миша використовують у комп'ютерному класі вашої школи? Наведіть значення їх властивостей.

4. З використанням яких пристроїв можна ввести графічне зображення в пам'ять комп'ютера?

5. Для чого використовують сканер? Які типи сканерів ви знаєте?

6. Для чого призначено пристрої введення, які входять до складу мультимедійного обладнання комп'ютерного класу?

Обговоріть і зробіть висновки

1. На значення яких властивостей слід звернути увагу, купуючи нову мишу для домашнього комп'ютера? Поясніть чому.

2. Які значення властивостей можуть мати сканери, які використовують у комп'ютерних класах школи? Який сканер ви б порекомендували для комп'ютерного класу?

3. Чим, на вашу думку, відрізняється відеокамера від вебкамери? Що в них спільного?

Виконайте завдання

1. На основі відомостей, розміщених у підручнику, підготуйте текстовий документ зі схемою класифікації пристроїв уведення даних. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.1**.

2. На основі відомостей, розміщених у підручнику, підготуйте комп'ютерну презентацію про пристрої керування роботою програм. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 3.2.2**.

3. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад *https://ek.ua/ua/*, визначте значення властивостей трьох різних маніпуляторів миша, що надійшли останнім часом у продаж у магазини України. Створіть у текстовому процесорі таблицю та заповніть її:

Властивість	*Значення властивості миші*		
Модель			
Тип підключення			
Тип датчиків руху			
Кількість кнопок			
Тип корпусу			
Призначення			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.3.**

4. Використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад *https://ek.ua/ua/*, визначте значення властивостей планшетних сканерів трьох різних виробників, що пропонуються магазинами України для використання вдома. Створіть у текстовому процесорі таблицю за зразком і заповніть її:

Властивість	*Значення властивості планшетного сканера*		
Модель, виробник			
Тип підключення			
Область сканування			
Роздільність			
Час попереднього сканування			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.4.**

5. За відомостями з інтернету або каталогами комп'ютерних магазинів запропонуйте для шкільної бібліотеки дві моделі проєкційних сканерів для створення електронної бібліотеки. Свої пропозиції обґрунтуйте в текстовому файлі. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.5.**

6. Порадьте своєму товаришу / подрузі, якому / якій подобається малювати, 2–3 моделі графічних планшетів. Пропозиції обґрунтуйте в текстовому файлі. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.6.**

7. Підготуйте повідомлення в текстовому документі про використання мікрофонів різних типів для створення мультимедійних об'єктів з використанням комп'ютерів. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.2.7.**

Готуємось до вивчення нового матеріалу

1. У якому вигляді результати опрацювання даних можуть подаватися користувачу комп'ютера?
2. Які пристрої ви використовували для виведення даних з комп'ютера?
3. Які пристрої використовують як для введення, так і для виведення даних?

3.3. ПРИСТРОЇ ВИВЕДЕННЯ ДАНИХ

У цьому пункті йтиметься про:

- основні види пристроїв виведення даних, що використовуються в персональних комп'ютерах;
- монітори та відеоадаптери, їх види та основні властивості;
- пристрої для друку, їх види та основні властивості;
- пристрої для виведення аудіо- та відеоданих.

ВИДИ ПРИСТРОЇВ ВИВЕДЕННЯ ДАНИХ

Пригадайте

● Дані яких видів зазвичай виводяться комп'ютером для їх сприйняття людиною? Які пристрої для цього використовуються? ● За значенням яких властивостей ви обираєте той чи інший пристрій виведення даних?

Як ви вже знаєте, до складу комп'ютерів уходять різноманітні пристрої виведення даних.

Як і пристрої введення даних, пристрої виведення даних можна розподілити за типом даних, для виведення яких вони призначені:

- текстових і графічних (**принтери, монітори**);
- звукових (**навушники, звукові колонки, синтезатори**);
- відеоданих (**монітори, проєктори, сенсорні екрани, мультимедійні (електронні) дошки**).

На малюнках 3.31 і 3.32 наведено зображення пристроїв виведення аудіо- та відеоданих.



Мал. 3.31. Пристрої виведення даних – мультимедійна дошка, навушники, звукові колонки



Мал. 3.32. Пристрої виведення даних – мультимедійний проєктор, мультимедійний екран

МОНІТОРИ



Пригадайте

- Для чого використовують монітор?
- Які ви знаєте типи моніторів?

Монітор є основним пристроєм для виведення даних у персональних комп'ютерах. Монітори поділяють за системою створення зображення на:

- **LCD-монітори** (англ. ***Liquid Crystal Display*** – рідиннокристалічний дисплей), або монітори на рідких кристалах;
- **плазмові;**
- **OLED-монітори** (англ. ***Organic Light Emitting Diode*** – органічний світлодіод);
- **e-ink** (англ. ***Electronic ink*** – електронне чорнило) – монітори на **електронному чорнилі** тощо.

Для більшості сучасних моніторів використовують рідиннокристалічну технологію. **LCD-монітори** поділяють за:

- принципом дії (*TN + film*, *PLS*, різні версії *IPS* та *VA*);
- довжиною діагоналі (від *19* до *85 дюймів*);
- роздільністю (від *1280 × 1024* до *5120 × 2880* точок);
- часом реакції – реагування на команду змінення кольору пікселя екрана із чорного на білий (від *1 мс* до *8 мс*);
- співвідношенням довжин сторін екрана (*16 : 9*, *16 : 10*, *21 : 9*, *5 : 4*).

Суттєвою характеристикою **LCD**-моніторів є тип джерела світла, яке використовується для підсвічування зображення на зовнішній поверхні екрана.

У переважній більшості сучасних **LCD**-моніторів використовується **LED-технологія** (англ. ***Light Emitting Diode*** – світлодіод) підсвічування екрана.

Плазмові монітори використовуються в основному для створення великих демонстраційних екранів, бо мають низьку роздільність, але високу яскравість зображення. Останнім часом ця технологія заміюється технологією на рідких кристалах.

Технологія **OLED** уже понад 15 років уважається перспективною для створення якісних моніторів завдяки високій контрастності та яскравості зображення, малій (до 1 см) товщині екрана. Однак висока вартість цих моніторів, значне зниження яскравості через 3–5 років роботи суттєво знижують їх ринок збуту.

Висока контрастність чорно-білого зображення в моніторах на основі **електронного чорнила**, значні терміни роботи без додаткової підзарядки (кілька тижнів), відсутність мерехтіння екрана забезпечують комфортні умови для читання з екрана. Тому монітори, які створено за вказаною технологією, в основному використовують в електронних книжках – планшетних комп'ютерах, що призначені для читання текстів. З 2016 року компанією **E Ink** розпочато випуск кольорових моніторів (мал. 3.33), однак їх



Мал. 3.33. Електронна книга на основі електронного чорнила

широкому розповсюдженню для більшості комп'ютерів заважає дуже низька швидкість оновлення екрана, мала кількість (4096) відтворюваних кольорів і потреба додаткового підсвічування.

ВІДЕОАДАПТЕР

Пригадайте

• Що таке *материнська плата*? • Який пристрій комп'ютера виконує опрацювання графічних даних?

Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовують спеціальний пристрій – **відеоадаптер.**

Розрізняють *інтегровані* відеоадаптери та *у вигляді окремої плати*. Інтегрований відеоадаптер розміщується в мікросхемі процесора або на материнській платі. Його використовують у персональних комп'ютерах, які не призначені для якісного опрацювання великих обсягів графічних даних. Для цих цілей використовують відеоадаптери у вигляді окремої плати (мал. 3.34), що вставляється в один зі слотів материнської плати.

Такі відеоадаптери містять спеціальний графічний процесор і додаткову оперативну пам'ять – ***графічну (відео-) пам'ять.*** Приклади значень властивостей відеоадаптера подано в таблиці 3.5.

Інтегрований (лат. *integer* – цілий) – нерозривно зв'язаний, цільний.
Інтеграція (лат. *integratio* – поповнення) – об'єднання в ціле окремих частин.
Чипсет (англ. *chip* – мікросхема, *set* – набір) – набір мікросхем, що забезпечує обмін даними між пристроями комп'ютера.



Мал. 3.34. Відеоадаптер

Таблиця 3.5

Приклади значень властивостей відеоадаптерів

Властивість	***Значення властивостей***	
Модель	PowerColor Radeon RX 7900 GRE 16 GB Fighter	ASUS DUAL-RTX4070S-O12G-EVO-WHITE
Графічний процесор	Radeon RX 7900 GRE	GeForce RTX 4070 SUPER
Обсяг пам'яті	16 ГБ	12 ГБ
Тип під'єднання до материнської плати	PCI Express 4.0	PCI Express 4.0
Інтерфейси виведення даних	HDMI 2.1; DisplayPort 2.1	HDMI 2.1; DisplayPort 1.4

ПРИСТРОЇ ДЛЯ ДРУКУВАННЯ



Пригадайте

● Які пристрої використовують для друкування текстових даних? ● Які види друкувальних пристроїв ви знаєте?

Для виведення текстових і графічних даних на папір, плівку або інші матеріали використовують **принтери** та **плотери.**

Основними властивостями принтерів є:

- розмір матеріалу, на який здійснюється друкування, – *А4*, *А3*;
- кількість кольорів – *монохромні*, *багатоколірні*;
- роздільність – від *600 × 1200* до *5760 × 1440 точок на дюйм* і більше;
- швидкість друкування – від *1* до *80 сторінок за хвилину*;
- тип підключення до комп'ютера – *дротове* або *бездротове*

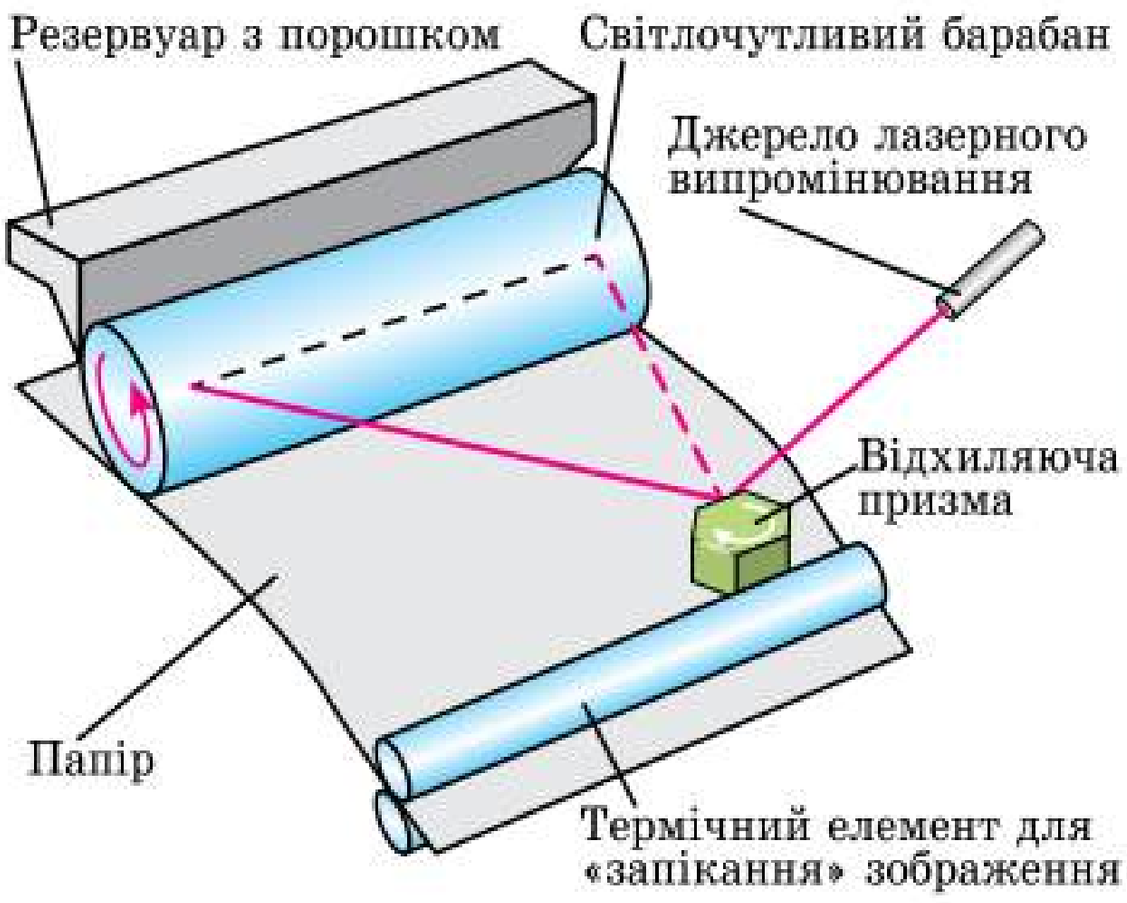
тощо.

Принтери поділяють за принципом дії на: **матричні, лазерні, струменеві, термосублімаційні, термічні, 3D-принтери** тощо. Найрозповсюдженішими є принтери, що використовують струменеву та лазерну технології, однак дедалі більшої популярності набувають термосублімаційні та 3D-принтери.

Лазерний принтер. Основним елементом лазерного принтера є циліндр з нанесеним на нього світлочутливим матеріалом (мал. 3.35). Лазерна установка генерує тонкий світловий промінь, який потрапляє на барабан і змінює його електричний заряд у точці падіння. Далі на барабан наноситься **тонер** – спеціальна фарба в порошковому вигляді. Кількість порошку, що прилипає до поверхні барабана, залежить від величини заряду в певній точці. Під час обертання барабан притискається до аркуша паперу. Тонер прилипає до поверхні паперу й утворює на ньому потрібне зображення.

Тонер (англ. *toner* – той, що надає відтінок) – порошок або крем, з використанням якого змінюють колір поверхні, на яку він наноситься.

Фюзер (англ. *fuse* – плавитися) – пристрій для плавлення, запікання.



Мал. 3.35. Схема дії лазерного принтера



Мал. 3.36. Струменевий принтер

Для фіксації порошку на папері аркуш пропускають між роликами, які розігріті до температури приблизно 180 °С (так звана **пічка**, або **фюзер**). Тонер «запікається» на папері. Після цього йому ні волога, ні пряме сонячне проміння не завдають шкоди.

Для кольорового друкування використовують кольорові тонери.

Струменевий принтер (мал. 3.36). Більш доцільно називати ці принтери **крапельними**. Технологія створення зображення у принтерах цього типу базується на нанесенні дуже малих крапель чорнил різного кольору на папір або інший матеріал. Залежно від способу генерації крапель розрізняють:

- *п'єзоелектричну технологію* – крапля видавлюється пластиною, що деформується під дією електричного струму (наприклад, принтери **Epson**);
- *бульбашкову*, або *термічну*, технологію – крапля видавлюється бульбашкою парів чорнила, що утворюються внаслідок швидкого нагрівання електротермічного елемента (наприклад, принтери **Hewlett Packard, Canon**).

Суттєвим недоліком струменевих принтерів є вигорання чорнила під дією прямого сонячного проміння та нестійкість до вологи.

Для тих, хто хоче знати більше

Сублімаційний друк може використовуватися для створення яскравих повноколірних зображень на різних поверхнях – на папері, картоні, а головне на тканині, склі, фаянсі тощо. Існує кілька варіантів сублімаційного друку:

- ***прямий «мокрий» друк*** спеціальними (сублімаційними) чорнилами на поверхню, що містить поліестер, або поверхню, на яку попередньо нанесено шар речовини з поліестером;
- ***прямий «сухий» друк*** з використанням плівок різного кольору (мал. 3.37), з яких під час нагрівання тверда фарба випаровується і переноситься на поверхню, на якій утворюється малюнок;

Поліестер – (грец. πολύ – багато та нім. *Essig-Aether* – оцтовий ефір) – речовина, що використовується в тому числі для створення синтетичних тканин.

Сублімація (лат. *sublimiter* – угору, прямо) – це фізичне явище переходу речовини з твердого стану в газоподібний, минаючи рідкий стан.



Мал. 3.37. Сублімаційний принтер і набір кольорових плівок до нього



Мал. 3.38. Сублімаційний принтер

- ***непрямий друк***, на першому етапі якого відбувається друк з використанням струменевого принтера і сублімаційних чорнил на спеціальний сублімаційний папір. На

другому етапі створене на папері зображення прикладається до поверхні об'єкта, на якій потрібно створити зображення, і нагрівається до температури понад 100 °С. Фарба з паперу випаровується та утворює зображення на чашках, футболках, табличках тощо. У цьому випадку також використовуються тканини з поліестером або поверхні з попередньо нанесеним шаром речовини з поліестером.

Прямий «мокрий» друк зазвичай застосовують для великих тиражів зображень на промислових поліграфічних підприємствах, прямий «сухий» друк – для друку фотографій високої якості (мал. 3.38), непрямий друк – для створення сувенірної продукції малими тиражами навіть у домашніх умовах.

Основна перевага сублімаційної технології друку – висока якість зображень і стійкість до вологи та світла та можливість нанесення на неплоскі поверхні.

Цікаві факти з історії

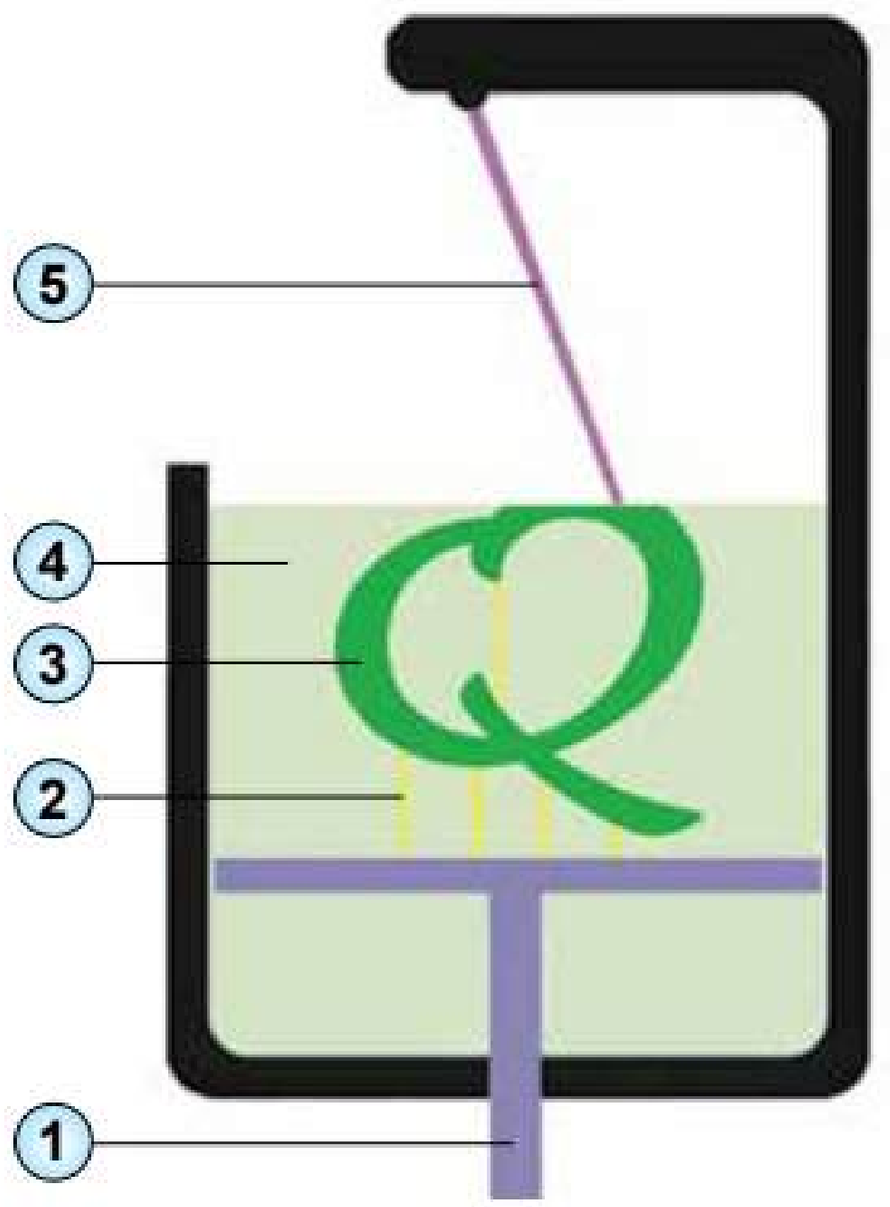
Одним з перших ідею виводити на тверду основу результати обчислень запропонував відомий англійський учений і винахідник **Чарлз Беббідж** (1792–1871). Під час проєктування своєї першої обчислювальної машини Беббідж приєднав до неї друкувальний пристрій, який видавлював результат обчислень на мідній пластині. Таким чином отримане число могло довго зберігатись, і це унеможливлювало помилки, які могли траплятися через неуважність тих, хто записував ці результати на папері.

Плотери (графобудівники) також можуть виводити на тверді носії текстові або графічні дані. Але в основному вони призначені для роботи з графікою. Використовують для друкування креслень, ескізів, плакатів та інших зображень великих розмірів, зазвичай більших ніж 297 мм × 420 мм (формат А3), на папері, різноманітних плівках, тканині тощо.

Принципи, які лягли в основу конструкцій сучасних плотерів, мало чим відрізняються від принципів роботи принтерів. Найрозповсюдженішими є плотери, які використовують струменеву технологію.

Тривимірні, або ***3D-принтери*** використовують для створення об'ємних об'єктів складної форми (мал. 3.39). Розрізняють тривимірні принтери за типом речовини, з якої «будується» об'єкт, і способом її отвердіння. Використовують кілька основних технологій:

- отвердіння спеціальної смоли під дією світла (лазерне випромінення або світлодіодне) – **SLA** (англ. *Stereolithography Apparatus* – стереолітографічна апаратура), або стереолітографічна технологія (мал. 3.39);



Мал. 3.39. Схема роботи **SLA** 3D-принтера:
1 – стіл для побудови об'єкта;
2 – додаткові елементи для підтримки частин об'єкта, що не мають опори; *3* – об'єкт, що друкується (літера Q);
4 – речовина, що твердіє під дією випромінення; *5* – промінь ультрафіолетового лазера

- запікання (плавлення та отвердіння) спеціального порошку під дією променя лазера – **SLS**-технологія (англ. ***Selective Laser Sintering*** – вибіркове лазерне спікання);
- отвердіння попередньо розплавлених пластичних мас – **FDM-технологія** (англ. ***Fused Deposition Modeling*** – моделювання методом наплавлення).

FDM-технологія – найдешевша. Це зумовило її широке розповсюдження у принтерах, які використовують у домашніх умовах і в навчальних закладах. Однак **SLA**- та **SLS**-технології мають суттєву перевагу в точності виготовлення об'єктів і широко використовуються у промисловому виробництві.

Також розрізняють 3D-принтери за максимальними розмірами об'єктів (довжиною, шириною, висотою), які можна виготовити з їх використанням.

Порівняння характеристик трьох основних технологій за п'ятибальною шкалою подано в таблиці 3.6.

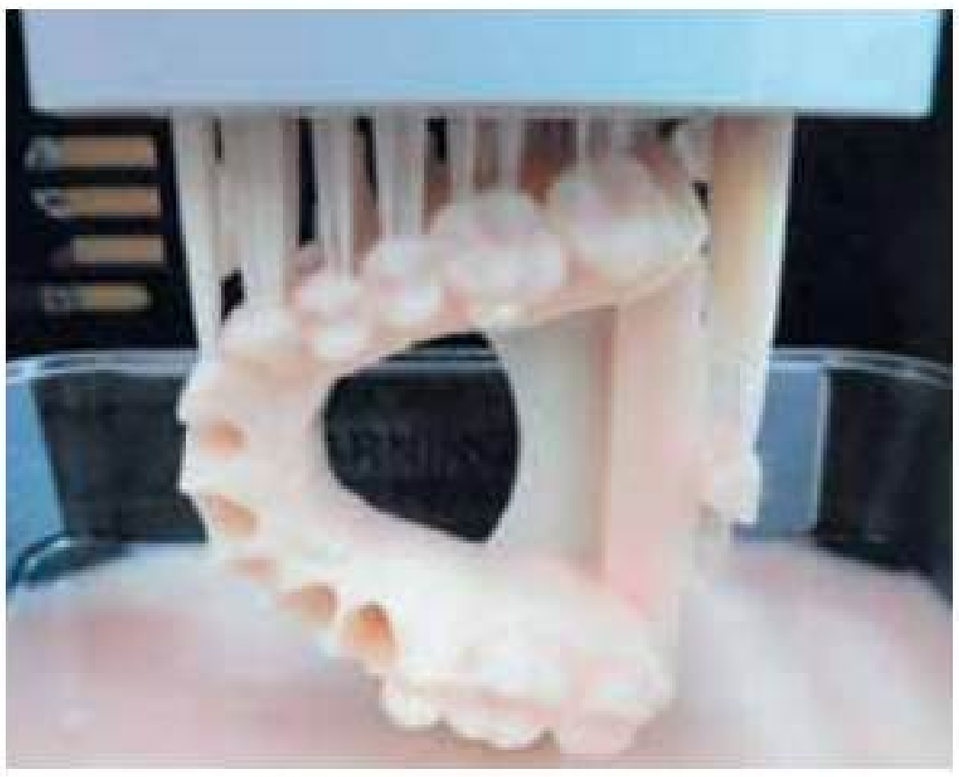
Таблиця 3.6

Порівняння характеристик технологій 3D-друку

Властивості	***Назва технології***		
	SLA	***SLS***	***FDM***
Роздільність (крок)	●●●●●	●●●●○	●●○○○
Точність виготовлення	●●●●●	●●●●●	●●●●○
Шерехатість поверхні	●●●●●	●●●●○	●●○○○
Можливість створення складних виробів	●●●●○	●●●●●	●●●○○
Простота використання	●●●●●	●●●●○	●●●●●
Розміри об'єктів друку для настільних принтерів, максимально (мм)	300 x 335 x 200	165 x 165 x 320	200 x 200 x 300
Вимоги до приміщення	Достатньо офісного приміщення	Бажано розміщення в окремій майстерні	Кондиціонер або спеціальна вентиляція

3D-принтери мають широке застосування в різних галузях людської діяльності (мал. 3.40–3.43). Особливо активно їх використовують у медицині, військовій справі, будівництві, під час виготовлення сувенірної продукції тощо.

Значення властивостей пристроїв постійно змінюються, вони вдосконалюються за дизайном і якістю здійснення операцій з виведення даних. Варто перед придбанням пристроїв ознайомитися з думками експертів і користувачів щодо їх використання. Такі матеріали регулярно розміщують в інтернеті.



Мал. 3.40. Створення зубного протеза у 3D-принтері



Мал. 3.41. Створення протеза грудної клітки у 3D-принтері



Мал. 3.42. Модель будинку, створена на 3D-принтері



Мал. 3.43. Створення моделі блока циліндрів автомобіля у 3D-принтері



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ye2cAOE2* або QR-кодом.



Найважливіше в цьому пункті

До **пристроїв виведення даних** належать пристрої виведення текстових і графічних даних (**монітор**, **принтер**, **плотер**), звукових (**навушники**, **звукові колонки**), відеоданих (**мультимедійні проєктори**, **екранні панелі**) тощо.

Монітори поділяють за системою створення зображення на: монітори на *рідких кристалах*, або *LCD-монітори*, *плазмові*, *OLED*, на *електронному чорнилі – e-ink* тощо. Для більшості сучасних моніторів використовують рідиннокристалічну технологію.

Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовують спеціальний пристрій – **відеоадаптер**. Відеоадаптер містить спеціальний графічний процесор і додаткову оперативну пам'ять – **графічну** (**відео-**) **пам'ять**.

Для виведення даних на папір, плівку або інші матеріали використовують **принтери** і **плотери**.

Принтери поділяють за принципом дії на: *матричні*, *лазерні*, *струменеві*, *термосублімаційні*, *термічні*, *3D-принтери* тощо. Найбільш розповсюдженими є принтери, що використовують *струменеву* та *лазерну* технології.

В останні роки активно розвиваються технології створення тривимірних об'єктів з використанням *3D-принтерів*.

Дайте відповіді на запитання

1. Які пристрої належать до пристроїв виведення даних?

2. За значенням яких властивостей класифікують монітори? Наведіть приклади значень властивостей сучасних моніторів.

3. Які види принтерів використовують у вашій школі? Як визначити значення їх властивостей?

4. Які принтери використовують для створення сувенірної продукції малими тиражами?

5. Для чого призначено пристрої виведення даних, що входять до складу мультимедійного обладнання комп'ютерного класу?

Обговоріть і зробіть висновки

1. На значення яких властивостей слід звернути увагу, купуючи новий відеоадаптер для домашнього комп'ютера? Поясніть чому.

2. Чим відрізняється принтер, який працює за лазерною технологією, від принтера на основі струменевої технології? Яка технологія дає змогу створити більш якісне кольорове зображення?

3. Які значення властивостей повинен, на вашу думку, мати принтер для домашнього використання?

Виконайте завдання

1. На основі відомостей, розміщених у підручнику, підготуйте текстовий документ зі схемою класифікації пристроїв виведення даних. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.3.1.**

2. На основі відомостей, розміщених у підручнику, підготуйте комп'ютерну презентацію зі схемою класифікації моніторів і фотографіями моніторів різного типу. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 3.3.2.**

3. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад *https://ek.ua/ua/*, визначте значення властивостей трьох різних моніторів, що надійшли останнім часом у продаж у магазини України. Створіть у текстовому процесорі таблицю та заповніть її.

Властивість	***Значення властивості монітора***		
Модель			
Виробник			
Довжина діагоналі, *дюймів*			
Технологія			
Максимальна роздільність, *пікселів*			
Час реакції, *мс*			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.3.3.**

4. На основі відомостей, розміщених у підручнику, підготуйте в одній з прикладних програм схему класифікації сучасних принтерів. Збережіть схему у вашій папці у файлі з іменем **завдання 3.3.4.**

5. Дізнайтеся значення властивостей пристроїв виведення вашого домашнього комп'ютера. Створіть у текстовому процесорі та заповніть таблиці результатів аналогічно до таблиць рубрики **Працюємо з комп'ютером.** Збережіть таблиці у вашій папці у файлі з іменем **завдання 3.3.5.**

6. За відомостями з інтернету або каталогами комп'ютерних магазинів визначте значення властивостей принтерів (струменевих і лазерних) трьох різних виробників, що запропоновані в магазинах України. Створіть у текстовому процесорі таблицю та заповніть її.

Властивість	***Значення властивості принтера***		
Модель			
Виробник			
Принцип дії			
Роздільність, *пікселів*			
Швидкість друку, *сторінок за хвилину*			

7. Підготуйте презентацію про галузі використання сублімаційних принтерів. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 3.3.7.**

8. Підготуйте презентацію про галузі використання 3D-принтерів. Зверніть особливу увагу на промислове використання технології 3D-друку. Збережіть комп'ютерну презентацію у вашій папці у файлі з іменем **завдання 3.3.8.**

Готуємось до вивчення нового матеріалу

1. Які види комп'ютерів ви знаєте?
2. Які чинники впливають на розвиток комп'ютерних технологій?
3. Які пристрої є складовими комп'ютера?

3.4. ВИДИ СУЧАСНИХ КОМП'ЮТЕРІВ ТА ЇХ ЗАСТОСУВАННЯ

У цьому пункті йтиметься про:

- види сучасних комп'ютерів;
- застосування комп'ютерів;
- основні напрями використання комп'ютерів;
- діагностику апаратної та програмної складової комп'ютера, усунення збоїв у їх роботі;
- добір складових комп'ютера залежно від його призначення.

ВИДИ СУЧАСНИХ КОМП'ЮТЕРІВ

Поміркуйте

● Розгляньте запропоновані зображення комп'ютерів (3.44). Як можна назвати ці комп'ютери? За значеннями яких властивостей вони відрізняються?



а) б) в) г)

Мал. 3.44. Комп'ютери різних видів

Вам уже відомо, що найпоширенішими на сьогодні є **персональні комп'ютери** (ПК), серед яких розрізняють **стаціонарні** та **мобільні (портативні)**. Стаціонарні ПК, у свою чергу, поділяються за основним призначенням на **офісні, домашні, ігрові** тощо. А мобільні, залежно від особливостей конструкції, – на **ноутбуки, нетбуки, планшетні ПК, смартфони** тощо. Вони використовуються для побутових і професійних задач.

Для розв'язування задач, що потребують великих обсягів складних обчислень, наприклад для розрахунків прогнозу погоди на декілька днів чи тижнів, використовують більш потужні комп'ютери. Найбільші з них називають **суперкомп'ютерами.** Одночасно суперкомп'ютери можуть розв'язувати мільйони різноманітних задач і обслуговувати сотні тисяч користувачів по всьому світу, використовуючи різноманітні комп'ютерні мережі.

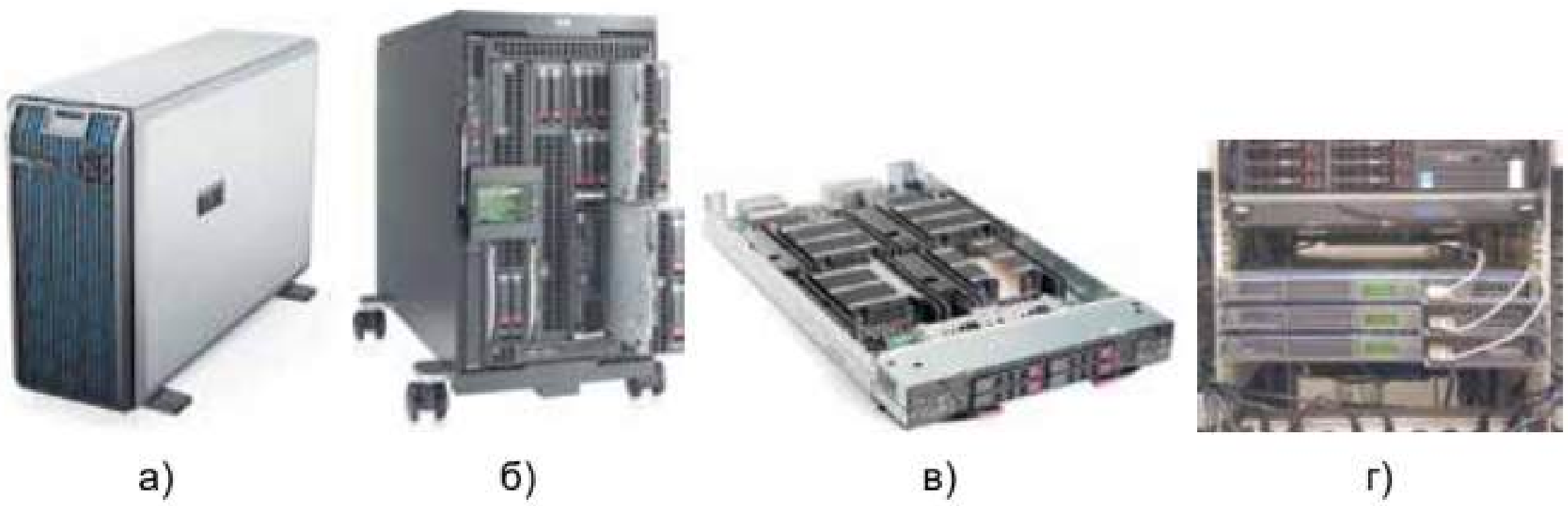
На середину 2024 року найпотужнішим комп'ютером у світі був комп'ютер **Frontier** (англ. *frontier* – кордон) (мал. 3.45) Національної лабораторії Оук-Рідж у штаті Теннессі, США. Останні роки він є найпотужнішим у списку 500 найпродуктивніших суперкомп'ютерів. Продуктивність таких комп'ютерів вимірюється у **флопсах** (англ. **FLOPS** – ***Floating Point Operations Per Second*** – кількість операцій з рухомою крапкою на секунду). Суперкомп'ютер **Frontier** – перший з комп'ютерів, що досяг продуктивності, яка перевищує 1 EFlops. Префікс кратності Е (екса) відповідає 10^{18}.



Мал. 3.45. Суперкомп'ютер **Frontier**

Frontier має загалом 8 699 904 комбінованих ядра центральних і графічних процесорів корпорації AMD. Цей суперкомп'ютер використовується для розв'язування різноманітних задач, таких як моделювання взаємодії мільйонів космічних галактик, проведення розрахунків з використанням штучного інтелекту конструкції низькотемпературного двигуна, прогнозування розвитку економіки на десятки років уперед, створення комп'ютерної моделі керованого термоядерного синтезу (що прискорить його використання як основи для генерації електроенергії, а «паливом» для генерації буде звичайна морська вода), створення моделі хімічних процесів для перетворення біомаси у придатне для використання паливо тощо.

Менш потужними, ніж суперкомп'ютери, але потужнішими за персональні комп'ютери, є **сервери** – комп'ютери, що надають користувачам певні послуги зберігання великих обсягів даних, проведення складних обчислень, які на звичайному ПК можуть тривати тижні та місяці, надання послуг електронної пошти, комунікаційних послуг, організація передавання даних комп'ютерними мережами, захист комп'ютерних мереж, використання програмного забезпечення серверів тощо. Їх потужності та вартість значно різняться. Ціна простіших серверів може складати кілька десятків тисяч гривень, а потужних – десятки мільйонів. Вигляд серверів різного призначення та будови подано на малюнку 3.46.



а) б) в) г)

Мал. 3.46. Сервери в різних варіантах виконання

Залежно від призначення та кількості користувачів, яким надають послуги сервери, вони можуть використовувати один спеціальний серверний процесор (наприклад, шкільний сервер доступу до інтернету) або кілька сотень чи тисяч таких процесорів (наприклад, сервер банку або інтернет-провайдера).

Серед найпопулярніших типів персональних комп'ютерів варто виділити комп'ютери, що поєднують функції мобільного телефона та комп'ютера, – **смартфони** (мал. 3.44, в). Смартфон став універсальним помічником не тільки в забезпеченні спілкування, але й у використанні сервісів інтернету. За даними сайту **WORLDOMETERS** (*https://www.worldometers.info/uk/*), щоденно у світі продається понад 7 млн мобільних телефонів.

Основні властивості смартфонів і приблизні їх значення на середину 2024 року наведено в таблиці 3.7.

Таблиця 3.7

Приклади значень основних властивостей смартфонів

Властивість	***Значення властивості***
Оперативна пам'ять, *ГБ*	4–24
Вбудована зовнішня пам'ять, *ГБ*	64–1000
Модель процесора	MediaTek MT8788; Samsung Exynos 2400; Qualcomm Snapdragon 8 Gen 3; Apple A16 Bionic
Діагональ дисплея, *дюйма*	4,6–8,3
Основна камера, *Мпіксель*	16–108
Стандарти зв'язку	4G; 5G
Операційна система	Android 14; iOS 17
Виробник	Samsung; Xiaomi; Apple; HUAWEI; realme

У 5-му класі ви вже ознайомилися з ще одним видом комп'ютерів – **вбудованими комп'ютерами.** Вбудовані комп'ютери є ключовими компонентами в багатьох сферах, забезпечуючи автоматизацію, ефективність і безпеку роботи різних систем і пристроїв:

- побутових пристроїв (пральні машини, холодильники, кондиціонери, телевізори, музичні центри);
- транспортних засобів (автомобілі, локомотиви, літаки);
- робототехніки

тощо.

ЗАСТОСУВАННЯ КОМП'ЮТЕРІВ



Пригадайте

- Чи відомо вам про застосування комп'ютерів у фінансовій справі, медицині, промисловості, транспорті? Наведіть ці приклади.
- Чи використовувалися комп'ютери у школі на заняттях з інших предметів, крім інформатики? З якою метою?

У наш час комп'ютери застосовують у науці, промисловості, торгівлі, управлінні, банківській системі, освіті, медицині, транспорті, зв'язку, сільському господарстві, системі соціального забезпечення та інших галузях господарства, у побуті.

Одне з багатьох застосувань комп'ютерів у науковій сфері – це проведення так званих комп'ютерних експериментів. Адже проведення багатьох наукових досліджень пов'язано зі значними труднощами – матеріальними, технічними, енергетичними тощо. Особливості комп'ютерного моделювання і проведення комп'ютерних експериментів ви вивчали у 6-му класі.

Важливу роль відіграє комп'ютер і на виробництві. Моделювання і конструювання різноманітних виробів з використанням комп'ютерів значно скорочує термін їх розробки, підвищує їх ефективність і якість, знижує вартість. Наприклад, якщо раніше, до застосування комп'ютерів у проєктуванні нової моделі автомобіля, від моменту ви-

никнення ідеї до її виробництва проходило 5–6 років, то тепер цей час становить менше ніж один рік.

У сфері обслуговування комп'ютери використовують для зберігання та опрацювання різноманітних даних: текстів, таблиць, баз даних, малюнків і фотографій, мультимедійних даних.

Важко сьогодні назвати таку установу, де б не використовували комп'ютери. Картотеки в бібліотеках і лікарнях, виконані на основі комп'ютерної бази даних, у багато разів надійніші та зручніші в роботі, ніж традиційні паперові. Бухгалтер теж сьогодні використовує комп'ютер і за кілька хвилин отримує результати, на обчислення яких раніше йому були потрібні години або дні. Фінансист, не виходячи зі свого кабінету, має змогу стежити за станом справ на біржі, простим натисненням на клавіші перевести на будь-який рахунок певну суму грошей.

Усе більшу роль у комп'ютеризації всіх сфер життя людини відіграють смартфони. З їх використанням здійснюється більшість покупок в інтернеті, керування «розумними» пристроями дома, обмін даними в мережах тощо.

Не виходячи з дому або ще по дорозі на роботу чи додому можна здійснити замовлення їжі, товарів, квитків на транспортні засоби, бронювання готелів, туристичних поїздок, сплатити комунальні послуги, записатися на прийом до лікаря тощо.

В Україні з 2019 року реалізується програма «Держава у смартфоні» (мал. 3.47). У рамках цієї програми передбачено, що користувачі без відвідувань різноманітних офісів державних і фінансових установ зможуть:



Мал. 3.47. Держава у смартфоні

- відкрити банківський рахунок;
- здійснити безготівкові платежі за різноманітні послуги державних і комунальних установ;
- надіслати звіти до державних органів про свою комерційну діяльність в електронному вигляді, з використанням електронного цифрового підпису;
- отримати в інтернет-магазині гарантійні талони на різноманітну техніку та обладнання в електронному вигляді;
- отримати доступ до різноманітних реєстрів:
  - дозволів на міжнародні перевезення;
  - земельних ресурсів;
  - лісових ресурсів;
  - оновлення даних про військовий облік;
  - лікарських препаратів і термінів дії сертифікатів на них

тощо.

Основними напрямами використання комп'ютерної техніки є:

Для особистих потреб:

- ***Спілкування***: електронна пошта, соціальні мережі, відеодзвінки, обмін повідомленнями.
- ***Розваги***: створення комп'ютерних ігор, музики, відеофільмів, потокових трансляцій.

- ***Доступ до контенту***: перегляд інтернету, онлайн-дослідження, навчання.
- ***Продуктивність***: опрацювання текстів, електронних таблиць, презентацій, управління особистими фінансами.

Бізнес, фінанси і промисловість:

- ***Обробка даних***: ведення документації, нарахування заробітної плати, управління запасами, управління відносинами з клієнтами (CRM).
- ***Спілкування***: електронна пошта, відеоконференції, інструменти для співпраці.
- ***Електронна комерція***: онлайн-продажі, обробка платежів, керування замовленнями.
- ***Проєктування та розробка***: автоматизоване проєктування виробів, моделювання, розробка документації продуктів.
- ***Автоматизація***: робототехніка, керування машинами, оптимізація виробничих процесів.
- ***Бухгалтерські справи***: облік матеріальних цінностей, нарахування заробітної плати, аудит.
- ***Фінанси***: банківська справа, фінансовий аналіз, управління ризиками фінансових операцій.

Освіта:

- ***Системи управління навчанням***: онлайн-курси, віртуальні класи та освітнє програмне забезпечення, тести, завдання.
- ***Дослідження***: доступ до баз даних, аналіз даних, наукове моделювання.
- ***Спілкування***: електронна пошта, онлайн-форуми, відеоконференції.
- ***Адміністративні завдання***: реєстрація здобувачів освіти, ведення особових справ учнів і студентів, складання розкладу, виставлення оцінок, підготовка до друку документів про освіту, організація обміну даними між учасниками освітнього процесу.

Уряд і державний сектор:

- ***Обслуговування громадян***: подання податкової декларації онлайн, податкові звіти, державні заяви, цифрове посвідчення особи.
- ***Управління даними***: дані перепису населення, реєстрація виборців, публічні записи.
- ***Комунікації***: внутрішні та зовнішні комунікації, антикризове управління (наприклад, сповіщення про загрози).
- ***Безпека***: спостереження, прикордонний контроль, правоохоронні органи.

Охорона здоров'я:

- ***Записи пацієнтів***: електронні медичні записи, медична візуалізація.
- ***Дослідження***: аналіз медичних даних, відкриття ліків.
- ***Телемедицина***: дистанційна консультація пацієнтів, діагностика.
- ***Медичне обладнання***: медичні прилади з комп'ютерним керуванням, комп'ютерна томографія, магнітно-резонансна томографія.

Наука та дослідження:

- ***Аналіз даних***: наукові експерименти, моделювання.

- ***Високопродуктивні обчислення***: створення суперкомп'ютерів, складні обчислення, прогноз погоди.
- ***Обробка зображень***: медичні зображення, супутникові зображення.
- ***Штучний інтелект***: машинне навчання, опрацювання природної мови, генерація нових даних (текстових, графічних, відео тощо).

Розваги та засоби масової інформації:

- ***Створення контенту***: редагування відео, графічний дизайн, створення музики, цифрове мистецтво.
- ***Розповсюдження***: онлайн-трансляція, цифрове завантаження.
- ***Ігри***: розробка ігор, ігрові онлайн-платформи.

Інші напрямки:

- ***Сільське господарство***: точне землеробство, аналіз даних щодо врожайності.
- ***Транспорт***: навігація автомобіля, управління дорожнім рухом, автономні транспортні засоби, відстеження та оптимізація потоку товарів і матеріалів.
- ***Військові***: командування, розвідка, безпілотники.

Цей список не є вичерпним, оскільки комп'ютери застосовуються практично в усіх аспектах сучасного життя.

КОНФІГУРАЦІЯ КОМП'ЮТЕРА ПІД ПОТРЕБУ



Поміркуйте

- Чи доводилось вам добирати комп'ютер чи окремі пристрої комп'ютера для власних потреб або радити друзям у їх доборі? Наведіть приклади.
- Залежно від чого, на вашу думку, потрібно здійснювати вибір комп'ютера або його складових?

Залежно від того, для яких цілей буде використовуватися комп'ютер, добирають його складові (конфігурацію). При цьому особливу увагу звертають на значення таких властивостей:

- ***потужність процесора*** (опрацювання математичних моделей, аналіз великих обсягів даних, включаючи машинне навчання та штучний інтелект, шифрування та дешифрування даних, створення та редагування комп'ютерної графіки, відеомонтаж та анімація, комп'ютерні ігри);
- ***обсяг оперативної пам'яті*** (редагування графічних зображень, відео та аудіо, 3D-моделювання та візуалізація, комп'ютерні ігри, компіляція великих проєктів, робота з великими базами даних, імітації, моделювання та аналіз великих даних);
- ***ємність пристроїв зовнішньої пам'яті*** (зберігання резервних копій, великих медіафайлів, баз даних, даних відеомонтажу та рендерингу, даних відеоспостережень);
- ***наявність окремого відеоадаптера*** (комп'ютерні ігри, відеомонтаж та анімація, рендеринг 3D-сцен, тривимірне проєктування в архітектурі, машинобудуванні та інших галузях, імітації та моделювання, наприклад у фізиці, біології, що вимагають візуалізації даних, системи опрацювання карт у реальному часі);

- ***якість відображення даних монітором*** (графічний дизайн і фотографія, відеомонтаж, тривимірне проєктування в архітектурі, машинобудуванні та інших галузях, медичні пристрої, створення макетів поліграфічних видань, перегляд відеозображень).

Звичайно, що на вибір комп'ютера можуть суттєво впливати і значення інших властивостей, таких як наявність кількох пристроїв зовнішньої пам'яті, значення властивостей материнської плати, тривалість роботи в автономному режимі (для мобільних пристроїв), наявність засобів підключення до різного типу комп'ютерних мереж тощо. На добір комп'ютерів буде суттєво впливати і їх вартість.

Зазвичай під час добору комп'ютерів, як і інших товарів, говорять про оптимальне співвідношення між ціною та функціональністю (якістю). Немає сенсу використовувати ігровий комп'ютер, який має потужну відеокарту і високоякісний монітор, для офісної роботи – підготовки текстових документів, опрацювання ділової графіки, роботи з електронними таблицями, презентаціями. Ресурси такого потужного комп'ютера не будуть використані навіть на 10 %, а вартість може бути в 10 разів більшою.

Комп'ютери, які використовують удома, призначені для виконання задач, що притаманні офісним, а також для опрацювання мультимедійного контенту: відео, музики, фото. Також вони активно використовуються для спілкування в інтернеті, дистанційного навчання тощо. Тому варто передбачити у складі цього типу комп'ютерів певні пристрої з відповідними значеннями властивостей.

Значна частина інтернет-магазинів пропонує добирати комп'ютери, використовуючи так звані «розумні» фільтри. Наприклад, сайт порівняння цін в інтернет-магазинах залежно від призначення пропонує такі групи комп'ютерів: *Компактний комп'ютер*, *Комп'ютер для роботи / навчання*, *Робоча станція*, *Базовий ігровий комп'ютер*, *Прогресивний ігровий комп'ютер*. На сайті одного з найбільших інтернет-магазинів у розділі комп'ютерної техніки ми знаходимо всього три групи комп'ютерів: *Початковий рівень*, *Для роботи та навчання*, *Ігрові комп'ютери (геймерські)*. Є сайти, де вибір комп'ютерів пропонують за зовнішніми ознаками Моноблок, Міні-ПК тощо.

Вартість системних блоків стаціонарних комп'ютерів при цьому змінюється в доволі великому діапазоні – від 5,5 тис. грн до 500 тис. грн (за цінами 2024 року). Наведемо приклади значень основних властивостей настільних персональних комп'ютерів для запропонованої вище класифікації (табл. 3.8).

Таблиця 3.8

Приклади значень властивостей комп'ютерів різного призначення

Властивість	***Значення властивості для комп'ютерів різного призначення***		
	Офісний	***Домашній***	***Ігровий***
Тип комп'ютера	Моноблок	Настільний комп'ютер	Настільний комп'ютер

Продовження таблиці 3.8

Властивість	***Значення властивості для комп'ютерів різного призначення***		
	Офісний	***Домашній***	***Ігровий***
Модель процесора	AMD Ryzen 3 7335U	Intel Core i5-12400F	Intel Core i7-14700KF
Тактова частота процесора, *ГГц*	3	2,5	3,4
Обсяг оперативної пам'яті, *ГБ*	8	16	32
Тип основного накопичувача	SSD	SSD	SSD
Ємність основного диска, *ГБ*	480	1000	2000
Тип відеоадаптера	інтегрований Intel HD	NVIDIA GeForce RTX 3050	NVIDIA GeForce RTX 4080
Обсяг відеопам'яті, *ГБ*	-	8	16
Тип звукового адаптера	інтегрований	інтегрований	інтегрований
Монітор	BenQ BL2490 (9H.LM5LJ.LBE)	QUBE Overlord G27Q180	Alienware AW2725DF (210-BLHH)
Діагональ монітора, *дюйма*	23,8	27	26,7
Частота оновлення, *Гц*	100	180	360
Час реакції, *мс*	5	1	0,03
Орієнтовна ціна, *грн*	15 тис.	25 тис.	40 тис.

ДІАГНОСТИКА АПАРАТНОЇ ТА ПРОГРАМНОЇ СКЛАДОВОЇ КОМП'ЮТЕРА, УСУНЕННЯ ЗБОЇВ У ЇХ РОБОТІ



Пригадайте

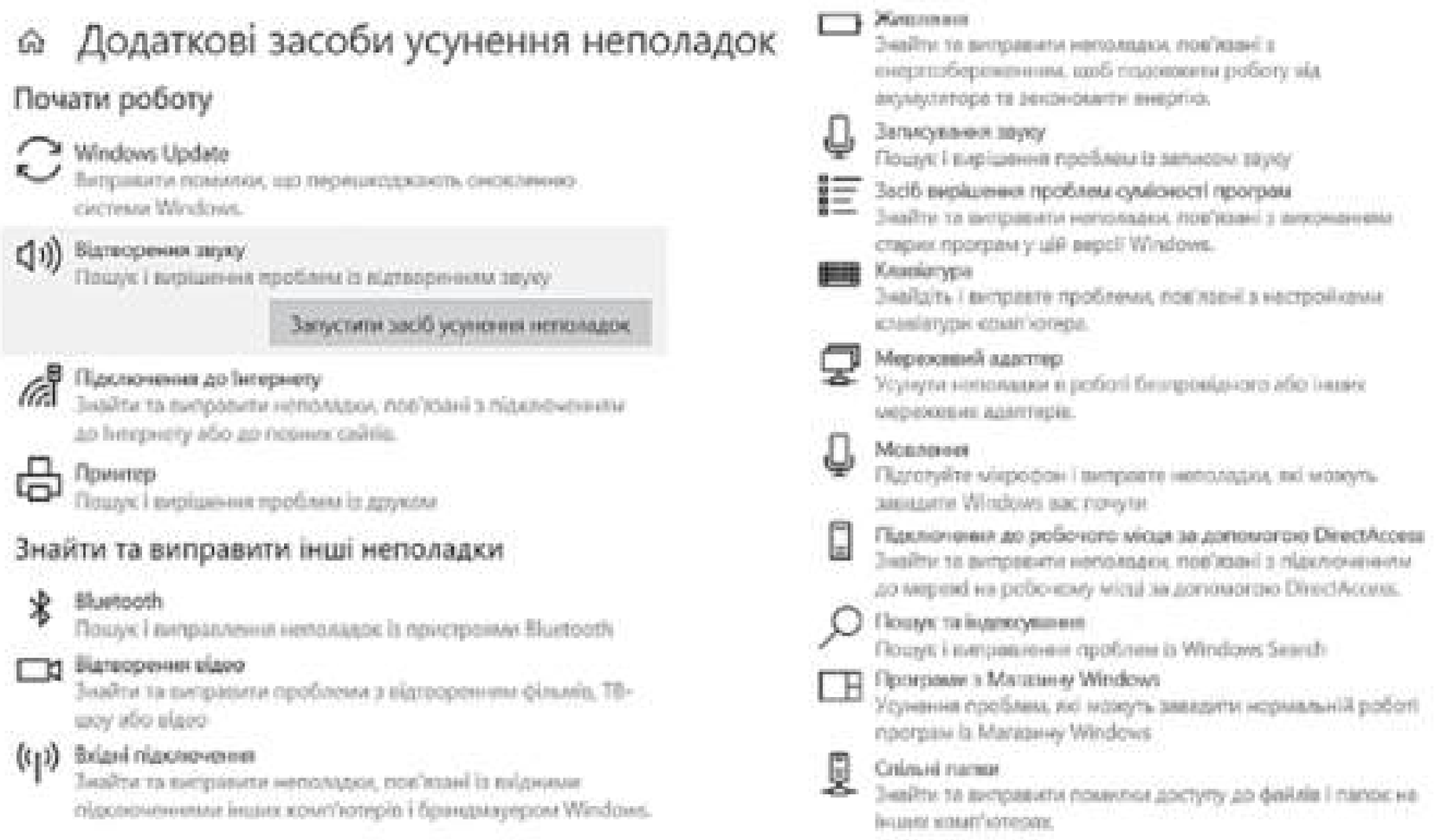
- Чи стикалися ви з перебоями в роботі ваших комп'ютерних пристроїв? Із чим це було пов'язано?
- Які дії ви здійснювали для усунення неполадок у роботі комп'ютера?

У сучасних комп'ютерах значна частина операцій з діагностики апаратної та програмної складової комп'ютера здійснюється автоматично з використанням програм операційної системи та додаткових. Ці програми регулярно проводять перевірку роботи пристроїв комп'ютера та окремих програм, включаючи операційну систему.

Діагностика – процес визначення стану організму, технічного пристрою, рослини. Взаємне розташування, розміщення якихось предметів або їх частин.

Наприклад, після включення комп'ютера виконується система команд, які об'єднані назвою **POST** (англ. ***Power On Self Test*** – самоперевірка під час увімкнення). Метою **POST** є визначити початковий стан пристроїв комп'ютера та виявити, чи правильно працюють основні апаратні компоненти. Якщо виявлено некритичні помилки, то виводяться повідомлення про їх наявність і пропозиція – виконати додаткову перевірку та усунути неполадки або ігнорувати проблеми та перейти до завантаження ОС. А якщо критичні – варто їх усунути (підключити певний пристрій, наприклад клавіатуру), а за неможливості – звернутися до відповідного фахівця.

Ще одним засобом пошуку та виправлення помилок у роботі програмного забезпечення є компонент **Windows**, який має назву **Засіб виправлення неполадок Windows (*Пуск* ⇒ *Налаштування* ⇒ *Оновлення та захист* ⇒ *Виправити неполадки*)**. Для розв'язання конкретної проблеми необхідно обрати посилання **Додаткові засоби усунення неполадок**, обрати вид неполадок (мал. 3.48) та кнопку **Запустити засіб усунення неполадок**.



Мал. 3.48. Елементи керування **Засобу усунення неполадок**

Залежно від вибору виду неполадок після натиснення кнопки **Запустити засіб усунення неполадок** відкриється діалогове вікно, у якому потрібно буде обрати способи діагностики та усунення неполадок. Наприклад, після вибору засобу усунення неполадок мережного адаптера в діалоговому вікні буде запропоновано обрати адаптер для певної мережі: **Ethernet** (англ. *ether net* – ефірна мережа, технологія передавання даних, що використовується в локальних мережах), **Wi-Fi, Підключення через мережу Bluetooth, Усі мережні адаптери**. Після вибору адаптера та кнопки **Далі** буде запущено сканування відповідних пристроїв зв'язку та намагання автоматично виправити неполадки. У випадку, коли програма не змогла виправити неполадки, буде виведено відповідне повідомлення. У такому разі варто звернутися до фахівця.

Існує цілий ряд додаткових програм для діагностики та усунення збоїв у роботі комп'ютера, таких як **AIDA64, SiSoftware Sandra, CPU-Z, ESET** та інші.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ye2cSwVY* або QR-кодом.



Найважливіше в цьому пункті

Серед комп'ютерів залежно від призначення та швидкості опрацювання даних виділяють **суперкомп'ютери**, **сервери**, **вбудовані** та **персональні** комп'ютери. Останні, у свою чергу, поділяються на **стаціонарні** та **мобільні**. Стаціонарні комп'ютери поділяють на **офісні**, **домашні**, **ігрові** тощо. А мобільні – на **ноутбуки**, **нетбуки**, **планшетні ПК**, **смартфони** тощо.

Суперкомп'ютери забезпечують виконання мільйонів різноманітних задач і обслуговування сотень тисяч користувачів по всьому світу, використовуючи різноманітні підключення, включаючи інтернет.

Дедалі більшу роль у комп'ютеризації всіх сфер життя людини відіграють смартфони. З їх використанням здійснюється більшість покупок в інтернеті, керування «розумними» пристроями будинку, обмін даними в мережах тощо. Використовуючи певні програми для смартфонів, можна проводити різноманітні фізичні та математичні дослідження, отримувати доступ до освітніх ресурсів у інтернеті, здійснювати фінансові операції, реалізовувати свої громадянські права.

Вбудовані комп'ютери – це невід'ємна частина сучасного світу. Вони оточують нас усюди і забезпечують комфорт, безпеку та ефективність багатьох процесів.

Залежно від того, для яких цілей буде використовуватися комп'ютер, добирають його складові (конфігурацію). При цьому особливу увагу звертають на значення таких властивостей: потужність процесора, обсяг оперативної пам'яті, ємність зовнішнього накопичувача (для смартфонів – обсяг зовнішньої пам'яті), наявність окремого відеоадаптера (для смартфонів не враховується), якість відображення даних монітором.

У сучасних комп'ютерах значна частина операцій з діагностики апаратної та програмної складової комп'ютера здійснюється автоматично з використанням програм операційної системи, таких як **POST**, а також сервісних програм операційної системи, таких як **Засіб усунення неполадок** тощо.



Дайте відповіді на запитання

1. Як поділяються персональні комп'ютери залежно від призначення?

2. Що таке *суперкомп'ютер*? Для яких цілей використовують такі комп'ютери? Які значення властивостей цих комп'ютерів є основними?

3. Для яких цілей використовують смартфон?

4. Для яких цілей використовують сервери? Наведіть приклади.

5. Що впливає на вибір комп'ютера для певних сфер застосування?

6. Які напрямки використання комп'ютерної техніки ви знаєте?

7. Які інструменти операційної системи надають користувачу можливість діагностувати та усувати недоліки апаратної та програмної складової комп'ютера? Як їх запустити на виконання?



Обговоріть і зробіть висновки

1. Чому дедалі більше користувачів у всьому світі використовують для підключення до мережі «Інтернет» смартфони? Чи продовжиться, на вашу думку, ця тенденція в майбутньому?

2. Чому існує значна відмінність у значеннях властивостей різних комп'ютерів? Наведіть приклади класифікації різних видів комп'ютерів.

3. Які ви знаєте приклади використання смартфонів у навчанні? Чи використовували ви смартфон під час проведення досліджень?

Виконайте завдання

1. Підготуйте текстовий документ зі схемою класифікації комп'ютерів. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.1.**

2. Доберіть комп'ютер для використання як домашній мультимедійний центр для відтворення аудіо та відео. Внесіть дані про складові такого комп'ютера в текстовий документ. Свій вибір обґрунтуйте. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.2.**

3. Підготуйте презентацію (5–10 слайдів) про основні напрямки використання комп'ютерної техніки. Збережіть презентацію у вашій папці у файлі з іменем **завдання 3.4.3.**

4. Знайдіть відомості про найпотужніші суперкомп'ютери на даний час. Створіть у текстовому документі таблицю зі значеннями основних властивостей п'яти найпотужніших суперкомп'ютерів. Використайте для цього матеріали сайту *https://www.top500.org*. Опишіть, де використовується найпотужніший суперкомп'ютер. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.4.4.**

5. Перевірте, використовуючи **Засіб усунення неполадок**, чи наявні неполадки в роботі вашого комп'ютера. Якщо є, то застосуйте вказаний засіб для усунення неполадок. Внесіть у текстовий документ відомості про послідовність та результати перевірки, а також опис дій під час усунення неполадок. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.4.5.**

6. Визначте призначення службової програми **Діагностика пам'яті Windows.** Проведіть діагностику пам'яті. Внесіть у текстовий документ відомості про послідовність і результати перевірки. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.4.6.**

7. Доберіть, використовуючи матеріали цього пункту, конфігурацію комп'ютера для дистанційного навчання учня / учениці 8-го класу. Вибір комплектуючих обґрунтуйте. Внесіть у текстовий документ відомості про обрані складові та обґрунтування вибору. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.4.7.**

Готуємось до вивчення нового матеріалу

1. Коли і ким було створено перший комп'ютер?
2. Які пристрої використовували люди для обчислень до створення комп'ютерів?

3.5. ІСТОРІЯ ОБЧИСЛЮВАЛЬНИХ І КОМП'ЮТЕРНИХ ПРИСТРОЇВ

У цьому пункті йтиметься про:

- етапи розвитку засобів реалізації інформаційних процесів;
- історію засобів опрацювання повідомлень;
- історію розвитку обчислювальної техніки в Україні.

ЕТАПИ РОЗВИТКУ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ



Пригадайте

• Які пристрої для проведення математичних обчислень використовували люди в давні часи? • Які перші механічні пристрої винайшло людство для прискорення математичних обчислень? Для яких потреб вони використовувались?

Розвиток людства безпосередньо пов'язаний з розвитком засобів передавання, опрацювання та зберігання повідомлень, у якому можна виділити кілька етапів (табл. 3.9).

Таблиця 3.9

Етапи розвитку засобів реалізації інформаційних процесів

Назва етапу	*Період історії людства*	*Приклади носіїв даних, засобів передавання та опрацювання даних*
Етап ручних засобів	Від стародавніх часів до середини XV ст.	*Носії даних* – глиняні дощечки, папіруси, береста, палиці із зарубками, картини тощо. *Засоби передавання* – сигнальні вогнища, барабанний бій, персональні посланці, голубина пошта, перші поштові служби для передавання державних документів. *Засоби опрацювання* – мозок і пальці людини, ручні обчислювальні прилади (абак, рахівниця, вузлики на мотузці тощо)
Етап механічних засобів	Від середини XV ст. до середини XIX ст.	*Носії даних* – книжки, газети, журнали, фотопластини, перфокарти. *Засоби передавання* – поштові служби. *Засоби опрацювання* – друкарські машинки, фотоапарати, арифмометри, машинки для обчислень, музичні машинки (шарманки, музичні скриньки), пристрої для відтворення анімації, ткацькі верстати з програмуванням з використанням перфокарт
Етап електричних засобів	Від середини XIX ст. до 40-х років XX ст.	*Носії даних* – платівки, кіноплівки, магнітні плівки. *Засоби передавання* – телефон, телеграф, радіо, телетайп, телебачення. *Засоби опрацювання* – фонографи, електричні друкарські машинки, табулятори, електричні арифмометри та інші обчислювальні машини, магнітофони, ротапринти
Етап електронних засобів	Від 40-х років XX ст. до наших днів	*Носії даних* – магнітні й оптичні диски, електронні мікросхеми. *Засоби передавання* – телебачення, комп'ютерні мережі, засоби мобільного зв'язку. *Засоби опрацювання* – електронні калькулятори, комп'ютери

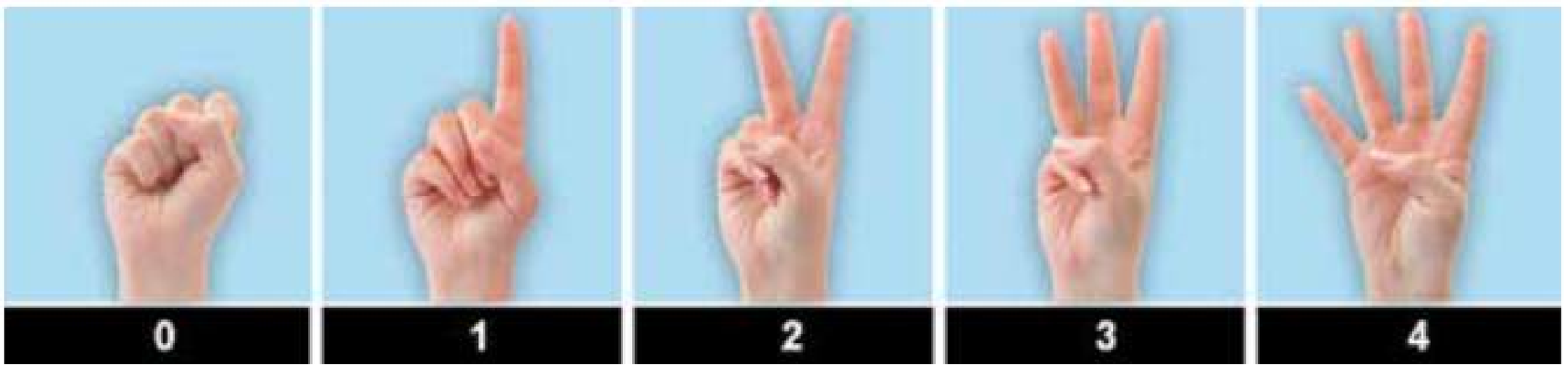
ІСТОРІЯ ЗАСОБІВ ОПРАЦЮВАННЯ ПОВІДОМЛЕНЬ



Поміркуйте

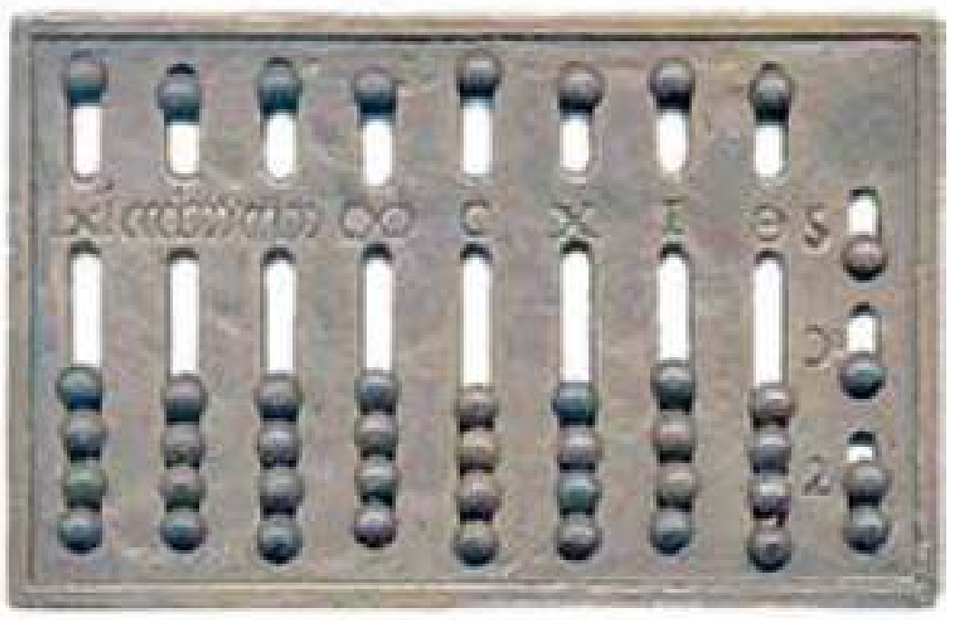
• До якого виду інформаційних процесів можна віднести обчислення? • Для яких потреб людство використовувало пристрої опрацювання повідомлень у давні та середні віки? • З якою метою людина вдосконалює пристрої опрацювання повідомлень?

Розглянемо детальніше історію розвитку засобів опрацювання повідомлень, призначених для виконання обчислень. Першими засобами опрацювання числових повідомлень були пальці (мал. 3.49). Про це свідчать, наприклад, римські цифри (I, V, X). У стародавній слов'янській нумерації одиниці називалися «перстами», тобто пальцями, а назва цифри «5» походить від слова «п'ясть» (кисть).

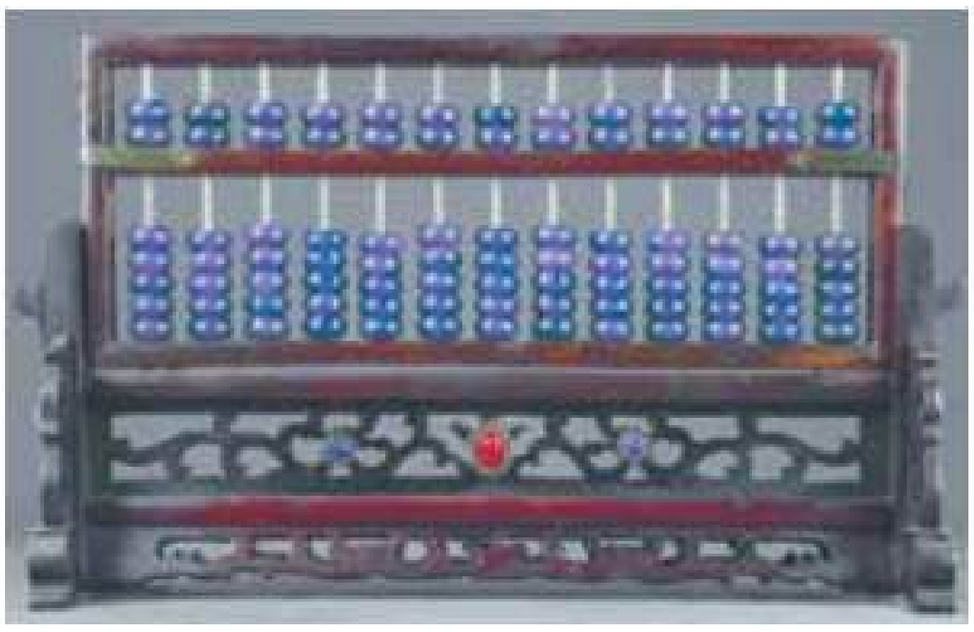


Мал. 3.49. Рахунок за допомогою пальців руки

Значним кроком уперед у розвитку засобів обчислювання стало створення абака в V ст. до н. е. у Стародавній Греції. Ідею такого пристрою греки запозичили у стародавніх вавилонян. Більш пізній римський варіант абака зображено на малюнку 3.50. **Абак** – це дошка з паралельними лунками, у які вкладали якісь предмети, найчастіше камінці. Від камінців пішла назва як пристрою для обчислень – **калькулятор** (лат. *calculus* – камінці), так і процесу – калькулювати, тобто підраховувати.



Мал. 3.50. Римський абак



Мал. 3.51. Суаньпань

Подібні до абака пристрої в подальшому було створено в різних країнах: Китаї **(суаньпань)** (мал. 3.51), Японії **(соробан)**, Кореї **(джупан)**, Туреччині **(кулба)** і використовувалися майже до кінця XX ст.

Разом з простими й доволі розповсюдженими пристроями типу абака ще в Давній Греції та Римі використовували різноманітні об-

числювальні пристрої з використанням коліщат із зубцями. Про це свідчить знахідка із затонулого античного судна. Її назвали «Антикітерський механізм» за назвою грецького острова, біля якого знайшли судно в 1902 р. Але тільки у другій половині XX ст. вчені змогли за уламками механізму відновити його будову і визначити призначення (мал. 3.52). Створений за 100–150 років до нашої ери пристрій використовувався як календар, а також для механічного визначення положення небесних тіл – Сонця, Місяця та відомих грекам планет – Меркурія, Венери, Марса, Юпітера, Сатурна, для передбачень затемнення Сонця та Місяця, моделювання орбіти Місяця. Механізм мав 37 зачеплених бронзових шестерень. Пристрої подібної високої складності не будувалися у Європі аж до XIV ст.



Мал. 3.52. Антикітерський механізм: фрагмент знахідки та його комп'ютерна модель

На жаль, пристрої, подібні до антикітерського механізму, і наукові знання, потрібні для їх виготовлення, були на тривалий час утрачені. Активні роботи щодо вдосконалення обчислювальних пристроїв розпочалися тільки в XVII ст. з розвитком промисловості, торгівлі, будівництва, військової справи, морської навігації.

У 1642 р. французький математик, фізик, інженер і філософ **Блез Паскаль** (1623–1662) (мал. 3.53) подав на розгляд королівської ради свій механічний обчислювальний пристрій. У цьому пристрої Б. Паскалем було вперше використано механізм перенесення одиниці в наступний розряд під час додавання. У подальшому він створив кілька десятків таких пристроїв (мал. 3.54). Їх використовували для виконання додавання та віднімання чисел.



Мал. 3.53. **Блез Паскаль**



Мал. 3.54. Обчислювальний пристрій **«Паскаліна»**

Згодом ідеї Паскаля були розвинуті та вдосконалені багатьма математиками та інженерами. Зокрема, німецький учений **Готфрід Лейбніц** (1646–1716) (мал. 3.55) створив перший у світі **арифмометр** – механічний обчислювальний пристрій, що виконував усі чотири арифметичні дії (мал. 3.56).



Мал. 3.55. **Готфрід Лейбніц**



Мал. 3.56. Арифмометр **Готфріда Лейбніца**

Удосконалені варіанти арифмометрів різних конструкцій активно використовували впродовж кількох століть; вони були замінені електронними калькуляторами лише у другій половині XX ст.

Одночасно з удосконаленням арифмометрів учені намагалися створити обчислювальні пристрої, які б могли автоматично здійснювати обчислення, виконуючи команди заздалегідь розробленої програми (мал. 3.57). Одним з перших ідею створення автоматизованої машини для обчислень висловив англійський математик і конструктор **Чарлз Беббідж** (1791–1871) (мал. 3.58). Він назвав її **аналітичною машиною.**



Мал. 3.57. Машинка для обчислень **І. Г. Мюлера**



Мал. 3.58. **Чарлз Беббідж**

За проєктом Ч. Беббіджа аналітична машина мала б такі складові частини:

1. **«Склад»** для зберігання чисел (у сучасній термінології – **пам'ять**).
2. **«Млин»** для виконання арифметичних операцій над числами (у сучасній термінології – **арифметичний пристрій**).
3. Пристрій, який керує послідовністю виконання арифметичних операцій (у сучасній термінології – **пристрій керування**).
4. Пристрій для введення початкових (вхідних) даних.
5. Пристрій виведення результатів.

Ця обчислювальна машина за життя Ч. Беббіджа так і не була створена.

Ада Лавлейс (1815–1852) (мал. 3.59), дочка відомого англійського поета Джорджа Байрона, працювала разом з Беббіджем над створенням проєкту аналітичної машини та його реалізацією. Вона вперше описала основні принципи розробки програм для обчислювальних машин. Ада Лавлейс стояла у витоків програмування, і на її честь названо одну із сучасних мов програмування **Ada**.

До кінця XIX ст. пристрої для обчислень були ручними або механічними. І тільки в кінці XIX ст. американський учений **Герман Голлеріт** (1860–1929) запропонував новий електромеханічний пристрій – **табулятор** (мал. 3.60).



Мал. 3.59. **Ада Лавлейс**



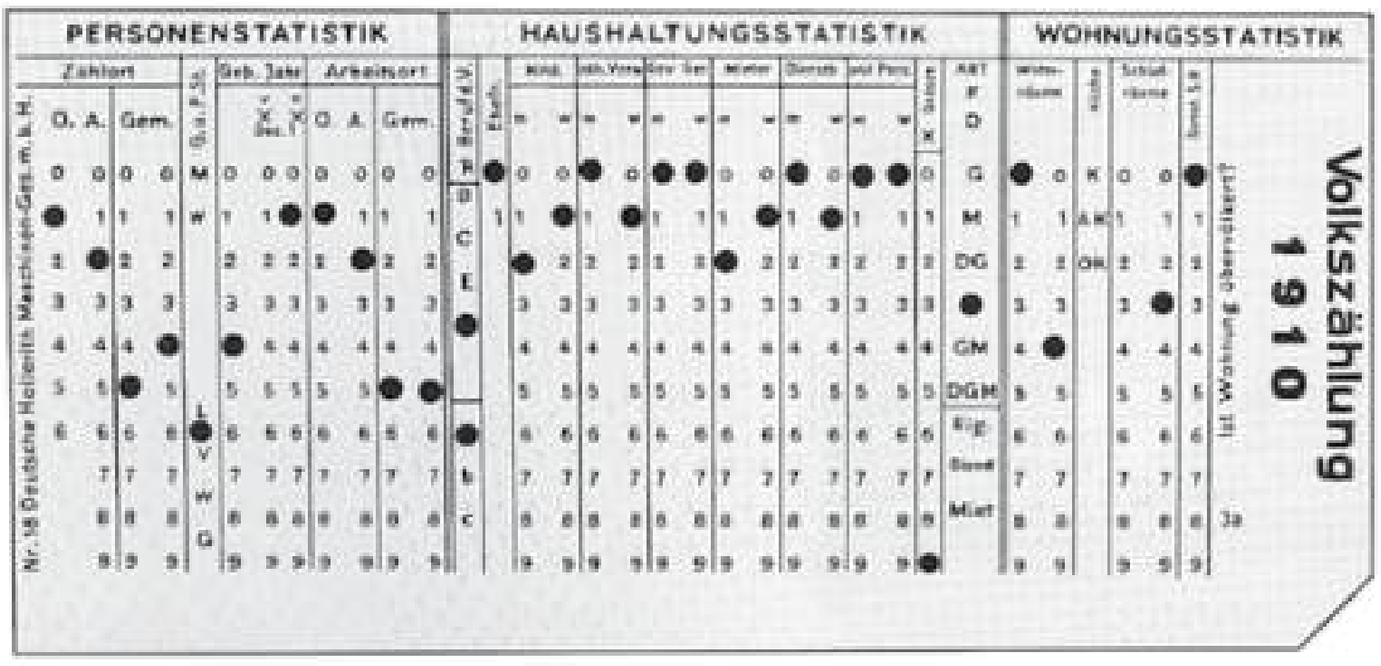
Мал. 3.60. Табулятор **Германа Голлеріта**

Перфорація (лат. *perforate* – продірявлювати) – пробивка отворів, сукупність отворів.

Він був призначений для опрацювання даних перепису населення. Дані про конкретну людину не записувалися на аркуші паперу, а позначалися отворами у строго визначених місцях персональної карти – **перфокарти** (мал. 3.61).

Заснована Голлерітом у 1896 р. фірма з масового випуску табуляторів, зазнавши низки реорганізацій і зміни власників, з 14 лютого 1924 р. стала називатись **IBM Corporation**. Сьогодні – це один з основних виробників потужних комп'ютерів.

До середини XX ст. розроблялися різноманітні конструкції механічних і електромеханічних обчислювальних пристроїв. Вони давали змогу значно пришвидшити процес обчислення.



Мал. 3.61. Перфокарта, що використовувалася під час перепису населення в Німеччині в 1910 році



Мал. 3.62.
К. Цузе

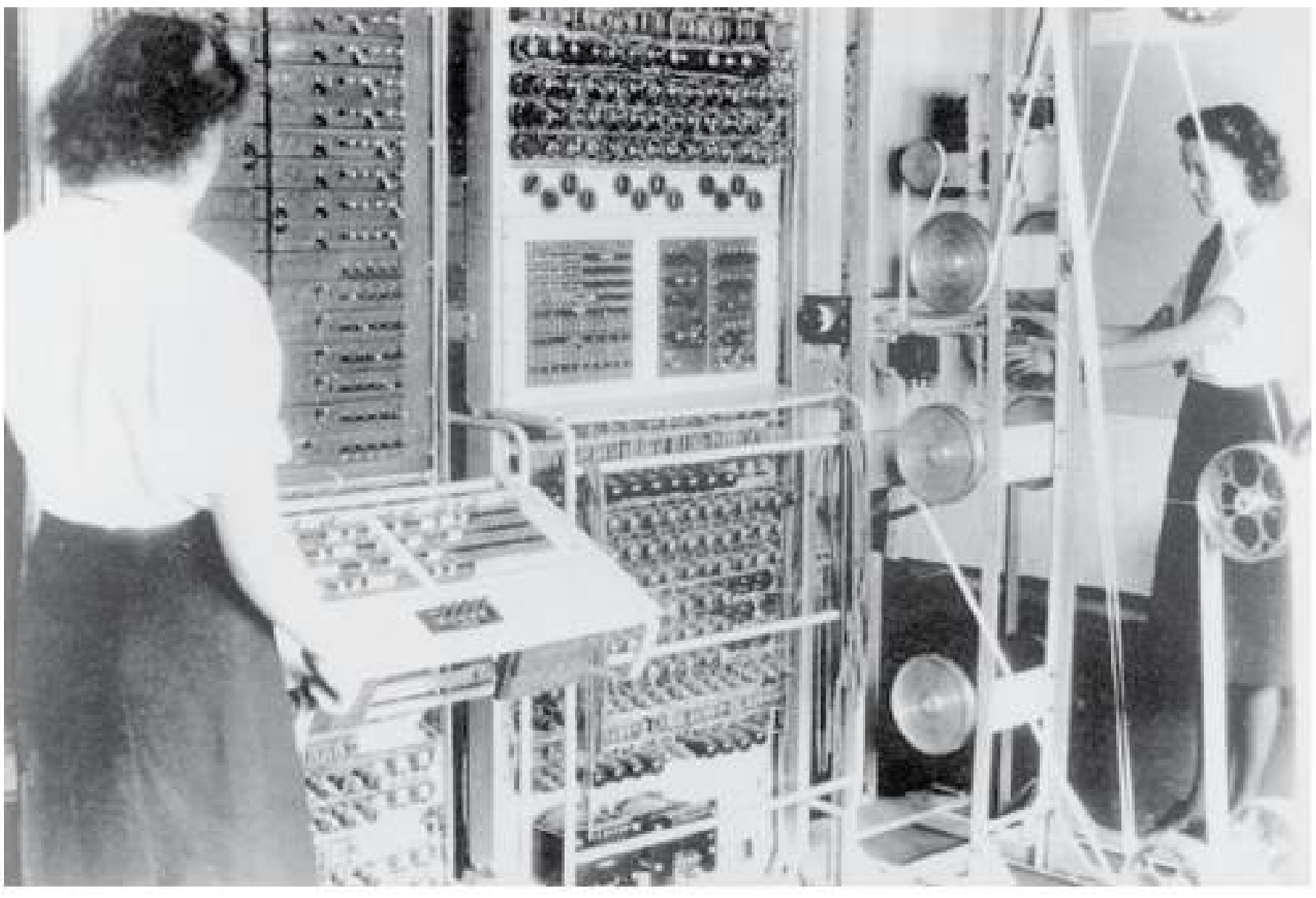
У кінці 30-х і в 40-х роках XX ст. учені та конструктори різних країн створили якісно нові обчислювальні машини, у яких зазвичай використовувалися електричні пристрої. Так, у Німеччині в 1941 р. **Конрад Цузе** (1910–1995) (мал. 3.62) створив першу обчислювальну машину на електромеханічних реле, яку назвав **Z3**. У ній використовувалося двійкове кодування. У 1950 р. він створив електронний комп'ютер – **Z4**, який став першим комерційним продуктом і був проданий Федеральному технологічному інституту у Цюриху (Швейцарія).

У 1941 р. у США **Джон Вінсент Атанасов** (1903–1995) разом зі своїм асистентом **Кліфордом Беррі** (1918–1963) створили першу обчислювальну машину на електронних лампах з використанням двійкового кодування – **ABC** (англ. *Atanasoff Berry Computer*).

У 1944 р. у США **Говард Ейкен** (1900–1973) на замовлення **IBM** створив обчислювальну машину **Mark-1** на електромагнітних реле з автоматичним керуванням послідовністю операцій.

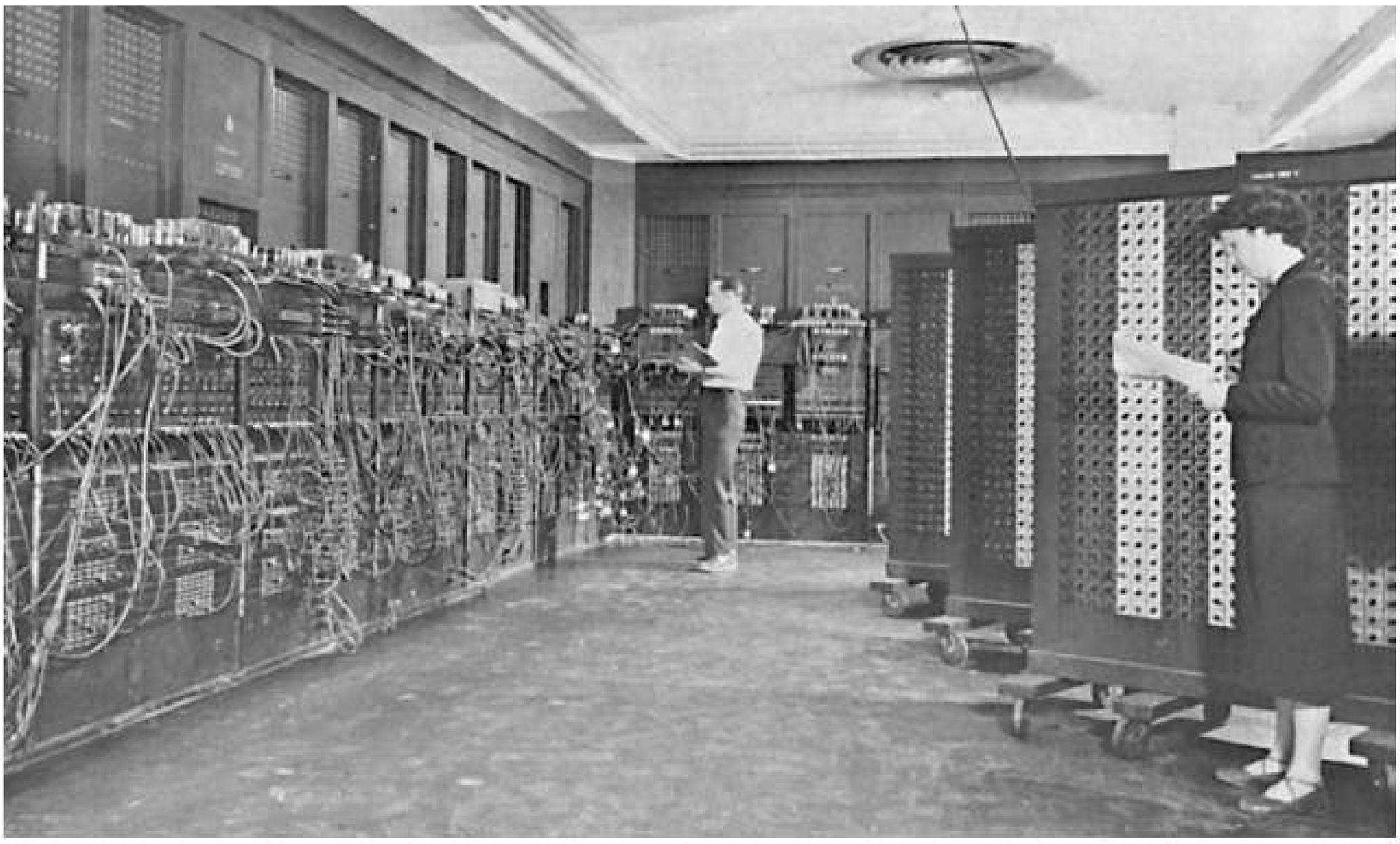
Із середини 40-х років XX ст. розпочинається ера **електронних обчислювальних машин (ЕОМ)**. Із цього часу прийнято розглядати як синоніми терміни **ЕОМ** та **комп'ютер**.

У 1943–1944 рр. в умовах цілковитої таємності вчені Великої Британії створили електронну обчислювальну машину **Colossus** (мал. 3.63), призначену для дешифрування радіограм нацистської Німеччини. Від інших машин того часу її відрізняло те, що програма, за якою вона працювала, зберігалася в пам'яті самої машини, на інших машинах вводили по черзі команди із зовнішніх пристроїв. Розробку здійснила група англійських учених під керівництвом **М. Ньюмена** (1897–1984) та інженера **Т. Флауерса** (1905–1998), з урахуванням математичних підходів до дешифрування даних, які запропонував визначний англійський математик **Алан Тюрінг** (1912–1954).



Мал. 3.63. ЕОМ **Colossus**

У 1943–1946 рр. у США **Джон Моклі** (1907–1980) та **Преспер Еккерт** (1919–1995) створили **ENIAC** (англ. *Electronic Numerical Integrator and Calculator* – електронний цифровий інтегратор і обчислювач) – **електронну обчислювальну машину** (ЕОМ), що містила 18 000 електронних ламп, важила 30 тонн і виконувала п'ять тисяч операцій за секунду (мал. 3.64). Після завершення цього проєкту вони відразу почали роботу над новим комп'ютером на замовлення військового відомства США – **EDVAC** (англ. *Electronic Discrete Variable Automatic Computer* – електронний дискретно змінний автоматичний обчислювач). До розробки було залучено відомого американського математика **Джона фон Неймана** (1903–1957), який сформулював основні принципи побудови універсальної обчислювальної машини. Незважаючи на те, що конструкцію комп'ютера та головні принципи його побудови розробили П. Еккерт і Д. Моклі, а Д. фон Нейман лише узагальнив ці положення, усьому світові принципи побудови комп'ютера стали відомі як ***«принципи фон Неймана»***.



Мал. 3.64. Обчислювальна машина **ENIAC**

Широкого розповсюдження комп'ютери отримали після появи великої кількості порівняно дешевих обчислювальних машин – так званих **міні-ЕОМ**, які в подальшому стали називатися персональними комп'ютерами. Перші з них продавались у вигляді електронних конструкторів, з деталей яких користувач міг самостійно зібрати свій комп'ютер, використавши телевізор як монітор. Першими персональними стали комп'ютери, розроблені фірмами:

- **MITS** (англ. *Micro Instrumentation and Telemetry Systems* – системи мікроінструментів і телеметрії) – комп'ютер ***Altair 8800*** (1975 р., мал. 3.65);
- **Apple Computer Company** – комп'ютер ***Apple I*** (1976 р., мал. 3.66);
- **IBM** – комп'ютер ***IBM PC*** (1981 р., мал. 3.67).



Мал. 3.65. **Altair 8800**



Мал. 3.66. **Apple I**



Мал. 3.67. **IBM PC**

Широке розповсюдження персональних комп'ютерів значно розширило коло задач, які розв'язуються з використанням ЕОМ.

Детальніше історію розвитку обчислювальної техніки можна розглянути за хронологічною таблицею (***Додаток 2***).

ІСТОРІЯ РОЗВИТКУ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ В УКРАЇНІ

Пригадайте

• Коли на території України створили першу електронно-обчислювальну машину? Яку назву вона мала? • Назвіть учених, які зробили суттєвий внесок у розвиток комп'ютерної галузі в Україні.

У 1951 р. в Києві під керівництвом **Сергія Олексійовича Лебедєва** (1902–1974) було введено в дію універсальну ЕОМ з програмою, що зберігалася в пам'яті, – **МЕОМ** (***Мала Електронна Обчислювальна Машина***). Це була перша ЕОМ у Радянському Союзі. Вона мала такі значення властивостей:

- загальна кількість електронних ламп – *приблизно 6000*;
- система кодування даних – *двійкова*;
- кількість розрядів – *16 бітів для кодування числа й один – для кодування знака* (+/–);
- ємність пристрою пам'яті – *31 число* і *63 команди*;
- швидкодія – *приблизно 3000 операцій за хвилину*.

Перші програми для цієї ЕОМ склала **Катерина Логвинівна Ющенко** (1919–2001) (мал. 3.68).



Мал. 3.68. **К. Л. Ющенко**

У Києві розробку обчислювальних машин було продовжено в Інституті кібернетики, який очолив видатний учений **Віктор Михайлович Глушков** (1923–1982). Під його керівництвом:

- завершено розробку ЕОМ **«Київ»** (1959 р., безпосередні розробники Б. В. Гнеденко, Л. М. Дашевський, К. Л. Ющенко);
- створено серію універсальних ЕОМ **«Дніпро»** (1961 р., головний конструктор Б. М. Малиновський);
- створено ЕОМ для інженерних розрахунків **«Промінь»** (1963 р., керівник розробки В. М. Глушков);
- розроблено серію ЕОМ для інженерних розрахунків **МІР** (***Машина для інженерних розрахунків***, 1960-ті роки);
- створено цілий ряд ЕОМ для військових цілей.

За визначні досягнення в розвитку інформатики в 1996 р. Міжнародне комп'ютерне товариство (IEEE Computer Society) нагородило В. М. Глушкова медаллю **«Комп'ютерний піонер»** (мал. 3.69).



Мал. 3.69. Лицьова та зворотна сторона медалі В. М. Глушкова **«Комп'ютерний піонер»**

Вагомий внесок у розвиток комп'ютерної техніки для ракетної галузі зробили колективи науково-виробничих об'єднань, конструкторських бюро з різних міст України. Так, у Сєвєродонецькому НВО «Імпульс» вироблялися електронні обчислювальні машини для систем автоматизованого керування виробництвом **М6000–М7000**, клавішні ЕОМ **«Іскра»**, засоби для автоматичного керування балістичними ракетами.

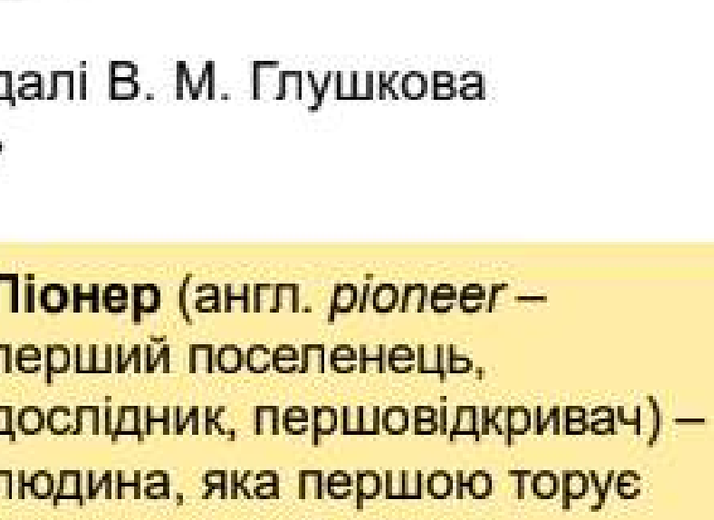
У Київському НВО «Кристал» було спроєктовано, а в м. Світловодську вироблено перші в колишньому Радянському Союзі та Європі мікрокалькулятори **«Електроніка»** (мал. 3.70).

У 70–90-ті роки XX ст. Харківське НВО «Хартрон» і Київський радіозавод розробляли і виготовляли ЕОМ для ракетно-космічних комплексів.

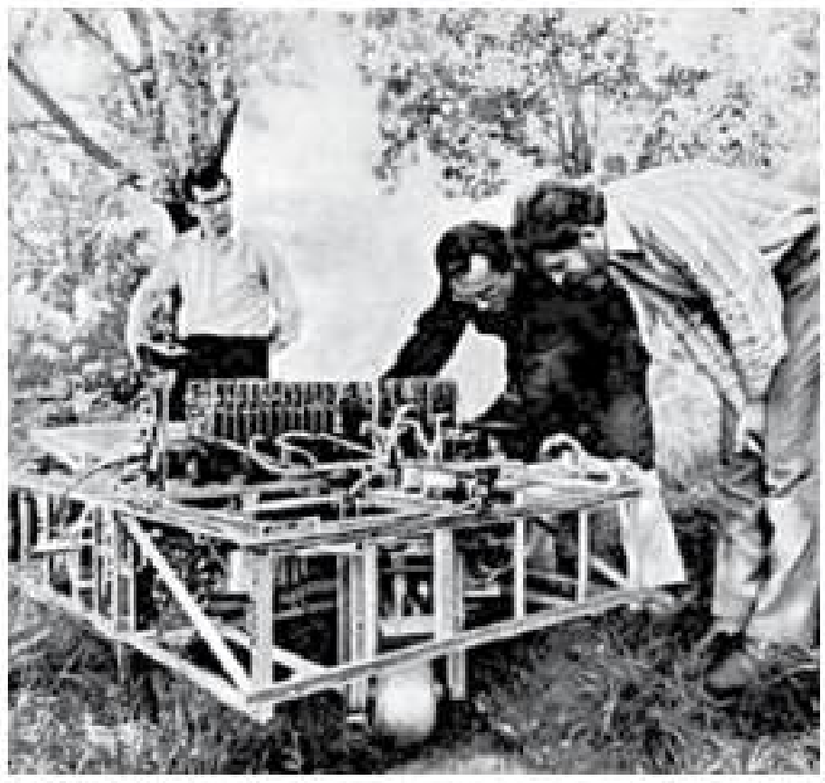
Було розроблено і налагоджено (Київський науково-дослідний інститут радіоелектроніки та Київський завод «Буревісник», 70–80-ті роки XX ст.) серійний випуск ЕОМ **«Карат»** для керування кораблями та підводними човнами.

У 1975 р. в Інституті кібернетики АН УРСР під керівництвом М. М. Амосова був спроєктований перший у СРСР автономний транспортний ро́бот **«ТАІР»** (мал. 3.71), який міг рухатися у природному середовищі, оминаючи перепони.

Піонер (англ. *pioneer* – перший поселенець, дослідник, першовідкривач) – людина, яка першою торує стежку в новій галузі діяльності.



Мал. 3.70. Мікрокалькулятор **«Електроніка»**



Мал. 3.71. Конструктори налагоджують ро́бота **«ТАІР»**



Мал. 3.72. Пуск ракети **Р-36М2**

Учені та інженери України протягом 60–80-х років XX ст. розробили багато комп'ютерів для різноманітних ракетних комплексів. Так, для найбільшої у світі балістичної ракети **Р-36М2** («Сатана») виробництва Дніпропетровського НВО «Південний машинобудівний завод» спеціалістами Харківського НВО «Електроприлад» і ВО «Київський радіозавод» створено ЕОМ, яка забезпечувала керування її запуском і польотом (мал. 3.72).

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Je2cSvVA* або QR-кодом.

Найважливіше в цьому пункті

Перші пристрої для проведення обчислень були створені людиною кілька тисячоліть тому. Виділяють кілька етапів розвитку засобів реалізації інформаційних процесів: етап ручних засобів, етап механічних засобів, етап електричних засобів і етап електронних засобів.

Із середини 40-х років XX ст. розпочинається ера **електронних обчислювальних машин (ЕОМ)**. Із цього часу прийнято розглядати як синоніми терміни **ЕОМ** та **комп'ютер**.

Вагомий внесок у розвиток обчислювальної техніки зробили Б. Паскаль, Г. Лейбніц, Ч. Беббідж, А. Лавлейс, Г. Голлеріт. Перші електронні обчислювальні машини були створені в 40–50-х роках XX ст.: К. Цузе (**Z4**), Д. Атанасовим і К. Беррі (**АВС**), Г. Ейкеном (**Mark-1**), групою англійських учених (**Colossus**), Д. Моклі та П. Еккертом (**ENIAC**), колективом радянських учених під керівництвом С. Лебедєва (**МЭСМ**).

В Україні розроблено перші в Радянському Союзі комп'ютери. До розробки ЕОМ для різних галузей економіки та для керування військовою технікою були залучені десятки підприємств і організацій з різних куточків України. Провідною організацією став Інститут кібернетики Академії наук України під керівництвом В. Глушкова.

Дайте відповіді на запитання

1. Які етапи розвитку інформаційних технологій ви знаєте?

2. Які носії даних використовувались на кожному з етапів розвитку ІТ?

3. Як змінювалися засоби опрацювання повідомлень на різних етапах розвитку інформаційних технологій?

4. У чому заслуга Б. Паскаля в розвитку пристроїв для проведення обчислень?

5. Чим відрізнявся арифмометр Г. Лейбніца від пристрою Б. Паскаля?

6. Які перші персональні комп'ютери ви знаєте? Коли вони були створені?

Обговоріть і зробіть висновки

1. Які складові універсальної машини для обчислень, що запропонував Ч. Беббідж? Що спільного в машини Ч. Беббіджа та сучасного комп'ютера?

2. Чим характеризувався період створення перших електронних обчислювальних машин (40–50 рр. XX ст.)? Що, на вашу думку, сприяло появі цих машин?

3. Як розвиток комп'ютерної техніки в Україні був пов'язаний з її промисловим потенціалом?

Виконайте завдання

1. Підготуйте хронологічну схему розвитку комп'ютерів, починаючи із середини XX ст. до кінця XX ст. Використайте для цього текстовий процесор. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.5.1.**

2. Знайдіть в інтернеті або друкованих ресурсах відомості про перші електронні обчислювальні машини. Створіть у текстовому процесорі порівняльну таблицю та заповніть її.

Властивості	***Значення властивостей ЕОМ***		
	Z4	***Colossus***	***ABC***
Розробники			
Дата створення			
Країна			
Використання програмного способу керування			
Використання двійкового кодування			
Зберігання програми в пам'яті			

Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.5.2.**

3. Порівняйте значення основних властивостей перших електронних обчислювальних машин: американської **ENIAC** і створеної в Україні **МЭСМ.** Створіть у текстовому процесорі порівняльну таблицю та заповніть її. Збережіть текстовий документ у вашій папці у файлі з іменем **завдання 3.5.3.**

4. Підготуйте відеоповідомлення про розробку українськими вченими ЕОМ для керування атомними криголамами. Збережіть відео у вашій папці у файлі з іменем **завдання 3.5.4.avi.**

5. Підготуйте повідомлення про особливості використання програмування в перших комп'ютерах, створених в Україні. Використайте в повідомленні інформацію про К. Ющенко (наприклад, за посиланням *https://dou.ua/lenta/interviews/about-kateryna-yushchenko/*). Збе-

режіть повідомлення в текстовому документі у вашій папці у файлі з іменем **завдання 3.5.5.**

6. Підготуйте повідомлення про перші персональні комп'ютери, що створювалися в Україні у 80-ті роки XX ст., наприклад комп'ютери **«Нейрон»**, **«Пошук»**, **«СМ 1810»**, **«Олімпік-С»**, **«УМПК-Р32»**, **«ЛиК»**, **«ПК-01» «Львів»**. Збережіть повідомлення в текстовому документі у вашій папці у файлі з іменем **завдання 3.5.6.**



Готуємось до вивчення нового матеріалу

1. Що таке *комп'ютерна програма*?
2. Які засоби використовують для створення комп'ютерних програм?
3. Яку мову програмування ви вивчали? У якому середовищі працювали? Як створити та виконати проєкт мовою програмування?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Апаратне забезпечення персонального комп'ютера»** *(початковий, середній, достатній, високий рівень).*

- Я можу назвати професії та галузі, у яких використовуються цифрові технології.
- Я можу аргументувати добір комп'ютерів для розв'язування задач з різних галузей.
- Я розумію, які етапи розвитку засобів опрацювання, передавання та зберігання даних пройшло людство та який вплив на освіту, виробництво, суспільство, культуру мав їх розвиток.
- Я можу назвати приклади пристроїв, що використовувалися на різних етапах історичних змін інформаційних технологій.
- Я періодично цікавлюся новими комп'ютерними пристроями, значеннями їх властивостей, їх новими функціями.
- Я можу обґрунтувати вибір апаратного чи програмного засобу для розв'язування певної задачі.
- Я вмію добирати компоненти комп'ютерної системи для заданої задачі, використовуючи розроблені критерії.
- Я вмію обирати, поєднувати та налаштовувати програмні та технічні засоби відповідно до потреб, характеристик / параметрів задачі та наявних обмежень.
- Я використовую стандартні засоби діагностики для виявлення джерела апаратної і/або програмної проблеми комп'ютерної системи.

Повторіть той матеріал, який ви знаєте недостатньо.



Практична робота № 3. «Добір складових персонального комп'ютера залежно від його призначення»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Доберіть конфігурацію комп'ютерів для людей різних професій і різних сфер застосування:
   - письменник – працює у своєму кабінеті, використовує комп'ютер для створення прозових творів, пошуку відомостей, для надсилання й отримання електронних листів тощо;

- студент – використовує комп'ютер для виконання навчальних завдань з гуманітарних дисциплін, а також для створення власної фонотеки, створення та редагування навчальних відеофільмів;
- кореспондент інтернет-видання – використовує комп'ютер для написання та своєчасного надсилання в редакцію повідомлень з новинами під час відряджень у різні частини світу.

2. Створіть у текстовому процесорі документ з таблицею значень властивостей комп'ютерів. Заповніть цю таблицю дібраними значеннями.

Властивість	***Значення властивості для комп'ютера***		
	письменника	***студента***	***кореспондента***
Тип комп'ютера			
Модель процесора			
Тактова частота процесора, *ГГц*			
Обсяг оперативної пам'яті, *ГБ*			
Ємність вбудованого пристрою зовнішньої пам'яті, *ГБ*			
Монітор			
Діагональ монітора, *дюйма*			
Роздільність			
Час реакції, *мс*			
Орієнтовна ціна, *грн*			

3. У цьому самому файлі обґрунтуйте ваш вибір конфігурації комп'ютерів.
4. Документ з таблицею та обґрунтуванням збережіть у вашій папці у файлі з іменем **практична 3.docx** і надішліть на електронну пошту вчителя / вчительки.

РОЗДІЛ 4

АЛГОРИТМИ ТА ПРОГРАМИ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- етапи розв'язування задач з використанням алгоритмів і проєктів, створених мовою програмування;
- елементи керування **кнопки**, **написи**, **поля**, **прапорці**, **перемикачі**, їх властивості, події, що можуть відбуватися з ними, їх використання у проєктах;
- встановлення та змінення значень властивостей елементів керування під час виконання проєкту;
- лінійні алгоритми та проєкти;
- логічні величини, логічні вирази, логічні операції: заперечення, кон'юнкція, диз'юнкція;
- алгоритми та проєкти з розгалуженнями;
- алгоритми та проєкти із циклами;
- алгоритми та проєкти з розгалуженнями та циклами;
- використання проєктів для проведення комп'ютерних експериментів.

У підручнику цей розділ викладено з використанням мови програмування **Python** і середовища створення та редагування проєктів **IDLE**.

Текст цього розділу з використанням мови програмування **Object Pascal** і середовища створення та редагування проєктів **Lazarus** розміщено за посиланням *https://cutt.ly/ceL9HV2G* або QR-кодом.

4.1. ЕТАПИ РОЗВ'ЯЗУВАННЯ ЗАДАЧ З ВИКОРИСТАННЯМ АЛГОРИТМІВ І ПРОЄКТІВ

У цьому пункті йтиметься про:

- етапи розв'язування задач з використанням алгоритмів і проєктів, створених мовою програмування на основі складених алгоритмів;
- тестові набори даних;
- види помилок під час виконання проєкту;
- тестування проєктів, налагодження проєктів, проведення комп'ютерних експериментів.

ЕТАПИ РОЗВ'ЯЗУВАННЯ ЗАДАЧ З ВИКОРИСТАННЯМ АЛГОРИТМІВ І ПРОЄКТІВ МОВОЮ ПРОГРАМУВАННЯ, СТВОРЕНИХ НА ОСНОВІ СКЛАДЕНИХ АЛГОРИТМІВ

Пригадайте

● Які ви знаєте етапи роботи над проєктом, що створюється мовою програмування? ● Що таке *модель об'єкта*? Для чого її створюють? ● Які види моделей ви знаєте? Чим характерна математична модель?

У попередніх класах ви вже розв'язували різноманітні задачі, створюючи алгоритми та проєкти мовою програмування на основі цих алгоритмів.

Розглянемо детальніше, з яких етапів має складатися розв'язування задач на обчислення з використанням алгоритмів і проєктів мовою програмування на основі цих алгоритмів.



Етап 1. Уважно прочитати умову задачі, проаналізувати її та створити математичну модель задачі. Тобто визначити:

1. Які дані є вхідними?
2. Які дані будуть кінцевими результатами?
3. Формули, використовуючи які можна отримати значення кінцевих результатів.

Етап 2. Створити алгоритм розв'язування задачі. Найчастіше алгоритм подають у вигляді блок-схеми. Але використовують й інші способи подання алгоритму, зокрема словесний або комбінований.

Етап 3. Створити комп'ютерний проєкт мовою програмування, у якому використовується створений алгоритм.

Етап 4. Запустити проєкт на виконання для кількох довільних наборів значень вхідних даних, щоб перевірити, чи видає він значення кінцевих результатів. Якщо виконання проєкту переривається, то це означає, що проєкт містить **синтаксичні помилки.** У такому разі потрібно їх виправити і знову запустити проєкт на виконання.

Етап 5. Запустити проєкт на виконання для таких наборів значень вхідних даних, щоб можна було б знайти значення кінцевих результатів без виконання проєкту і порівняти їх з тими значеннями кінцевих результатів, які видає комп'ютер у результаті виконання проєкту.

Якщо очікувані нами значення кінцевих результатів не збігаються з тими, які видає комп'ютер, то це означає, що проєкт містить **логічні помилки.** Потрібно їх визначити, виправити і знову запустити проєкт на виконання для тих самих наборів значень вхідних даних.

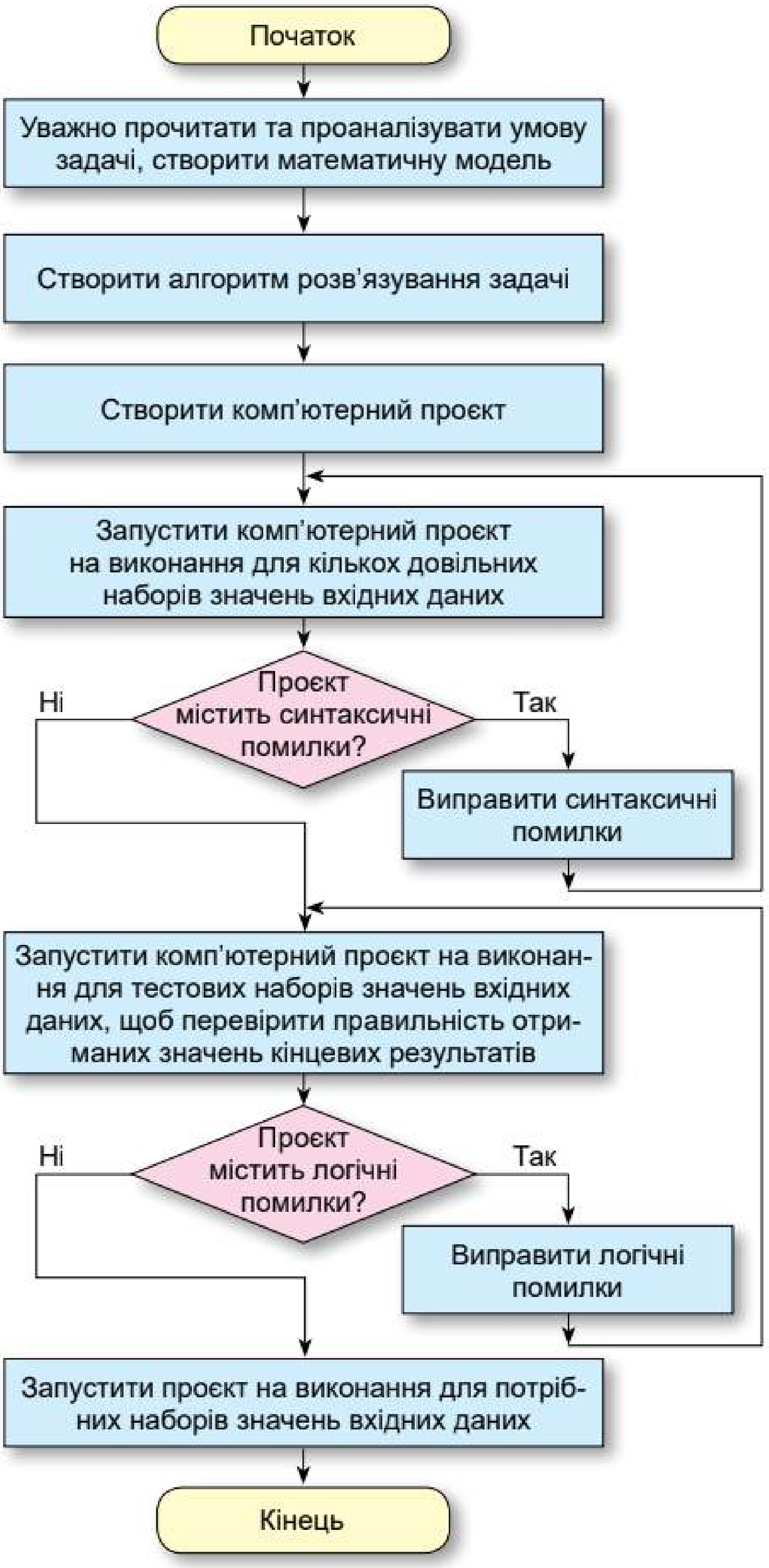
Такі набори значень вхідних даних мають бути дібрані так, щоб перевірити правильність виконання всіх складових частин проєкту. Зокрема, якщо проєкт містить розгалуження, то потрібно дібрати такі набори значень вхідних даних, щоб перевірити правильність роботи проєкту для всіх можливих випадків виконання розгалужень.

Набори значень вхідних даних, які використовуються для перевірки правильності проєкту, називаються **тестовими наборами.**

Етап 6. Запустити проєкт на виконання для наборів значень вхідних даних, які вказані в умові задачі, або інших значень і отримати значення кінцевих результатів.

Звертаємо вашу увагу: під час проведення **Етапу 6** можна висловлювати гіпотези і, виконуючи проєкт для різних наборів значень вхідних даних, підтвердити їх або заперечити. Тобто при проведенні **Етапу 6** можна здійснювати **комп'ютерний експеримент.**

Перераховані вище етапи та послідовність їх виконання можна подати у вигляді блок-схеми (мал. 4.1).



Мал. 4.1. Етапи розв'язування задач з використанням алгоритмів і проєктів



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Se2cSXqw* або QR-кодом.



Найважливіше в цьому пункті

Етапи розв'язування задач з використанням алгоритмів і проєктів мовою програмування, створених на основі складених алгоритмів:

Етап 1. Уважно прочитати умову задачі, проаналізувати її та створити математичну модель задачі. Тобто визначити:

1. Які дані є вхідними?
2. Які дані будуть кінцевими результатами?
3. Формули, використовуючи які можна отримати значення кінцевих результатів.

Етап 2. Створити алгоритм розв'язування задачі. Найчастіше алгоритм подають у вигляді блок-схеми.

Етап 3. Створити комп'ютерний проєкт мовою програмування, у якому використовується створений алгоритм.

Етап 4. Запустити проєкт на виконання для кількох довільних наборів значень вхідних даних, щоб перевірити, чи видає він значення кінцевих результатів.

Етап 5. Запустити проєкт на виконання для таких наборів значень вхідних даних, щоб можна було б знайти значення кінцевих результатів без виконання проєкту і порівняти їх з тими значеннями кінцевих результатів, які видає комп'ютер у результаті виконання проєкту.

Етап 6. Запустити проєкт на виконання для наборів значень вхідних даних, які вказані в умові задачі, або інших значень і отримати значення кінцевих результатів.

Якщо виконання проєкту переривається, то проєкт містить **синтаксичні помилки**.

Якщо значення результатів виконання проєкту не збігаються з очікуваними, то проєкт містить **логічні помилки**.

Набори значень вхідних даних, які використовуються для визначення правильності роботи проєкту, називаються **тестовими наборами**.



Дайте відповіді на запитання

1. З яких етапів складається розв'язування задач на обчислення з використанням алгоритмів і проєктів мовою програмування, створених на основі відповідних алгоритмів?

2. Що потрібно виконати на кожному з етапів розв'язування задач на обчислення з використанням алгоритмів і проєктів мовою програмування, створених на основі відповідних алгоритмів? Коротко поясніть.

3. Помилки яких видів може містити проєкт? У чому полягає кожний з них?

4. Які набори значень вхідних даних називають тестовими?



Обговоріть і зробіть висновки

1. Як визначити мінімальну кількість тестових наборів значень вхідних даних, щоб якісно протестувати проєкт?

2. Чи можна дібрати набори значень вхідних даних і скільки їх потрібно, щоб на 100 % гарантувати, що проєкт завжди працюватиме правильно?

3. Що буде, якщо не виправити синтаксичну помилку? Логічну помилку?

Виконайте завдання

1. Щоб дістатися з дому на тренування, Василю потрібно дійти до автобуса і потім проїхати ним до стадіону. Який час він витратить на шлях від дому до стадіону, якщо відомо відстані, які проходить Василь і проїжджає автобус, і швидкості руху Василя й автобуса? Виконайте всі етапи розв'язування задачі. Математичну модель і алгоритм запишіть у зошит. Висловіть гіпотезу. Перевірте її, використовуючи створений проєкт. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.1.1**.

2. Туристам потрібно за три дні пройти ***s*** км. У І день вони йшли ***t1*** год зі швидкістю ***v1*** км/год, у ІІ день вони йшли ***t2*** год зі швидкістю ***v2*** км/год. З якою швидкістю їм потрібно рухатися в ІІІ день, щоб за ***t3*** год пройти потрібний шлях? Виконайте всі етапи розв'язування задачі. Математичну модель і алгоритм запишіть у зошит. Висловіть гіпотезу. Перевірте її, використовуючи створений проєкт. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.1.2**.

3. В Улянки є кулька, для якої відомі її радіус ***R*** і маса ***m1***. Улянка занурює кульку в посудину з рідиною. Відомі маса витісненої рідини ***m2*** та її об'єм ***v2***. Визначити, чи буде плавати кулька в рідині, чи потоне, чи спливатиме. Об'єм кульки обчислюється за формулою $4/3\pi R^3$. Виконайте всі етапи розв'язування задачі. Математичну модель і алгоритм запишіть у зошит. Висловіть гіпотезу. Перевірте її, використовуючи створений проєкт. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.1.3**.

4. Петрик хоче запросити свого друга в кіно і пригостити його морозивом. Мати дала йому для цього ***x*** грн. Чи вистачить грошей у Петрика на 2 квитки в кіно і на 2 порції морозива? Виконайте всі етапи розв'язування задачі. Математичну модель і алгоритм запишіть у зошит. Висловіть гіпотезу. Перевірте її, використовуючи створений проєкт. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.1.4**.

Готуємось до вивчення нового матеріалу

1. Наведіть приклади властивостей вікна виконання проєкту та їх значень.
2. Які події можуть відбуватися з вікном виконання проєкту?
3. Що таке *обробник події*? Як він записується і виконується у проєкті?
4. Для чого у проєктах використовується елемент керування *кнопка*?

4.2. КНОПКА

У цьому пункті йтиметься про:

- кнопку як елемент керування, її властивості та їх значення;
- створення кнопки та способи її розміщення у вікні;
- події, які можуть відбуватися з кнопкою;
- обробники подій з кнопкою.

ВЛАСТИВОСТІ КНОПКИ



Пригадайте

● Які ви знаєте властивості вікна? Наведіть приклади їх значень. ● Якою командою створюється та відкривається вікно? ● Як змінити значення властивостей вікна? ● Для чого використовують *кнопку* в проєктах?

Одним з елементів керування, який можна використати у віконних проєктах, є **кнопка**.

Щоб розмістити кнопку в уже створеному та відкритому вікні, потрібно:

1. Створити новий об'єкт типу **Button** (англ. *button* – кнопка), пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
2. Установити значення властивостей кнопки або залишити їх за замовчуванням.
3. Розмістити створений об'єкт у вікні.

Для створення кнопки та пов'язування її зі змінною використовують команду вигляду:

<ім'я_кнопки> = Button()

або

<ім'я_кнопки> = Button(<набір властивостей та їх значень>)

У першому випадку буде створено кнопку зі значеннями властивостей **висота**, **ширина** і **колір**, установленими за замовчуванням, без тексту на ній. У другому випадку буде створено кнопку, значення властивостей якої задано у цій команді. У дужках через кому вказують пари **<властивість> = <значення>**.

Для кнопки можуть бути встановлені значення таких властивостей:

- **text** (англ. *text* – текст) – текст на кнопці; значенням цієї властивості може бути довільний текст; цей текст потрібно брати в лапки; наприклад, **text = 'Обчислити'**;
- **width** (англ. *width* – ширина) – ширина кнопки; значенням цієї властивості може бути натуральне число – кількість символів, які можуть бути розміщені поруч по горизонталі; наприклад, **width = 20**;
- **height** (англ. *height* – висота) – висота кнопки; значенням цієї властивості може бути натуральне число – кількість символів, які можуть бути розміщені поруч по вертикалі (кількість рядків); наприклад, **height = 2**;
- **bg**, або **background** (англ. *background* – задній план, фон), – колір фону; значенням цієї властивості може бути назва кольору англійською мовою; цю назву потрібно брати в лапки; наприклад, **bg = 'red'**;
- **fg**, або **foreground** (англ. *foreground* – передній план), – колір символів тексту; значенням цієї властивості може бути назва кольору англійською мовою; цю назву потрібно брати в лапки; наприклад, **fg = 'green'**;
- **font** (англ. *font* – шрифт) – шрифт символів тексту на кнопці; значення цієї властивості може містити назву шрифту, розмір

символів і накреслення: **bold** (англ. *bold* – жирний) – напівжирний, **italic** (англ. *italic* – курсив) – курсивний, **underline** (англ. *underline* – підкреслений) – підкреслений;
наприклад, **font = 'Arial 14'** або **font = ('Verdana', 13, 'bold')**;

- **state** (англ. *state* – стан) – стан кнопки; значеннями цієї властивості можуть бути: **normal** (англ. *normal* – нормальний) – звичайний, **disable** (англ. *disable* – відключити, зробити неспроможним) – недоступний; ці значення потрібно брати в лапки, наприклад, **state = 'normal'**

та інших.

Наприклад, кнопка з іменем **bt** з текстом *Змінити* завширшки *15* символів може бути створена командою:

bt = Button(text = 'Змінити', width = 15).

Звертаємо вашу увагу: якщо під час створення кнопки значення властивості **width** не задається, то ширина кнопки автоматично визначається довжиною тексту, який є значенням властивості **text.**

Значення властивостей кнопки також можна задавати окремими командами виду: **<ім'я кнопки>['<назва властивості>'] = <значення властивості>**. Наприклад, установити для кнопки з іменем **btn** ширину 25 символів можна командою **btn['width'] = 25**, установити шрифт *Times New Roman* з висотою символів 28 можна командою **btn['font'] = 'Times New Roman, 28'**, задати синій колір символів – командою **btn['fg'] = 'blue'**. Такого виду команди можна використовувати і для змінення значень властивостей кнопки під час виконання проєкту.

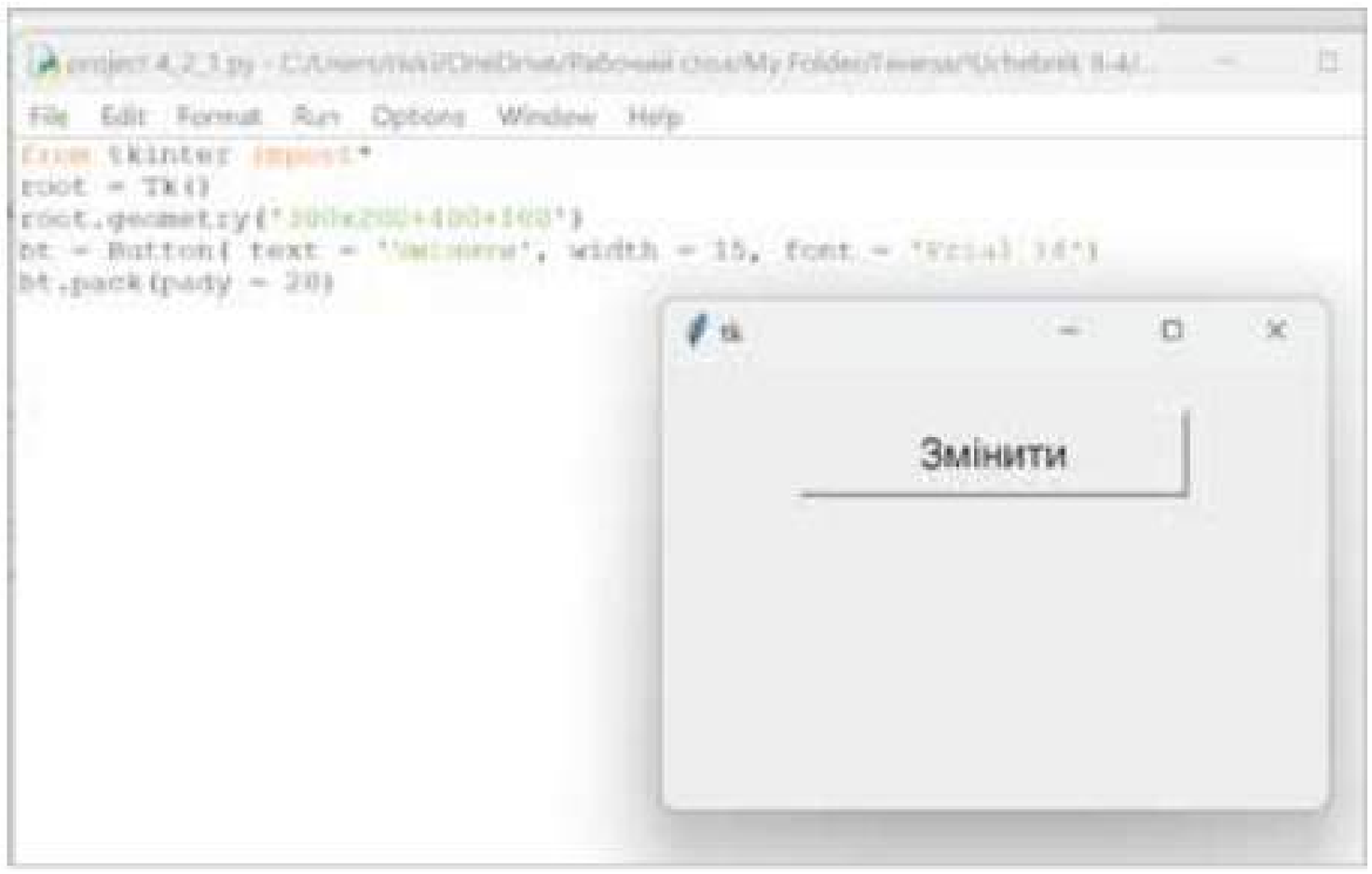
Для розміщення кнопки у вікні можна використати один з двох методів:

- **pack()** (англ. *pack* – упаковка) – кнопка автоматично розміщується під рядком заголовка вікна з однаковими відступами від лівої та правої межі вікна. Якщо у вікні розміщується кілька кнопок, то вони будуть розташовані одна під одною без відступів між собою. Якщо потрібно розмістити кнопку з деяким відступом від рядка заголовка або від попередньої кнопки, то в дужках можна вказати цей відступ у пікселях як значення властивості **pady** (англ. *pady* – відступ). Наприклад, після виконання команди **button.pack(pady = 20)** кнопка буде розміщена з відступом 20 пікселів від верхньої межі вікна або від попередньої кнопки;
- **place(x = <відступ від лівої межі вікна>, y = <відступ від верхньої межі вікна>)** (англ. *place* – місце) – верхня ліва вершина кнопки розміщується в точці вікна із заданими координатами **x** та **y**.

Нагадуємо: початок системи координат у вікні розміщується в його лівій верхній вершині, вісь ***x*** розташована горизонтально вправо, вісь ***y*** – вертикально вниз.

Наприклад, після виконання команди **button.place(x = 200, y = 100)** верхня ліва вершина кнопки з іменем **button** буде розміщена в точці вікна з координатами (200, 100).

На малюнку 4.2 наведено текст проєкту зі створення та розміщення кнопки з іменем **bt** у вікні та результат його виконання.



Мал. 4.2. Кнопка у вікні

Для тих, хто хоче знати більше

Під час виконання проєкту може виникнути потреба, наприклад у іграх, зробити кнопку невидимою. Для цього, залежно від методу розміщення кнопки, можна використати методи **pack_forget()** або **place_forget()**. Щоб зробити кнопку знову видимою, слід повторно застосувати метод **pack()** або **place()**.

ПОДІЇ З КНОПКОЮ. ОБРОБНИКИ ПОДІЙ ДЛЯ КНОПКИ

Пригадайте

- Які події можуть відбуватися з вікном проєкту?
- Що таке *обробник події*?
- Як записується обробник події у проєкті та як він виконується?

Як і з вікном, з кнопкою можуть відбуватися різні події. І з кожною такою подією можна пов'язати її обробник, який буде виконуватися одразу після настання відповідної події.

Пов'язати з кнопкою обробник будь-якої події можна так само, як з вікном, використовуючи метод **bind()**. Наприклад, на малюнку 4.3 наведено приклад обробника події **Click** для кнопки з іменем **button**, який містить команду встановлення жовтого кольору вікна.

```
from tkinter import *
def click(event):
    root['bg'] = 'yellow'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Змінити', width = 15)
button.pack(pady = 30)
button.bind('<1>', click)
```

Мал. 4.3. Обробник події **Click** для кнопки

Для кнопки подією за замовчуванням є *клацання лівої кнопки миші*. Тому пов'язати з кнопкою обробник саме цієї події можна простіше:

- під час створення кнопки вказати ім'я відповідного обробника події як значення властивості **command**;
- обробник події з указаним іменем описати перед створенням кнопки; у заголовку обробника події не потрібно вказувати посилання на подію, що відбулася, – **event**.

На малюнку 4.4 наведено приклад обробника події **Click** для кнопки – процедури з іменем *click( )*, яка містить команди:

- установлення для кнопки *червоного* кольору фону;
- установлення шрифту *Arial* розміром *14* пікселів;
- виведення на кнопці вказаного тексту;
- установлення для кнопки стану *недоступна*.

```
from tkinter import *
def click():
    button['bg'] = 'red'
    button['font'] = 'Arial 14'
    button['text'] = 'Змінюються значення властивостей кнопки'
    button['state'] = 'disable'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Вибери мене', command = click)
button.pack(pady = 30)
```

Мал. 4.4. Обробник події **Click** для кнопки

Звертаємо вашу увагу:

1. Ім'я обробника події – значення властивості **command** – записувати потрібно обов'язково без лапок.
2. Значення властивостей кнопки в обробнику подій можна змінювати так само, як значення властивостей вікна (указати назву властивості у квадратних дужках у лапках, а нове значення властивості – після знаку =).
3. В обробнику події можна змінювати значення властивостей як самої кнопки, так й інших об'єктів (вікна, інших кнопок).
4. Розташування кнопки у вікні можна змінювати в обробнику події, якщо застосувати до кнопки метод **place** з новими значеннями x та y.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/We2cDohZ* або QR-кодом.



Найважливіше в цьому пункті

Щоб розмістити кнопку у вікні, потрібно створити новий об'єкт типу **Button**, пов'язати його зі змінною, що визначатиме ім'я кнопки, установити значення властивостей кнопки, розмістити створений об'єкт у вікні.

Для створення кнопки та пов'язування її зі змінною використовують команду вигляду **<ім'я_кнопки> = Button()** або **<ім'я кнопки> = Button(<набір властивостей та їх значень>)**.

Під час створення кнопки можуть бути встановлені значення її властивостей **text**, **width**, **height**, **bg**, **fg**, **font**, **state**. Для цього потрібно ввести в дужках через кому пари **<властивість> = <значення>**. Для кнопки можуть бути встановлені значення властивостей **text**, **width**, **height**, **bg**, **fg**, **font**, **state** та інших.

Для розміщення кнопки у вікні можуть бути використані методи **pack()** або **place()**.

Пов'язати з кнопкою обробник будь-якої події можна так само, як з вікном, використовуючи метод **bind()**.

Щоб пов'язати з кнопкою метод опрацювання події **Click**, можна в команді створення кнопки вказати ім'я відповідної процедури – обробника події як значення властивості **command**. У заголовку цієї процедури не потрібно вводити ключове слово **event**.



Дайте відповіді на запитання

1. Як створити кнопку?
2. Як розмістити створену кнопку у вікні?
3. Які ви знаєте властивості кнопки? Що визначає кожна з них?
4. Як задати значення властивостей кнопки?
5. Якими способами можна призначити для кнопки обробники подій?
6. Як можна призначити для кнопки обробник події **Click**?
7. Як у проєкті змінити значення властивостей кнопки?



Обговоріть і зробіть висновки

1. Чим відрізняється у проєкті створення та відкриття вікна від створення та розміщення кнопки?
2. Що спільного і відмінного в обробниках події для вікна і для кнопки?



Виконайте завдання

1. Створіть проєкт, у якому в результаті настання події **Click** для кнопки її ширина стане 30 символів, висота 3 символи і зміниться текст на кнопці на назву вашого населеного пункту. Початкові значення властивостей кнопки залиште за замовчуванням. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.1**.
2. Створіть проєкт, у якому в результаті настання події **MouseMove** для кнопки її ширина стане 20 символів, висота 4 символи і зміниться текст на кнопці на ваше прізвище. Початкові значення властивостей кнопки виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.2**.
3. Створіть проєкт, у якому в результаті настання події **DoubleClick** для кнопки зміниться її ширина і висота, текст на кнопці, колір фону і колір тексту, а для вікна – його розміри, колір фону, текст у рядку заголовка. Значення цих властивостей виберіть самостійно. Початкові значення властивостей кнопки залиште за замовчуванням. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.3**.
4. Створіть проєкт, у якому розташуйте у вікні три кнопки однакових розмірів одна під одною. Початковий колір верхньої кнопки – червоний, двох інших – чорний. Створіть обробники події **Click** для кожної кнопки такі, щоб:
   - вибір середньої кнопки робив її жовтою, а дві інші – чорними;
   - вибір нижньої кнопки робив її зеленою, а дві інші – чорними;
   - вибір верхньої кнопки робив її червоною, а дві інші – чорними.

   Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.4**.
5. Створіть проєкт, у якому розташуйте у вікні три кнопки горизонтально одну поруч з іншою. Установіть такі значення їх властивостей, щоб на початку виконання проєкту перша кнопка була видима і доступна, друга – видима і недоступна, третя – невидима і недоступна. Для кожної кнопки створіть обробник події **Click**, щоб:

- вибір першої кнопки робив її невидимою і недоступною, другу кнопку – видимою і доступною, третю кнопку – видимою і недоступною;
- вибір другої кнопки робив її невидимою і недоступною, третю кнопку – видимою і доступною, першу кнопку – видимою і недоступною;
- вибір третьої кнопки робив її невидимою і недоступною, першу кнопку – видимою і доступною, другу кнопку – видимою і недоступною.

Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.2.5.**



Готуємось до вивчення нового матеріалу

1. Які ще елементи керування, крім кнопки, ви використовували під час роботи з різними програмами?
2. Для чого використовується графічний об'єкт **Напис** у текстовому процесорі **Word**?



4.3. НАПИС

У цьому пункті йтиметься про:

- напис як елемент керування у вікні, його властивості та їх значення;
- створення напису та способи його розміщення у вікні;
- події, які можуть відбуватися з написом;
- обробники подій з написом.

НАПИС. ВЛАСТИВОСТІ НАПИСУ



Пригадайте

- Які ви знаєте властивості кнопки? Наведіть приклади їх значень.
- Якими командами створюється кнопка і розміщується у вікні?
- Як змінити значення властивостей кнопки?

Ще одним елементом керування, який можна використати у проєкті, є **напис. Напис** – це елемент керування, який використовується для виведення текстових повідомлень.

Щоб розмістити напис у вже створеному і відкритому вікні, потрібно:

1. Створити новий об'єкт типу **Label** (англ. *label* – напис) і пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
2. Установити значення властивостей напису або залишити їх за замовчуванням.
3. Розмістити створений об'єкт у вікні.

Для створення напису та пов'язування його зі змінною використовують команду вигляду:

<ім'я_напису> = Label()

або

<ім'я_напису> = Label(<набір властивостей та їх значень>)

У першому випадку буде створено напис зі значеннями властивостей **висота**, **ширина** і **колір**, установленими за замовчуванням, без тексту в ньому. У другому випадку буде створено напис, значення

властивостей якого задані у цій команді. У дужках через кому вказують пари **<властивість> = <значення>**.

Для напису можуть бути встановлені значення властивостей **text, width, height, bg, fg, font, state** та інших аналогічно до того, як вони встановлювалися для кнопки.

Наприклад, після виконання команди:

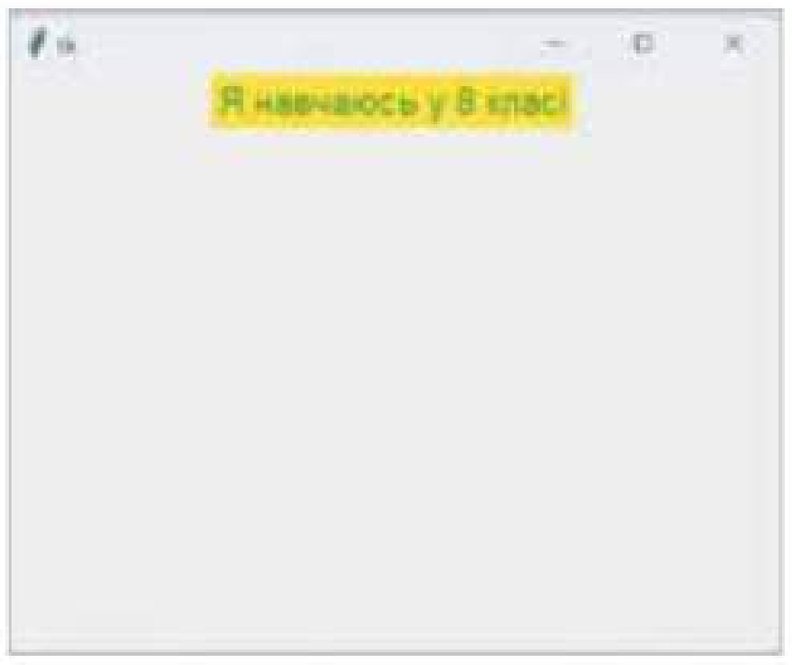
lb = Label(text = 'Я навчаюсь у 8 класі', bg = 'yellow', fg = 'green', font = 'Arial 14') буде створено напис з іменем ***lb***, у якому буде зеленими літерами на жовтому фоні виведено текст *Я навчаюсь у 8 класі*, шрифт символів *Arial*, розмір символів *14* (мал. 4.5).

Звертаємо вашу увагу: якщо під час створення напису значення властивості **width** не задається, то ширина напису автоматично визначається довжиною тексту, який є значенням властивості **text**.

Значення властивостей напису також можна задавати окремими командами виду: **<ім'я напису>['<назва властивості>'] = <значення властивості>.**

Наприклад, установити для напису з іменем **label** ширину 25 символів можна командою **label['width'] = 25**, установити шрифт *Times New Roman* з висотою символів 10 можна командою **label['font'] = ('Times New Roman, 10')**, задати синій колір символів – командою **label['fg'] = 'blue'**. Такого виду команди можна використовувати і для змінення значень властивостей напису під час виконання проєкту.

Для розміщення напису у вікні проєкту використовують вже відомі вам методи **pack()** і **place()**. Наприклад, для розміщення у вікні напису з іменем **label** з відступом 20 пікселів від рядка заголовка вікна можна використати команду **label.pack(pady = 20)** (мал. 4.6).



Мал. 4.5. Напис у вікні



Мал. 4.6. Вікно проєкту з написом

ОБРОБНИКИ ПОДІЙ ДЛЯ НАПИСУ

Пригадайте

- Які події можуть відбуватися з кнопкою?
- Що таке *обробник події*?
- Як записується обробник події у проєкті та як він виконується?

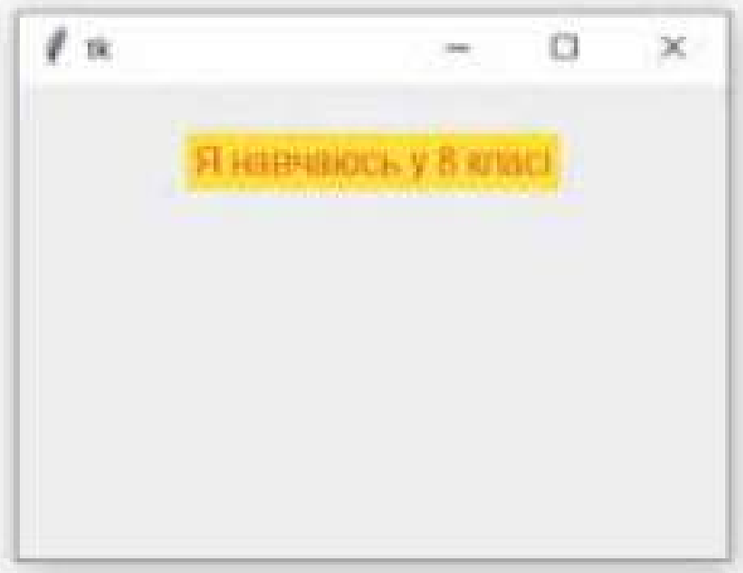
Хоча написи призначені переважно для виведення текстів, але з ними, як і з вікнами і кнопками, можна пов'язувати обробники різних подій.

З написом, як і з вікном проєкту або кнопкою, можна пов'язати процедуру – обробник деякої події, використовуючи метод **bind()**:

<ім'я напису>.bind('<подія>', <ім'я процедури>)

Наприклад, потрібно, щоб після настання події **Click** з написом з іменем **label** колір фону напису ставав *жовтим*, колір символів – *червоним*, розмір символів *14*, у напис виводився заданий текст *Я навчаюсь у 8 класі*. На малюнку 4.7 наведено текст відповідного проєкту та результат його виконання, у тому числі результат виконання процедури – обробника події **Click** для напису.

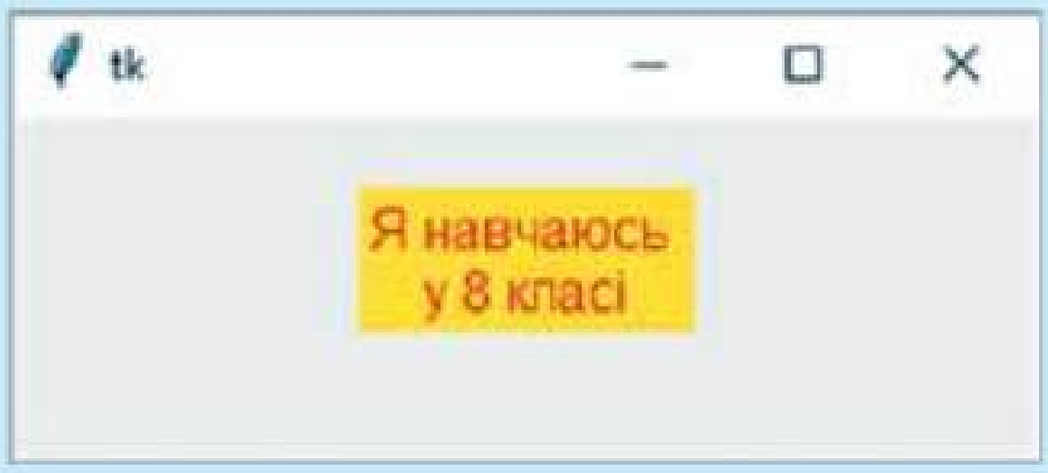
```
from tkinter import *
def label_click(event):
    label['text'] = 'Я навчаюсь у 8 класі'
    label['font'] = 14
    label['fg'] = 'red'
    label['bg'] = 'yellow'
root=Tk()
root.geometry('300x200')
label = Label(text = 'Це напис')
label.bind('<1>', label_click)
label.pack(pady = 20)
```



Мал. 4.7. Обробник події **Click** для напису та результат його виконання

Для тих, хто хоче знати більше

Текст у написі можна вивести в кілька рядків. Для цього на місці, де повинен розпочинатися новий рядок, потрібно ввести символи **\n**. Наприклад, у написі, створеному командою **label = Label(text = 'Я навчаюсь \n у 8 класі')**, у першому рядку буде відображатися текст *Я навчаюсь*, а у другому – *у 8 класі*. Результат виконання цієї команди наведено на малюнку 4.8.



Мал. 4.8. Виведення тексту в написі у два рядки

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/0e2cDIXk* або QR-кодом.

Найважливіше в цьому пункті

Напис – це елемент керування, який використовується для виведення у вікні текстових повідомлень.

Для створення напису та пов'язування його зі змінною використовують команду вигляду:

<ім'я напису>= Label()

або

<ім'я напису> = Label(<набір властивостей та їх значень>)

У першому випадку буде створено напис зі значеннями властивостей **висота**, **ширина** і **колір**, установленими за замовчуванням, без тексту в ньому. У другому випадку буде створено напис, значення властивостей якого задано у цій команді. У дужках через кому вказують пари **<властивість> = <значення>**.

Для напису можуть бути встановлені значення властивостей **text**, **width**, **height**, **bg**, **fg**, **font**, **state** та інших аналогічно до того, як вони встановлювалися для кнопки.

Значення властивостей також можна задавати та змінювати окремими командами виду:

<ім'я напису>['<назва властивості>'] = <значення властивості>

Для розміщення напису у вікні проєкту використовують методи **pack()** і **place()**.

З написом можна пов'язати процедуру – обробник деякої події, використовуючи метод **bind()**:

<ім'я напису>.bind('<подія>', <ім'я процедури>)



Дайте відповіді на запитання

1. Для чого використовують напис?
2. Як створити напис?
3. Як можна встановити значення властивостей напису?
4. Які ви знаєте властивості напису? Поясніть їх.
5. Як розмістити напис у вікні?



Обговоріть і зробіть висновки

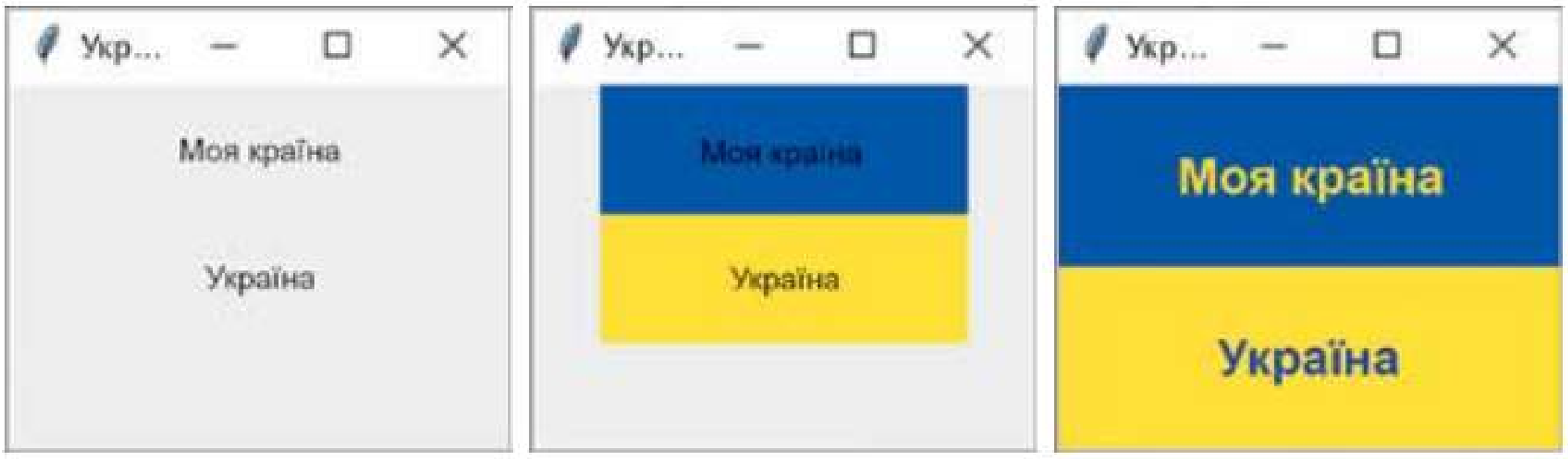
1. Що відбуватиметься, якщо довжина тексту, який виводиться на кнопку і в напис, більша за їх ширину?
2. З якою метою у вікні розміщують порожній напис?



Виконайте завдання

1. Створіть проєкт, у якому в результаті настання події **Click** для напису буде встановлено ширину напису 30 символів і зміниться текст у написі на назву вулиці, на якій розміщено ваш заклад освіти. Початкові значення властивостей напису залиште за замовчуванням. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.1**.
2. Створіть проєкт, у якому в результаті настання події **DblClick** для напису буде встановлено висоту напису 3 символи, збільшиться розмір символів і зміниться текст у написі на ваше ім'я. Початкові значення властивостей напису залиште за замовчуванням. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.2**.
3. Створіть проєкт, у якому в результаті настання події **MouseMove** для напису буде встановлено новий текст у ньому, іншим шрифтом, іншим розміром символів, іншим кольором символів. Початкові значення властивостей напису виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.3**.
4. Створіть проєкт, у якому в результаті настання події **Click** для вікна ширина напису збільшиться на 5 символів, у напис виведеться назва вашого закладу освіти жовтим кольором на червоному фоні. Початкові значення властивостей напису виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.4**.
5. Створіть проєкт, у якому в результаті настання події **Click** для кнопки розмір вікна проєкту стане 500 на 400 і в напис виведеться назва вашого закладу освіти синім кольором на жовтому фоні. Початкові значення властивостей вікна і напису виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.5**.

6. Створіть проєкт, у якому розташуйте у вікні два написи *Моя країна* та *Україна*. Установіть заголовок вікна – *Україна*. Після настання події **Click** з першим написом колір фону першого напису повинен стати *синім*, другого напису – *жовтим*. Після настання події **Click** з другим написом колір символів першого напису повинен стати *жовтим*, другого напису – *синім*, шрифт символів у обох написах – *Arial*, *14*, *напівжирний*. Вигляд вікна проєкту наведено на малюнку 4.9. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.6**.



а) б) в)

Мал. 4.9. Вигляд вікна проєкту:
а) після запуску;
б) після настання події **Click** для першого напису;
в) після настання події **Click** для другого напису

7. Створіть проєкт, у якому розташуйте у вікні два написи *Ніч* і *День*, задайте заголовок вікна – *Світло*. Після запуску проєкту колір фону вікна – *чорний*, колір фону першого напису – *сірий*, другий напис *не відображається*. Після настання події **Click** з першим написом колір фону вікна стає *білим*, перший напис *зникає*, другий напис відображається на *сірому* фоні. Після настання події **Click** з другим написом вікно проєкту повертається до початкового стану. Вигляд вікна проєкту наведено на малюнку 4.10. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.3.7**.



а) б)

Мал. 4.10. Вікно проєкту: а) початковий стан і після настання події **Click** для другого напису; б) після настання події **Click** для першого напису



Готуємось до вивчення нового матеріалу

1. Для чого використовують поля в текстовому процесорі **Word**?
2. Чим відрізняються об'єкти *поле* та *напис*?

4.4. ПОЛЕ. ПРОЄКТИ З ВХІДНИМИ ДАНИМИ ТА РЕЗУЛЬТАТАМИ

У цьому пункті йтиметься про:

- поле, його властивості та їх значення;
- створення поля та способи його розміщення у вікні;
- події, які можуть відбуватися з полем;
- обробники подій з полем;
- використання поля у проєктах з вхідними даними та результатами;
- використання у проєктах констант.

ПОЛЕ. ВЛАСТИВОСТІ ПОЛЯ

Пригадайте

- Які ви знаєте властивості напису? Наведіть приклади їх значень.
- Якими командами створюється напис і розміщується у вікні?
- Як змінити значення властивості напису?

Ще одним елементом керування, який можна використати у проєкті, є *поле*. **Поле** – це елемент керування, який використовується в основному для введення даних. Поле можна використовувати також для виведення результатів.

Щоб розмістити поле в уже створеному і відкритому вікні, потрібно:

1. Створити новий об'єкт типу **Entry** (англ. *entry* – запис, вхід), пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
2. Установити значення властивостей поля або залишити їх за замовчуванням.
3. Розмістити створений об'єкт у вікні.

Для створення поля та пов'язування його зі змінною використовують команду вигляду:

<ім'я_поля> = Entry()

або

<ім'я_поля> = Entry(<набір властивостей та їх значень>)

У першому випадку буде створено поле зі значеннями властивостей **висота**, **ширина** і **колір**, установленими за замовчуванням, без тексту в ньому. У другому випадку буде створено поле, значення властивостей якого задано у цій команді. У дужках через кому вказують пари **<властивість> = <значення>**.

Властивості поля **width**, **bg**, **fg**, **font** аналогічні до відповідних властивостей кнопки та напису, а властивості **text** і **height** відсутні (висота поля визначається висотою символів у його вмісті). Для поля можна задавати товщину рамки як значення властивості **bd** (англ. *border* – межа).

Для розміщення поля у вікні проєкту можна використовувати відомі вам методи **pack()** і **place()**.

На малюнку 4.11 наведено вигляд вікна проєкту з двома полями. Верхнє поле створено командою **ent1 = Entry()** зі значеннями властивостей за замовчуванням, нижнє поле створено командою:

ent2 = Entry(width = 15, bg = 'yellow', fg = 'green', font = 'Arial, 20', bd = 3)



Мал. 4.11. Вікно проєкту з двома полями

Щоб вивести в поле деякий текст, використовують команду виду:

<ім'я поля>.insert(<позиція>, <текст>)

(англ. *insert* – вставити, вставка),



Мал. 4.12. Поле з текстом у вікні

де *позиція* – це позиція символа в полі, починаючи з якої буде вставлено текст. Якщо поле порожнє, то текст уставляється, починаючи з позиції 0, навіть якщо в команді указано іншу позицію. Якщо ж у полі вже є деякий уміст, то текст можна вставити, починаючи з будь-якої позиції у цьому вмісті. Для вставлення тексту в кінець існуючого вмісту вказують позицію **END**.

Наприклад, щоб вивести в порожнє поле з іменем **entry** текст *Я навчаюсь у 8 класі*, потрібно виконати команду **entry.insert(0, 'Я навчаюсь у 8 класі')** (мал. 4.12).

Для змінення вмісту цього поля на *Я навчаюсь у 8-А класі* можна після попередньої команди додати команду **entry.insert(14, '-А')**.

А щоб отримати в полі текст *Я навчаюсь у 8-А класі школи № 1*, можна після попередніх команд додати команду

entry.insert(END, ' школи № 1').

Для очищення вмісту поля з іменем **entry** використовують команду:

entry.delete(0, END).

Отримати з поля його вміст можна методом **get()** (англ. *get* – отримати).

Наприклад, у результаті виконання команди **x = entry.get()** значенням змінної **x** стане текст, який міститься в полі з іменем **entry**.

А щоб у напис з іменем **label** вивести текст, що введено в поле з іменем **entry**, потрібно виконати команду **label['text'] = entry.get()**.

Під час виконання проєкту буває зручно, щоб одразу після запуску проєкту в полі був установлений курсор. У такому разі кажуть, що поле знаходиться **у фокусі**. Для цього використовують метод **focus_set()** (англ. *focus* – фокус, *set* – установити). Наприклад, виконання команди **entry.focus_set()** приводить до встановлення курсора всередині поля з іменем **entry**.

ОБРОБНИКИ ПОДІЙ ДЛЯ ПОЛЯ



Пригадайте

- Які події можуть відбуватися з написом?
- Що таке *обробник події*?
- Як записується обробник події у проєкті та як він виконується?

Як і з іншими об'єктами вікна, з полем можна пов'язати процедури – обробники подій, використовуючи метод **bind()**.

Наприклад, на малюнку 4.13 наведено текст проєкту з процедурою – обробником події **Click** для поля. Ця процедура містить команди:

- установлення *жовтого* кольору фону поля;
- установлення *червоного* кольору для символів тексту в полі;
- установлення розміру *12* для символів тексту в полі;
- виведення тексту, уведеного в полі, у напис;
- присвоювання змінній ***x*** тексту з поля;
- виведення у вікно повідомлень тексту – значення змінної ***x***, яке є текстом з поля, до якого в кінці додано знак оклику.

```
from tkinter import*
from tkinter.messagebox import*

def click(event):
    entry['bg']='yellow'
    entry['fg']='red'
    entry['font']=12
    label['text'] = entry.get()
    x = entry.get()
    showinfo('Поле', x+'!')

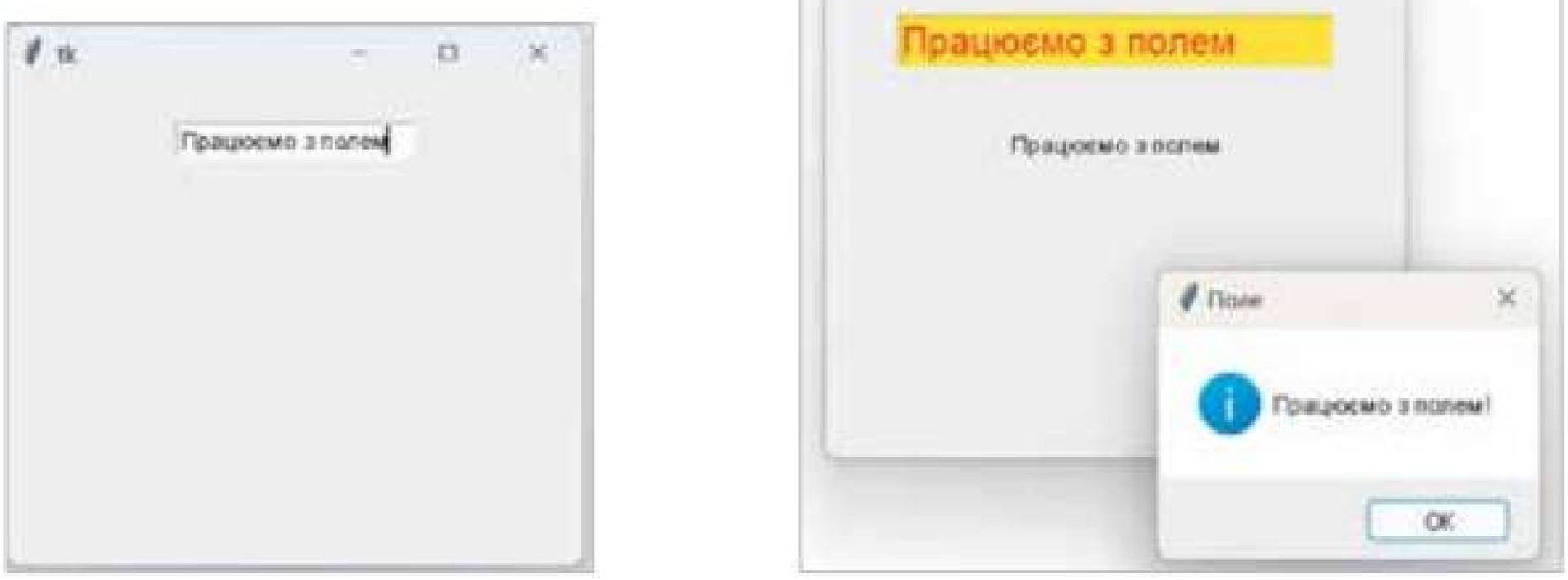
root = Tk()
root.geometry('300x250')

entry = Entry()
entry.pack(pady = 20)
entry.focus_set()
entry.bind('<1>', click)

label = Label()
label.pack(pady = 10)
```

Мал. 4.13. Текст проєкту з обробником події **Click** для поля

Після запуску проєкту на виконання в поле можна ввести довільний текст (на малюнку 4.14, *а* в поле введено текст *Працюємо з полем*) і вибрати поле. Результат виконання обробника події **Click** для поля наведено на малюнку 4.14, *б*.



а) б)

Мал. 4.14. Вікно виконання проєкту після введення тексту в поле і до настання події **Click** для поля (а) та після настання події (б)

ПРОЄКТИ З УВЕДЕННЯМ ДАНИХ І ВИВЕДЕННЯМ РЕЗУЛЬТАТІВ



Пригадайте

● Для чого використовують команду присвоювання? Який її загальний вигляд? ● Як вивести текст у напис; у поле; у вікно повідомлень? ● Які типи змінних ви знаєте? Які значення можуть мати змінні цих типів? Що визначає тип змінної?

У ході виконання проєктів можна вводити значення вхідних даних і виводити значення результатів.

Для введення значень вхідних даних використовується **поле**, а для виведення значень результатів можна використовувати **поле**, **напис**, **вікно повідомлень**.

Створимо для прикладу проєкт, у якому обчислюватиметься сума двох довільних чисел.

Створимо математичну модель для цього проєкту:
вхідні дані: два довільних числа x і y;
кінцеві результати: їх сума z;
формули: $z = x + y$.

Оскільки в математичній моделі два вхідних даних, використаємо у цьому проєкті два поля, у які користувач під час виконання проєкту вводитиме доданки (вони можуть бути і цілі, і дробові). Кінцевий результат виведемо в напис. Розмістимо також у вікні кнопку, після вибору якої відбуватиметься додавання введених у поля чисел і виведення суми в напис.

Текст проєкту, що створює вікно з розміщеними в ньому необхідними об'єктами, процедура для додавання двох чисел, а також вікно виконання проєкту наведені на малюнку 4.15.

Розглянемо наведену процедуру.

Хоча користувач під час виконання проєкту вводить у поля числа, система тим не менше сприймає їх як тексти. Ці тексти є вмістом полів з іменами **entry1** і **entry2**. Тому перші дві команди процедури

```
from tkinter import *

def click():
    x = float(entry1.get())
    y = float(entry2.get())
    z = x + y
    label['text'] = str(z)

root = Tk()
root.geometry("300x250")

entry1 = Entry()
entry1.pack(pady = 10)
entry2 = Entry()
entry2.pack(pady = 10)

label = Label()
label.pack(pady = 10)
```



Мал. 4.15. Вікно з двома полями, написом і кнопкою та текст проєкту з процедурою для додавання двох чисел

x = float(entry1.get()) і ***y = float(entry2.get())*** переводять текстове подання чисел безпосередньо в числа і присвоюють ці значення змінним ***x*** і ***y*** відповідно.

Третя команда додає ці числа та присвоює суму змінній ***z***.

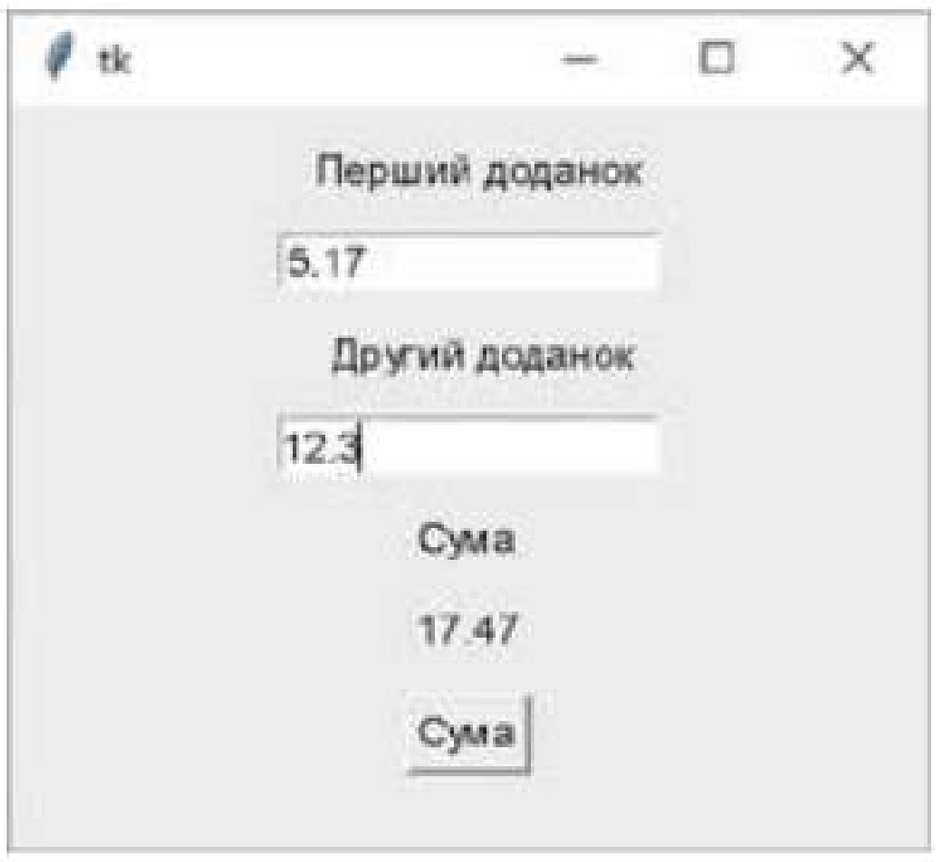
Четверта команда **label['text'] = str(z)** (англ. *string* – рядок) переводить обчислену суму з числа в його текстове подання та присвоює цей текст значенню властивості **text** напису з іменем **label**.

Значення змінних ***x***, ***y***, ***z***, які використані у процедурі, за умовою задачі можуть бути довільними числами, тобто і цілими, і дробовими. Тому для таких змінних використано тип **float**.

Під час виконання створеного проєкту користувач уводить у поля два числа (на малюнку 4.15 це числа −2,7 і 15,38), вибирає кнопку *Сума*, і результат (12,68) виводиться в напис. Не закриваючи вікно виконання проєкту, користувач може ввести в поля інші числа, знову вибрати кнопку *Сума* та одержати новий результат. І так можна повторювати довільну кількість разів.

Звертаємо вашу увагу: під час уведення в поля дробових чисел ціла і дробова частини відокремлюються крапкою.

Щоб під час виконання проєкту користувачеві було зрозуміліше, у які поля що саме потрібно вводити і де виводитиметься результат, доцільно у вікні розмістити поруч з кожним полем і написом, призначеними для виведення результату, написи з пояснювальними текстами, наприклад, *Перший доданок*, *Другий доданок*, *Сума* (мал. 4.16).



Мал. 4.16. Вікно виконання проєкту додавання двох чисел з пояснювальними текстами

Звертаємо вашу увагу.

1. Для виведення значення кінцевого результату (змінна ***z***) можна замість напису розмістити у вікні ще одне поле, наприклад з іменем **entry3**. І тоді у процедурі команда виведення була б такою: **entry3.insert(0, str(z))**.
2. Перед використанням у проєкті команди виведення в поле доцільно включати у проєкт команду очищення поля від можливих попередніх результатів виконання проєкту: **entry3.delete(0, END)**.
3. Можна було б також вивести значення змінної ***z*** у вікно повідомлень. Тоді команда виведення була б такою: **showinfo('Результат', str(z))**.

Якщо у проєкті вхідними даними могли би бути тільки цілі числа, то для читання їх з полів можна використати команди **x = int(entry1.get())** і **y = int(entry2.get())**.

У проєкті на малюнку 4.13 у команді **x = entry.get()** використана змінна ***x***, значенням якої є текст. Така змінна має тип **string**.

ВИКОРИСТАННЯ У ПРОЄКТІ КОНСТАНТ (ПОСТІЙНИХ ВЕЛИЧИН)

Пригадайте

- Чим константи (постійні величини) відрізняються від змінних?
- Які константи ви знаєте? Наведіть їх значення.

У проєктах, крім змінних, часто використовують *константи* (постійні величини). **Константа** – це величина, значення якої не змінюється в ході виконання проєкту.

Наприклад, якщо потрібно створити проєкт для обчислення маси сосни, то потрібно знати її об'єм і густину. Це може бути важливим для визначення, які вантажівки та скільки їх потрібно для перевезення сосен на деревообробний завод.

І якщо об'єм сосни може бути різним (змінна величина), то густина сосни є постійною величиною. Значення густини можна знайти в таблиці густин речовин.

Створимо математичну модель для цього проєкту:
вхідні дані: об'єм сосни V;
кінцеві результати: маса сосни m;
формула: $m = \rho V$, де ρ – густина сосни – константа, що дорівнює 520 кг/м^3.

У мові програмування **Python** прийнято записувати імена констант (постійних величин) великими літерами.

Уважатимемо, що у цьому проєкті значення об'єму вводитиметься в кубічних метрах у поле з іменем **entry**, а значення маси виводитиметься в кілограмах у напис з іменем **label**.

Тоді відповідний фрагмент проєкту виглядатиме так:

```
RO = 520
v = float(entry.get())
m = v * RO
label['text'] = str(m)
```

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Ae2cD0Vl* або QR-кодом.

Найважливіше в цьому пункті

Поле – це елемент керування, який використовується для введення значень вхідних даних і виведення значень результатів.

Щоб розмістити поле в уже створеному і відкритому вікні, потрібно:

1. Створити новий об'єкт типу **Entry** (англ. *entry* – запис, вхід), пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
2. Установити значення властивостей поля або залишити їх за замовчуванням.
3. Розмістити створений об'єкт у вікні.

Для створення поля та пов'язування його зі змінною використовують команду вигляду:

<ім'я_поля> = Entry()
або
<ім'я_поля> = Entry(<набір властивостей та їх значень>)

У першому випадку буде створено поле зі значеннями властивостей **висота**, **ширина** і **колір**, установленими за замовчуванням, без тексту в ньому. У другому випадку буде створено поле, значення властивостей якого задано у цій команді. У дужках через кому вказують пари **<властивість> = <значення>**.

Властивості поля **width**, **bg**, **fg**, **font** аналогічні до відповідних властивостей кнопки та напису, а властивості **text** і **height** відсутні (висота поля визначається висотою символів у його вмісті). Для поля можна задавати товщину рамки як значення властивості **bd** (англ. ***border*** – межа).

Для розміщення поля у вікні проєкту можна використовувати відомі вам методи **pack()** і **place()**.

Щоб вивести в поле деякий текст, використовують команду виду:

<ім'я поля>.insert(<позиція>, <текст>)

Отримати з поля його вміст можна, використовуючи метод **get()**, наприклад **label['text'] = entry.get()**.

Для очищення вмісту поля з іменем **entry** використовують команду:

entry.delete(0, END)

Як і з іншими об'єктами вікна, з полем можна пов'язати процедури – обробники подій, використовуючи метод **bind()**.

Прочитати число з поля з іменем **entry1** і присвоїти його змінній можна командою **x = float(entry1.get())** або **x = int(entry1.get())**.

Вивести значення змінної **z** у напис можна командою **label['text'] = str(z)**, а в поле – командою **entry3.insert(0, str(z))**.

Константа – це величина, значення якої не змінюється в ході виконання проєкту. У мові програмування **Python** прийнято записувати імена констант (постійних величин) великими літерами.

Дайте відповіді на запитання

1. Для чого може використовуватися поле?
2. Як створити поле і розмістити його у вікні?
3. Які властивості поля ви знаєте? Схарактеризуйте їх.
4. Як задати значення властивості поля в команді його створення?
5. Як змінити значення властивості поля в ході виконання проєкту?
6. Як отримати значення з поля; записати в поле; видалити дані з поля?
7. Якими командами можна перевести текстове подання числа безпосередньо в число і навпаки?
8. Чим константа відрізняється від змінної?

Обговоріть і зробіть висновки

1. Чим відрізняються команди виведення в напис і в поле?
2. Чим відрізняються команди виведення в поле й у вікно повідомлень?

Виконайте завдання

1. Створіть проєкт, у якому в результаті настання події **Click** для поля буде встановлено ширину поля 15 символів і виведено текст у поле «Інформатика» синім кольором на жовтому фоні. Початкові значення властивостей поля залиште за замовчуванням. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.1**.
2. Створіть проєкт, у якому в результаті настання події **DoubleClick** для поля буде встановлено товщину рамки поля 3 пікселі

і змінено текст у полі на ваше ім'я синім кольором. Початкові значення властивостей поля виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.2**.

3. Створіть проєкт, у якому в результаті настання події **MouseMove** для поля буде змінено текст у ньому, установлено червоний колір тексту, сірий колір фону і ширина поля збільшиться на 5 символів. Початкові значення властивостей поля виберіть самостійно. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.3**.

4. Створіть математичну модель і проєкт для обчислення значення виразу $(2*a + b)*(a - b)$. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.4**.

5. Створіть математичну модель і проєкт для обчислення значення виразу $(3*a - b)*(a + 2*b)$. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.5**.

6. Створіть математичну модель і проєкт для обчислення довжини кола за даним радіусом, використовуючи константу. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.6**.

7. Створіть математичну модель і проєкт для обчислення шляху, який проходить світло у вакуумі за певний час. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.7**.

8. Створіть математичну модель і проєкт для визначення, яку суму грошей має заплатити споживач за використану електроенергію. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.8**.

9. Клієнт поклав у банк x грн під p % річних. За умовою банку прибуток за рік додається до вкладу. Створіть математичну модель і проєкт для визначення, який прибуток клієнт отримає через 2 роки. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.4.9**.



Практична робота № 4. «Проєкти з полями, написами і кнопками, з уведенням даних і виведенням результатів»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

Задача 1. Створіть проєкт з використанням поля і кнопки. Установіть такі початкові значення властивостей:

- поля:
  - ширина – *15 символів*;
  - товщина рамки – *3 пікселі*;
  - відступ від верхньої межі вікна – *30 пікселів*;
- кнопки:
  - ширина – *10 символів*;
  - відступ від поля – *10 символів*;
  - текст на кнопці – *назва вашого класу*;
  - колір тексту – *синій*;
  - колір фону – *жовтий*.

Створіть обробник події **Click** для кнопки, виконання якого встановить *червоний* колір фону поля, шрифт символів *Arial*, розмір символів *14*, збільшить ширину поля на 20 символів, установить колір тексту *білий* і виведе в поле текст *Ми використовуємо властивості поля!*.

Збережіть проєкт у вашій папці у файлі з іменем **практична 4.1.**

Задача 2. Створіть проєкт для обчислення значення виразу ***a*(4*b* – *a*)**.

1. Створіть математичну модель для розв'язування задачі. Запишіть її в зошит.
2. Визначте, які компоненти потрібно розмістити у вікні для введення значень вхідних даних, виведення значень кінцевих результатів. Розмістіть їх у вікні. Підпишіть їх.
3. Створіть обробник події **Click** для кнопки для обчислення значення виразу і виведення результату.
4. Збережіть проєкт у вашій папці у файлі з іменем **практична 4.2.**

Задача 3. Мати дала Марійці певну суму грошей і попросила купити 400 г печива, 2 пакети молока і хліб. Складіть проєкт для обчислення суми грошей, яка залишиться в Марійки.

1. Створіть математичну модель для розв'язування задачі. Запишіть її в зошит.
2. Визначте, які компоненти потрібно розмістити у вікні для введення значень вхідних даних, виведення результатів. Розмістіть їх у вікні. Підпишіть їх.
3. Створіть обробник події **Click** для кнопки для обчислення результату і виведення його.
4. Збережіть проєкт у вашій папці у файлі з іменем **практична 4.3.**



Готуємось до вивчення нового матеріалу

1. Які типи змінних ви знаєте? Наведіть приклади їх значень.
2. Що визначають типи змінних?

4.5. ЛОГІЧНІ ВИРАЗИ. ЗМІННІ ЛОГІЧНОГО ТИПУ. ЛОГІЧНІ ОПЕРАЦІЇ

У цьому пункті йтиметься про:

- логічні вирази;
- змінні логічного типу;
- логічні операції: заперечення, кон'юнкція, диз'юнкція.

ЛОГІЧНІ ВИРАЗИ



Пригадайте

● Що таке *висловлювання*? ● На яких уроках ви використовували істинні та хибні висловлювання? Наведіть приклади. ● Чи будь-яке речення є висловлюванням? Наведіть приклади.



Поміркуйте

- Як можна визначити, що висловлювання істинне?
- Як можна визначити, що висловлювання хибне?

Ви вже знаєте, що **висловлювання** – це речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами та про яке можна однозначно сказати, **істинне** воно чи **хибне**. Приклади **істинних висловлювань**: *«Київ – столиця України»*, *2*5 – 4 = 6*, *4 < 12*. Приклади **хибних висловлювань**: *«У січні 30 днів»*, *2*7 + 3 = 12*, *3 > 5*. Приклади речень, які не є висловлюваннями: *«Зачини, будь ласка, вікно»*, *«Тобі сподобався цей фільм?»*.

Висловлювання можна розглядати як **логічний вираз**.

Логіка (грец. λόγος – слово, смисл, думка, мова) – наука про форми, методи і закони правильного мислення, про способи міркування.

Логічним виразом називається вираз, який може набувати одного з двох значень: **True** (англ. *true* – істина, істинність, правда) або **False** (англ. *false* – хиба, хибність, неправда). Значення **True** та **False** називають **логічними константами**.

Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **True**. Тобто, *«Київ – столиця України»* = **True**, (*2*5 – 4 = 6*) = **True**, (*4 < 12*) = **True**.

Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **False**. Тобто, *«Цей підручник для учнів сьомого класу»* = **False**, (*2*7 + 3 = 12*) = **False**, (*3 > 5*) = **False**.

В алгоритмах логічні вирази використовують у розгалуженнях і циклах. Ми розглянемо таке використання в наступних пунктах.

ЗМІННІ ЛОГІЧНОГО ТИПУ



Пригадайте

● Які типи змінних ви знаєте? Наведіть приклади значень змінної кожного з цих типів. ● Що визначає тип змінної?

У попередніх пунктах ви використовували змінні типів **float** та **int**, значеннями яких були числа, та типу **str**, значеннями яких були рядки тексту.

В інформатиці використовують також змінні, значенням яких можуть бути **True** або **False**. У мові програмування **Python** ці значення потрібно обов'язково писати з великої літери.

Змінні, які можуть набувати одного з двох значень, **True** або **False**, називаються **змінними логічного типу**.

Логічний тип змінних у мові програмування **Python** позначається **bool** (англ. ***boolean*** – логічний).

Звертаємо вашу увагу: самі змінні логічного типу і логічні константи також є логічними виразами.

Надавати змінним логічного типу відповідні значення можна командою присвоювання. Наприклад, ***x*** = **True** або ***x*** = **False**.

Змінній логічного типу можна присвоювати значення іншої змінної логічного типу, наприклад ***y*** = ***x***.

Змінній логічного типу можна також присвоювати значення логічного виразу. Тобто можна використовувати такі команди присвоювання: ***x*** = **(32 < 13)**, ***y*** = **(2 + 12/3 == 6)**. У першій команді значення логічного виразу ***32 < 13*** дорівнює **False**, тому і значення змінної ***x*** теж **False**. У другій команді знак == використано для порівняння, чи

дорівнює значення виразу $2 + 12/3$ числу 6. Оскільки вони рівні, то значення логічного виразу $2 + 12/3 == 6$ дорівнює **True**, і тому значення змінної y теж **True**.

ЛОГІЧНІ ОПЕРАЦІЇ



Пригадайте

● Які ви знаєте арифметичні операції? Над якими об'єктами вони виконуються? Що є результатом цих операцій? ● Яку операцію над текстами ви використовували? Що є результатом такої операції?

Найпростіші логічні операції ви вже знаєте. Це **операції порівняння:** дорівнює, не дорівнює, більше, менше, більше або дорівнює, менше або дорівнює. У мові програмування **Python** ці операції позначають так:

==	дорівнює	>	більше	>=	більше або дорівнює
!=	не дорівнює	<	менше	<=	менше або дорівнює

Розглянемо ще три логічні операції: **заперечення, кон'юнкція** (лат. *conjunctio* – об'єднання) і **диз'юнкція** (лат. *disjunctio* – роз'єднання).

Запереченням логічного виразу x називається логічна операція, результат якої дорівнює **True**, якщо значення логічного виразу x дорівнює **False**, і дорівнює **False**, якщо значення логічного виразу x дорівнює **True**.

Заперечення логічного виразу x у мові програмування **Python** позначають **not** x. Інколи в математиці заперечення логічного виразу x позначається $\bar{x}$ або $\neg x$.

Тобто, якщо x = **True**, то **not** x = **False**, і якщо x = **False**, то **not** x = **True**. Для операції заперечення можна скласти таку таблицю, яку називають **таблицею істинності:**

x	not x
True	False
False	True

Для заперечення висловлювання часто використовують частку **не** до дієслова або до узагальнюючих слів *усі*, *кожен*. Наприклад, результатом заперечення висловлювання A = *«Існує найбільше натуральне число»* є висловлювання **not** A = *«Не існує найбільшого натурального числа»*, а результатом заперечення висловлювання B = *«Усі натуральні числа парні»* є висловлювання **not** B = *«Не всі натуральні числа парні»*.

Кон'юнкцією двох логічних виразів x і y називається логічна операція, результат якої дорівнює **True**, якщо значення кожного з логічних виразів x і y дорівнюють **True**, і результат якої дорівнює **False**, якщо значення хоча б одного з логічних виразів x або y дорівнює **False**.

Кон'юнкція двох логічних виразів x і y у мові програмування **Python** позначається x ***and*** y (англ. *and* – і, та). Інколи в математиці кон'юнкція двох логічних виразів x і y позначається $x \wedge y$.

Таблиця істинності для кон'юнкції виглядає так:

x	*y*	*x and y*
True	True	True
True	False	False
False	True	False
False	False	False

Для утворення кон'юнкції двох висловлювань часто використовують сполучники **і, та**. Наприклад, висловлювання *«Число 36 кратне 3* **і** *число 36 кратне 2»* є кон'юнкцією двох висловлювань: *«Число 36 кратне 3»* **і** *«Число 36 кратне 2»*, а висловлювання *«По місту курсують автобуси* **та** *трамваї»* є кон'юнкцією двох висловлювань: *«По місту курсують автобуси»* **та** *«По місту курсують трамваї»*.

Диз'юнкцією двох логічних виразів ***x*** і ***y*** називається логічна операція, результат якої дорівнює **True**, якщо значення хоча б одного з логічних виразів ***x*** або ***y*** дорівнює **True**, і результат якої дорівнює **False**, якщо значення кожного з логічних виразів ***x*** і ***y*** дорівнює **False**.

Диз'юнкція двох логічних виразів ***x*** і ***y*** у мові програмування **Python** позначається ***x* or *y*** (англ. *or* – або). Інколи в математиці диз'юнкція двох логічних виразів ***x*** і ***y*** позначається $x \vee y$.

Таблиця істинності для диз'юнкції виглядає так:

x	*y*	*x or y*
True	True	True
True	False	True
False	True	True
False	False	False

Для утворення диз'юнкції висловлювань часто використовують сполучник **або**. Наприклад, висловлювання *11 ≤ 21 (11 менше* **або** *дорівнює 21)* є диз'юнкцією двох висловлювань: *(11 < 21)* **або** *(11 = 21)*.

Часто використовують логічні вирази, що містять кілька логічних операцій. Наприклад, ***x or y and not z***. У таких виразах для вищерозглянутих логічних операцій визначено такий **пріоритет виконання логічних операцій:**

- спочатку виконуються операції заперечення;
- потім операції кон'юнкції;
- потім операції диз'юнкції.

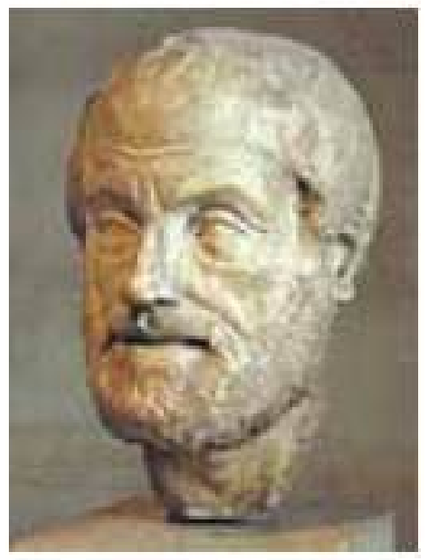
Для визначення значень логічного виразу можна скласти **таблицю істинності**. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу, та виконати, враховуючи пріоритет операцій, усі логічні операції, які входять до цього виразу. Наприклад, для логічного виразу ***x or y and not z*** вона буде такою:

x	*y*	*z*	*not z*	*y and not z*	*x or y and not z*
True	True	True	False	False	True
True	True	False	True	True	True
True	False	True	False	False	True
False	True	True	False	False	False
True	False	False	True	False	True
False	True	False	True	True	True
False	False	True	False	False	False
False	False	False	True	False	False

Для змінення порядку виконання логічних операцій використовують дужки. Так, наприклад, у логічному виразі ***(x or y) and not z*** спочатку буде виконано операцію диз'юнкції в дужках, потім операцію заперечення, а потім операцію кон'юнкції.

Цікаві факти з історії

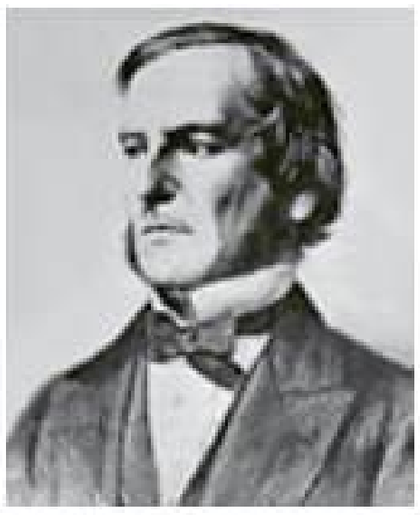
Логіка як наука зародилася у Давній Греції. Засновником цієї науки вважається давньогрецький філософ Арістотель (384–322 рр. до н. е) (мал. 4.17).



Мал. 4.17. Арістотель

Означення логічних операцій і формулювання їх властивостей були сформульовані англійським математиком і філософом **Джорджем Булем** (1815–1864) (мал. 4.18). У 1854 році вийшла його основна робота «Дослідження законів мислення, на яких засновані математичні теорії логіки й імовірності». У ній досліджується система, яку сьогодні називають *Алгеброю висловлювань*, або *Булевою алгеброю*.

Булева алгебра стала фундаментальною науковою основою для створення комп'ютерів і використовується в усіх мовах програмування. Саме іменем цього вченого названо логічний тип даних **bool** у програмуванні. Існують мови логічного програмування, наприклад **PROLOG** *https://uk.wikipedia.org/wiki/Пролог_(мова_програмування)*.



Мал. 4.18. Джордж Буль



Мал. 4.19. Ірина Хоменко

Потрібно також відзначити вагомий внесок українських учених у розвиток логіки як науки: Яна Лукасевича (1878–1956), Павла Копніна (1922–1971), Клавдії Руденко (1919–1974), Ірини Хоменко (нар. 1961 р.) (мал. 4.19).

У 2019 році **UNESCO** (англ. ***U**nited **N**ations **E**ducational, **S**cientific and **C**ultural Organization* – Організація Об'єднаних Націй з питань освіти, науки і культури) оголосило 14 січня **Всесвітнім днем логіки**.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Ge2cFgsq* або QR-кодом.



Найважливіше в цьому пункті

Висловлювання – це речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами і про яке можна однозначно сказати, **істинне** воно чи **хибне**.

Логічним виразом називається вираз, який може набувати одного з двох значень: **True** або **False**. Ці значення називаються **логічними константами**.

Висловлювання можна розглядати як **логічний вираз**. Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **True**. Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **False**.

Змінні, які можуть набувати одного з двох значень, **True** або **False**, називаються **змінними логічного типу**. Логічний тип змінних у мові програмування **Python** позначається **bool**. Змінні логічного типу і логічні константи також уважаються логічними виразами.

Над логічними виразами можна виконувати **логічні операції: заперечення**, **кон'юнкція**, **диз'юнкція**.

Запереченням логічного виразу ***x*** називається логічна операція, результат якої дорівнює **True**, якщо значення логічного виразу ***x*** дорівнює **False**, і дорівнює **False**, якщо значення логічного виразу ***x*** дорівнює **True**.

Кон'юнкцією двох логічних виразів ***x*** і ***y*** називається логічна операція, результат якої дорівнює **True**, якщо значення кожного з логічних виразів ***x*** і ***y*** дорівнюють **True**, і дорівнює **False**, якщо значення хоча б одного з логічних виразів ***x*** або ***y*** дорівнює **False**.

Диз'юнкцією двох логічних виразів ***x*** і ***y*** називається логічна операція, результат якої дорівнює **True**, якщо значення хоча б одного з логічних виразів ***x*** або ***y*** дорівнює **True**, і дорівнює **False**, якщо значення кожного з логічних виразів ***x*** і ***y*** дорівнює **False**.

У логічних виразах для вищерозглянутих логічних операцій визначено такий **пріоритет виконання операцій**: спочатку виконуються операції заперечення, потім операції кон'юнкції, потім операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки.

Для обчислення значень логічного виразу можна скласти **таблицю істинності**. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу, і виконати, враховуючи пріоритет операцій, усі логічні операції, які до нього входять.



Дайте відповіді на запитання

1. Який вираз називається логічним? Наведіть приклади.
2. Які змінні називаються змінними логічного типу?
3. Що таке *заперечення логічного виразу*?
4. Що таке *кон'юнкція двох логічних виразів*?
5. Що таке *диз'юнкція двох логічних виразів*?
6. Яким є пріоритет виконання логічних операцій?



Обговоріть і зробіть висновки

1. Як би ви назвали логічну операцію **«Якщо A, то B»**? Чи використовували ви таку логічну операцію на уроках? Якщо так, наведіть приклади. Чи використовували ви таку логічну операцію в повсякденному житті? Якщо так, наведіть приклади.



Виконайте завдання

1. Наведіть приклади використання логічних виразів і логічних операцій у математиці, українській мові, інших предметах, у життєвих ситуаціях.

2. Сформулюйте заперечення висловлювань і з'ясуйте, істинні вони чи хибні:

а) Число 3 є дільником числа 545.
б) Автомобіль не має права їхати на червоне світло.
в) Існують паралелограми з прямими кутами.
г) Рівняння $2x - 3 = 0$ має цілий корінь.
д) Не існує натурального числа, що ділиться на 2.
е) Існує ціле число, яке ділиться на всі цілі числа.
ж) Усі люди вміють грати в шахи.

3. Серед наступних висловлювань знайдіть кон'юнкції та диз'юнкції та визначте, істинні вони чи хибні:

а) Число 27 кратне 3 і кратне 9.
б) $17 < 42 < 18$.
в) Число 2 просте або парне.
г) ΔABC є гострокутним, прямокутним або тупокутним.
д) Діагоналі будь-якого паралелограму перпендикулярні та точкою перетину поділяються навпіл.
е) $7^2 = 49$ і $(-7)^2 = 49$.
ж) $21 \leq 21$.
и) $21 \leq 18$.

4. Нехай A = *«Сергієнко вивчає англійську мову»*, B = *«Сергієнко має оцінку 8 з інформатики»*. Сформулюйте висловлювання:

а) ***A and not B***;
б) ***not A or B***;
в) ***A or not A and B***.

5. Запишіть у зошит порядок виконання логічних операцій у логічних виразах зі змінними логічного типу ***a*** і ***b***:

а) ***a or not a and b***;
б) ***not a or not b and a***;
в) ***(a or not b) and (not a or not b)***;
г) ***not a and not b or c and d***.

6. Створіть у зошиті таблиці істинності для логічних виразів зі змінними логічного типу ***a*** і ***b***:

а) ***a or a and b***;
б) ***a or not a and b***;
в) ***a or b or not b***;
г) ***not a or not b***;
д) ***(a or not b) and (a or b)***;
е) ***(x or y) and not y***;
ж) ***a and (not b or a) and b***;
и) ***(x or y) and not z***.

7. Створіть проєкт для визначення значення логічного виразу ***(x or not y) and z***, де ***x***, ***y***, ***z*** – змінні логічного типу. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.5.7**. Використайте створений проєкт для заповнення в зошиті таблиці істинності для даного виразу за всіх можливих наборів значень змінних ***x***, ***y***, ***z***.

8. Створіть проєкт для визначення значення логічного виразу ***x* > 0 *and not* (*y* < 0 *or x* < 0) *and y* > 0**, де ***x***, ***y*** – змінні числового типу. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.5.8**. Використайте створений проєкт для визначення значення даного логічного виразу для наведених наборів значень змінних ***x***, ***y***, ***z***:

а) $x = 5, y = 8$;
б) $x = 5, y = -8$;
в) $x = -5, y = 8$;
г) $x = -5, y = -8$.

9. На нафтопереробному заводі є три установки: A, B, C. На кожній з них установлено датчики. Якщо одночасно буде перевищено граничні показники датчиків на першій установці й на будь-якій іншій або на всіх трьох установках, то виникає надзвичайна ситуація. У такому разі повинен пролунати попереджувальний сигнал. Для створення автоматизованої системи подавання цього сигналу складіть у зошиті логічний вираз і для нього заповніть таблицю істинності, на основі якої буде працювати ця система.

10. Михайлу дозволяють пограти на комп'ютері в комп'ютерні ігри, якщо він виконав домашні завдання *або* навів лад у своїй кімнаті *і* завершила свою роботу з комп'ютером його старша сестра. У системі *«Розумного» будинку* можна фіксувати виконання кожного виду роботи. Це робить можливим автоматично отримувати повідомлення про надання доступу до комп'ютера. Складіть у зошиті логічний вираз і для нього заповніть таблицю істинності, на основі якої працює система надсилання повідомлень.

Готуємось до вивчення нового матеріалу

1. У яких випадках в алгоритмі потрібно використати розгалуження?
2. Які види розгалуження ви знаєте? Як вони виконуються?

4.6. ЛОГІЧНІ ВИРАЗИ В РОЗГАЛУЖЕННЯХ. ВІКОННІ ПРОЄКТИ З РОЗГАЛУЖЕННЯМИ

У цьому пункті йтиметься про:

- використання логічних виразів і логічних операцій у розгалуженнях;
- віконні проєкти з розгалуженнями.

ЛОГІЧНІ ВИРАЗИ В РОЗГАЛУЖЕННЯХ

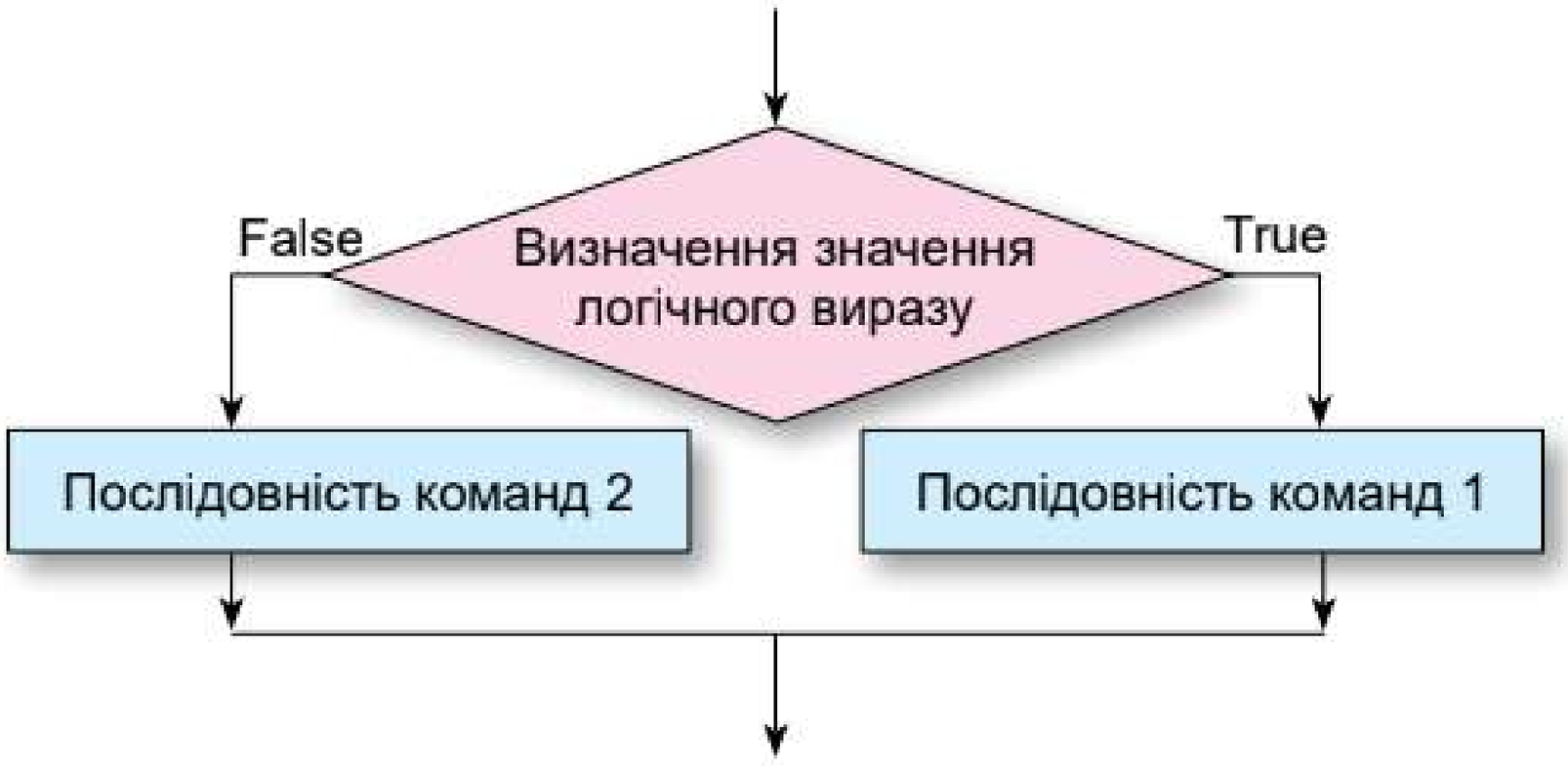
Пригадайте

• У яких випадках в алгоритмах використовують розгалуження? • Які ви знаєте види розгалужень? Як виконується розгалуження кожного з цих видів?

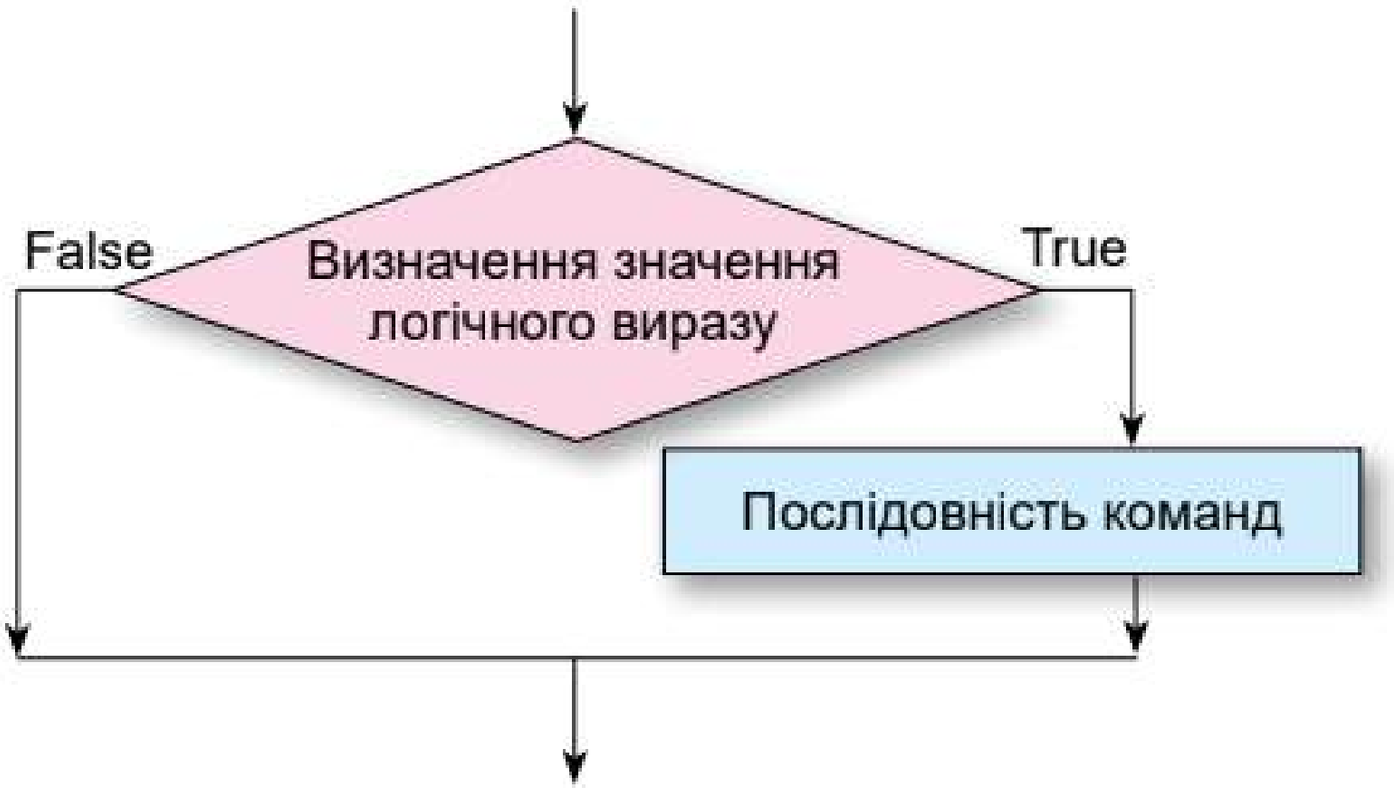
Ви вже знаєте, що кожне розгалуження розпочинається з **команди перевірки умови**, результатом виконання якої може бути або **Так (істина, True)**, або **Ні (хиба, False)**. І залежно від результату виконання команди перевірки умови будуть виконуватися ті чи інші команди алгоритму.

Як умови ви використовували в розгалуженнях висловлювання, математичні рівності та нерівності, запитання, на які потрібно відповісти **Так** чи **Ні**. Усі такі умови є логічними виразами. У розгалуженнях можна використовувати й інші логічні вирази: логічні змінні, логічні вирази з логічними операціями тощо.

Тому в загальному випадку кожне розгалуження розпочинається з **команди визначення значення логічного виразу**. Результатом виконання такої команди можуть бути значення **True** або **False**. І залежно від результату виконання цієї команди будуть виконуватися ті чи інші команди алгоритму (мал. 4.20, 4.21).



Мал. 4.20. Блок-схема повного розгалуження



Мал. 4.21. Блок-схема неповного розгалуження

За використання поняття логічного виразу команди розгалуження в мові програмування **Python** виглядатимуть так:

а) **команда повного розгалуження:**

```
if <логічний вираз>:
    <послідовність команд 1>
else:
    <послідовність команд 2>
```

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **True**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалужен-

ням; якщо це значення **False**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

б) **команда неповного розгалуження:**

```
if <логічний вираз>:
    <послідовність команд>
```

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **True**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **False**, одразу виконується команда, наступна за розгалуженням.

в) **команда багатоальтернативного розгалуження:**

```
if <логічний вираз 1>:
    <послідовність команд 1>
elif <логічний вираз 2>:
    <послідовність команд 2>
elif <логічний вираз 3>:
    <послідовність команд 3>
...
else:
    <послідовність команд N>
```

Виконання цієї команди відбувається так:

- визначається значення **логічного виразу 1**;
- якщо це значення **True**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням;
- якщо це значення **False**, то визначається значення **логічного виразу 2**;
- якщо це значення **True**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням;
- якщо це значення **False**, то визначається значення наступного логічного виразу і так далі;
- якщо всі логічні вирази в команді мають значення **False**, то виконується **послідовність команд N**, розміщена після **else**, і після цього виконується команда, наступна за розгалуженням.

Звертаємо вашу увагу:

- Команда багатоальтернативного розгалуження може не містити

```
else:
    <послідовність команд N>
```

У такому разі якщо всі логічні вирази в команді мають значення **False**, то одразу виконується команда, наступна за розгалуженням.

- Якщо послідовності команд містять тільки по одній команді, то цю команду можна записувати без переходу на новий рядок після двокрапки.

ПРОЄКТИ З РОЗГАЛУЖЕННЯМИ



Пригадайте

● У яких випадках використовується команда повного розгалуження? ● У яких випадках використовується команда неповного розгалуження? ● У яких випадках використовується команда багатоальтернативного розгалуження?

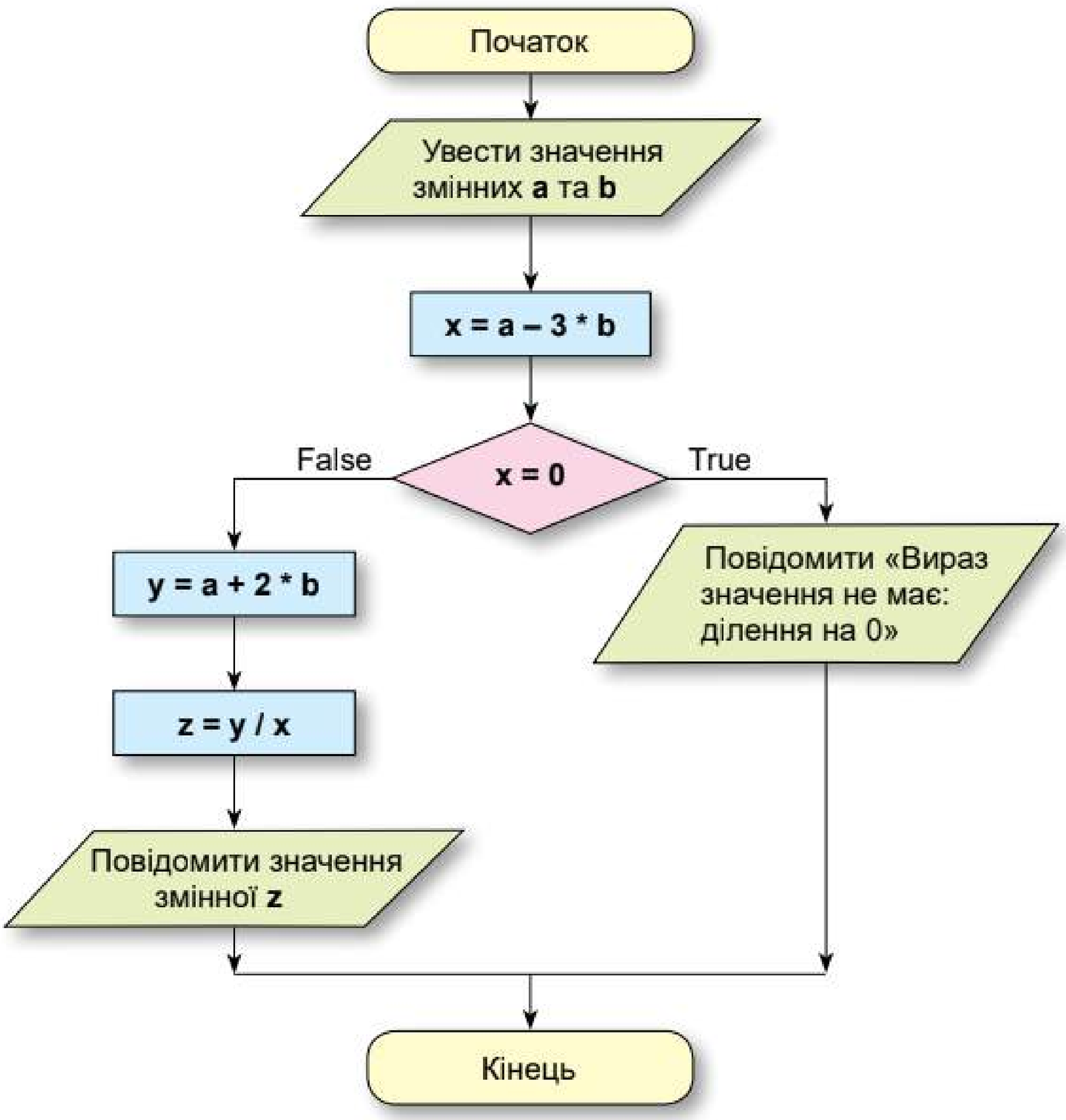
Розглянемо кілька задач, для розв'язування яких створюватимемо віконні проєкти з розгалуженнями, у яких використовуватимуться різні логічні вирази, у тому числі з логічними операціями.

Задача 1. Обчислити значення виразу $\frac{a+2b}{a-3b}$, де a, b – довільні числа.

Створимо математичну модель для цієї задачі:
вхідні дані: два довільних числа ***a*** і ***b***;
кінцеві результати: значення виразу ***z***;
формули: ***z*** **= (*a* + 2*b) / (*a* − 3*b)**, якщо $a - 3*b \neq 0$, або «Вираз значення не має: ділення на нуль», якщо $a - 3*b = 0$.

Оскільки вираз містить дію ділення на вираз, який може дорівнювати нулю (наприклад, при ***a*** **= 3** і ***b*** **= 1**), алгоритм розв'язування цієї задачі повинен містити розгалуження, у якому будуть враховані два можливих випадки: значення дільника ***a*** **− 3*b** дорівнює 0 і значення дільника ***a*** **− 3*b** не дорівнює 0.

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 4.22.



Мал. 4.22. Блок-схема алгоритму розв'язування **Задачі 1**

Оскільки математична модель задачі має два вхідних даних і один кінцевий результат, розмістимо у вікні два поля для введення значень вхідних даних, підпишемо їх у відповідних написах, і кнопку, після вибору якої відбуватимуться відповідні обчислення. Результат виводитимемо у вікно повідомлень.

Текст проєкту буде такий:

```
from tkinter import*
from tkinter.messagebox import*
def click():
    a = float(entry1.get())
    b = float(entry2.get())
    x = a - 3 * b
    if x == 0:
        showinfo('Результат', 'Вираз значення не має: ділення на нуль')
    else:
        y = a + 2 * b
        z = y / x
        showinfo('Результат', str(z))
root = Tk()
root.geometry('300x250')
label1 = Label(text = 'Значення a')
label1.pack(pady = 20)
entry1 = Entry()
entry1.pack()
entry1.focus_set()
label2 = Label(text = 'Значення b')
label2.pack(pady = 20)
entry2 = Entry()
entry2.pack()
button = Button(text = 'Обчислити', command = click)
button.pack(pady = 30)
```

Для перевірки правильності проєкту можна виконати його, наприклад, для таких наборів вхідних даних: $a = 6$, $b = 2$; $a = 2$, $b = 1$. Результати його виконання при цих значеннях вхідних даних наведено на малюнку 4.23.

Звертаємо вашу увагу.

Обчислення значення дільника $a - 3*b$ можна було б не здійснювати в окремій команді та не запам'ятовувати як значення змінної x, а здійснити безпосередньо в команді розгалуження. Також можна обчислення значення виразу $(a + 2*b) / (a - 3*b)$ записати не двома командами, а однією. Тоді обробник події був би таким:



Мал. 4.23. Результати виконання проєкту для значень вхідних даних $a = 6$, $b = 2$ (а) і $a = 2$, $b = 1$ (б)

```
def click():
    a = float(entry1.get())
    b = float(entry2.get())
    if a - 3 * b == 0:
        showinfo('Результат', 'Вираз значення не має: ділення на нуль')
    else:
        z = (a + 2 * b ) / (a - 3 * b)
        showinfo('Результат', str(z))
```

Розглянемо задачу, під час розв'язування якої в логічному виразі команди розгалуження використовується диз'юнкція.

Задача 2. Обчислити значення функції

$$y = \begin{cases} 2x - 12, \text{ якщо } x < -3 \text{ або } x > 5, \\ 7 - 8x \text{ для всіх інших } x. \end{cases}$$

Побудуємо математичну модель для цієї задачі:

вхідні дані: довільне число x;

кінцеві результати: значення функції y;

формули: $y = \begin{cases} 2x - 12, \text{ якщо } x < -3 \text{ або } x > 5, \\ 7 - 8x \text{ для всіх інших } x. \end{cases}$

Оскільки математична модель задачі має одне вхідне дане й один кінцевий результат, розмістимо у вікні два поля: одне для введення значень вхідного даного, друге – для виведення значення кінцевого результату, підпишемо їх у відповідних написах, і кнопку, після вибору якої відбуватимуться відповідні обчислення.

Текст проєкту буде такий:

```
from tkinter import*
def click():
    x = float(entry1.get())
    if x < -3 or x > 5: y = 2 * x - 12
    else: y = 7 - 8 * x
    entry2.delete(0, END)
    entry2.insert(0, str(y))
root = Tk()
root.geometry('300x250')
label1 = Label(text = 'Значення x')
label1.pack(pady = 10)
entry1 = Entry()
entry1.pack()
entry1.focus_set()
label2 = Label(text = 'Значення y')
label2.pack(pady = 10)
entry2 = Entry()
entry2.pack()
button = Button(text = 'Обчислити значення функції', command = click)
button.pack(pady = 20)
```

Для перевірки правильності проєкту можна виконати його, наприклад, для таких значень: $x = -4$; $x = 0$; $x = 7$.

Нагадуємо вам: тестувати проєкт з розгалуженням потрібно для таких наборів вхідних даних, щоб перевірити правильність його роботи при всіх можливих варіантах виконання.

Так, наприклад, проєкт для обчислення значення виразу $y = \dfrac{x-5}{x^2-9}$ містить команду розгалуження, оскільки його виконання залежить від того, дорівнює знаменник нулю чи не дорівнює. Тому його потрібно тестувати для значень x, за яких знаменник дорівнює нулю, тобто 3 і −3, і за яких знаменник не дорівнює нулю, наприклад 5 і 2.

Працюємо з комп'ютером

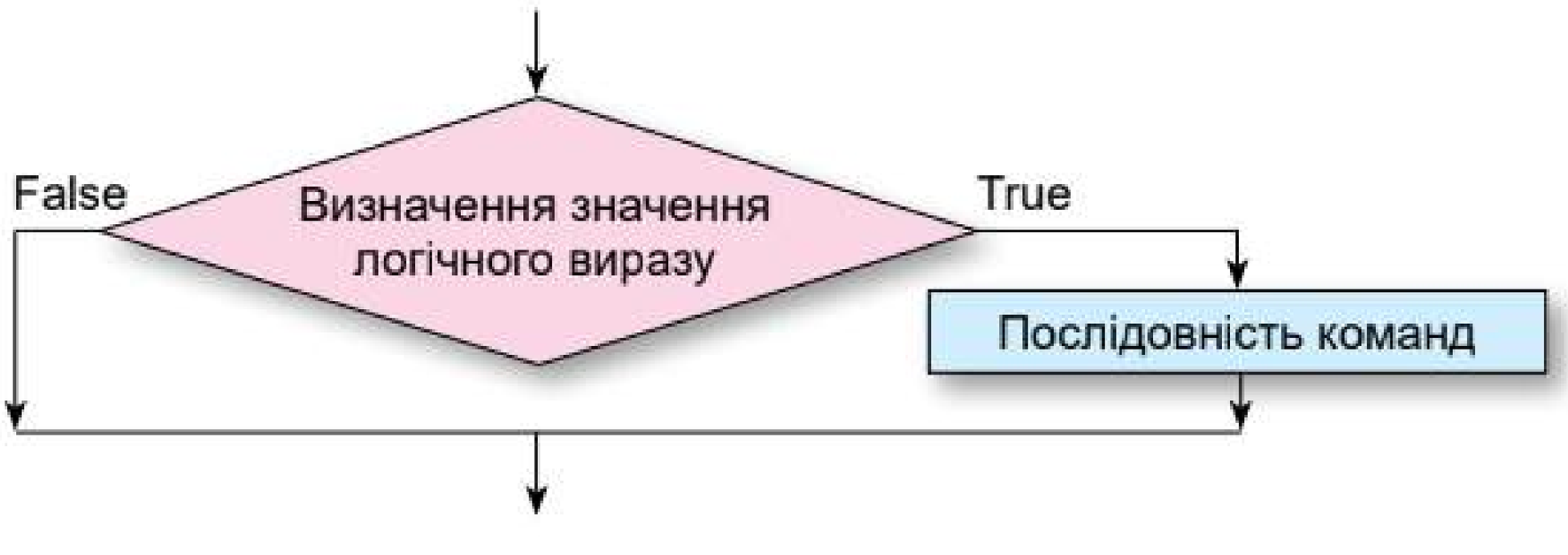
Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ue2cFJWD* або QR-кодом.

Найважливіше в цьому пункті

У загальному випадку кожне розгалуження розпочинається з **команди визначення значення логічного виразу**. Результатом виконання такої команди можуть бути значення **True** або **False**. І залежно від результату виконання цієї команди будуть виконуватися ті чи інші команди алгоритму (мал. 4.24, 4.25).



Мал. 4.24. Блок-схема повного розгалуження



Мал. 4.25. Блок-схема неповного розгалуження

За використання поняття логічного виразу команди розгалуження в мові програмування **Python** виглядатимуть так:

а) **команда повного розгалуження:**

```
if <логічний вираз>:
    <послідовність команд 1>
else:
    <послідовність команд 2>
```

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **True**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо це значення **False**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

б) **команда неповного розгалуження:**

```
if <логічний вираз>:
    <послідовність команд 1>
```

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **True**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **False**, одразу виконується команда, наступна за розгалуженням.

в) **команда багатоальтернативного розгалуження:**

```
if <логічний вираз 1>:
    <послідовність команд 1>
elif <логічний вираз 2>:
    <послідовність команд 2>
elif <логічний вираз 3>:
    <послідовність команд 3>
...
else:
    <послідовність команд N>
```

Виконання цієї команди відбувається так:

- визначається значення **логічного виразу 1**;
- якщо це значення **True**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням;
- якщо це значення **False**, то визначається значення **логічного виразу 2**;
- якщо це значення **True**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням;
- якщо це значення **False**, то визначається значення наступного логічного виразу і так далі;
- якщо всі логічні вирази в команді мають значення **False**, то виконується **послідовність команд N**, розміщена після **else**, і після цього виконується команда, наступна за розгалуженням.

Дайте відповіді на запитання

1. Який вигляд має команда повного розгалуження з використанням логічних виразів? Як вона виконується?

2. Який вигляд має команда неповного розгалуження з використанням логічних виразів? Як вона виконується?

3. Який вигляд має команда багатоальтернативного розгалуження з використанням логічних виразів? Як вона виконується?

4. У яких випадках в алгоритмах потрібно використовувати повне розгалуження, а у яких – неповне?

5. Як потрібно підбирати тестові набори значень вхідних даних для тестування проєкту з розгалуженням?

Обговоріть і зробіть висновки

1. Чи можна команду багатоальтернативного розгалуження замінити на інші команди розгалуження? Якщо так, наведіть приклади.



Виконайте завдання

1. Складіть блок-схему і поясніть виконання фрагмента проєкту:

```
if <логічний вираз>:
    <команди 1>
else:
    if <логічний вираз 2>:
        <команди 2>
    else:
        <команди 3>
```

2. Складіть блок-схему і поясніть виконання фрагмента проєкту:

```
if <логічний вираз>:
    <команди 1>
if <логічний вираз 2>:
    <команди 2>
```

Поясніть, чим цей фрагмент проєкту відрізняється від фрагмента проєкту в завданні № 1 і чим відрізняються їх виконання.

3. f – змінна логічного типу. Чи можна розгалуження **if f == True: <Команда>** замінити розгалуженням **if f: <Команда>**? Відповідь поясніть.

4. Створіть математичну модель і проєкт для обчислення значення виразу $(a + b) - c : (a - b)$. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.4**. Запишіть у зошит набори вхідних даних, для яких ви тестували цей проєкт.

5. Створіть математичну модель і проєкт для обчислення значення виразу $a + (b - c) : (a + 2b)$. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.5**. Запишіть у зошит набори вхідних даних, для яких ви тестували цей проєкт.

6. Створіть математичну модель і проєкт, який визначає менше з двох довільних чисел або виводить повідомлення, що числа рівні. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.6**. Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

7. Створіть математичну модель і проєкт, який визначає найменше з чотирьох довільних чисел. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.7**. Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

8. Створіть математичну модель і проєкт, який обчислює модуль довільного числа. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.8**. Запишіть у зошит значення вхідних даних, для яких ви тестували цей проєкт.

9. Створіть проєкт, у якому у вікні розміщується кнопка завширшки 40 символів. Після кожного вибору кнопки її ширина зменшується на 5 символів. А коли ширина кнопки стає менше, ніж 10 символів, її ширина знову стає 40 символів. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.9**.

10. Створіть математичну модель і проєкт для обчислення значення функції $y = \begin{cases} 15 - 3x, \text{ якщо } -1 < x < 4, \\ 6 + 4x \text{ для всіх інших } x. \end{cases}$ Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем

завдання 4.6.10. Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

11. Створіть математичну модель і проєкт для обчислення значення функції $y = \begin{cases} 2x - 4, \text{ якщо } x < 2, \\ x^2 + 5, \text{ якщо } 2 \leqslant x \leqslant 7, \\ 3 - 5x, \text{ якщо } x > 7. \end{cases}$ Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.11.** Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

12. Створіть математичну модель і проєкт для обчислення значення функції $y = \begin{cases} 2x - 4, \text{ якщо } x < 2, \\ x^2 + 5, \text{ якщо } 5 \leqslant x \leqslant 12, \\ 3 - 5x \text{ для всіх інших } x. \end{cases}$ Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.12.** Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

13. Створіть математичну модель і проєкт для визначення, чи можна з трьох відрізків із заданими довжинами утворити трикутник. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.13.** Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

14. Банк пропонує вкладнику два види депозитів: 15 % прибутку за рік і 7 % прибутку за пів року. В обох депозитах прибуток за вказаний час додається до вкладу. Створіть математичну модель і проєкт для порівняння прибутку за 1 рік по цих депозитах, якщо на обидва покласти одну й ту саму суму грошей. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.6.14.** Запишіть у зошит набори значень вхідних даних, для яких ви тестували цей проєкт.

Готуємось до вивчення нового матеріалу

1. Які елементи керування у віконних проєктах ви знаєте?
2. Для чого використовувався кожний з них?
3. Як використовувалися події з цими елементами керування?

4.7. ПРАПОРЦІ ТА ПЕРЕМИКАЧІ У ВІКОННИХ ПРОЄКТАХ

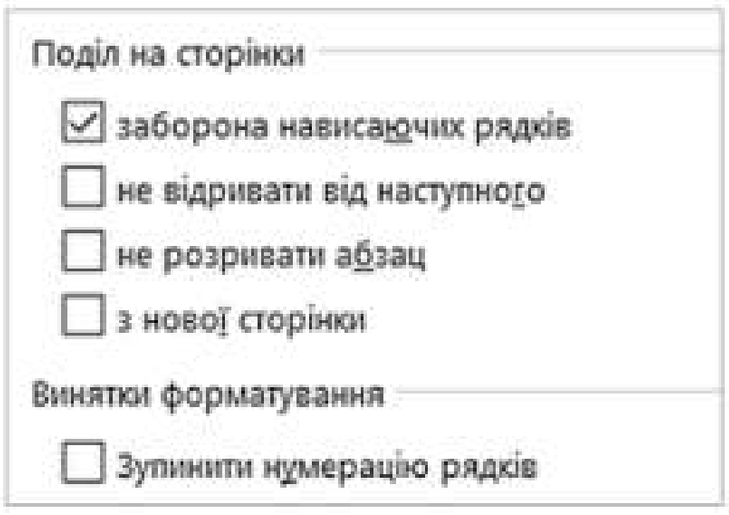
У цьому пункті йтиметься про:

- прапорці як елементи керування у віконних проєктах;
- перемикачі як елементи керування у віконних проєктах.

ПРАПОРЦІ ТА ЇХ ВЛАСТИВОСТІ

Пригадайте

● Який загальний вигляд команди неповного розгалуження в мові програмування **Python** і як вона виконується? ● Для чого використовували у програмах, з якими ви працювали, прапорці як елементи керування? Наведіть приклади.



Мал. 4.26. Приклад використання прапорців

У різних програмах прапорці використовують для вибору одного, або кількох варіантів з набору можливих, або жодного (мал. 4.26). Якщо прапорець вибраний, то в нього всередині з'являється позначка ☑.

Команда для створення прапорця у вікні проєкту мовою **Python** така:

<ім'я прапорця> = Checkbutton(text = '<текст підпису>'), (англ. *checkbutton* – прапорець).

Після розміщення у вікні прапорця, створеного у такий спосіб, він матиме визначений користувачем підпис. Наприклад, після виконання команд:

```
flag = Checkbutton(text = 'Ваш вибір')
flag.pack()
```

буде створено та розміщено у вікні прапорець з іменем **flag** і підписом *Ваш вибір* (мал. 4.27).



Мал. 4.27. Прапорець у вікні проєкту

Властивості прапорця **width**, **height**, **bg**, **fg**, **font** аналогічні до відповідних властивостей напису, поля та кнопки.

Щоб мати можливість дізнаватися під час виконання проєкту стан прапорця (вибраний або невибраний), потрібно пов'язати з ним спеціальну змінну логічного типу. Для створення такої змінної використовують команду виду:

<ім'я змінної> = BooleanVar()

Пов'язують цю змінну з прапорцем під час його створення командою, що має вигляд:

<ім'я прапорця> = Checkbutton(text = '<текст підпису>', variable = <ім'я змінної>), (англ. *variable* – змінна).

Наприклад, після виконання команд:

```
choice = BooleanVar()
flag = Checkbutton(text = 'Ваш вибір', variable = choice)
flag.pack()
```

буде створено та розміщено у вікні прапорець з іменем **flag** і підписом *Ваш вибір*, а змінна з іменем **choice** зберігатиме стан цього прапорця: **True** – якщо прапорець вибраний, **False** – якщо прапорець не вибраний.

Значення цієї змінної за замовчуванням – **False**, і після запуску проєкту на виконання позначка прапорця буде відсутня. Якщо під час виконання проєкту буде вибрано прапорець, то значення змінної стане **True**. За повторного вибору прапорця його стан зміниться на протилежний, і також зміниться значення змінної.

Щоб під час запуску проєкту прапорець одразу був у стані *вибраний*, потрібно використати команду виду **<ім'я змінної>.set(True)** (англ. *set* – установити), наприклад **choice.set(True)**.

Щоб визначити значення цієї змінної, під час виконання проєкту потрібно виконати команду виду **<ім'я змінної>.get()**, наприклад **choice.get()**. Значення цієї змінної можна використовувати у проєктах у команді розгалуження як значення логічного виразу.

Якщо у проєкті використовується кілька прапорців, то для кожного з них повинна бути створена окрема змінна, наприклад:

```
choice1 = BooleanVar()
flag1 = Checkbutton(text = 'Заголовок', variable = choice1)
flag1.pack()
choice2 = BooleanVar()
flag2 = Checkbutton(text = 'Колір', variable = choice2)
flag2.pack()
choice3 = BooleanVar()
flag3 = Checkbutton(text = 'Розміри', variable = choice3)
flag3.pack()
```

Розглянемо, як можна використати прапорці у проєктах з розгалуженнями.

Нехай, наприклад, нам потрібно в ході виконання проєкту змінювати значення або тільки однієї з властивостей вікна (або заголовок, або колір, або розміри), або кількох з них, або не змінювати значення жодної з них.

Для цього використаємо кнопку та три прапорці (мал. 4.28). Оскільки при виконанні проєкту жодний прапорець може бути не вибраний, то доцільно під час запуску проєкту на виконання для кожного з них залишити значення логічної змінної **False**, яке встановлюється за замовчуванням.



Мал. 4.28. Вікно проєкту з кількома прапорцями

Створимо процедуру для опрацювання події **Click** для кнопки і до її тексту помістимо такі три послідовні (не вкладені одна в одну) команди розгалуження:

```
if choice1.get(): root.title('Заголовок змінено')
if choice2.get(): root['bg'] = 'red'
if choice3.get(): root.geometry('500x200')
```

Якщо при настанні події **Click** для кнопки перший прапорець буде вибраним, то значення змінної **choice1** буде **True** і відповідно виконуватиметься команда **root.title('Заголовок змінено')**. Якщо ж перший прапорець не буде вибраним, то значення змінної **choice1** буде **False** і відповідно команда **root.title('Заголовок змінено')** не виконуватиметься, а виконуватиметься наступна команда, тобто наступна команда розгалуження для другого прапорця.

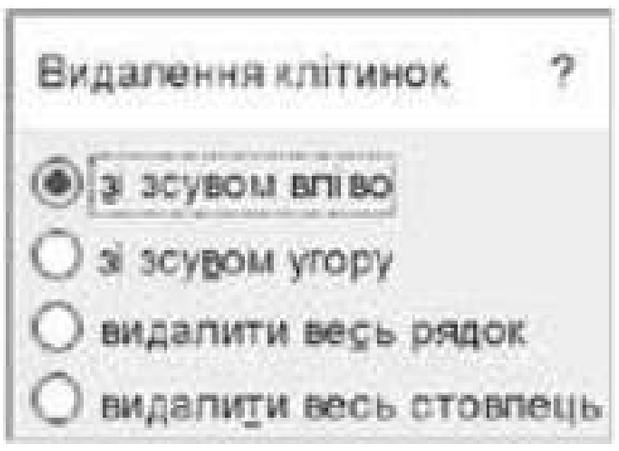
Аналогічно виконуватимуться і дві інші команди розгалуження.

ПЕРЕМИКАЧІ ТА ЇХ ВЛАСТИВОСТІ



Пригадайте

● Який загальний вигляд команди повного розгалуження в мові програмування **Python** і як вона виконується? ● Який загальний вигляд команди багатоальтернативного розгалуження в мові програмування **Python** і як вона виконується? ● Для чого використовувалися у програмах, з якими ви працювали, перемикачі? Наведіть приклади.



Мал. 4.29. Вікно з перемикачами

Перемикачі як елементи керування використовують для вибору тільки одного варіанта з кількох можливих (мал. 4.29).

Якщо у вікні у групі перемикачів розміщено кілька перемикачів, то серед них можна вибрати тільки один. Якщо перемикач вибраний, то в нього всередині з'являється позначка ◉.

Оскільки, на відміну від прапорців, з однієї групи перемикачів може бути вибраний тільки один, то важливо, який саме перемикач вибраний. Тому всі перемикачі з однієї групи повинні бути пов'язані з однією змінною цілого типу. Команда створення цієї змінної така:

<ім'я змінної> = IntVar()

Команда створення кожного з перемикачів групи має такий вигляд:

<ім'я перемикача> = Radiobutton(text = '<текст підпису>', variable = <ім'я змінної>, value = <значення змінної>), де:

- властивість **text** визначає текст підпису перемикача;
- властивість **variable** визначає ім'я змінної, яка пов'язана з перемикачем. Повинна бути одна й та сама змінна для всіх перемикачів однієї групи;
- властивість **value** (англ. *value* – значення) визначає значення, якого набуде змінна, якщо буде вибраний саме цей перемикач. Значення повинні бути різними для кожного з перемикачів однієї групи.

На малюнку 4.30 наведено фрагмент проєкту, у якому створено три перемикачі однієї групи з іменами **switch1**, **switch2** та **switch3**, для яких змінна **choice** набуває відповідно значень *1*, *2* та *3*.

```
choice = IntVar()
switch1 = Radiobutton(text = "Заголовок", variable = choice, value = 1)
switch1.pack(pady = 10)
switch2 = Radiobutton(text = "Колір", variable = choice, value = 2)
switch2.pack(pady = 10)
switch3 = Radiobutton(text = "Розмір", variable = choice, value = 3)
switch3.pack(pady = 10)
```

Мал. 4.30. Фрагмент проєкту зі створення трьох перемикачів однієї групи

Властивості перемикача **width**, **height**, **bg**, **fg**, **font** аналогічні до відповідних властивостей прапорця та інших елементів керування.

За замовчуванням змінна **choice** має значення 0. Оскільки це значення не відповідає жодному перемикачу, то після запуску проєкту на виконання жоден перемикач не буде вибраним. Щоб один з перемика-

чів одразу був вибраним, потрібно встановити для змінної відповідне значення командою виду:

<ім'я змінної>.set(<значення змінної>).

Наприклад, після виконання команди **choice.set(1)** стане вибраним перший перемикач.

Щоб дізнатися в ході виконання проєкту, який з перемикачів вибрано, потрібно виконати команду виду **<ім'я змінної>.get()**, наприклад **choice.get()**.

Якщо потрібно розмістити у вікні кілька незалежних груп перемикачів, то кожну групу перемикачів потрібно пов'язати з окремою змінною.

Розглянемо, як можна, використовуючи перемикачі, змінювати значення однієї з властивостей вікна: або заголовок, або колір, або розміри.

Використаємо у проєкті три перемикачі та кнопку (мал. 4.31).



Мал. 4.31. Вікно проєкту з перемикачами

Створимо процедуру – обробник події **Click** для кнопки і до її тексту помістимо таку команду:

```
if choice.get() == 1:
    root.title('Заголовок змінено')
elif choice.get() == 2:
    root['bg'] = 'red'
else:
    root.geometry('600x300')
```

Розпочинатися виконання цієї команди буде з визначення значення змінної **choice.**

Якщо при настанні події **Click** для кнопки вибраним буде перший перемикач, то значення логічного виразу **choice.get() == 1** дорівнюватиме **True.** Тому виконається команда змінення заголовка вікна, а інші частини цієї команди розгалуження виконуватися не будуть.

Якщо при настанні події **Click** для кнопки вибраним буде другий перемикач, то значення логічного виразу **choice.get() == 1** дорівнюватиме **False.** Тому буде виконуватися команда після ключового слова **elif.** За її виконання визначається значення змінної **choice.** Оскільки буде вибраний другий перемикач, то це значення дорівнює **2.** Тому виконається команда змінення кольору вікна, а інші частини команди розгалуження виконуватися не будуть.

Якщо ж при настанні події **Click** для кнопки вибраним буде третій перемикач, то значення логічних виразів **choice.get() == 1** і **choice.get() == 2** дорівнюватимуть **False.** Тому не будуть виконуватися команди змінення заголовка вікна та кольору вікна, а виконуватиметься команда після ключового слова **else,** тобто команда змінення розмірів вікна.

Якщо під час виконання проєкту сталося так, що не буде вибраний жодний з перемикачів, то після вибору кнопки зміняться розміри вікна (*подумайте самостійно, чому відбудуться саме такі змінення*). Щоб так не сталося, потрібно замість ключового слова **else** використати ще одну команду **elif:**

```
if choice.get() == 1:
    root.title('Заголовок змінено')
elif choice.get() == 2:
    root['bg'] = 'red'
elif choice.get() == 3:
    root.geometry('600x300')
```

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/He2cGipi* або QR-кодом.

Найважливіше в цьому пункті

Прапорці використовують для вибору одного, або кількох варіантів з набору можливих, або жодного. З кількох прапорців може бути вибраний або один, або кілька, або всі, або жодний.

Команда для створення прапорця у мові **Python** така:
<ім'я прапорця> = Checkbutton(text = '<текст підпису>', variable = <ім'я змінної>), де змінна логічного типу використовується для визначення стану прапорця та створюється командою **<ім'я змінної> = BooleanVar()**.

Щоб під час запуску проєкту прапорець одразу був у стані *вибраний*, потрібно використати команду виду **<ім'я змінної>.set(True)**. Щоб дізнатися значення цієї змінної під час виконання проєкту, потрібно виконати команду виду **<ім'я змінної>.get()**. Значення цієї змінної можна використовувати у проєктах у команді розгалуження як логічний вираз.

Для опрацювання стану прапорців (вибраний або невибраний) у проєкті використовують для кожного з них послідовно записані команди розгалуження, які незалежні одна від іншої.

Перемикачі використовують для вибору тільки одного з кількох можливих варіантів. З кількох перемикачів однієї групи може бути вибраний тільки один.

Команда для створення перемикача в мові **Python** має вигляд:
<ім'я перемикача> = Radiobutton(text = '<текст підпису>', variable = <ім'я змінної>, value = <значення змінної>), де змінна використовується для визначення вибраного перемикача та створюється командою **<ім'я змінної> = IntVar()**.

Якщо потрібно, щоб один з перемикачів групи відразу був вибраним, то слід установити для змінної відповідне значення командою виду **<ім'я змінної>.set(True)**.

Щоб дізнатися в ході виконання проєкту, який з перемикачів однієї групи вибрано, потрібно виконати команду виду **<ім'я змінної>.get()**.

У вікні можна також розмістити кілька незалежних одна від іншої груп перемикачів. Для цього потрібно кожну групу перемикачів пов'язати з окремою змінною. Тоді в кожній групі можна вибрати по одному перемикачу.

Для опрацювання стану прапорців (вибраний або невибраний) у проєкті використовують команду багатоальтернативного розгалуження.

Дайте відповіді на запитання

1. Для чого використовують прапорці, а для чого – перемикачі?
2. Як визначити стан прапорця у проєкті мовою **Python**?
3. Як потрібно записувати команди розгалуження, у яких перевіряється, вибрано прапорець чи ні?
4. Як визначити, який перемикач вибрано у проєкті мовою **Python**?
5. Як доцільно записувати команди розгалуження, у яких перевіряється, вибрано перемикач чи ні?



Обговоріть і зробіть висновки

1. Чи можна для прапорців використати команду багатоальтернативного розгалуження?

2. Чи можна для перемикачів замість команди багатоальтернативного розгалуження використати команди розгалуження, вкладені одна в одну?

3. Чи можна для перемикачів замість команди багатоальтернативного розгалуження використати команди розгалуження, записані послідовно, одна за одною?



Виконайте завдання

1. Створіть блок-схему та поясніть хід виконання послідовності команд:

```
if choice 1.get():
    <команди 1>
if choice 2.get():
    <команди 2>
```

де **choice1** і **choice2** – змінні логічного типу, пов'язані з прапорцями.

2. Створіть блок-схему та поясніть хід виконання послідовності команд:

```
if choice.get() == 1:
    <команди 1>
elif choice.get() == 2:
    <команди 2>
else:
    <команди 3>
```

де **choice** – змінна цілого типу, пов'язана з групою перемикачів.

3. Поясніть відмінності у виконанні команд розгалуження із завдання № 1 та із завдання № 2.

4. Сформулюйте життєву задачу, у проєкті для розв'язування якої можна використати фрагмент із завдання № 1.

5. Сформулюйте життєву задачу, у проєкті для розв'язування якої можна використати фрагмент із завдання № 2.

6. Створіть математичну модель і проєкт, у якому можна обчислити суму, різницю, добуток, частку від ділення двох довільних чисел залежно від установлених позначок прапорців. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.7.6**.

7. Створіть математичну модель і проєкт, у якому можна обчислити або суму, або різницю, або добуток, або частку від ділення двох довільних чисел залежно від вибраного перемикача. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.7.7**.

8. Робітник виготовляє деталі. Відомо кількість деталей, виготовлених робітником за місяць, і вартість однієї деталі. Якщо робітник виконав план, то він одержує премію в розмірі 10 % від зароблені в результаті виготовлення деталей суми. Якщо робітник перевиконав план, то його премія становить 30 % від зароблені суми. Створіть математичну модель і проєкт з використанням перемикачів, який визначає суму грошей, зароблену робітником за місяць. Математичну

модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.7.8.**

9. Робітниця має фіксовану заробітну плату за роботу протягом місяця. Якщо в робітниці є діти, вона отримує доплату в розмірі 20 % від її місячної заробітної плати. Якщо в робітниці протягом місяця були порушення трудової дисципліни, то її заробітна плата зменшується на 15 %. Створіть математичну модель і проєкт з використанням прапорців, який визначає суму грошей, зароблену робітницею за місяць. Математичну модель запишіть у зошит. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.7.9.**

10. Створіть проєкт з використанням двох груп перемикачів і кнопки. Перемикачі першої групи визначають колір, у який зафарбовується вікно, а перемикачі другої групи визначають, якими мають стати розміри вікна. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.7.10.**



Практична робота № 5. «Проєкти з розгалуженнями»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Створіть математичну модель і проєкт для обчислення значення виразу $x = (a + 3*b)/(2*a - b)$. Збережіть проєкт у вашій папці у файлі з іменем **практична 5.1**.
2. Створіть проєкт для знаходження найменшого із заданих чотирьох чисел. Збережіть проєкт у вашій папці у файлі з іменем **практична 5.2.**
3. Вкладник хоче покласти в банк на депозит певну суму грошей на 2 роки. Банк пропонує два види депозитів: під 15 % річних і під 17 % річних. У першому депозиті прибуток додається до вкладу кожного року, у другому – ні. Створіть математичну модель і проєкт з використанням перемикачів для обчислення суми грошей, яку зможе отримати вкладник через два роки, якщо він вибере один з цих депозитів. Збережіть проєкт у вашій папці у файлі з іменем **практична 5.3.**



Готуємось до вивчення нового матеріалу

1. Які процеси називаються циклічними? Наведіть приклади.
2. Які ви знаєте команди циклу?



4.8. ЦИКЛИ З ЛІЧИЛЬНИКОМ

У цьому пункті йтиметься про:

- цикли;
- команду циклу з лічильником;
- використання команди циклу з лічильником у проєктах.

КОМАНДА ЦИКЛУ З ЛІЧИЛЬНИКОМ



Пригадайте

● Які процеси називають циклічними? Наведіть приклади. ● Що таке *цикл в алгоритмі*? ● Який вигляд має команда циклу з лічильником у мові програмування, яку ви вивчали раніше? Як вона виконується?

Ви вже знаєте, що в багатьох алгоритмах одна або кілька команд повинні повторюватися багато разів поспіль. Недоцільно включати в алгоритм ці команди стільки разів, скільки вони повинні повторюватися, тому що кількість разів повторення може бути дуже велика, або навіть заздалегідь невідома.

У таких випадках потрібно в алгоритмах використовувати цикли. Нагадаємо, що **цикл у алгоритмі** – це фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз.

У мовах програмування є кілька команд, які можуть реалізувати цикл. Одна з них – **команда циклу з лічильником.** Її доцільно використовувати в тих випадках, коли кількість повторень команд тіла циклу відома ще до початку виконання циклу. Наприклад, якщо відомо, скільки років гроші будуть зберігатися на рахунку в банку.

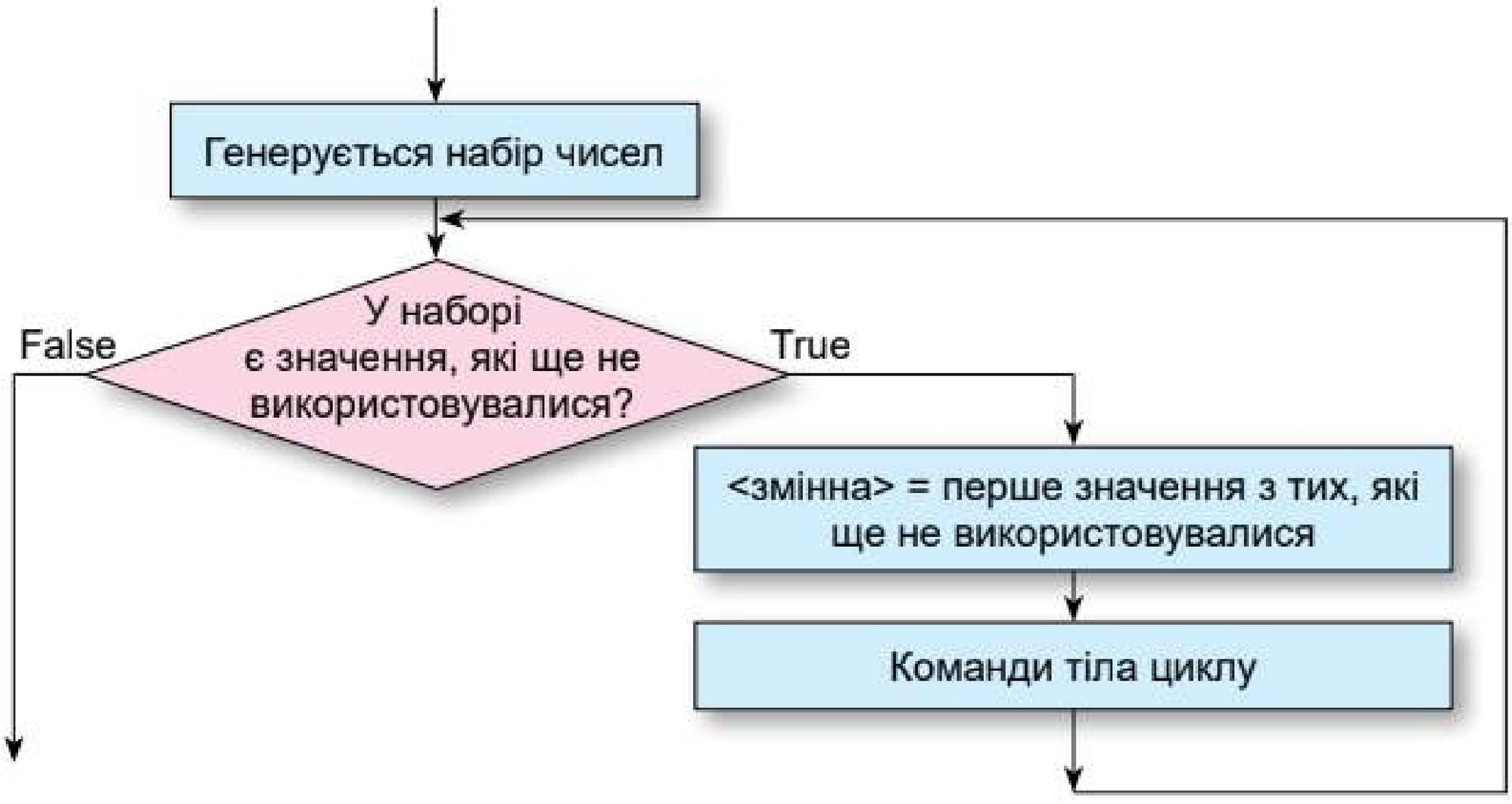
Загальний вигляд команди циклу з лічильником у мові **Python** такий:

for <ім'я змінної> in range(<кількість повторень>):
    <команди тіла циклу>

Рядок **for <ім'я змінної> in range(<кількість повторень>)** (англ. *for* – для, *in range* – у діапазоні) називається **рядком заголовка команди циклу з лічильником.** Змінна в рядку заголовка команди циклу з лічильником називається **лічильником циклу.**

Функція **range(<кількість повторень>)** утворює (генерує) набір послідовних цілих чисел, перше з яких 0, а останнє – на 1 менше від заданої кількості повторень (усього чисел стільки, як задано кількість повторень). Наприклад, функція *range(3)* генерує набір з 3 чисел: 0, 1, 2. Змінна – лічильник циклу послідовно набуває значень із цього набору чисел. Тіло циклу виконується, поки лічильник отримує нові значення. Коли нового значення в наборі немає, виконання циклу припиняється.

Блок-схему виконання команди циклу з лічильником в **Python** наведено на малюнку 4.32.



Мал. 4.32. Блок-схема виконання команди циклу з лічильником

Виконується команда циклу з лічильником так:

1. Функція **range(<кількість повторень>)** генерує набір цілих чисел від 0 до **кількість повторень − 1**.
2. Виконується перевірка, чи є в наборі числа, які ще не використовувалися і могли б стати новим значенням **змінної − лічильника циклу**.
3. Якщо результат перевірки **True**, то **змінна − лічильник циклу** набуває значення першого числа з набору чисел, які ще не використовувалися, і після цього виконується команда 4. Якщо результат перевірки **False**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Виконуються команди тіла циклу.
5. Повторюється виконання дії команди 2.

У записі команди циклу з лічильником усі команди тіла циклу повинні бути записані з однаковим відступом праворуч від першої літери слова **for**.

Розглянемо як приклад проєкт для розв'язування задачі, у якому використовується цикл з лічильником.

Задача. Створити проєкт для обчислення кількості кульок у піраміді (мал. 4.33).



Мал. 4.33. Пірамідка

Створимо математичну модель для цієї задачі, враховуючи, що в першому зверху шарі пірамідки 1 кулька, у другому − 2 ряди по 2 кульки, у третьому − 3 ряди по 3 кульки і так далі. Врахуємо також, що кількість шарів у пірамідці може бути довільним натуральним числом.

Математична модель:

вхідні дані: кількість шарів у пірамідці (n);

кінцеві результати: кількість кульок у пірамідці (s);

формули: $s = 1^2 + 2^2 + 3^2 + 4^2 + \ldots + n^2$.

Проєкт для розв'язування цієї задачі та вікно його виконання наведено на малюнку 4.34.



Мал. 4.34. Проєкт розв'язування **Задачі** та вікно його виконання

Виконаємо цей проєкт для $n = 4$.

Команда	Результат виконання				
	Крок 1	Крок 2	Крок 3	Крок 4	Крок 5
n = int(ent1.get())	n = 4				
s = 0	s = 0				
range(4)	0, 1, 2, 3				
У наборі ще є невикористані числа?	**True**	**True**	**True**	**True**	**False**
Змінна **i** набуває першого значення з набору невикористаних чисел	*i = 0*	*i = 1*	*i = 2*	*i = 3*	
s = s + (i + 1) * (i + 1)	s = 0 + + $(0 + 1)^2$ = = 1	s = 1 + + $(1 + 1)^2$ = = 5	s = 5 + + $(2 + 1)^2$ = = 14	s = 14 + + $(3 + 1)^2$ = = 30	
ent2.insert(0, str(s))					30

Звертаємо вашу увагу: після завершення циклу змінна – лічильник циклу *i* матиме значення 3. Це останнє значення з набору чисел, згенерованого функцією *range(4)*.

Окрім функції **range(<кількість повторень>)**, у команді циклу з лічильником можна використовувати функцію **range(<початок діапазону>, <кінець діапазону>, <крок>)**, яка має 3 аргументи цілого типу:

- **початок діапазону** – перше ціле число, яке потрапить до згенерованого набору;
- **кінець діапазону** – ціле число, яке не увійде до згенерованого набору та яке більше за всі числа з набору, якщо **крок** > 0, і менше всіх чисел з набору, якщо **крок** < 0;
- **крок** – різниця між кожними двома сусідніми цілими числами у згенерованому наборі чисел.

Наприклад, функція **range(3, 8, 2)** згенерує такий набір чисел: *3*, 3 + 2 = *5*, 5 + 2 = *7*. Наступне число 7 + 2 = *9* не увійде в набір чисел, оскільки більше за другий аргумент функції.

Кожен з аргументів функції **range** може бути як додатним, так і від'ємним цілим числом. **Початок діапазону** та **кінець діапазону** можуть також набувати значення 0.

ТЕСТУВАННЯ ПРОЄКТУ, ЩО МІСТИТЬ ЦИКЛ З ЛІЧИЛЬНИКОМ



Пригадайте

- Для чого тестують проєкти?
- У чому полягає процес тестування проєкту?
- Як потрібно тестувати проєкт з розгалуженнями?

Під час тестування проєкту, що містить цикл з лічильником, доцільно добирати такі набори вхідних даних, щоб тіло циклу виконувалося 1 раз; 2 рази; 3 рази; 5 разів та за потреби й іншу кількість разів.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Ie2cGz5m* або QR-кодом.

Найважливіше в цьому пункті

Цикл в алгоритмі – це фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз. **Цикл з лічильником** використовується в тих випадках, коли перед його виконанням відомо, скільки разів повинні виконатися команди тіла циклу.

Загальний вигляд команди циклу з лічильником у мові **Python** такий:

for <ім'я змінної> in range(<кількість повторень>):
<команди тіла циклу>

Рядок **for <ім'я змінної> in range(<кількість повторень>)** називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником називається **лічильником циклу**.

Функція **range(<кількість повторень>)** утворює (генерує) набір послідовних цілих чисел, перше з яких 0, а останнє – на 1 менше від заданої кількості повторень. Змінна – лічильник циклу послідовно набуває значень із цього набору чисел.

Виконується команда циклу з лічильником так:

1. Функція **range(<кількість повторень>)** генерує набір цілих чисел від 0 до **кількість повторень – 1**.
2. Виконується перевірка, чи є в наборі числа, які ще не використовувалися і могли б стати новим значенням **змінної – лічильника циклу**.
3. Якщо результат перевірки **True**, то **змінна – лічильник циклу** набуває значення першого з набору чисел, які ще не використовувалися, і після цього виконується дія 4; якщо **False**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Виконуються команди тіла циклу.
5. Повторюється виконання дії 2.

У записі команди циклу з лічильником усі команди тіла циклу повинні бути записані з однаковим відступом праворуч від першої літери у слові **for**.

У команді циклу з лічильником можна також використовувати функцію
range(<початок діапазону>, <кінець діапазону>, <крок>),
яка має 3 аргументи цілого типу:

- **початок діапазону** – перше ціле число, яке потрапить до згенерованого набору;
- **кінець діапазону** – ціле число, яке не увійде до згенерованого набору й яке більше за всі числа з набору, якщо **крок** > 0, і менше від усіх чисел з набору, якщо **крок** < 0;
- **крок** – різниця між кожними двома сусідніми цілими числами у згенерованому наборі чисел.

Дайте відповіді на запитання

1. Що таке *цикл в алгоритмі*?

2. Який вигляд має команда циклу з лічильником у мові програмування **Python**?

3. Як виконується команда циклу з лічильником у мові програмування **Python**?

4. Які ви знаєте приклади задач, що можна розв'язати з використанням циклу з лічильником?

5. На яких тестових наборах вхідних даних доцільно тестувати проєкт із циклом з лічильником?



Обговоріть і зробіть висновки

1. Чи можуть команди тіла циклу з лічильником виконуватися тільки один раз? Наведіть приклад.

2. Чи можуть команди тіла циклу з лічильником не виконуватись жодного разу? Наведіть приклад.

3. Чи може виконання циклу з лічильником ніколи не закінчитися? Поясніть свою відповідь.

4. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, циклу?



Виконайте завдання

1. Створіть у зошиті блок-схему фрагмента проєкту і виконайте його:

а)
```
n = 1
for i in range(4):
    n = n * (i + 1)
```

б)
```
p = 0
a = 2
for i in range(3, 7, 1):
    a = 3 * a + 1
    p = p + a
```

2. Створіть блок-схему фрагмента проєкту і виконайте його:

а)
```
n = 1
for i in range(4):
    n = n + (i + 1)**3
```

б)
```
p = 0
a = 4
for i in range(2, 6, 1):
    a = 2 * a - 1
    p = p + a
```

3. Створіть математичну модель і блок-схему алгоритму знаходження суми 5 чисел, перше з яких дорівнює 7, а кожне з чотирьох наступних на 3 більше за попереднє. Виконайте складений алгоритм.

4. Створіть математичну модель і блок-схему алгоритму знаходження суми 6 чисел, перше з яких дорівнює 100, а кожне з п'яти наступних дорівнює попередньому, поділеному на 5. Виконайте складений алгоритм.

5. Створіть математичну модель і проєкт для обчислення суми n чисел, перше з яких дорівнює додатному числу x, а кожне наступне у 2 рази більше за попереднє. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт, і отриманий результат. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.8.5**.

6. Створіть математичну модель і проєкт для обчислення суми n чисел, перше з яких дорівнює x, а кожне наступне дорівнює квадрату попереднього. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт, і отриманий результат. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.8.6**.

7. Людина поклала в банк певну суму грошей. Щорічно банк додає до вкладу 20 % від суми, що зберігається на рахунку після попереднього року. Створіть математичну модель і проєкт для визначення суми, яка буде на рахунку через n років. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт, і отриманий результат. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.8.7**.

Висловіть гіпотезу: на скільки років потрібно покласти гроші, щоб їх сума збільшилася вдвічі? Запишіть гіпотезу в зошит. Використовуючи проєкт, перевірте свою гіпотезу. Запишіть у зошит, чи підтвердилася ваша гіпотеза і чому.

8. Людина поклала в банк певну суму грошей. Щорічно банк додає до вкладу заздалегідь визначений відсоток від суми, що зберігається на рахунку на кінець поточного року. Створіть математичну модель і проєкт для визначення, на скільки зросте сума на рахунку через n років. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт, і отриманий результат. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.8.8**.

9. Спортсмен, готуючись до участі в марафоні, у перший день тренувань подолав дистанцію в s км. Кожного наступного дня його денна дистанція збільшувалася на p % від дистанції попереднього дня. Створіть математичну модель і проєкт для визначення:

1) дистанції, яку спортсмен здолає на n-й день тренувань;
2) сумарної дистанції за n днів тренувань;
3) наскільки більшою буде дистанція в n-й день тренувань за дистанцію першого дня?

Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.8.9**.

Висловіть гіпотезу: скільки кілометрів потрібно пробігти в перший день і на скільки відсотків потрібно збільшувати дистанцію щодня, щоб за тиждень сумарно пробігти не менше ніж 200 км? Запишіть гіпотезу в зошит. Використовуючи проєкт, перевірте свою гіпотезу. Запишіть у зошит, чи підтвердилася ваша гіпотеза і чому.

Готуємось до вивчення нового матеріалу

1. Чи знаєте ви інші види циклів, крім циклу з лічильником? Наведіть приклади таких циклів з життя. Як вони записуються, як виконуються?
2. У яких випадках в алгоритмах потрібно використовувати інші цикли, а не цикл з лічильником?

4.9. ЦИКЛИ З ПЕРЕДУМОВОЮ

У цьому пункті йтиметься про:

- команду циклу з передумовою;
- використання команди циклу з передумовою у проєктах.

ЦИКЛИ З ПЕРЕДУМОВОЮ

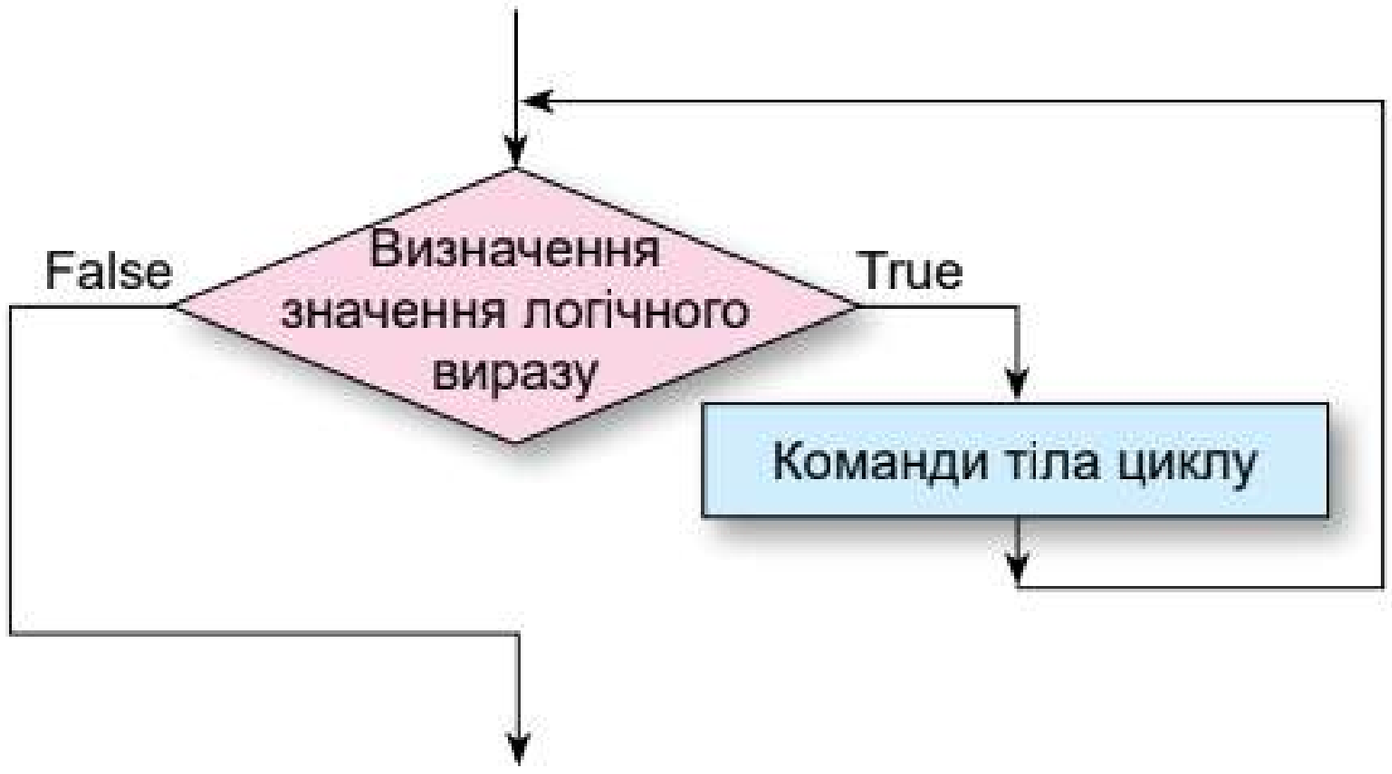
Пригадайте

• У яких випадках в алгоритмах використовують цикл з лічильником? • Який загальний вигляд команди циклу з лічильником у мові програмування **Python** і як вона виконується? • Як виконується команда **повтори до**  в мові програмування **Scratch**?

Ви вже знаєте, якщо перед виконанням циклу невідомо, скільки разів мають виконатися команди тіла циклу, то використовувати цикл з лічильником не можна.

У таких випадках використовують інші цикли, зокрема **цикл з передумовою.**

Блок-схему циклу з передумовою наведено на малюнку 4.35.



Мал. 4.35. Блок-схема циклу з передумовою

Команда циклу з передумовою в мові програмування **Python:**

while<логічний вираз>:
    <команди тіла циклу>
(англ. *while* – поки).

Виконання команди циклу з передумовою відбувається так: визначається значення **логічного виразу**; якщо це значення дорівнює **True**, то виконуються **команди тіла циклу**, після чого знову визначається значення **логічного виразу**, і якщо це значення знову дорівнює **True**, то знову виконуються **команди тіла циклу**, після чого знову визначається значення **логічного виразу**; якщо значення **логічного виразу** дорівнює **False**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.

Звертаємо вашу увагу: серед команд тіла циклу можуть бути лінійні фрагменти, розгалуження, інші цикли.

Для тих, хто хоче знати більше

Якщо потрібно достроково перервати виконання циклу, то можна використати команду **break** (англ. *break* – переривати, перерва).

Розглянемо задачу, алгоритм розв'язування якої містить цикл з передумовою.

Задача. Фірма щомісячно переказує кошти на купівлю дронів. Першого місяця вона переказала x грн, а кожного наступного – на y грн більше. Створити математичну модель і проєкт для визначення кількості місяців, через які загальна переказана сума всіх коштів перевищить **100 000** грн.

Створимо математичну модель для цієї задачі:

вхідні дані: кошти, переведені в першому місяці (x); число, на яке сума коштів у наступному місяці більша за суму коштів за попе-

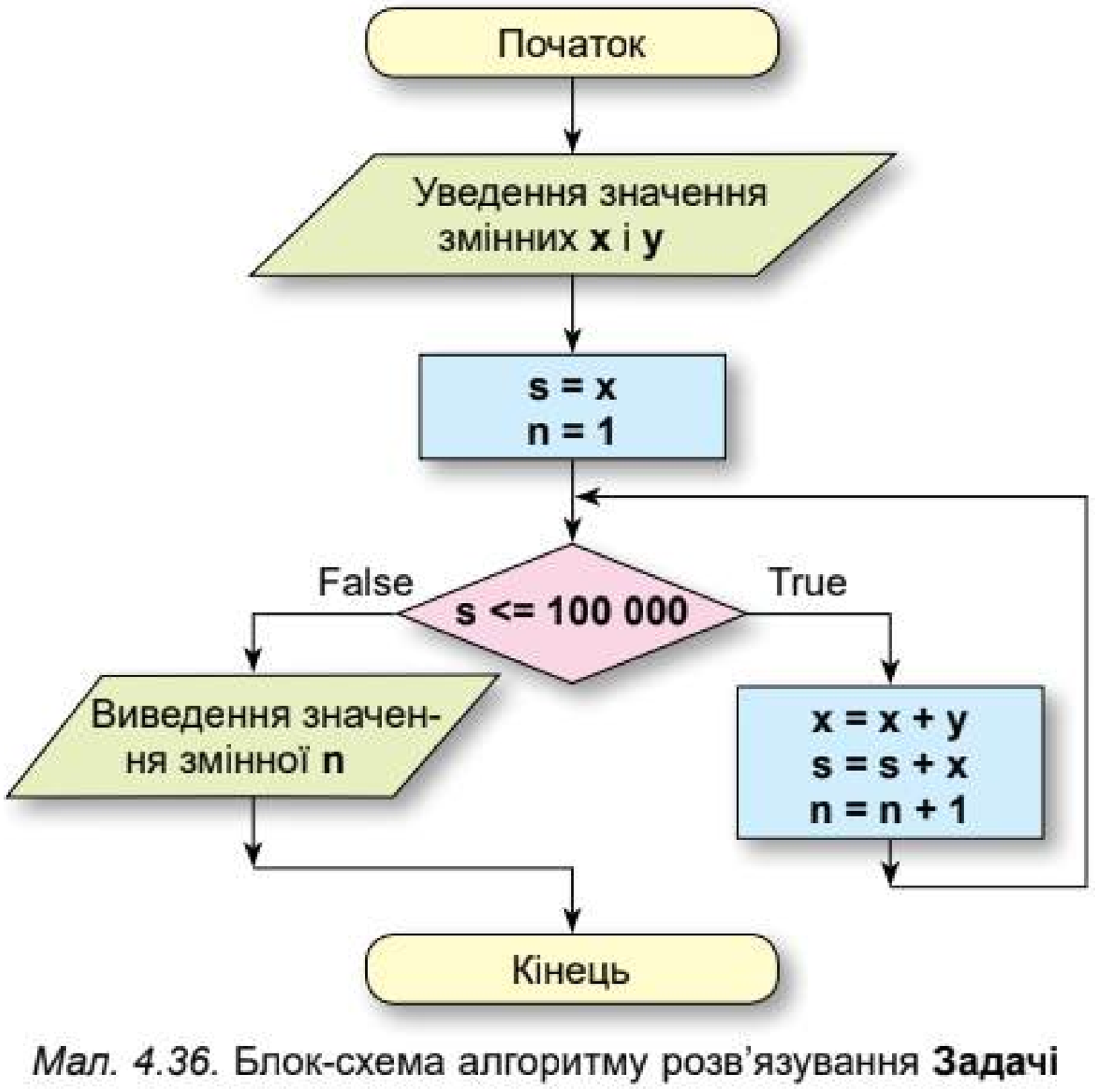
редній місяць (**y**); число, яке має перевищити загальна сума грошей (**100 000**);
кінцеві результати: кількість місяців (**n**);
формули: $x = x + y$, $s = s + x$, $s <= 100\ 000$.

З трьох вхідних даних два потрібно вводити (**x** і **y**), а третє – конкретне число. У задачі кінцевий результат, який потрібно виводити, один.

Перед початком циклу присвоїмо змінній **s**, яку використаємо для збереження загальної суми, значення першого доданка (**x**). А змінній **n**, яку використаємо для збереження кількості потрібних місяців, присвоїмо значення 1, оскільки в сумі вже враховано один (перший) доданок.

До суми потрібно додавати наступні доданки, поки ця сума буде менша або дорівнюватиме 100 000. Тому логічний вираз у заголовку циклу матиме вигляд $s <= 100\ 000$. У тілі циклу шукатимемо наступний доданок ($x = x + y$), додаватимемо його до поточного значення суми ($s = s + x$) і збільшуватимемо кількість доданих доданків на 1 ($n = n + 1$).

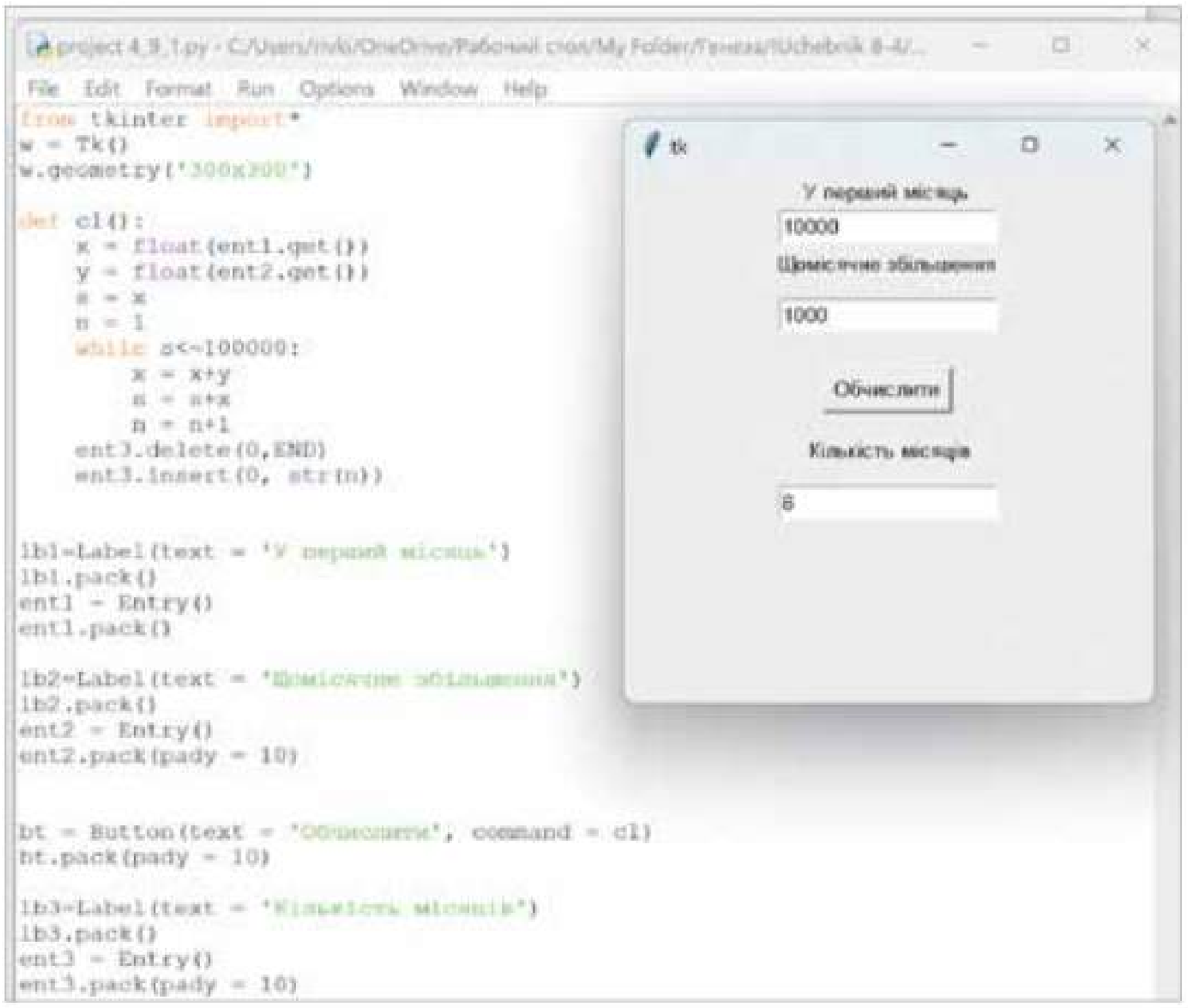
Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 4.36.



Мал. 4.36. Блок-схема алгоритму розв'язування **Задачі**

Проєкт для розв'язування цієї задачі та вікно його виконання наведено на малюнку 4.37.

Тестувати проєкт, що містить цикл з передумовою, доцільно для таких наборів значень вхідних даних, щоб тіло циклу не виконувалося жодного разу, виконувалося 1 раз; 2 рази; 3 рази; 5 разів і за потреби й іншу кількість разів.



Мал. 4.37. Проєкт розв'язування **Задачі** та вікно його виконання

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/4e2cGDfs* або QR-кодом.

Найважливіше в цьому пункті

Команда циклу з передумовою в мові програмування **Python**:

while<логічний вираз>:

<команди тіла циклу>

(англ. *while* – поки).

Виконання команди циклу з передумовою відбувається так: визначається значення **логічного виразу**; якщо це значення дорівнює **True**, то виконуються **команди тіла циклу**, після чого знову визначається значення **логічного виразу**, і якщо це значення знову дорівнює **True**, то знову виконуються **команди тіла циклу**, після чого знову визначається значення **логічного виразу**; якщо значення **логічного виразу** дорівнює **False**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.

Дайте відповіді на запитання

1. Який вигляд має блок-схема циклу з передумовою? Поясніть виконання цього циклу.

2. Який вигляд має команда циклу з передумовою в мові програмування **Python**? Як вона виконується?

3. Чи можуть команди тіла циклу з передумовою не виконуватися жодного разу? Поясніть свою відповідь. Наведіть приклади.

4. Чи може виконання циклу з передумовою ніколи не закінчитися? Поясніть свою відповідь.

Обговоріть і зробіть висновки

1. Що спільного і що відмінного у циклів з лічильником і з передумовою?

2. Чи можна команду циклу з лічильником замінити командою циклу з передумовою? Якщо так, наведіть приклад.

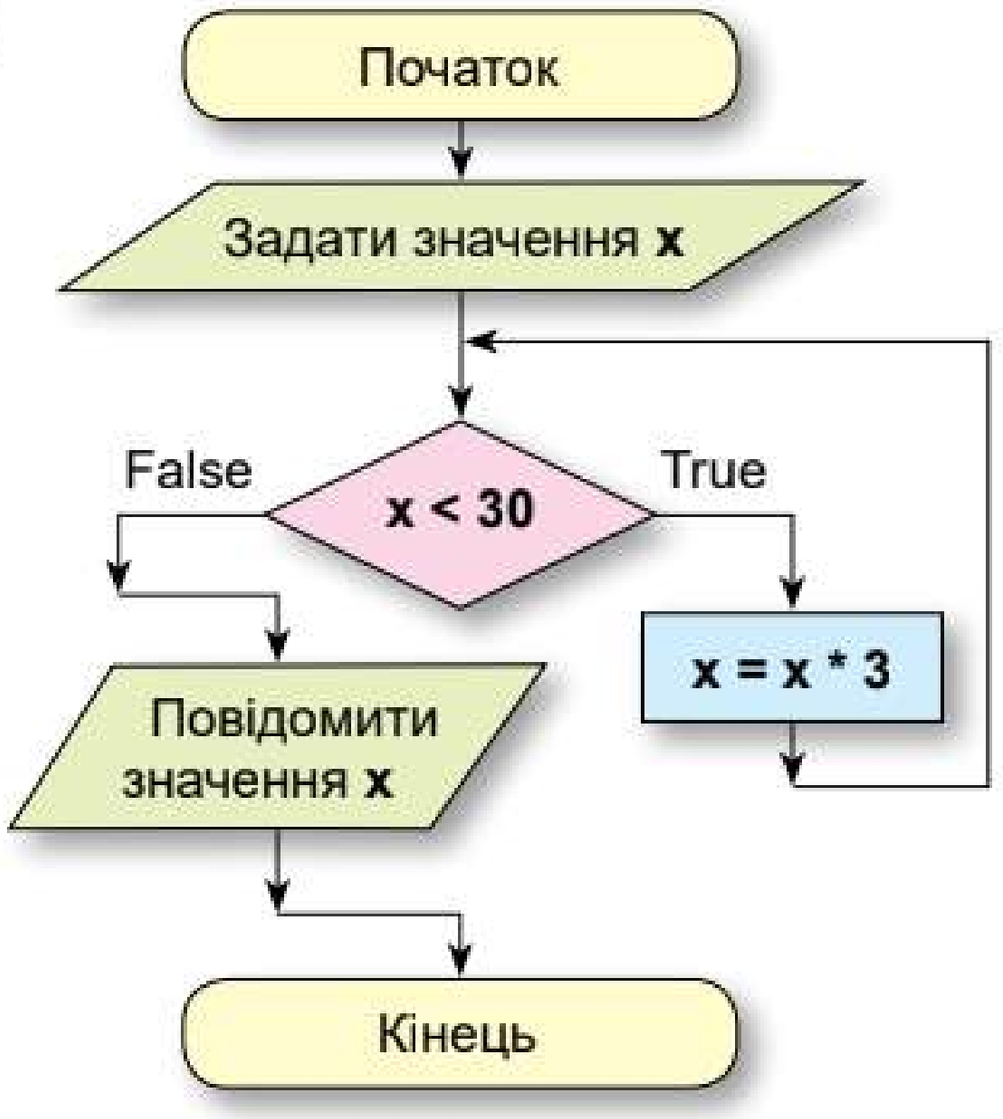
3. Чи можна команду циклу з передумовою замінити командою циклу з лічильником? Якщо так, наведіть приклад.

4. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, циклу з лічильником і циклу з передумовою?

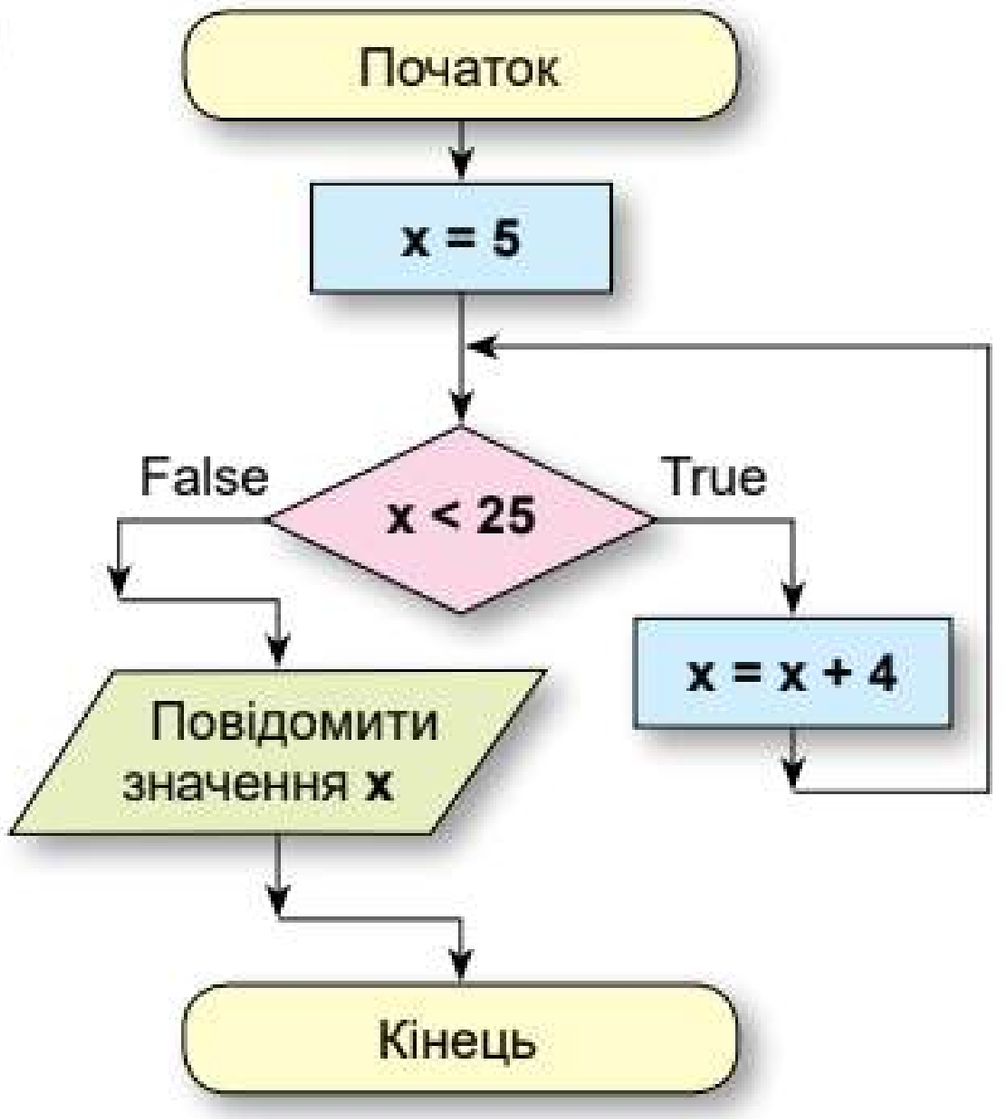
Виконайте завдання

1. Виконайте в зошиті алгоритми:

а)



б)



2. Виконайте в зошиті фрагменти проєктів:

а)
```
k = 1
a = 12
while a < 100:
    a = 2 * a - 4
    k += 1
```

б)
```
s = 0
a = 1
k = 1
while a < 50:
    s = s + a
    k += 1
    a = k * k
```

3. Виконайте в зошиті фрагменти проєктів:

а)
```
k = 1
a = 100
while a > 10:
    a = a / 2
    k += 1
```

б)
```
s = 0
a = 5
while a <= 100:
    s = s + a
    a = a * 2
```

4. Складіть текстові задачі за алгоритмами їх розв'язування із завдання № 1.

5. Складіть текстові задачі за алгоритмами їх розв'язування із завдання № 2.
6. Створіть математичну модель і проєкт для визначення, за скільки годин робітник виконає план у ***x*** деталей, якщо за першу годину роботи він виготовив ***y*** деталей, а за кожну наступну годину виготовляє на 1 деталь більше, ніж за попередню. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.6.**
7. Створіть математичну модель і проєкт для визначення, за скільки годин туристка подолає відстань ***x*** км, якщо за першу годину вона пройшла 6 км, а за кожну наступну годину проходить на 1 км менше, ніж за попередню. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.7.**
8. Створіть проєкт для обчислення середнього арифметичного додатних чисел, перше з яких дорівнює 100, а кожне наступне менше від попереднього на задане додатне число. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.8.**
9. Створіть математичну модель і проєкт для визначення, на скільки років вкладник має покласти ***x*** грн до банку, щоб отримати від банку більше за ***a*** грн прибутку, якщо банк щорічно збільшує вклад на ***p*** %. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.9.**
10. Створіть проєкт для знаходження найбільшого спільного дільника (НДС) двох натуральних чисел за алгоритмом Евкліда: поки числа різні, більше з них замінювати різницею більшого та меншого; коли числа стануть рівними, то кожне з них дорівнюватиме їх НСД. Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.10.**
11. Спортсменка, готуючись до участі в марафоні, у перший день тренувань подолала дистанцію в ***s*** км. Кожного наступного дня її денна дистанція збільшувалася на ***p*** % від дистанції попереднього дня. Створіть математичну модель і проєкт для визначення:
    1) у який день тренувань спортсменка подолає дистанцію в ***s1*** км;
    2) у який день тренувань дистанція збільшиться на ***s2*** км порівняно з попереднім днем.

    Збережіть проєкт у вашій папці у файлі з іменем **завдання 4.9.11.**



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу ***«Алгоритми та програми»*** *(початковий, середній, достатній, високий рівень).*

- Я знаю етапи розв'язування задач з використанням алгоритмів і проєктів.
- Я вмію виконувати кожний етап розв'язування задач з використанням алгоритмів і проєктів.
- Я знаю властивості кнопки, напису, поля та яких значень вони можуть набувати.
- Я вмію створювати віконні проєкти з використанням кнопок, написів, полів та їх властивостей.
- Я вмію створювати віконні проєкти з обробниками подій для кнопок, написів, полів.
- Я знаю, які величини називаються логічними.
- Я знаю, що є результатами операцій: кон'юнкції, диз'юнкції, заперечення над значеннями логічних величин.

- Я вмію складати таблицю істинності для логічних виразів.
- Я вмію створювати віконні проєкти з використанням логічних величин і операцій над ними.
- Я вмію створювати віконні проєкти з використанням прапорців і перемикачів.
- Я знаю, що таке цикл з лічильником і випадки, у яких його можна застосовувати у проєктах.
- Я вмію створювати віконні проєкти з використанням циклу з лічильником.
- Я знаю, що таке цикл з передумовою і випадки, у яких його можна застосовувати у проєктах.
- Я вмію створювати віконні проєкти з використанням циклу з передумовою.
- Я вмію добирати тестові набори значень вхідних даних для тестування проєктів.

Повторіть той матеріал, який ви знаєте недостатньо.



Практична робота № 6. «Проєкти із циклами»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Створіть математичну модель і проєкт для розв'язування задачі: *Перед початком повені рівень води в річці становив H см. Під час повені кожну годину рівень води зростав на P см. Яким буде рівень води через N год після початку повені? Через скільки годин після початку повені рівень води буде не менше ніж K см?*
2. Використайте у проєкті поля для введення вхідних даних, поле для виведення відповіді на перше запитання задачі та дві кнопки. Біля кожного поля розмістіть написи для пояснювальних коментарів.
3. Складіть обробник події **Click** для першої кнопки, виконання якого приведе до виведення в поле відповіді на перше запитання задачі.
4. Складіть обробник події **Click** для другої кнопки, виконання якого приведе до знаходження відповіді на друге запитання задачі та виведення його у вікно повідомлень.
5. Збережіть проєкт у вашій папці у файлі з іменем **практична 6**.



Готуємось до вивчення нового матеріалу

1. Що таке *дизайн*? Які види дизайну вам відомі?
2. Які ви знаєте вимоги до дизайну слайдів комп'ютерної презентації?

РОЗДІЛ 5

СТВОРЕННЯ ТА ПУБЛІКАЦІЯ ВЕБРЕСУРСІВ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- етапи створення сайтів;
- тренди вебдизайну;
- використання гіпертекстових, графічних і мультимедійних елементів на вебсторінках;
- мова гіпертекстової розмітки;
- автоматизовані засоби створення та публікації вебресурсів.

5.1. ЕТАПИ СТВОРЕННЯ САЙТУ. ДИЗАЙН ВЕБСТОРІНОК. ТРЕНДИ ВЕБДИЗАЙНУ

У цьому пункті йтиметься про:

- етапи створення сайтів;
- дизайн вебсторінок;
- рекомендації до оформлення та структури вебсторінок;
- тренди вебдизайну;
- оцінювання вебсайтів.

ЕТАПИ СТВОРЕННЯ САЙТІВ

Пригадайте

- Що таке *вебсторінка*? Що таке *вебсайт*?
- Які об'єкти можуть бути розміщені на вебсторінках?
- Що потрібно знати, щоб відкрити вебсторінку?

Ви вже знаєте, що таке вебсайти, які об'єкти можуть розміщуватися на вебсторінках, з яких складаються сайти, як здійснювати пошук і перегляд вебресурсів інтернету. Тепер ви навчитеся їх створювати та розміщувати в інтернеті.

Створення вебсайту складається з кількох етапів.

1. **Постановка завдання.** Визначається мета створення сайту, його основна тематика, цільова аудиторія – користувачі, які можуть зацікавитись умістом сайту, аналізуються існуючі сайти такої самої або схожої тематики.
2. **Визначення структури сайту.** Складається перелік тематичних розділів сайту та сторінок у них, визначаються зв'язки між сторінками, утворюється **мапа (карта) сайту**, яка пояснює його логічну структуру, формулюються вимоги до функціонування інтерактивних елементів.

Наприклад, на малюнку 5.1 наведено частину мапи сайту, присвяченого Україні.



Мал. 5.1. Мапа сайту «Україна»

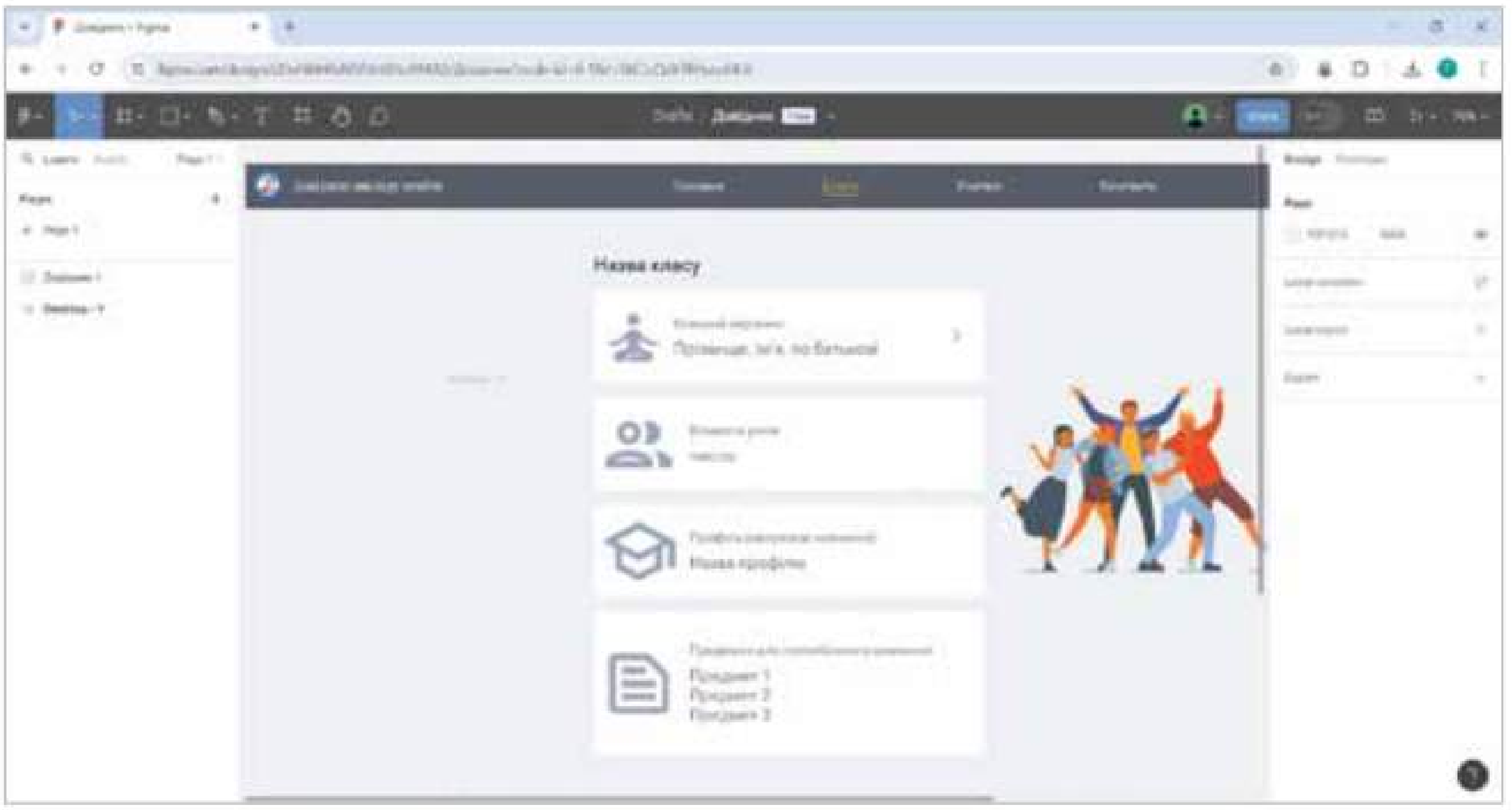
3. **Створення дизайн-макета сторінок сайту.** На цьому етапі обирають, які об'єкти будуть міститися на сторінках, розмір і розташування цих об'єктів, кольорову гаму, набір шрифтів та інше, тобто визначають **стиль сайту.**

У дизайні враховують три складові:

- **ергономіку** – зручність і зрозумілість користування сайтом;
- **композицію** – узгодженість розмірів об'єктів і їх розташування;
- **колористику** – гармонійність поєднання кольорів.

Дизайн-макет створюють, як правило, у спеціалізованому графічному редакторі, наприклад **Adobe XD**, **Sketch**, **Figma** або іншому.

На малюнку 5.2 наведено дизайн-макет вебсторінки шкільного довідника, який створено у спеціалізованому графічному онлайн-редакторі **Figma.**



Мал. 5.2. Дизайн-макет сторінки сайту в редакторі **Figma**

4. **Створення сторінок сайту.** На цьому етапі вибирають програмні засоби, які будуть використані для створення сторінок сайту. Під час створення вебсторінок на них розміщують і форматують

контент (англ. *content* – вміст) – текстові, графічні, мультимедійні та інтерактивні елементи, а також **елементи навігації** – кнопки, гіперпосилання та інше – таким чином, щоб сторінка отримала вигляд відповідно до розробленого дизайн-макета; забезпечують функціонування інтерактивних елементів – систем пошуку, форм опитування тощо.

На малюнку 5.3 наведено вигляд сторінки, створеної відповідно до наведеного дизайн-макета.



Мал. 5.3. Вебсторінка, створена відповідно до дизайн-макета

5. **Публікація сайту в інтернеті.** Призначення сторінкам сайту **URL-адрес** і надання дозволу користувачам інтернету на перегляд вебсторінок відбувається після **публікації** сайту. Перед публікацією сторінки сайту потрібно розмістити на спеціальному сервері, який надає власні апаратні та програмні ресурси для зберігання файлів користувача, забезпечення доступу до них, опрацювання запитів тощо. Такі послуги називають **хостингом** (англ. *hosting* – виявлення гостинності).

ОФОРМЛЕННЯ ТА СТРУКТУРА ВЕБСТОРІНОК

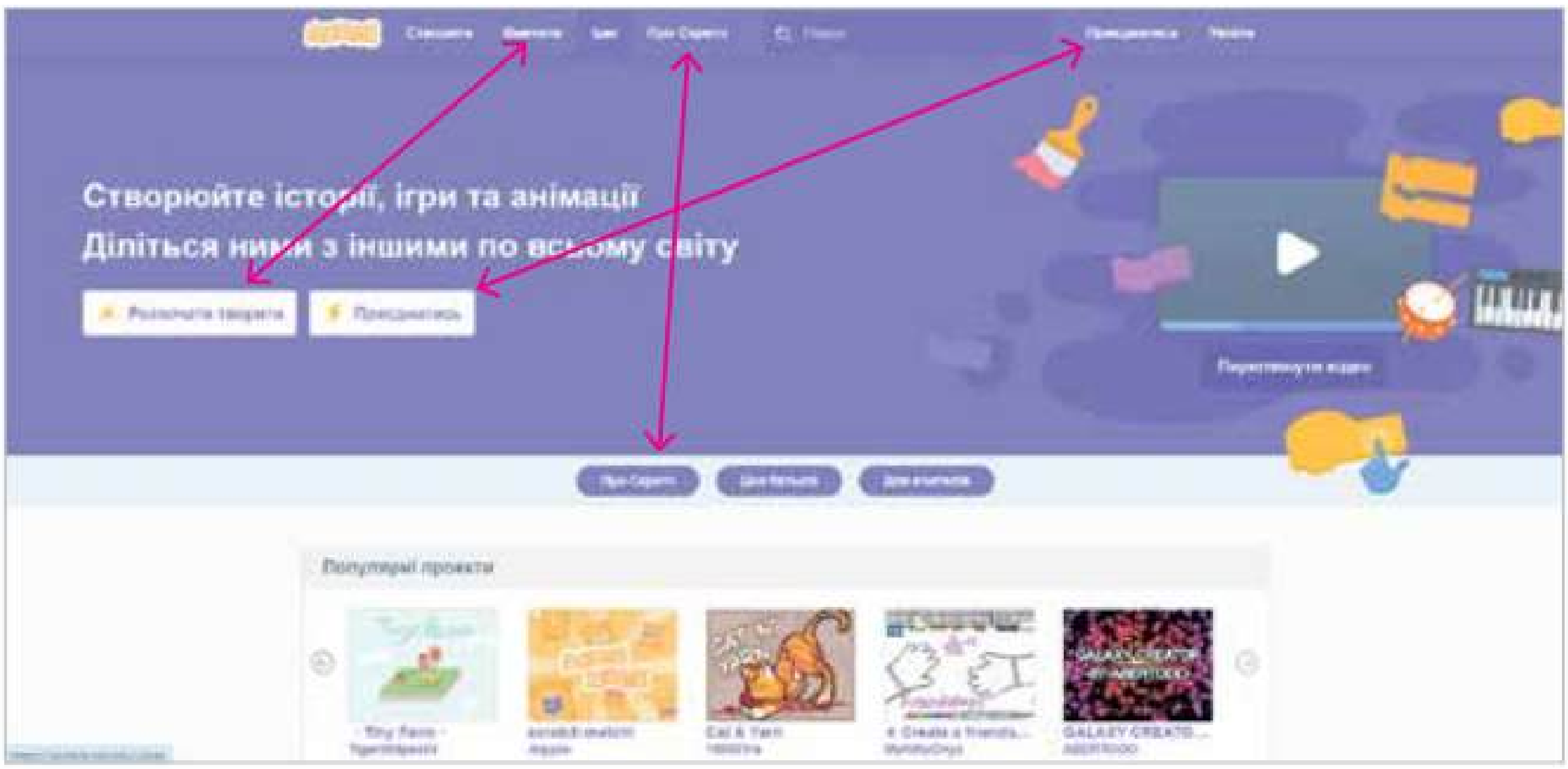
Поміркуйте

● Які елементи на вебсторінках слугують орієнтиром для користувача, щоб зрозуміти, який сайт і який його розділ переглядається? Чому це зручно? ● Які засоби використовують для перегляду сторінок, на яких контент не вміщується по висоті на одному екрані? ● Чи доцільно мати кілька різних елементів на вебсторінці, щоб перейти до перегляду іншої сторінки?

Створюючи дизайн-макет для власного сайту, варто мати на увазі рекомендації з оформлення та структури вебсторінок, пов'язані з **ергономікою, композицією** та **колористикою.**

Вдалий дизайн сайту передбачає зручність користування ним. Будь-яка людина повинна інтуїтивно пов'язувати дії, які їй потрібно виконати на вебсторінці, з об'єктами, які вона бачить. Тому під час розробки дизайну вебсторінок слід ураховувати такі правила *ергономіки*:

- розміщуйте назву сайту в заголовку всіх сторінок;
- зазначайте на кожній сторінці, до якого тематичного розділу вона входить;
- вибирайте розмір шрифту, за якого текст буде сприйматися комфортно;
- створіть можливість переходу з будь-якої сторінки сайту до перегляду головної сторінки;
- створюйте внутрішні гіперпосилання на сторінках з великим обсягом контенту для переходів усередині вебсторінки;
- не допускайте появи горизонтальних смуг прокручування на вебсторінках;
- пропонуйте кілька посилань для переходу на одну й ту саму сторінку, щоб відвідувачі сайту могли вибрати найзручніший для себе спосіб (мал. 5.4).



Мал. 5.4. Приклад сторінки з посиланнями, що дублюються

При розробці дизайну вебсторінок потрібно намагатися надавати додаткові зручності для користувачів з особливими потребами: можливість збільшувати шрифт тексту, розміри зображень, озвучувати призначення окремих елементів.

Будь-який відвідувач сайту завдяки дизайну повинен швидко зрозуміти тематику сайту та легко знайти корисний для себе контент. Тому слід дбати про доцільне розміщення об'єктів на вебсторінках, а для цього бажано дотримуватися таких правил *композиції*:

- великий текст поділяйте на частини, визначайте для них заголовки;
- шрифт у заголовках робіть крупнішим від шрифту основного тексту;
- розміщуйте найважливіші дані на початку сторінки;
- доповнюйте текстові відомості зображеннями, які ілюструють текст;

- не зловживайте анімацією, відео, музичними та графічними елементами – вони можуть відволікати увагу від корисних матеріалів та уповільнюватимуть завантаження сторінок;
- не копіюйте матеріали з інших інтернет-ресурсів, а пропонуйте гіперпосилання на них для уточнення відомостей – так ви забезпечите дотримання закону про захист авторських прав.

Обираючи *кольорову гаму* для вашого сайту:

- враховуйте ваші вподобання, тематику сайту та вподобання потенційних відвідувачів, що складають цільову аудиторію;
- вибирайте контрастні кольори для тексту та фону сторінки, щоб текст легше сприймався.

ТРЕНДИ ВЕБДИЗАЙНУ



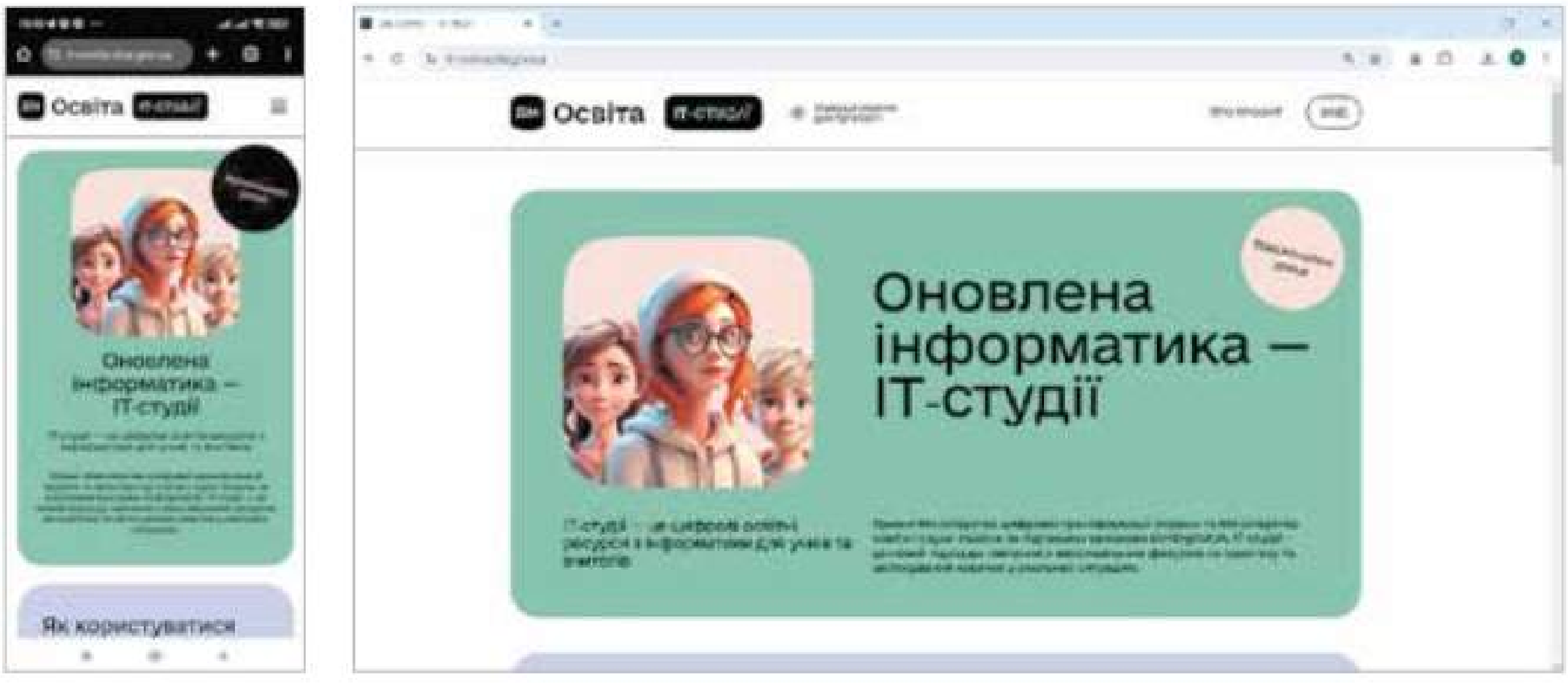
Поміркуйте

● Чи завжди зручно, щоб вебсторінка виглядала однаково під час її перегляду на екрані смартфона та стаціонарного комп'ютера? ● Сайти якого змісту та з яким оформленням вас найбільше приваблюють?

Під час вибору дизайну для власного вебсайту потрібно враховувати також сучасні **тренди вебдизайну.**

Тренд (англ. *trend* – тенденція) – основна тенденція змінення чого-небудь, загальний напрямок змін упродовж тривалого часу.
Адаптація (лат. *adaptō* – пристосовувати) – процес пристосування до мінливих умов зовнішнього середовища.

Змінення трендів у вебдизайні пов'язано з розвитком цифрових технологій. Так, за даними статистики в наш час більше половини користувачів звертаються до ресурсів інтернету з мобільних пристроїв. Тому в останні роки особливого значення набув так званий **адаптивний дизайн**, при забезпеченні якого одна й та сама сторінка, можливо, по-різному, але в одному стилі, коректно відображається і на великих, і на маленьких екранах (мал. 5.5).



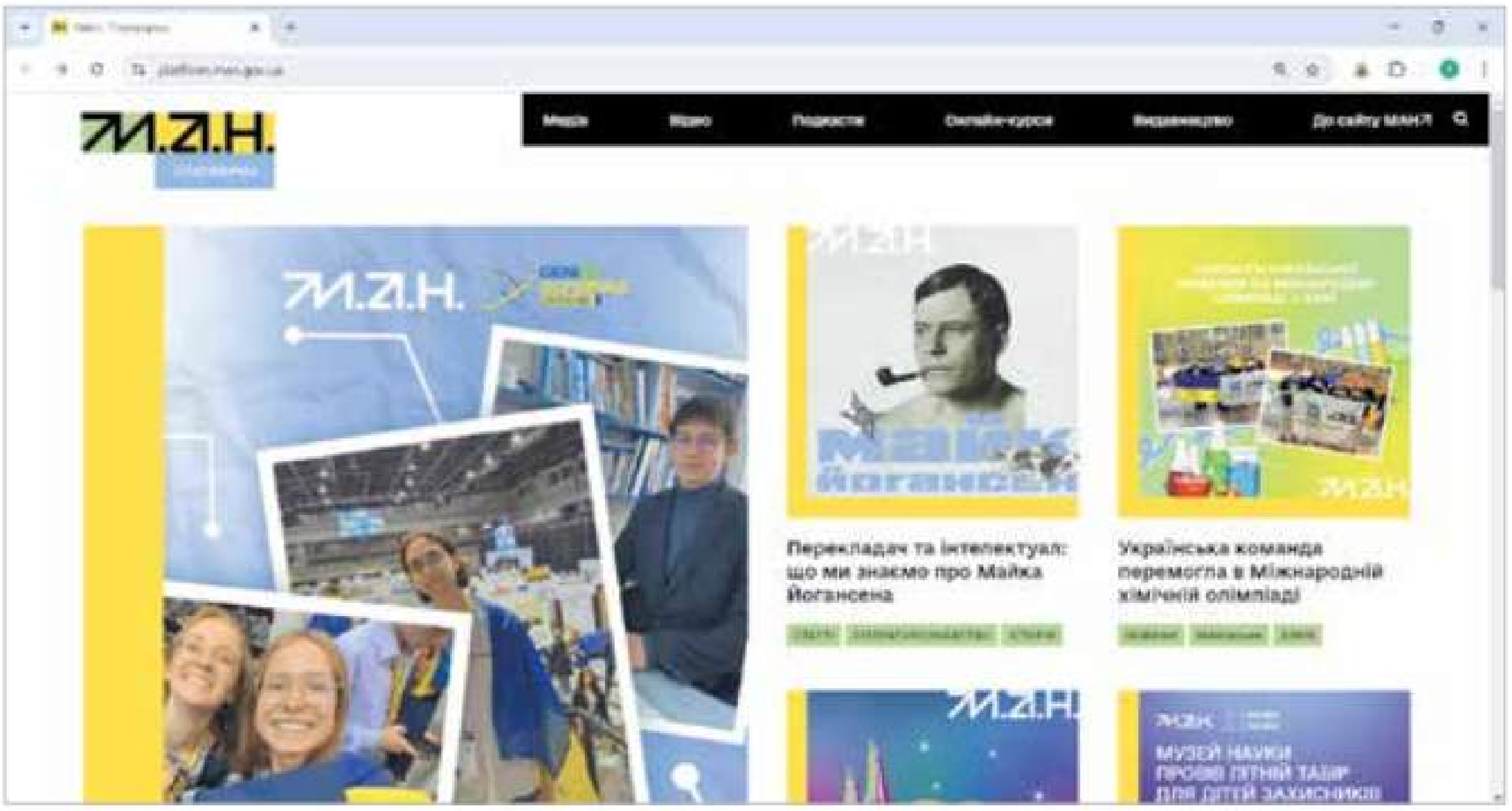
Мал. 5.5. Вигляд сторінки з адаптивним дизайном під час перегляду на смартфоні (ліворуч) і ноутбуці (праворуч)

Ще одним трендом, пов'язаним з особливостями мобільних пристроїв, стало використання **темного режиму**. Чорний колір на екранах деяких видів мобільних пристроїв не потребує затрат енергії, а тому в темному режимі збільшується час використання батареї пристрою. Крім того, темний фон сторінок зменшує напругу очей, оскільки забезпечує контраст між фоном і вмістом сторінки.

Стрімке поширення систем **штучного інтелекту** викликало появу ще одного тренду – використання у веброзробці механізму голосового пошуку та елементів штучного інтелекту.

Також у дизайні вебсторінок упродовж останніх кількох років спостерігаються такі тренди:

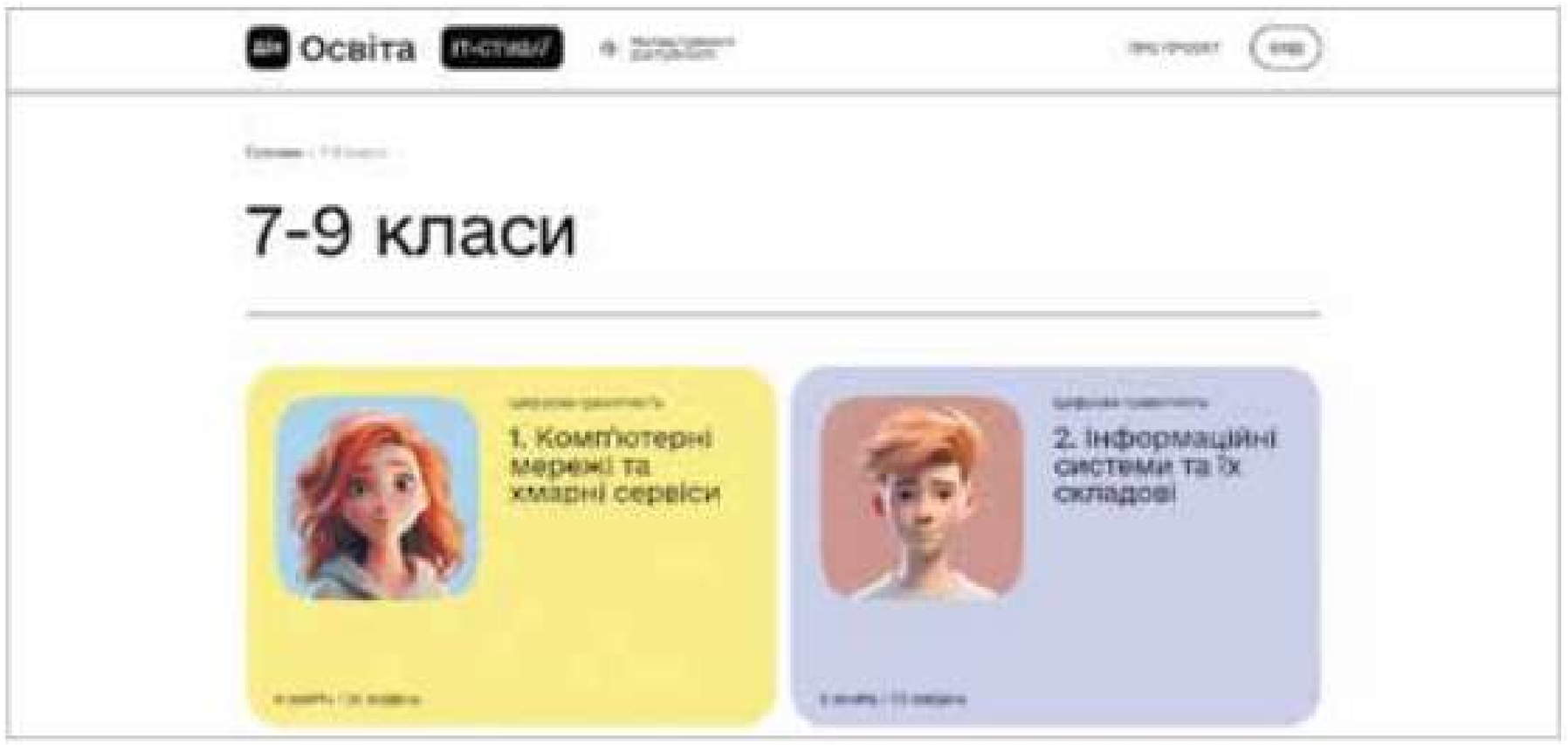
- **типографіка** – застосування нестандартних оригінальних шрифтів (мал. 5.6), великих заголовків, креативного розміщення тексту;



Мал. 5.6. Використання фірмового шрифту на сайті МАН

- **нестандартні колірні рішення** – використання яскравих кольорів і багатоколірної градієнтної заливки фону сторінки або її окремих елементів;
- **анімація та інтерактивність** – використання анімаційних ефектів для привертання уваги до важливих елементів сторінки, застосування плавних переходів під час прокручування вебсторінки, реалізація інтерактивної анімації та вставлення 3D-об'єктів;
- **мінімалізм** – обмеження кількості елементів на екрані, проста композиція, наявність великого порожнього простору. Популярним є структурування контенту на вебсторінках з використанням так званих **карток** – прямокутних блоків, у яких об'єднані взаємопов'язані графічні, текстові та навігаційні елементи (мал. 5.7).

Сайт, який відповідає трендам вебдизайну, виглядає стильно та сучасно, привертає увагу відвідувачів, є зручним для його перегляду на різних пристроях.



Мал. 5.7. Використання карток на вебсторінці

ОЦІНЮВАННЯ ВЕБСАЙТІВ



Поміркуйте

- Від чого залежить тривалість перегляду вами тієї чи іншої вебсторінки?
- Чи однаково сприймаються вами всі сайти однакової тематики? Від чого залежить ваше враження від сайту?

Здійснюючи навігацію вебсторінками, ви, напевно, звертали увагу на те, як по-різному сприймаються сайти однакової тематики та спрямованості. На одних сайтах ви затримувалися надовго, а з деяких – одразу переходили на інший ресурс.

Для відвідувача під час оцінювання вебсайту важливими показниками є:

- ступінь інформативності вебсайту, можливість отримання корисних матеріалів, які не повторюють відомості з інших ресурсів;
- візуальна привабливість сайту, дизайн, стиль оформлення;
- зрозумілість і зручність навігації, легкість знаходження потрібних даних, структурованість матеріалу, все те, що забезпечує простоту, комфортність та ефективність у процесі використання сайту;
- наявність інтерактивних засобів та ефектів;
- частота оновлення даних на сайті, що свідчить про увагу розробника до власного проєкту;
- мови, якими доступні матеріали на вебсайті;
- можливість залишити коментарі, отримати відповіді, звернутися до розробника за додатковими відомостями

та інше.

Тривалість перегляду відвідувачами сторінок на сайті, повторне звертання до його ресурсів визначають популярність сайту та його рейтинг у базах даних пошукових систем.

Набір стратегій та інструментів для просування сайту в інтернеті, щоб він відображався на верхніх місцях у видачах пошукових систем, називають **SEO-оптимізацією** сайтів (англ. ***Search Engine Optimization*** – пошукова оптимізація). Урахування трендів і рекомендацій з вебдизайну є однією зі стратегій пошукової оптимізації.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/de2cHoMb* або QR-кодом.

Найважливіше в цьому пункті

Створення вебсайту складається з кількох етапів: постановка завдання, визначення структури сайту, створення дизайн-макета сторінок сайту, створення сторінок, публікація сайту в інтернеті.

Створюючи дизайн-макет для власного сайту, варто мати на увазі рекомендації з оформлення та структури вебсторінок, пов'язані з ергономікою, композицією та колористикою. Під час розробки дизайну вебсторінок потрібно намагатися надавати доткові зручності для користувачів з особливими потребами.

У процесі вибору дизайну для власного вебсайту потрібно враховувати також сучасні **тренди вебдизайну**. За **адаптивного дизайну** одна й та сама сторінка, можливо, по-різному, але в одному стилі, коректно відображається і на великих, і на маленьких екранах. Сайт, який відповідає трендам вебдизайну, виглядає стильно та сучасно, привертає увагу відвідувачів, є зручним для його перегляду на різних пристроях.

Дайте відповіді на запитання

1. За якими етапами відбувається створення сайтів?
2. Що таке *дизайн-макет* сторінок сайту?
3. У чому полягає створення вебсторінок?
4. Які ви знаєте сучасні тренди вебдизайну?
5. Які рекомендації щодо оформлення та структури сторінок варто враховувати під час створення сайту?

Обговоріть і зробіть висновки

1. Які зручності для користувачів з різними видами особливих потреб можна створити на сайті?
2. Чи всі тренди вебдизайну повинні бути враховані на кожній вебсторінці, щоб вона виглядала сучасно? Відповідь поясніть.
3. Чим може відрізнятися дизайн вебсайтів однієї тематики залежно від цільової аудиторії – від віку, статі, інтересів користувачів?
4. У чому схожість і відмінність вимог до оформлення вебсторінок і слайдів комп'ютерних презентацій?

Виконайте завдання

1. Перегляньте сайт **Тут цікаво** (*tut-cikavo.com*). Оцініть дотримання рекомендацій щодо оформлення та розміщення об'єктів на сторінці та відповідність сучасним трендам. Проаналізуйте, як видозмінюються сторінки під час змінення масштабу та ширини області перегляду в браузері. Створіть текстовий документ, у який запишіть свої висновки. Збережіть документ у вашій папці у файлі з іменем **завдання 5.1.1**.

2. Порівняйте особливості оформлення та зручність користування сайтами **Український Ерудит** (*ukrerudyt.com*) і **Tezzt** (*tezzt.com*). Створіть текстовий документ, у якому дайте відповіді на запитання:

а) Яке основне призначення кожного із цих сайтів?
б) Наскільки зручно шукати потрібні відомості (наприклад, вікторини з історії України)?
в) Чи реалізований на сайтах адаптивний дизайн? У чому це виявляється?

г) Наскільки сучасними виглядають сайти? Які сучасні тренди вебдизайну прослідковуються на сторінках?

Зробіть і запишіть у текстовому документі висновок, який сайт здається вам більш привабливим і чому. Збережіть документ у вашій папці у файлі з іменем **завдання 5.1.2**.

3. Об'єднайтесь у пари та розробіть у графічному редакторі дизайн-макети двох варіантів головної сторінки сайту вашого закладу освіти – для відтворення на екрані смартфона та на екрані ноутбука. Враховуйте сучасні тренди вебдизайну. Збережіть дизайн-макети у вашій папці у файлах з іменами **завдання 5.1.3.1** та **завдання 5.1.3.2**. Презентуйте створені дизайн-макети однокласникам / однокласницям.

4. Виконайте пошук в інтернеті матеріалів про тренди вебдизайну поточного року та трирічної давнини. Проаналізуйте, які тренди змінились, а які залишились актуальними. Збережіть опис трендів і результати порівняння у вашій папці у файлі з іменем **завдання 5.1.4**. Презентуйте результати вашого дослідження однокласникам / однокласницям.

5. Перегляньте на сайті **Rentafont** (*rentafont.com.ua*) колекції шрифтів, які можуть бути використані на вебсайтах. Створіть у графічному редакторі власні зображення кількох українських літер, які можна було б використати для оформлення заголовків на сайті. Збережіть створене зображення у вашій папці у файлі з іменем **завдання 5.1.5**.

6. Знайдіть в інтернеті відомості про адаптивні логотипи сайтів і зображення логотипів відомих брендів, які використовуються в адаптивному дизайні. Створіть у графічному редакторі власні зображення адаптивного логотипу для вашого персонального сайту. Збережіть створене зображення у вашій папці у файлі з іменем **завдання 5.1.6**.

Готуємось до вивчення нового матеріалу

1. Для чого створюється розмітка на проїжджій частині дороги та які особливості цієї розмітки? У яких ще випадках використовують розмітку та з якою метою?
2. Як ви структуруєте та розміщуєте текст, коли пишете твір у зошиті або вводите в текстовому процесорі?
3. Як у мові програмування, яку ви знаєте, забезпечити виведення тексту в різних рядках?

5.2. ПОНЯТТЯ ПРО МОВУ HTML

У цьому пункті йтиметься про:

- мову **HTML**;
- поняття тегу, атрибута тегу, значення атрибута;
- базову структуру вебсторінки;
- створення та збереження вебсторінок.

МОВА HTML. ТЕГИ, АТРИБУТИ ТЕГІВ, ЗНАЧЕННЯ АТРИБУТІВ

Пригадайте

● Які об'єкти можуть бути розміщені на вебсторінках? Які ви знаєте властивості цих об'єктів? ● Як здійснюється форматування об'єктів у текстовому документі? ● Що розуміють під словом *розмітка*? Де та для чого використовується це поняття?

HTML (англ. ***Hyper Text Markup Language*** – мова розмітки гіпертексту) – це мова, якою описують структуру та контент вебсторінки для відображення цієї сторінки у вікні браузера.

Кожна вебсторінка має свій **HTML-код**. Коли ми відкриваємо для перегляду деяку вебсторінку з інтернету, браузер отримує з інтернет-сервера її HTML-код та супутні файли – графічні, відео та інші. На основі HTML-коду браузер розміщує у вікні об'єкти вебсторінки: заголовки, абзаци тексту, гіперпосилання, списки, таблиці, зображення, відео тощо.

Елементами мови **HTML** є **теги** (англ. *tag* – ярлик, ознака), що вказують браузеру, які об'єкти розмістити на сторінці та як їх відобразити. Імена тегів записують усередині кутових дужок < та >:

<ім'я_тегу>

Наприклад, тег **<input>** указує браузеру розмістити на сторінці елемент керування *поле*, а тег **<hr>** (англ. ***horizontal row*** – горизонтальний ряд) – *горизонтальну лінію* (мал. 5.8).



Мал. 5.8. HTML-теги (ліворуч) і відповідні їм об'єкти сторінки (праворуч)

Наведені теги **<input>** та **<hr>** є **непарними** тегами. Найчастіше непарні теги вказують браузеру вставити певний об'єкт на сторінку.

Існують також **парні** теги, що складаються з **відкриваючої** та **закриваючої** частин, які обмежують фрагмент тексту або HTML-коду. У закриваючій частині перед іменем тегу ставлять символ /. Загальний вигляд парного тегу такий:

<ім'я_тегу> фрагмент тексту або HTML-коду </ім'я_тегу>

Тут **<ім'я_тегу>** – відкриваюча частина тегу, а **</ім'я_тегу>** – відповідна йому закриваюча частина.

Парні теги вказують браузеру, як відобразити фрагмент, обмежений частинами парного тегу.

Наприклад:

<h1>Моя Україна**</h1>**

Парний тег з іменем **h1** (англ. ***header*** – заголовок) указує відобразити обмежений ним фрагмент тексту *Моя Україна* як заголовок першого рівня. Теги з іменами від **h1** до **h6** визначають 6 можливих рівнів заголовків на сторінці.

** ** Столиця – місто ****Київ****. ** **

Тут парний тег з іменем **p** (англ. *paragraph* – пункт) указує відобразити обмежений ним фрагмент HTML-коду *Столиця – місто* *******Київ*******, як абзац, а парний тег з іменем **strong** (англ. *strong* – сильний) указує відобразити слово *Київ* напівжирним накресленням (мал. 5.9).

```
<h1>Моя Україна</h1>

<p> Столиця - місто <strong>Київ</strong>. </p>
```

Моя Україна

Столиця - місто **Київ**.

Мал. 5.9. Використання парних тегів

Теги можуть містити **атрибути** (англ. *attribute* – характерна риса, ознака), які уточнюють дію цих тегів. У атрибутів можуть бути **значення**. Атрибути записують після імені тегу, відокремлюючи пропуском. Значення атрибутів записують у лапках після знака =. Загальний вигляд парного тегу з одним атрибутом такий:

<ім'я_тегу атрибут="значення"> текст або HTML-код </ім'я_тегу>

Наприклад:

<img src="https://yvu.com.ua/wp-content/uploads/2020/08/images.jpg">

Непарний тег з іменем **img** (англ. ***image*** – зображення) указує вставити на сторінку зображення, а значенням атрибуту **src** (англ. ***source*** – джерело) є URL-адреса файлу *https://yvu.com.ua/wp-content/uploads/2020/08/images.jpg*, з якого буде взято це зображення;

** Юридичний Вісник України ****

Парний тег з іменем **a** (англ. *anchor* – якір) указує створити гіперпосилання для переходу на сторінку з URL-адресою *https://yvu.com.ua/*, яка є значенням атрибута **href** (англ. ***hypertext reference*** – гіпертекстова виноска). Для переходу на цю сторінку потрібно вибрати текст *Юридичний Вісник України*, обмежений частинами парного тегу з іменем **a**.

<h2 style="color: blue;">Малий та великий герби України**</h2>**

Парний тег з іменем **h2** вказує вивести текст *Малий та великий герби України* як заголовок другого рівня. Атрибут **style** (англ. *style* – стиль) визначає стиль заголовка: колір символів тексту *синій*, відповідно до правила *color: blue* (англ. *color* – колір, *blue* – синій), яке є значенням цього атрибута.

Тег може мати кілька атрибутів, які відокремлюються один від одного пропусками. Наприклад, тег **<input type="**button" **value="**Ok"> указує розмістити на сторінці елемент керування типу *кнопка* з написом Ok (мал. 5.10).

```
<img src="https://yvu.com.ua/wp-content/uploads/2020/08/images.jpg">

<p>
З сайту <a href="https://yvu.com.ua/">Юридичний Вісник України</a>
</p>

<h4 style="color: blue;">Малий та великий герби України</h4>

<input type="button" value="Ok">
```



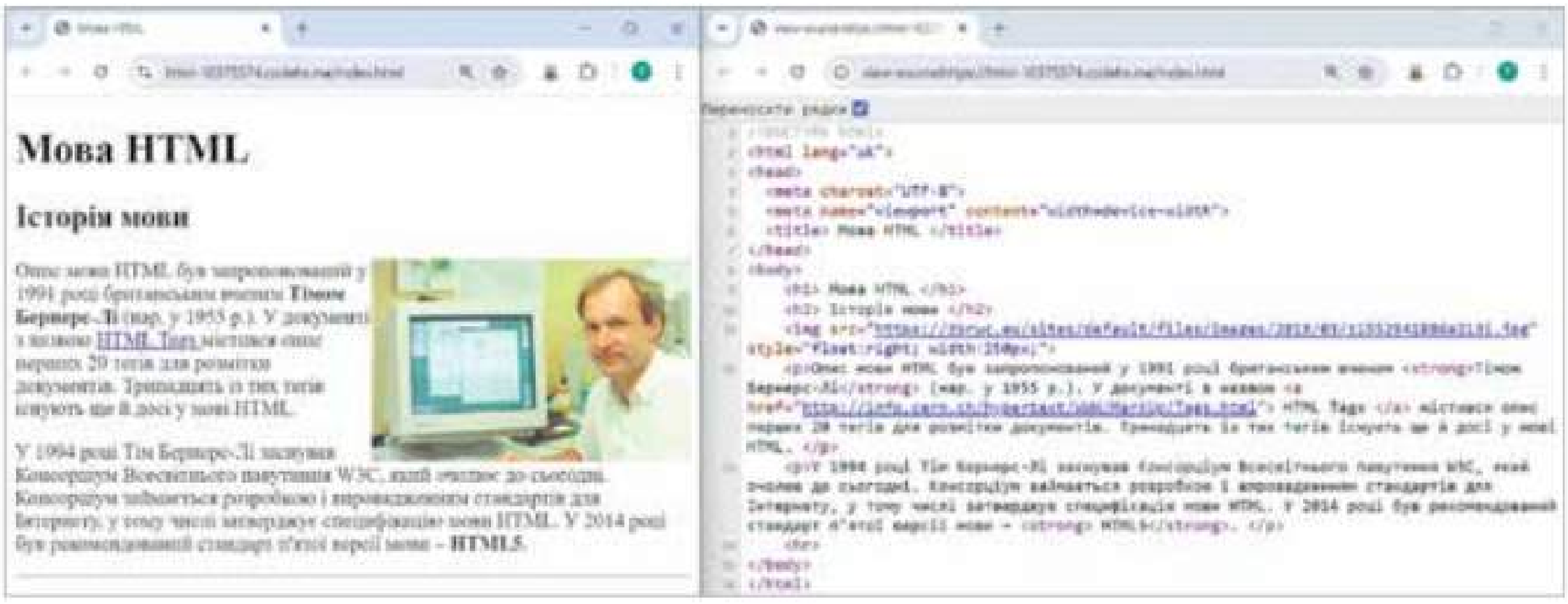
З сайту Юридичний Вісник України

Малий та великий герби України

Ok

Мал. 5.10. Використання тегів з атрибутами

Переглянути HTML-код будь-якої вебсторінки, відкритої у вікні браузера, можна вибравши в її контекстному меню команду **Переглянути джерело сторінки** або натиснувши сполучення клавіш **Ctrl + U** (мал. 5.11).



Мал. 5.11. Вебсторінка (ліворуч) і її HTML-код (праворуч)

Під час перегляду в браузері HTML-коду сторінки теги, їх атрибути та значення атрибутів виділено різними кольорами, а колір символів тексту чорний. Розглянемо детальніше HTML-код сторінки, наведеної на малюнку 5.11 (табл. 5.1).

Таблиця 5.1

Пояснення тегів HTML-коду вебсторінки (мал. 5.11)

Номери рядків у HTML-коді	*Теги*	*Пояснення*
1	Непарний тег **<!DOCTYPE html>** (англ. ***document type*** – тип документа)	Указує тип даного документа та версію мови **HTML**, яку використано. Атрибут **html** означає, що сторінка створена з використанням актуальної наразі версії мови **HTML5**
2, 16	Парний тег **<html lang**="uk"**>** **</html>** (англ. ***language*** – мова)	Указує на початок і закінчення HTML-коду сторінки. Атрибут **lang** зі значенням **uk** повідомляє, що основна мова в тексті сторінки – українська
3, 7	Парний тег **<head>** **</head>** (англ. *head* – голова)	Указує на початок і закінчення **службової** частини коду сторінки з відомостями, потрібними для її правильного відображення браузером
4	Непарний тег **<meta charset**="UTF-8"**>** (англ. *char set* – набір символів)	Указує, яку саме таблицю кодів символів використано в ході створення сторінки. Значення **UTF-8** атрибута **charset** означає використання кодової таблиці **Юнікод**

Продовження таблиці 5.1

Номери рядків у HTML-коді	***Теги***	***Пояснення***
5	Непарний тег **<meta name**="viewport" **content**="width=device-width"**>** (англ. *viewport* – вікно перегляду, *content* – зміст)	Указує браузеру розміщувати контент сторінки з урахуванням ширини екрана пристрою, який використовується
6	Парний тег **<title>** Мова HTML **</title>** (англ. *title* – назва)	Указує вивести на вкладці браузера назву сторінки *Мова HTML*
8, 15	Парний тег **<body>** **</body>** (англ. *body* – тіло)	Указує на початок і закінчення **інформаційної** частини, яка містить текст для відображення на сторінці та розмітку
9	Парний тег ** ** Мова HTML ** **	Указує відобразити текст *Мова HTML* як заголовок першого рівня
10	Парний тег ** ** Історія мови ** **	Указує відобразити текст *Історія мови* як заголовок другого рівня
11	Непарний тег **<img src=**"https://zbruc.eu/sites/default/files/images/2019/03/z1552941886a213i.jpg" **style**="float: right; width: 250px;"**>**	Указує вставити на сторінку зображення з файлу з адресою *https://zbruc.eu/sites/default/files/images/2019/03/z1552941886a213i.jpg*. Стиль зображення (атрибут **style**) – вирівнювання по правому краю з обтіканням текстом (**float: right;**), ширина 250 пікселів (**width: 250px;**)
12, 13	Парний тег ** … **	Указує відобразити текст, що розміщений між частинами парного тегу, як окремий абзац
12, 13	Парний тег ** … **	Указує відобразити текст, що розміщений між частинами парного тегу, напівжирним накресленням
12	Парний тег **** HTML Tags ****	Указує створити гіперпосилання для переходу на сторінку з адресою *http://info.cern.ch/hypertext/WWW/MarkUp/Tags.html*. Для переходу потрібно вибрати текст *HTML Tags*
14	Непарний тег **<hr>**	Указує браузеру вставити на сторінку горизонтальну лінію

Теги, розміщені в рядках з 1 по 8 та в 15 і 16, визначають так звану базову структуру вебсторінки. Вони використовуються в HTML-коді кожної вебсторінки з потрібними значеннями атрибутів, щоб браузери правильно відтворювали сторінку за її HTML-кодом.

Цікаві факти з історії

Опис мови **HTML** запропонував у 1991 році британський учений **Тім Бернерс-Лі** (нар. у 1955 р.) (мал. 5.12). Під його керівництвом було розроблено **концепцію Всесвітнього павутиння** (спільно з **Робертом Кайо**), систему URL-адрес ресурсів у мережі, створено перший браузер, вебсервер, вебсайт. Завдяки цьому стало можливим існування служби передавання гіпертекстових сторінок **World Wide Web** (англ. *World Wide Web* – всесвітнє павутиння) – **Всесвітнього павутиння.**



Мал. 5.12. **Тім Бернерс-Лі**

У 1994 році Тім Бернерс-Лі заснував **Консорціум Всесвітнього павутиння** (англ. ***World Wide Web Consortium***, ***W3C*** – консорціум Всесвітнього павутиння) (*https://www.w3.org*). Консорціум займається розробкою та впровадженням стандартів для різних напрямів веброзробки, у тому числі для мови **HTML**.

У 2014 році було рекомендовано до впровадження актуальний на сьогодні стандарт п'ятої версії мови – **HTML5**.

СТВОРЕННЯ HTML-КОДУ

Пригадайте

- Як під час збереження текстового документа змінити розширення імені файлу?
- Як у текстовому редакторі під час створення проєктів відомою вам мовою програмування виділяються ключові слова, імена змінних, тексти?

Файли з HTML-кодом можна створювати як текстові файли та зберігати з розширенням імені **html.** Файлу з HTML-кодом головної сторінки сайту зазвичай дають ім'я **index.html.**

Для створення та редагування HTML-коду можна використовувати текстові редактори, наприклад **Блокнот, Notepad++, Edit Plus** або спеціалізовані редактори коду **Visual Studio Code, Sublime Text, Atom** та інші.

Також створення HTML-коду можливе на онлайн-ресурсах, наприклад в онлайн-редакторах **Replit** (*replit.com*), **CodeHS** (*codehs.com*), **JS Bin** (*jsbin.com*) та інших. Під час створення HTML-коду в онлайн-редакторі можна переглядати на тій самій вкладці браузера, як виглядатиме вебсторінка.

Створення HTML-коду розпочинається з уведення базової структури вебсторінки з потрібними значеннями атрибутів. Після цього в інформаційній частині коду, що обмежена частинами парного тегу з іменем **body**, вводять текст і теги для відображення на сторінці контенту відповідно до дизайн-макета. Об'єкти на сторінці будуть розташовані в порядку їх опису в HTML-коді.

У багатьох редакторах є засоби для спрощення процесу написання HTML-коду: підсвітка синтаксису, розпізнавання введених тегів і автоматичне їх доповнення, вставлення готових фрагментів HTML-коду тощо.

Виконання HTML-коду відбувається під час відкривання відповідних файлів у вікні браузера. Якщо файли з HTML-кодом розміщено на носіях даних вашого комп'ютера, то вони не будуть доступні для перегляду користувачам інтернету до розміщення цих файлів на сервері та їх публікації. В онлайн-редакторах зазвичай існують спеціальні засоби для публікації створених сторінок.

Для тих, хто хоче знати більше

Стандарт останньої версії мови **HTML5** містить понад 100 тегів. Крім того, під час написання HTML-коду можуть бути використані окремі теги з попередніх версій мови, хоча вони вважатимуться застарілими. Особливо це стосується тегів і атрибутів для форматування об'єктів сторінки: установлення кольору символів і фону, вирівнювання, розмірів об'єктів тощо.

Замість тегів мови **HTML** для форматування об'єктів на сторінці використовують **мову опису стилів CSS** (англ. ***C**ascading **S**tyle **S**heets* – каскадні таблиці стилів).

Ознайомитися з тегами мови **HTML** та особливостями мови **CSS** можна на сайті **W3SchoolsUA українською** (*https://w3schoolsua.github.io*).

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Be2cHPJ1* або QR-кодом.

Найважливіше в цьому пункті

HTML – це мова, якою описують структуру та контент вебсторінки для відображення цієї сторінки у вікні браузера.

Кожна вебсторінка має свій **HTML-код**. На основі HTML-коду браузер розміщує у вікні об'єкти вебсторінки: заголовки, абзаци тексту, гіперпосилання, списки, таблиці, зображення, відео тощо.

Елементами мови **HTML** є **теги**, які вказують браузеру, які об'єкти розмістити на сторінці та як їх відобразити. Імена тегів записують усередині кутових дужок **<** та **>**.

Теги можуть бути **непарними** або **парними**. Найчастіше непарні теги вказують браузеру вставити певний об'єкт на сторінку. Парні теги складаються з **відкриваючої** та **закриваючої** частин, які обмежують фрагмент тексту або HTML-коду. У закриваючій частині перед іменем тегу ставлять символ /.

Теги можуть містити **атрибути**, які уточнюють дію цих тегів. У атрибутів можуть бути **значення**. Атрибути записують після імені тегу, відокремлюючи пропуском. Значення атрибутів записують у лапках після знаку =.

Файли з HTML-кодом можна створювати як текстові файли та зберігати з розширенням імені **html**. Файлу з HTML-кодом головної сторінки сайту зазвичай дають ім'я **index.html**.

Для створення та редагування HTML-коду можна використовувати текстові редактори або спеціалізовані редактори коду. Також створення HTML-коду можливе на онлайн-ресурсах. Виконання HTML-коду відбувається після відкривання відповідних файлів у вікні браузера.

Дайте відповіді на запитання

1. Що визначає HTML-код сторінки?

2. Що таке *тег*? Як записують теги? Як записують парні теги? Як записують теги з атрибутами?

3. Який тег призначено для вставлення зображення з файлу на вебсторінку?

4. Який тег призначено для створення гіперпосилання?

5. Як можна переглянути HTML-код сторінки?

Обговоріть і зробіть висновки

1. Із чим може бути пов'язана поява нових версій мови **HTML**?

2. Чи може бути доступна для перегляду користувачами інтернету ваша сторінка, створена засобами текстового редактора, під час розробки сторінки? Чи може бути доступна для перегляду сторінка, що створюється в онлайн-редакторі?

Виконайте завдання

1. Відкрийте у вікні браузера сторінку з адресою *kievoi.ippo.kubg.edu.ua*. Проаналізуйте HTML-код цієї вебсторінки. Відкрийте файл **Розділ 5\Пункт 5.2\завдання 5.2.1.docx**. Заповніть у файлі таблицю відповідями на запитання щодо вмісту HTML-коду сторінки. Для кожного запитання вкажіть ім'я тегу, який указує на відповідь. Збережіть файл у вашій папці з тим самим іменем.

2. Відкрийте файл **Розділ 5\Пункт 5.2\завдання 5.2.2.txt**. Збережіть файл у вашій папці з іменем **завдання 5.2.2.html**. Упишіть у HTML-код текст і теги для створення на сторінці заголовка першого рівня з назвою вашого населеного пункту, горизонтальної лінії, посилання на сайт адміністрації вашого населеного пункту в інтернеті. Збережіть змінений файл у тій самій папці з тим самим іменем. Перегляньте його вміст у вікні браузера. Перевірте правильність гіперпосилання.

3. Відкрийте файл **Розділ 5\Пункт 5.2\завдання 5.2.3.txt**. Збережіть файл у вашій папці з іменем **завдання 5.2.3.html**. Перегляньте вміст збереженого файлу у вікні браузера. Змініть HTML-код таким чином, щоб:

- у заголовку браузера відображалось ваше прізвище;
- у заголовку першого рівня відображалось ваше прізвище та ім'я;
- у заголовку другого рівня відображалась назва вашого закладу освіти;
- слова *Українська Вікіпедія* виводились у напівжирному накресленні;
- останнє гіперпосилання вказувало на головну сторінку **Вікіпедії**;
- горизонтальна лінія розмістилась під гіперпосиланням на **Вікіпедію**.

Збережіть змінений файл у тій самій папці з тим самим іменем. Перегляньте його вміст у вікні браузера. Перевірте правильність гіперпосилання.

4. Створіть вебсторінку за зразком (мал. 5.11) та описом (табл. 5.1). Текст і URL-адреси ресурсів містяться у файлі **Розділ 5\Пункт 5.2\завдання 5.2.4.txt**. Збережіть створену сторінку у вашій папці у файлі з іменем **завдання 5.2.4.html**.

5. Знайдіть в інтернеті відомості про створення таблиць засобами мови **HTML**. Запишіть у зошит імена тегів, що вказують створити таблицю, заголовок таблиці, рядок таблиці, клітинку таблиці.

6. Знайдіть в інтернеті відомості про створення списків засобами мови **HTML**. Запишіть у зошит імена тегів, що вказують створити нумерований список, список з маркерами, елемент списку. Створіть вебсторінку, що містить нумерований список з назвами днів тижня та список з маркерами з назвами пір року. Збережіть створену сторінку у вашій папці у файлі з іменем **завдання 5.2.6.html**.

Готуємось до вивчення нового матеріалу

1. Назвіть етапи розробки вебсайту.
2. Які ви знаєте сучасні тренди вебдизайну?
3. Документи яких типів можуть бути створені з використанням сервісів **Google**?

5.3. АВТОМАТИЗОВАНІ ЗАСОБИ СТВОРЕННЯ ТА ПУБЛІКАЦІЇ ВЕБСАЙТІВ

У цьому пункті йтиметься про:

- автоматизовані засоби створення вебсайтів;
- створення сайтів у сервісі **Google Сайти**;
- вибір теми оформлення сайту;
- створення вебсторінок і додавання елементів на сторінки;
- попередній перегляд і публікацію сайту.

АВТОМАТИЗОВАНІ ЗАСОБИ СТВОРЕННЯ ВЕБСАЙТІВ

Пригадайте

● Які етапи створення сайту? У чому вони полягають? ● Які програми можна використати для створення HTML-коду вебсторінок? ● Із чого складається та як створюється HTML-код вебсторінок?

Як ви знаєте, створення сторінок сайту розпочинається після визначення його структури та створення дизайн-макета. Для створення сайту можна самостійно вводити HTML-код його сторінок у текстовому редакторі або використати автоматизовані засоби створення та публікації вебсайтів. У таких засобах існують шаблони дизайнерського оформлення, а також варіанти структури сайтів.

Спеціалізовані вебредактори – це програми, що призначені для візуального конструювання вебсторінок, наприклад **Adobe Dreamweaver**, **Google Web Designer**, **WYSIWYG Web Builder** та інші. Розробник розміщує в окремому документі шаблони окремих елементів сторінки – заголовків, зображень, таблиць, списків, гіперпосилань та інших, вибираючи їх з бібліотеки об'єктів, заповнює шаблони потрібним умістом, а HTML-код сторінки генерується автоматично. Такі програми називають **WYSIWYG-редакторами** (англ. ***What You See Is What You Get*** – що ви бачите, то ви й отримуєте).

Іншим засобом створення сайтів є **системи управління вебконтентом WCMS** (англ. ***Web Content Management System*** – система управління вебконтентом) – програми, що призначені для створення, редагування та керування вмістом вебсайту. Популярними **WCMS** є системи **Joomla**, **WordPress**, **MediaWiki** та інші. Вони містять набори готових шаблонів вебсторінок і модулів, що роблять сайт динамічним: форумів, стрічок новин, каталогів файлів тощо. Контент сторінок, створених у цих системах, зберігається в базі даних, а HTML-код генерується автоматично на основі вибраного шаблону.

Спеціалізовані вебредактори та системи управління вебконтентом можна встановити на локальному комп'ютері. Створені в їх середовищі файли потрібно самостійно розміщувати в інтернеті.

Існують також **онлайн-сервіси конструювання сайтів**, які призначені для розробки сайтів безпосередньо в інтернеті. Такими є сервіси **Google Сайти** (*sites.google.com*), **WIX** (*uk.wix.com*), **Weblium** (*ua.weblium.com*) та інші. Створені в них на основі готових шаблонів сайти автоматично розміщуються на серверах інтернету, але розробник зазвичай не може безпосередньо редагувати HTML-код вебсторінок.

Створення вебсайту засобами онлайн-сервісів конструювання сайтів відбувається в кілька кроків:

1. Реєстрація облікового запису на сервері онлайн-сервісу конструювання сайтів.
2. Вибір шаблону сайту.
3. Створення та редагування сторінок сайту, заповнення їх контентом.
4. Публікація сайту в інтернеті.

СТВОРЕННЯ САЙТУ В СЕРВІСІ GOOGLE САЙТИ

Пригадайте

• Які сервіси **Google** ви використовували? З якою метою? • Як і на якому носії даних відбувається збереження електронних документів у сервісах **Google**? • Які переваги надає використання сервісів **Google** порівняно з використанням локальних програмних засобів? Які є недоліки їх використання?

Розглянемо, як відбувається створення сайту в сервісі **Google Сайти**. Для його використання потрібно мати обліковий запис **Google**.

Щоб створити сайт, слід:

1. Відкрити у вікні браузера сторінку **Google** (*google.com*) та ввести дані власного облікового запису.
2. Вибрати у списку кнопки **Додатки Google** додаток **Сайти**

   або відкрити сторінку з адресою *https://sites.google.com*.
3. У вікні **Сайти** вибрати шаблон, на основі якого буде створено новий сайт (мал. 5.13).



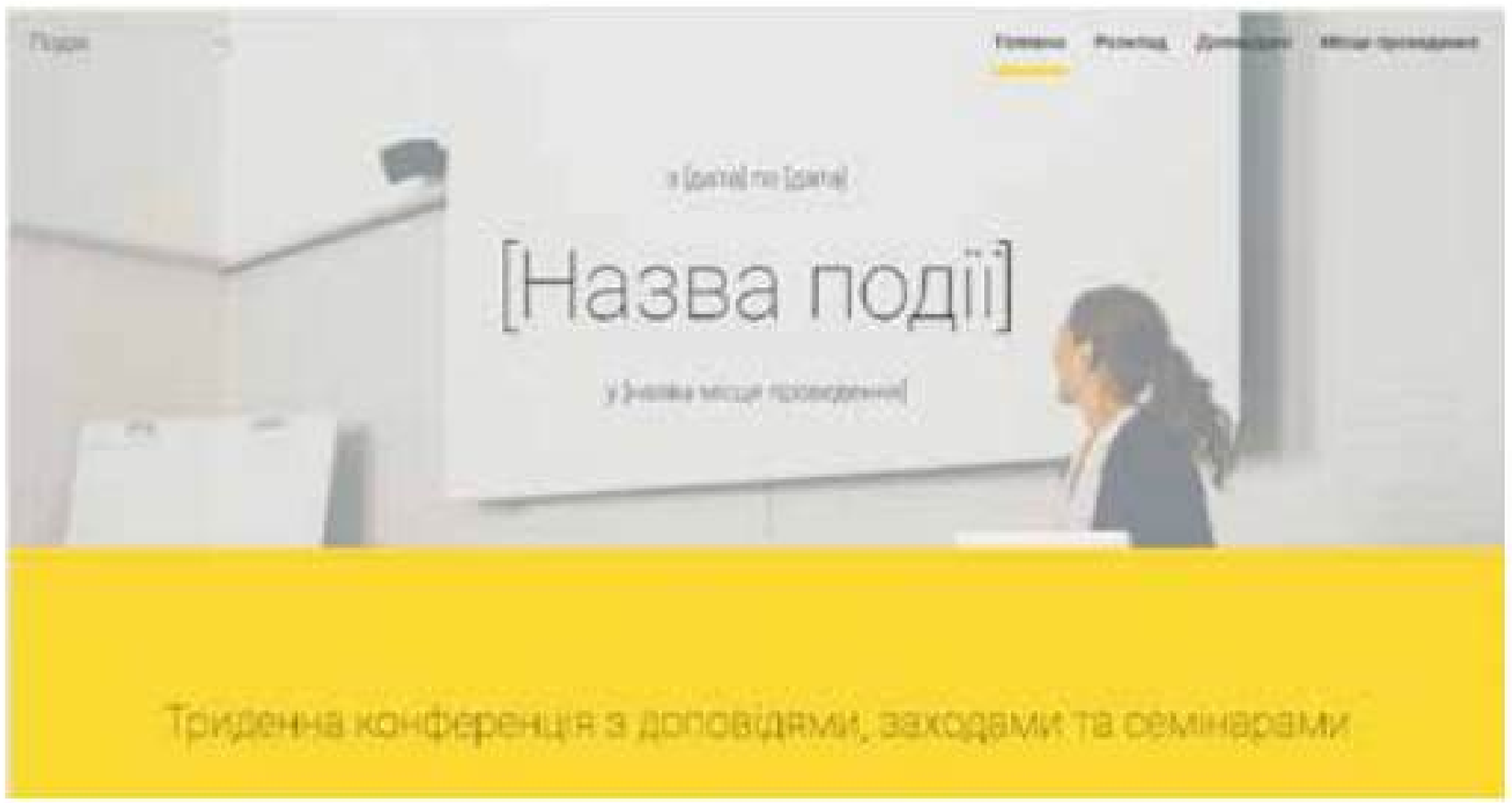
Мал. 5.13. Список шаблонів сайтів на сторінці **Сайти**

Усі шаблони, за виключенням шаблону **Порожній сайт**, містять кілька сторінок, на яких вже розміщено панелі навігації, елементи оформлення та рекомендації щодо заповнення сторінок контентом. На малюнку 5.14 наведено вигляд головної сторінки шаблону *Подія*.

Більше шаблонів можна побачити, якщо відкрити **Галерею шаблонів**, вибравши відповідну кнопку Галерея шаблонів у вікні **Сайти**.

Після вибору шаблону подальша робота буде полягати в заповненні його контентом відповідно до рекомендацій, за потребою – зміненні вмісту та оформлення, додаванні або вилученні сторінок, публікації готового сайту.

Якщо ви не знайшли шаблон, який вас задовольняє, то ви можете створити сайт на основі шаблону **Порожній сайт**. Для цього можна скористатися одним із способів:



Мал. 5.14. Головна сторінка шаблону *Подія*

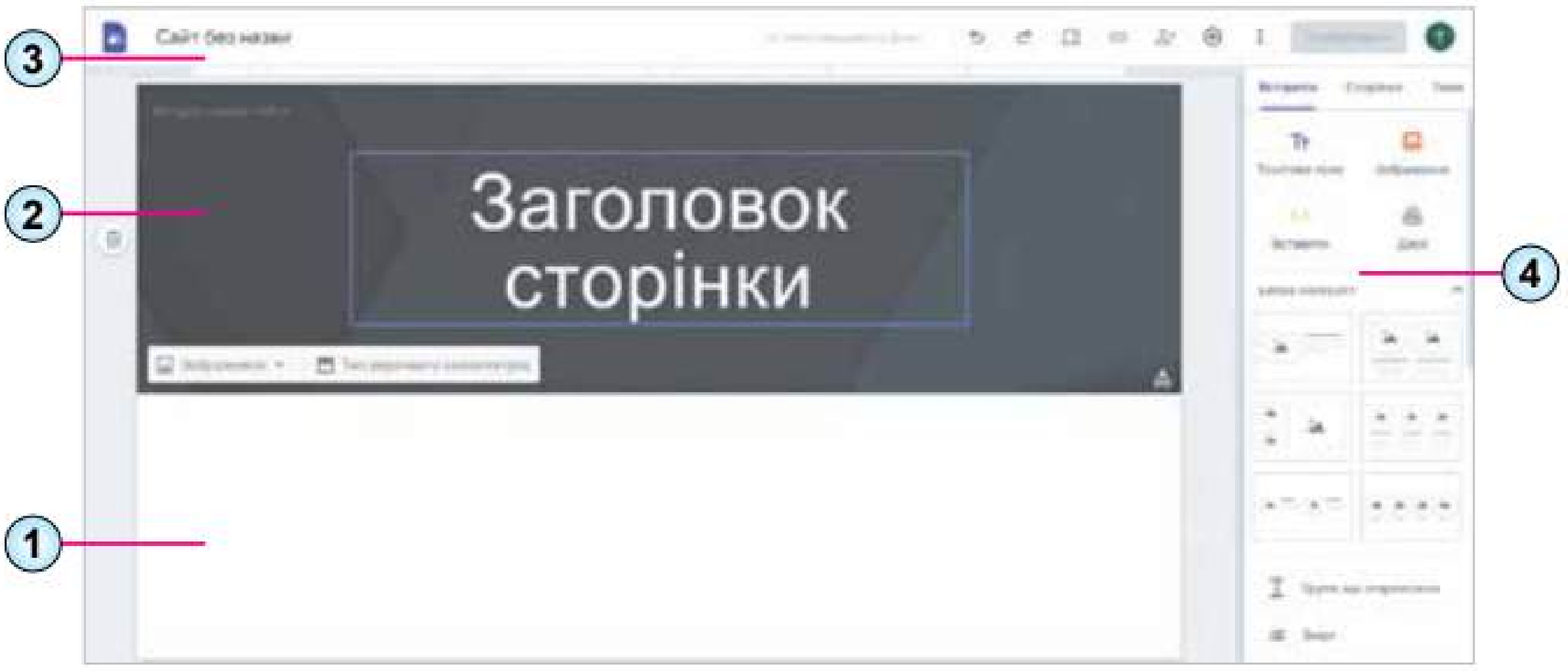
- вибрати у вікні **Сайти** шаблон **Порожній сайт;**
- вибрати в нижньому правому куті вікна **Сайти** кнопку **Створити новий сайт** ;
- відкрити вікно **Google Диска** та виконати ***Створити*** ⇒ ***Більше*** ⇒ ***Google Сайти***.

Незалежно від обраного способу у вікні браузера буде відкрито редактор сайтів **Google** з однією головною сторінкою нового сайту в режимі редагування (мал. 5.15).

Верхня панель редактора містить назву сайту та інструменти для роботи із сайтом.

Основну частину вікна займає **робоча область**, у якій розміщено:

- **верхній колонтитул** вебсторінки з фоновим зображенням, назвою (заголовком) сторінки та назвою сайту;
- **робоче поле** сторінки для розміщення текстових і нетекстових елементів.



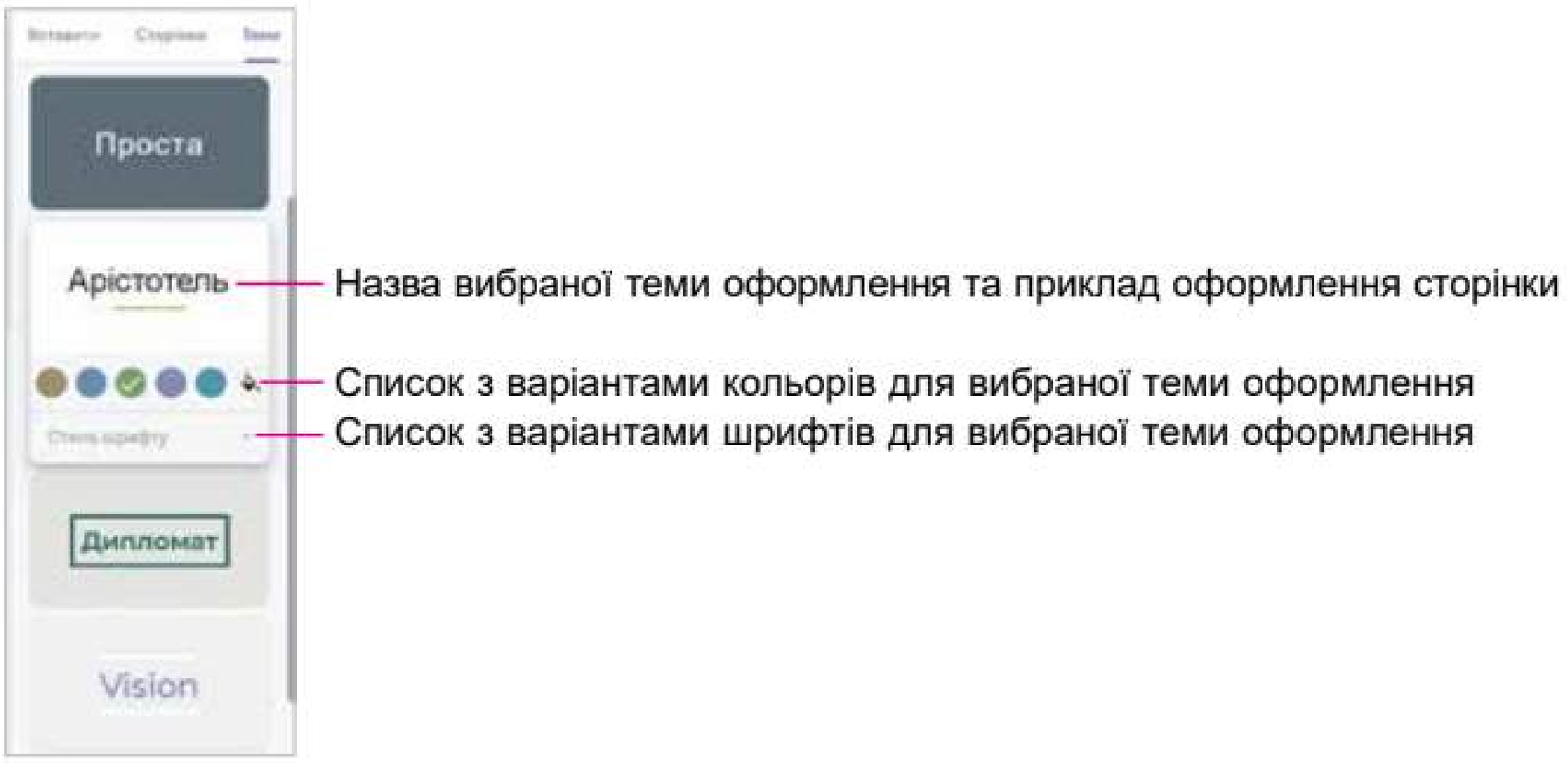
Мал. 5.15. Вікно редактора сайтів **Google** з головною сторінкою нового сайту:
1 – робоче поле вебсторінки; *2* – верхній колонтитул вебсторінки; *3* – верхня панель редактора сайтів; *4* – бічна панель редактора сайтів

Справа від робочої області сторінки розміщена **бічна панель** редактора сайтів з вкладками **Вставити, Сторінки, Теми.** Інструменти цієї панелі використовують для додавання, видалення та редагування сторінок, вставлення об'єктів на сторінки, вибору та змінення оформлення тощо.

Кожен сайт має назву, яка відображає його вміст або призначення. Щоб задати назву сайту, її потрібно ввести в поле **Введіть назву сайту** у верхньому колонтитулі головної сторінки.

Для змінення теми оформлення сайту слід:

1. Вибрати вкладку **Теми** на бічній панелі.
2. Вибрати назву теми оформлення.
3. Вибрати кольорову гаму та стиль шрифту з тих, що запропоновані для вибраної теми (мал. 5.16).



Мал. 5.16. Вкладка **Теми** редактора сайтів **Google**

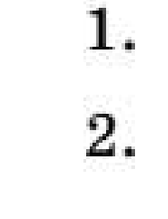
Вигляд верхнього колонтитула сторінки можна змінювати. Після наведення вказівника на верхній колонтитул з'являється панель з командами **Зображення** та **Тип верхнього колонтитула.** Після вибору першої команди можна додати зображення для верхнього колонтитула з носіїв даних комп'ютера або вибрати з розміщених в інтернеті. Вибір другої команди дає можливість змінити тип колонтитула – *Обкладинка*, *Великий банер*, *Банер* або *Лише заголовок*. Верхні колонтитули на різних сторінках сайту можуть відрізнятися.

Також на всіх сторінках можна розмістити нижній колонтитул. Для цього потрібно підвести вказівник до нижньої межі сторінки та вибрати кнопку **Додати нижній колонтитул.** У поле, що з'явиться, потрібно ввести дані, наприклад рік створення або оновлення сайту, відомості про його розробника та інші. Нижній колонтитул буде автоматично повторюватися на всіх сторінках сайту.

У процесі створення файл з проєктом сайту автоматично зберігається на **Google Диску.** Ім'я файлу збігається з назвою сайту, значок файлу має вигляд . Цей файл можна відкривати в подальшому

для редагування сайту. Також у вікні **Сайти** з'являється ескіз цього сайту, використовуючи який також можна відкрити сайт у режимі редагування.

СТВОРЕННЯ ВЕБСТОРІНОК У РЕДАКТОРІ САЙТІВ GOOGLE



Поміркуйте

- Від чого залежить кількість вебсторінок, з яких складається сайт?
- Чи можуть бути однаковими назви кількох вебсторінок одного сайту?
- Для чого вебсторінки сайту розподіляють за окремими розділами?

Створення вебсторінок сайту та його тематичних розділів здійснюється відповідно до мапи сайту, яка розроблена на етапі визначення структури сайту.

Сайт одразу після створення його на основі шаблону **Порожній сайт** містить лише одну сторінку, яка є *головною сторінкою сайту*. Назву головної сторінки потрібно ввести в поле **Заголовок сторінки** у верхньому колонтитулі. Часто назва головної сторінки збігається з назвою сайту.

Для створення нової сторінки на сайті слід:

1. Відкрити на бічній панелі вкладку **Сторінки**.
2. Навести вказівник на кнопку + в нижній частині бічної панелі.
3. Вибрати кнопку **Створити сторінку** .
4. Увести назву сторінки в поле **Назва** у вікні **Створити сторінку**.
5. Вибрати посилання **Готово**.

Після цього на вкладці **Сторінки** бічної панелі з'явиться блок з назвою створеної сторінки, а на панелі навігації сайту буде утворено посилання на цю сторінку (мал. 5.17).



Мал. 5.17. Панель навігації та вкладка **Сторінки** бічної панелі:
1 – панель навігації сайту; *2* – посилання на створену сторінку;
3 – блок створеної сторінки на бічній панелі

У структурі сайту можуть бути тематичні розділи, у яких згруповано всі сторінки певної теми. Для створення тематичного розділу слід:

1. Відкрити на бічній панелі вкладку **Сторінки**.
2. Навести вказівник на кнопку + в нижній частині бічної панелі.

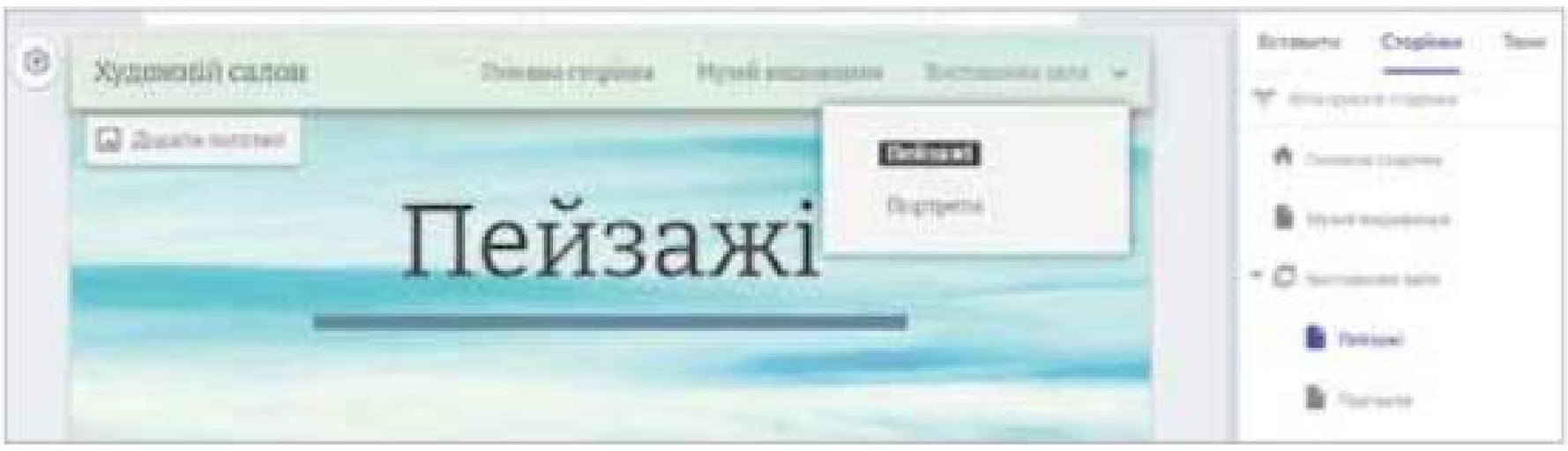
3. Вибрати кнопку **Розділ нового меню** .
4. Увести назву розділу в поле **Назва** у вікні **Розділ нового меню.**
5. Вибрати посилання **Готово.**

На бічній панелі блок тематичного розділу має вигляд Виставкова зала .

На панелі навігації посилання на розділ буде активним, якщо до розділу входить хоча б одна сторінка. Щоб створити в розділі сторінку, слід:

1. Вибрати кнопку у блоці розділу.
2. Вибрати в меню, що відкриється, команду **Додати підсторінку.**
3. Увести назву сторінки в поле **Назва** у вікні **Нова підсторінка.**
4. Вибрати посилання **Готово.**

На малюнку 5.18 наведено вигляд вкладки **Сторінки** та панелі навігації сайту з посиланнями на сторінки тематичного розділу.



Мал. 5.18. Вкладка **Сторінки** та панель навігації сайту з посиланнями на сторінки тематичного розділу

Розміщення посилань на панелі навігації можна змінити, перетягнувши на вкладці **Сторінки** бічної панелі блок з назвою сторінки в потрібне місце.

У деяких шаблонах сайтів панель навігації згорнута у значок так званого **бургер-меню** , після вибору якого список сторінок відкривається в лівій бічній панелі. Змінити вигляд і положення панелі навігації можна, вибравши кнопку **Налаштування навігації** ліворуч від верхнього колонтитула та у вікні **Налаштування** в розділі **Навігація** – відповідний режим: **За верхнім краєм** або **Збоку** (мал. 5.19).



Мал. 5.19. Вибір місця розташування панелі навігації

ВСТАВЛЕННЯ ОБ'ЄКТІВ НА ВЕБСТОРІНКУ



Пригадайте

● Які об'єкти можуть бути вставлені на вебсторінку? ● Що таке *макет слайда комп'ютерної презентації*? ● Як уставити об'єкт на слайд комп'ютерної презентації, використовуючи покажчик місця заповнення?

На вебсторінку можна вставити різні об'єкти: текстові поля, зображення, відео, документи, які розміщено на **Google Диску**, та інше. Для цього призначено вкладку **Вставити** на правій бічній панелі редактора сайтів.

Призначення деяких елементів керування вкладки **Вставити** наведено в таблиці 5.2.

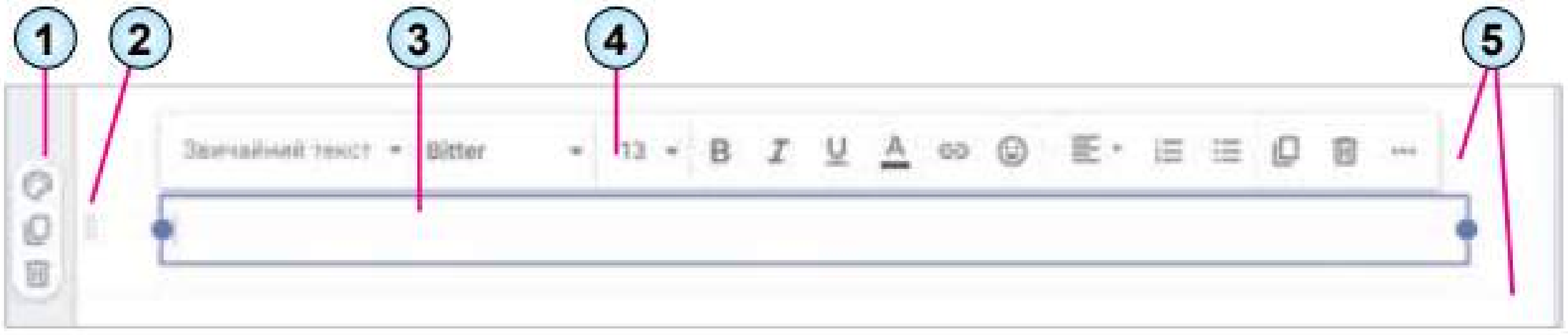
Таблиця 5.2

Призначення деяких елементів керування вкладки Вставити

Елемент керування	*Призначення*
Тт Текстове поле	Уставлення на сторінку текстового поля
Зображення	Уставлення на сторінку зображення з **Google Диска**, за URL-адресою з інтернету, з носіїв даних комп'ютера тощо
Вставити	Уставлення об'єктів різних типів з інтернету за їх URL-адресою або вставлення фрагментів HTML-коду
Диск	Уставлення документів різних типів з **Google Диска**
БЛОКИ КОНТЕНТУ	Вибір макета розміщення об'єктів на сторінці: у два, три, чотири стовпці та інші
Зміст	Уставлення змісту для швидкого переходу до вибраного заголовка на поточній сторінці, який автоматично формується із заголовків у текстових полях
Карусель зображень	Уставлення слайд-шоу з вибраних зображень
YouTube	Уставлення відео з сервісу **YouTube**
Форми	Уставлення **Google Форми**

Також на вебсторінку можна вставити **Google Карти**, **Google Календарі**, кнопки з посиланнями на вебсторінки, посилання на сторінки соціальних мереж у вигляді відповідних значків, діаграми з електронних таблиць та інші об'єкти.

Під час вставлення об'єкта на сторінці утворюється окремий її розділ завширшки на всю сторінку. Уставлений об'єкт розміщується всередині цього розділу сторінки у прямокутному блоці з маркерами на межі блока. На малюнку 5.20 наведено вигляд розділу сторінки, у якому міститься блок текстового поля.



Мал. 5.20. Розділ сторінки з блоком текстового поля:
1 – панель інструментів розділу сторінки; *2* – кнопка для переміщення розділу сторінки; *3* – блок текстового поля; *4* – панель інструментів блока текстового поля; *5* – верхня та нижня межа розділу сторінки

Кожний розділ сторінки має панель інструментів, призначенням яких є:

– вибір кольору фону для розділу сторінки;

– створення копії розділу сторінки;

– видалення розділу сторінки.

Блок з об'єктом можна перемістити всередині розділу сторінки або в інший її розділ, змінити розміри блока, використовуючи маркери на межах, або видалити. На малюнку 5.21 наведено вигляд розділу сторінки з трьома блоками текстових полів.



Мал. 5.21. Розділ сторінки з трьома блоками текстових полів

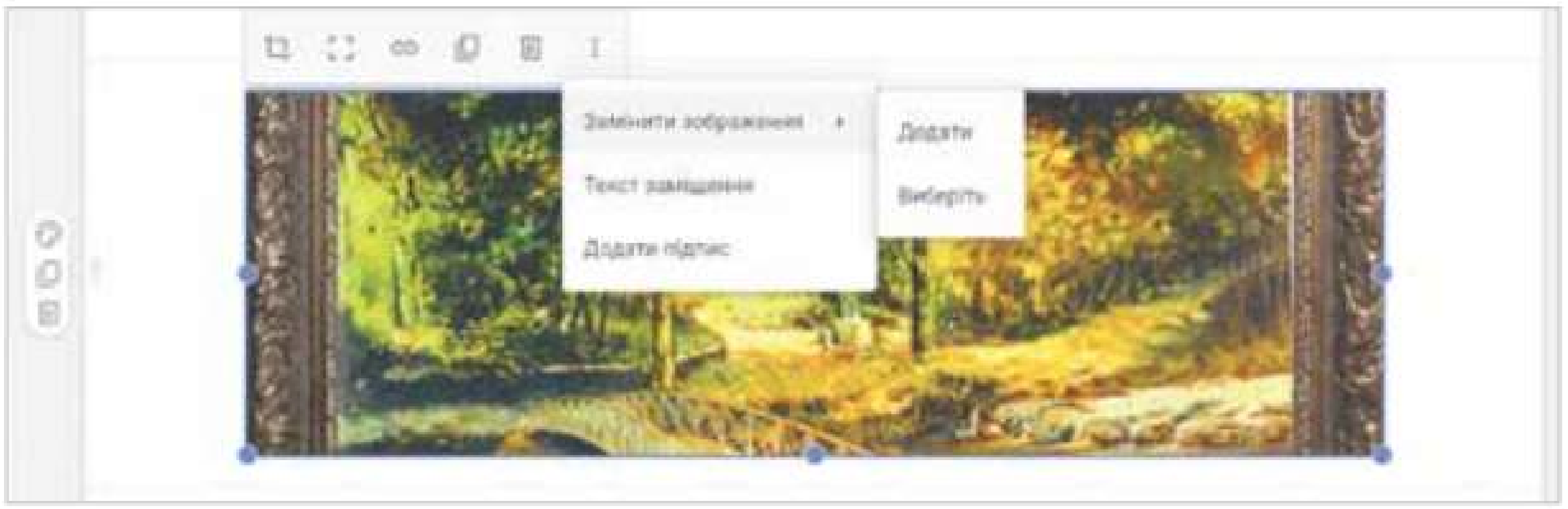
Для кожного блока, якщо його вибрати, відкривається окрема панель інструментів.

На панелі інструментів текстового поля (мал. 5.21) розміщено елементи керування, подібні до тих, що використовуються в текстовому процесорі. До тексту, уведеного в текстовому полі, може бути застосований один зі стандартних стилів – *Назва*, *Заголовок*, *Підзаголовок*, *Звичайний текст* і *Малий текст*. Редагування та форматування тексту в текстовому полі відбувається так само, як у текстовому процесорі.

Блок зображення створюється після вибору кнопки 

на бічній панелі та вибору файлу, зображення з якого буде відображатися на сторінці. Файл можна додати з носіїв даних вашого комп'ютера, вибрати з вашого **Google Диска** або сервісу **Google Фотографії**, уставити за відомою URL-адресою або знайти в **Google Зображеннях**.

Вигляд панелі інструментів блока зображення наведено на малюнку 5.22.

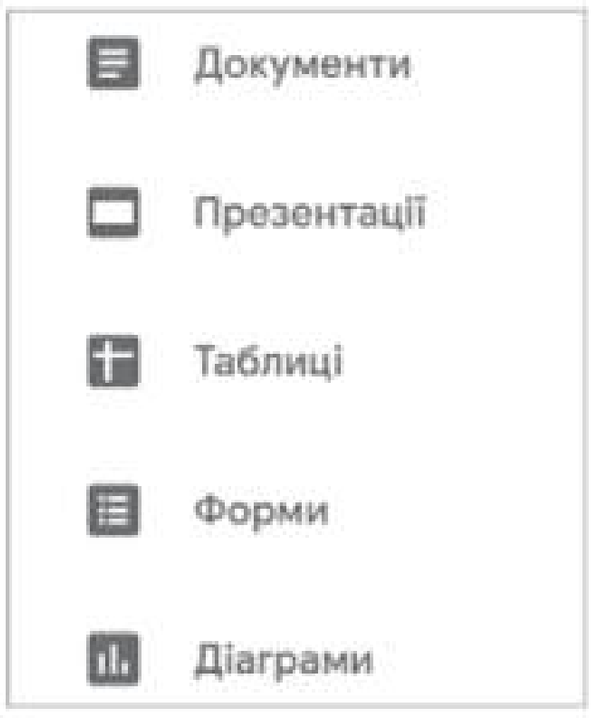


Мал. 5.22. Панель інструментів блока зображення

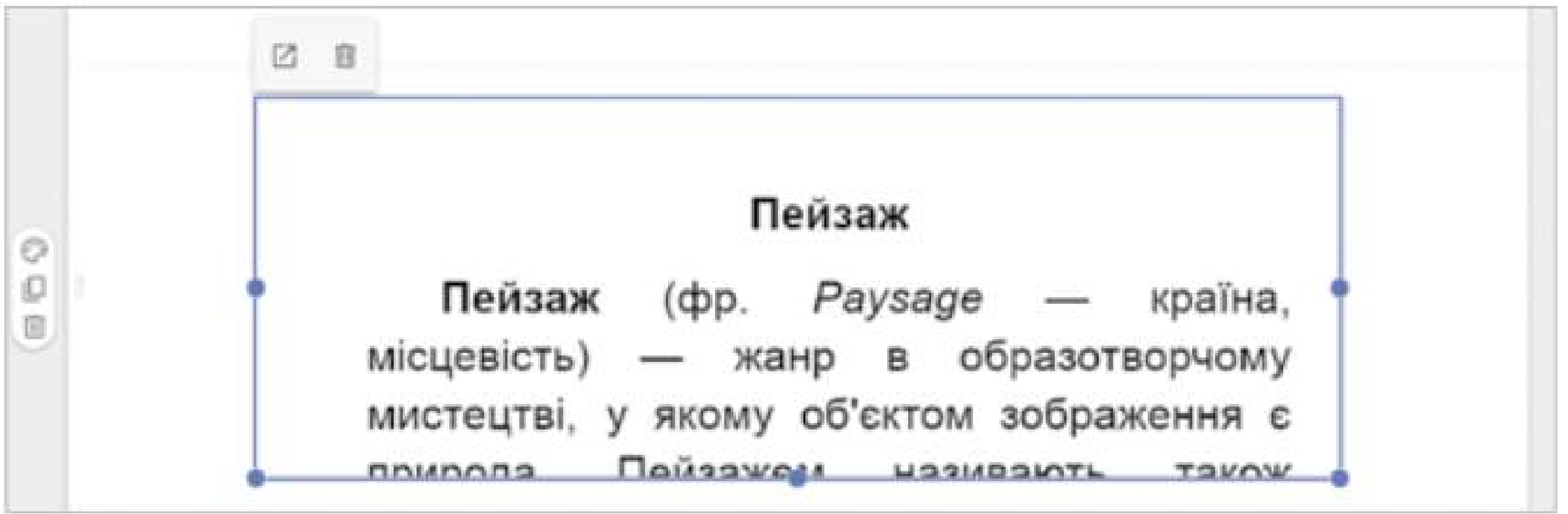
За замовчуванням ширина блока зображення складає третину ширини сторінки. Розміри блока можна змінювати. Використовуючи кнопку панелі інструментів **Обрізати (Ввести)** , можна вибрати, яку частину зображення буде відображено у блоці. Вибір кнопки **Скасувати обрізування** приводить до відображення у блоці всього зображення.

На вебсторінку можна вставляти файли різних типів, які зберігаються на вашому **Google Диску**. Якщо вибрати кнопку 

, то праворуч від робочої області сторінки відкриється панель **Google Диск**, на якій можна побачити вміст вашого **Google Диска**, знайти та вибрати файл, який потрібно додати на сторінку. Якщо потрібно вставити файл, створений з використанням одного з сервісів **Google**, то можна скористатися кнопками вкладки **Вставити** для вибору та вставлення відповідного файлу (мал. 5.23). Після вибору будь-якої з цих кнопок відкривається панель з переліком файлів лише вибраного типу на вашому **Google Диску**. Вибраний на цій панелі файл буде вставлений на сторінку в окремому блоці. За потреби буде запитано ваш дозвіл на перегляд цього файлу користувачами інтернету. На малюнку 5.24 наведено вигляд блока зі вставленим Google-документом. Побачити документ у повному обсязі можна буде після публікації сайту.



Мал. 5.23. Кнопки для вставлення файлів, створених у різних сервісах **Google**



Мал. 5.24. Блок Google-документа на сторінці

Якщо на вкладці **Вставити** вибрати будь-який елемент з групи **Блоки контенту**, то на сторінці буде створено розділ відповідного макета. У цьому розділі на визначених місцях можна вводити заголовки та абзаци тексту, вставляти зображення, документи різних типів, відео з сервісу **YouTube**, **Google Календарі** та **Google Карти**.

Для видалення вставленого об'єкта потрібно вибрати кнопку **Видалити** на панелі інструментів блока цього об'єкта.

ПОПЕРЕДНІЙ ПЕРЕГЛЯД І ПУБЛІКАЦІЯ САЙТУ

Пригадайте

● Який дизайн називають *адаптивним*? ● Які зміни в розміщенні та вигляді об'єктів вебсторінки можуть відбуватися під час змінення розміру області перегляду? ● Що таке *URL-адреса* сторінки?

Вебсторінки, що створюються в редакторі сайтів **Google**, мають адаптивний дизайн. Щоб побачити, як виглядатиме сайт для користувачів інтернету під час перегляду його на екрані смартфона, планшетного комп'ютера, ноутбука чи стаціонарного комп'ютера, можна вибрати кнопку **Попередній вигляд** на верхній панелі редактора сайтів і відповідну кнопку (*Телефон*, *Планшет*, *Великий екран*) на панелі керування попереднім переглядом (мал. 5.25), яка розміщується в нижньому правому куті екрана. Вийти з режиму попереднього перегляду можна вибором кнопки .



Мал. 5.25. Панель керування попереднім переглядом сайту

Створений вами сайт буде залишатися недоступним для користувачів інтернету, поки ви його не опублікуєте. Для публікації потрібно:

1. Вибрати кнопку **Опублікувати** на верхній панелі редактора сайтів.
2. Увести, використовуючи малі літери англійського алфавіту, цифри та дефіс, останню частину URL-адреси сайту. Перша частина

адреси для всіх сайтів буде однаковою: *sites.google.com/view/*. Остання частина повинна бути унікальною для кожного сайту. Якщо введена вами адреса вже існує, то про це буде повідомлено та її потрібно змінити.

3. Вибрати кнопку **Опублікувати**.

Опублікований сайт можна переглянути, вибравши команду **Переглянути опублікований сайт** у списку кнопки **Опублікувати**. URL-адреса, яку потрібно повідомити користувачам інтернету для перегляду вашого сайту, міститься в рядку адреси у вікні перегляду опублікованого сайту. Також цю адресу можна скопіювати, вибравши кнопку **Копіювати посилання на опублікований сайт** на панелі інструментів редактора сайтів.

Після внесення будь-яких змін до вашого сайту його потрібно повторно публікувати. Уведення URL-адреси сайту буде вимагатися лише під час першої публікації сайту.

Як і до інших електронних документів, створених у різних сервісах **Google**, до сайту можна надавати дозвіл іншим людям для спільного редагування. Для цього призначена кнопка **Надати доступ іншим** , яка розміщена на верхній панелі редактора сайтів.

Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/re2cH4s6* або QR-кодом.

Найважливіше в цьому пункті

Створення сайту може виконуватися з використанням автоматизованих засобів: спеціалізованих вебредакторів, систем управління вебконтентом **WCMS**, онлайн-систем конструювання сайтів.

Створення вебсайту засобами онлайн-систем конструювання сайтів відбувається у кілька кроків: реєстрація облікового запису на сервері онлайн-системи конструювання сайтів, вибір шаблону сайту, створення та редагування сторінок сайту, заповнення їх контентом, публікація сайту в інтернеті.

Для створення сайту в сервісі **Google Сайти** можна використати один із запропонованих шаблонів з рекомендаціями до вмісту або шаблон **Порожній сайт**. На вебсторінки можна вставляти різні об'єкти: текстові поля, зображення, відео, документи, які розміщено на **Google Диску**, та інше.

Вебсторінки, що створюються в редакторі сайтів **Google**, мають адаптивний дизайн.

Дайте відповіді на запитання

1. Які існують засоби автоматизованого створення сайтів? Коротко схарактеризуйте їх.

2. Які етапи створення вебсайтів засобами сервісу **Google Сайти**? Поясніть їх.

3. Як змінити вигляд верхнього та створити нижній колонтитул на сторінці?

4. Як вибрати тему оформлення для сайту? Значення яких властивостей можуть бути вибрані разом з темою оформлення?

5. Як створити нову вебсторінку на сайті? Як створити сторінку в тематичному розділі?

6. Які об'єкти можна вставити на вебсторінки сайту, створеного у сервісі **Google Сайти**?



Обговоріть і зробіть висновки

1. Чому, на вашу думку, потрібно виконувати публікацію сайту після внесення змін до нього?

2. Які, на ваш погляд, переваги та недоліки створення сайтів шляхом написання HTML-коду їх сторінок і шляхом використання онлайн-систем конструювання сайтів?



Виконайте завдання

1. Створіть у сервісі **Google Сайти** сайт відповідно до мапи, наведеної на малюнку 5.1 (див. с. 164). Надайте сайту назву *Україна*. Виберіть тему оформлення *Дипломат*. Уведіть на кожній сторінці розділу *Загальні відомості* по одному абзацу тексту. Вставте на кожній сторінці розділу *Символіка* по два зображення. Опублікуйте сайт. Повідомте вчителю / вчительці URL-адресу створеного вами сайту.

2. Доберіть назви сторінок і створіть у текстовому процесорі мапу сайту *Рекордсмени живої природи*. Збережіть її у вашій папці у файлі з іменем **завдання 5.3.2**. Створіть у сервісі **Google Сайти** сторінки сайту відповідно до розробленої мапи. Уставте на кожній сторінці по одному абзацу тексту та по одному зображенню. Опублікуйте сайт. Перегляньте опублікований сайт, відкрийте HTML-код головної сторінки. Переконайтеся в наявності тегів базової структури сторінки. Повідомте вчителю / вчительці URL-адресу створеного сайту.

3. Створіть у сервісі **Google Сайти** сайт *Наші космічні сусіди*. Мапа сайту, текст та зображення для розміщення на сторінках містяться в папці **Розділ 5\Пункт 5.3\Завдання 5.3.3**. Виберіть для сайту тему оформлення *Вражаюча*. Опублікуйте сайт. Повідомте вчителю / вчительці URL-адресу створеного вами сайту.



4. Створіть у сервісі **Google Сайти** сайт про себе на основі шаблону *Портфоліо учня*. Виберіть шаблон у **Галереї шаблонів**



, якщо він не відображається на сторінці **Сайти**. Уведіть текстові дані відповідно до рекомендацій шаблону, заповніть блоки-заповнювачі та сторінки предметів і позашкільних занять відповідно до вашого класу та ваших інтересів. Замініть зображення на власні. Видаліть блоки шаблону, які залишились не заповненими, або додайте та заповніть нові блоки за потреби. Опублікуйте сайт. Повідомте вчителю / вчительці URL-адресу створеного вами сайту.



5. Об'єднайтесь у групу з трьох осіб. Доберіть матеріали та створіть спільно сайт *Домашні улюбленці* у сервісі **Google Сайти**. Розробіть і розмістіть на сторінках сайту Google-форму для опитування відвідувачів сайту про їх домашніх улюбленців. Уставте відео з сервісу **YouTube** про домашніх тварин. Опублікуйте сайт. Повідомте вчителю / вчительці URL-адресу створеного вами сайту.

Готуємось до вивчення нового матеріалу

1. Які типи даних ви знаєте? Дані яких типів можуть бути опрацьовані в електронних таблицях?
2. Як утворюються адреси даних в електронних таблицях?
3. Як записують формули в електронних таблицях?

Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Створення та публікація вебресурсів»** *(початковий, середній, достатній, високий рівень).*

- Я розумію та вмію пояснити особливості нелінійного тексту.
- Я знаю та враховую критерії оформлення вебсторінок, їх відповідність сучасним трендам вебдизайну, можу запропонувати власні критерії.
- Я вмію використовувати гіпертекстові документи та створювати гіпертекстові посилання на ресурси інтернету.
- Я знаю можливості програмних засобів для створення гіпертекстових документів.
- Я вмію використовувати програмні засоби та технології для створення й редагування вебсторінок у різних режимах (онлайн, офлайн).
- Я вмію обирати та комбінувати тексти, зображення, мультимедійні дані для проєктування, створення, публікації вебсторінок.
- Я можу визначити відповідність вмісту та вигляду вебсторінки цільовій аудиторії, отримати відгуки користувачів для вдосконалення продукту.

Повторіть той матеріал, який ви знаєте недостатньо.

Практична робота № 7.
«Створення сайту з використанням онлайн-системи конструювання сайтів»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Створіть у сервісі **Google Сайти** сайт *Винаходи українців у ІТ-галузі*.
2. Виберіть тему оформлення *Vision*.
3. Створіть і розмістіть вебсторінки на основі такої мапи сайту:



4. Уставте на головній сторінці відео **Історії українського ІТ** із сервісу **YouTube** (*https://www.youtube.com/watch?v=PG8QoaTG1kc*).
5. Заповніть сторінки текстовими матеріалами, наведеними у файлі **Розділ 5\Пункт 5.3\практична 7.docx**, та зображеннями з інтернету з наданими URL-адресами файлів. Створіть гіперпосилання на інтернет-ресурси, з яких отримано відомості та фотографії.
6. Створіть нижній колонтитул і уведіть ваші дані як розробника сайту.
7. Опублікуйте сайт з адресою *discovery-прізвище*, де *прізвище* – ваше прізвище, записане англійською.
8. Повідомте вчителю / вчительці URL-адресу створеного вами сайту.

РОЗДІЛ 6

ОПРАЦЮВАННЯ ДАНИХ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- об'єкти електронної таблиці, їх властивості;
- формати даних в електронних таблицях;
- абсолютні та мішані посилання, їх модифікація та використання в електронних таблицях;
- призначення та використання логічних, математичних і статистичних функцій у табличному процесорі;
- умовне форматування даних в електронних таблицях;
- сортування та фільтрування даних в електронних таблицях;
- установлення параметрів сторінки та друкування електронної таблиці;
- розв'язування задач з різних дисциплін засобами табличного процесора.

6.1. ОБ'ЄКТИ ЕЛЕКТРОННОЇ ТАБЛИЦІ, ЇХ ВЛАСТИВОСТІ. ФОРМАТИ ДАНИХ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ. АДРЕСАЦІЯ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ

У цьому пункті йтиметься про:

- об'єкти електронної таблиці, їх властивості;
- формати числових даних в електронних таблицях;
- відносні, абсолютні та мішані посилання у формулах.

ОБ'ЄКТИ ЕЛЕКТРОННОЇ ТАБЛИЦІ, ЇХ ВЛАСТИВОСТІ

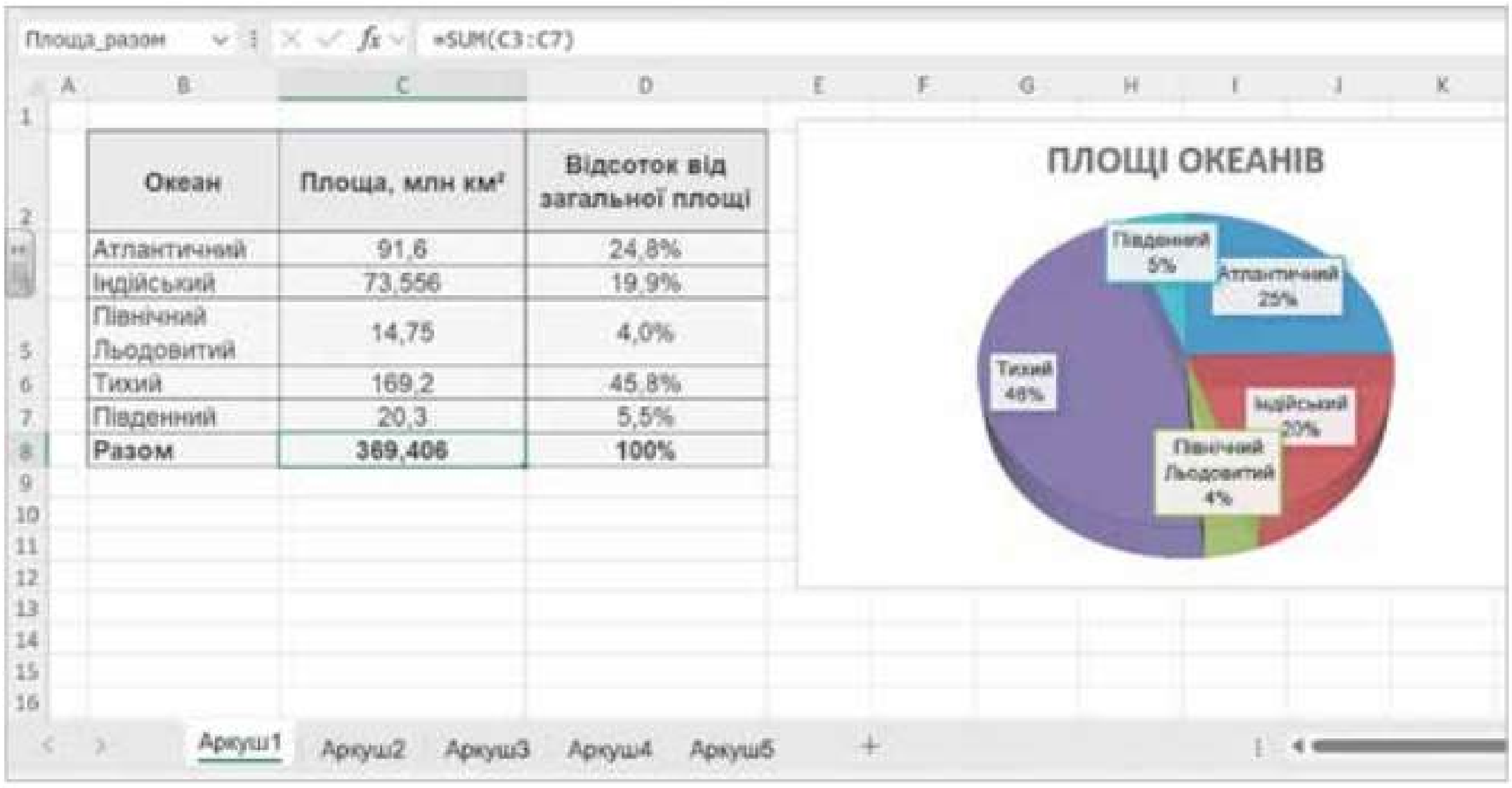
Пригадайте

● Що таке *табличний процесор*? Яке його призначення? ● Що таке *електронна таблиця*? Із чого вона складається? Які дані можуть міститися у клітинках електронної таблиці? ● Із чого складається адреса клітинки? Що таке *діапазон клітинок*? Як задається його адреса?

У 6-му класі ви почали вивчати та використовувати електронні таблиці для виконання математичних обчислень у табличному процесорі **Microsoft Excel**. У 8-му класі ви продовжите вивчення цього програмного забезпечення на прикладі програми **Excel** з пакета програм **Microsoft Office 365**.

Як вам уже відомо, основним об'єктом опрацювання табличного процесора **Microsoft Excel** є **електронна книга**, яка має ім'я, за замовчуванням це ім'я *Книга1*. Користувач може змінити ім'я книги під час її збереження у файлі. Стандартним типом файлу електронної книги є тип **Книга Excel**, а стандартним розширенням імені файлу є **xlsx** (значок).

Складається книга з **аркушів** з іменами **Аркуш1**, **Аркуш2** і так далі (мал. 6.1), які зазначено на **ярликах** аркушів. Користувач за



Мал. 6.1. Електронна книга та її об'єкти

бажанням може видаляти аркуші або додавати нові, змінювати їх імена на інші.

На аркуші електронної книги автоматично створюється **електронна таблиця**, яка складається з **рядків** і **стовпців**, на перетині яких утворюється **клітинка**. Кожна клітинка електронної таблиці має **адресу**, яка складається з номера стовпця та рядка, на перетині яких вона розміщена (наприклад, **C5**, **D2**). У клітинках електронної таблиці можуть зберігатися числа, тексти та формули.

Клітинці можна надати **ім'я**, наприклад, клітинка **C8** на малюнку 6.1 має ім'я *Площа_разом*. Імена клітинок легше запам'ятовувати, ніж їх адреси. Також їх зручніше використовувати у формулах, оскільки ім'я клітинки не змінюється у формулах під час їх копіювання. Якщо клітинку, яка має ім'я, зробити поточною, то в полі **Ім'я** буде відображатися не адреса клітинки, а її ім'я.

Щоб надати клітинці ім'я, потрібно виділити клітинку, увести її ім'я в поле **Ім'я** і натиснути **Enter**. Водночас використовувати це ім'я клітинки можна буде в усій електронній книзі. *Слід пам'ятати*, що в імені клітинки не можна використовувати пропуски, воно не може розпочинатися із цифри, не може збігатися з адресами клітинок і не може бути клітинок з однаковими іменами.

Для отримання списку всіх імен клітинок електронної таблиці можна виконати ***Формули*** ⇒ ***Визначені імена*** ⇒ ***Диспетчер імен***.

Деяка сукупність клітинок аркуша електронної книги утворює **діапазон клітинок**, який також має адресу. Адреса діапазону клітинок задається адресами двох клітинок, розташованих у його протилежних кутах і розділених двокрапкою (наприклад, **B2:D2**, **C3:D7**). Діапазону, як і окремій клітинці, можна надати ім'я.

Увесь рядок та ввесь стовпець електронної таблиці також є діапазонами клітинок. Наприклад, адресою діапазону клітинок, до якого входять усі клітинки десятого рядка, є **10:10**, а адресою діапазону

клітинок, до якого входять усі клітинки стовпця **B**, є **B:B**. Відповідно **6:8** – це адреса діапазону клітинок, що включає всі клітинки рядків з номерами **6**, **7**, **8**, а **H:L** – адреса діапазону клітинок, до якого входять усі клітинки стовпців **H**, **I**, **J**, **K**, **L**.

На аркуші, крім електронної таблиці, можуть розміщуватися й інші об'єкти: діаграми, малюнки тощо (мал. 6.1). *Звертаємо увагу*, що ці об'єкти не розміщуються у клітинках електронної таблиці, а знаходяться на аркуші. Деякі аркуші можуть містити тільки діаграми, без електронних таблиць. Такі аркуші за замовчуванням мають імена **Діаграма1**, **Діаграма2** і т. д.

У таблиці 6.1 наведено перелік основних об'єктів табличного процесора та їх властивості.

Таблиця 6.1

Властивості об'єктів табличного процесора

Об'єкт	*Властивості об'єкта*
Електронна книга	Ім'я, кількість аркушів, порядок розташування аркушів та ін.
Аркуш	Ім'я, кількість розміщених об'єктів та їх вид, колір ярлика, орієнтація аркуша, колір тла, наявність сітки та ін.
Електронна таблиця	Загальна кількість рядків, стовпців і клітинок; кількість рядків, стовпців і клітинок, що містять дані, та ін.
Рядок	Номер, висота, кількість заповнених даними клітинок та ін.
Стовпець	Номер, ширина, кількість заповнених даними клітинок та ін.
Клітинка	Адреса / ім'я, уміст, тип даних у клітинці, межі, заливка, шрифт символів та ін.
Діапазон клітинок	Адреса / ім'я, кількість клітинок та ін.
Діаграма	Тип, вид, назва, розмір області діаграми, колір заливки та ін.

Значення властивостей об'єктів електронної таблиці можна змінювати командами меню, кнопками **Стрічки**, командами контекстного меню.

Цікаві факти з історії

Перший табличний процесор створили в 1979 році студент Гарвардського університету (США) **Ден Бріклін** і його товариш – програміст **Боб Френкстон** (мал. 6.2). Він був розроблений для комп'ютера **Apple II** та отримав назву **VisiCalc** (англ. ***Visible Calculator*** – наочний калькулятор). Електронна таблиця мала 5 стовпців і 20 рядків, а програма виконувала 4 арифметичні дії. Переобчислення формул у 100 клітинках таблиці після змінення даних відбувалося за 20 с.

Після **VisiCalc** з'явились аналогічні програми від інших розробників і для інших моделей комп'ютерів, такі як **SuperCalc**, **Lotus 1-2-3**, **Multiplan** та інші.

Перша версія табличного процесора **Excel** з'явилася в 1985 році. Вона також була створена для комп'ютера **Apple**. Його розробники – американські програмісти **Дуг Кландер** і **Філіп Флоренце**.

Серед сучасних найбільш поширених табличних процесорів можна назвати **Microsoft Office Excel**, **LibreOffice Calc**, **Google Таблиці**.



Мал. 6.2. Вікно першого табличного процесора **VisiCalc** та його автори **Ден Бріклін** (праворуч) і **Боб Френкстон**

Для тих, хто хоче знати більше

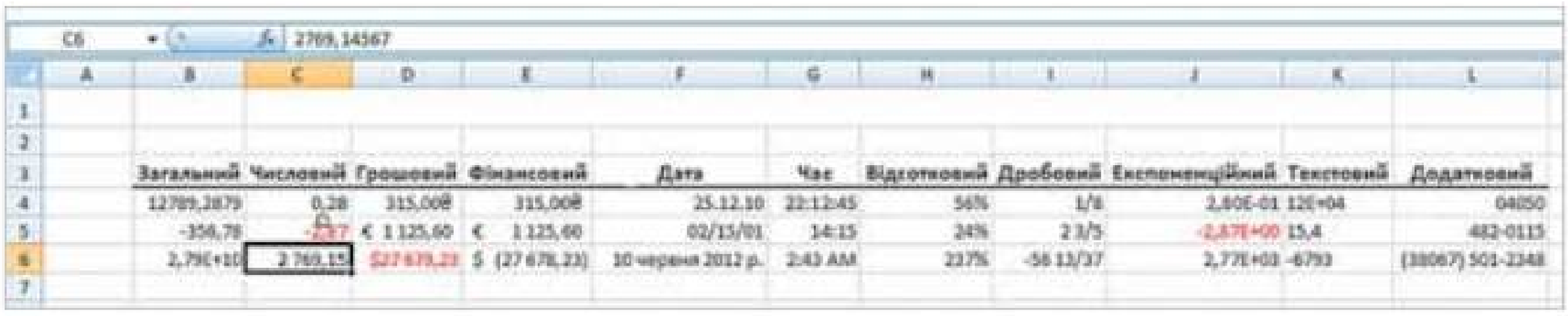
Стовпці електронної таблиці можуть також нумеруватися натуральними числами від 1 до 16 384. У цьому разі адреси клітинок записуються, наприклад, так: **R1C1**, **R5C2**, **R17C4**, **R27C26**, – де після літери **R** (англ. ***Row*** – рядок) указується номер рядка, а після **C** (англ. ***Column*** – стовпець) – номер стовпця. Тобто адресу **R5C2** потрібно розуміти так: клітинка на перетині **п'ятого рядка** і **другого стовпця**. Для змінення способу нумерації стовпців потрібно виконати ***Файл*** ⇒ ***Параметри*** ⇒ ***Формули*** та встановити позначку прапорця **Стиль посилань R1C1**.

ФОРМАТИ ЧИСЛОВИХ ДАНИХ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ

Пригадайте

● Які операції форматування можна виконати над даними у клітинках електронної таблиці? ● У яких форматах може бути подано числові дані в електронній таблиці? ● Що таке *стандартний вигляд числа*?

В 6-му класі ви вже використовували для обчислення в табличному процесорі *числовий* формат подання даних. Але в табличному процесорі **Excel** числові дані можуть бути подані й у інших форматах (мал. 6.3)

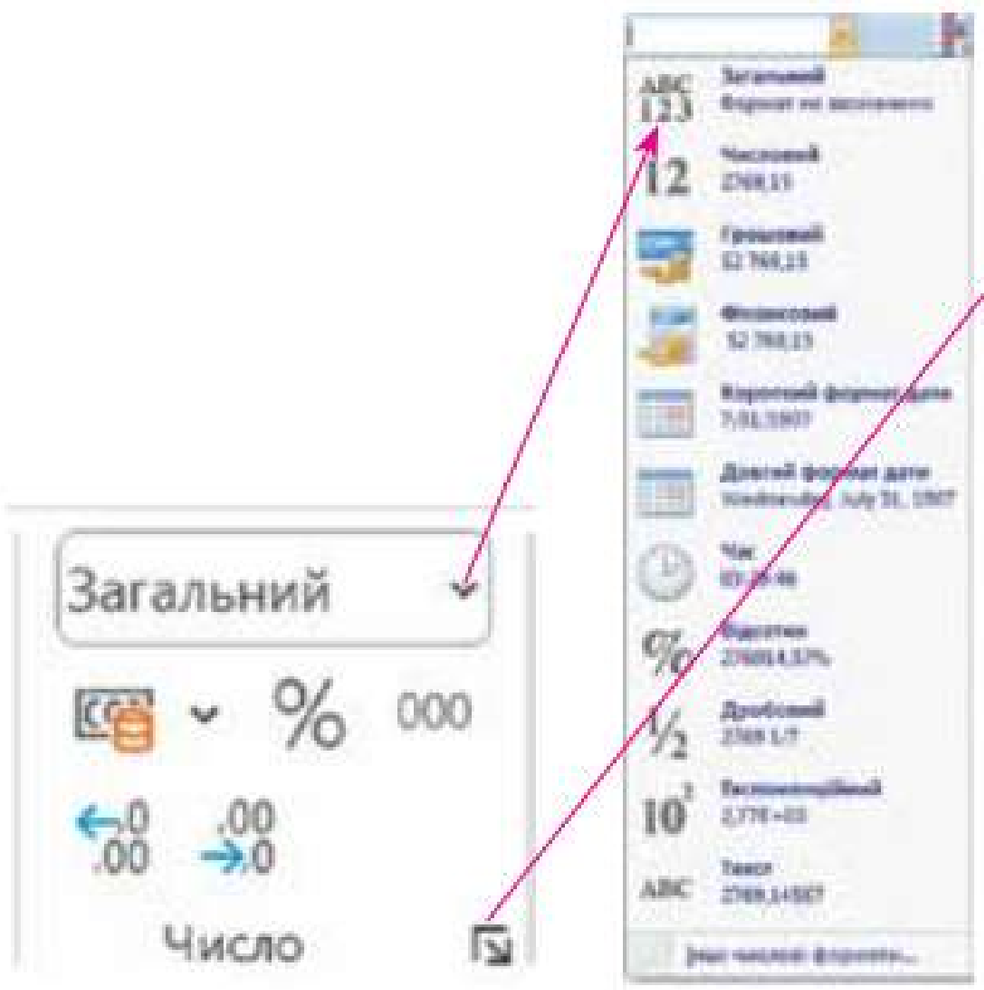


Мал. 6.3. Приклади числових форматів даних

Нагадаємо, що змінення формату подання даних не змінює значення цих даних у пам'яті комп'ютера, а лише встановлює певний вигляд їх подання у клітинці. Реальне значення даних можна побачити в **Рядку формул**, зробивши відповідну клітинку поточною.

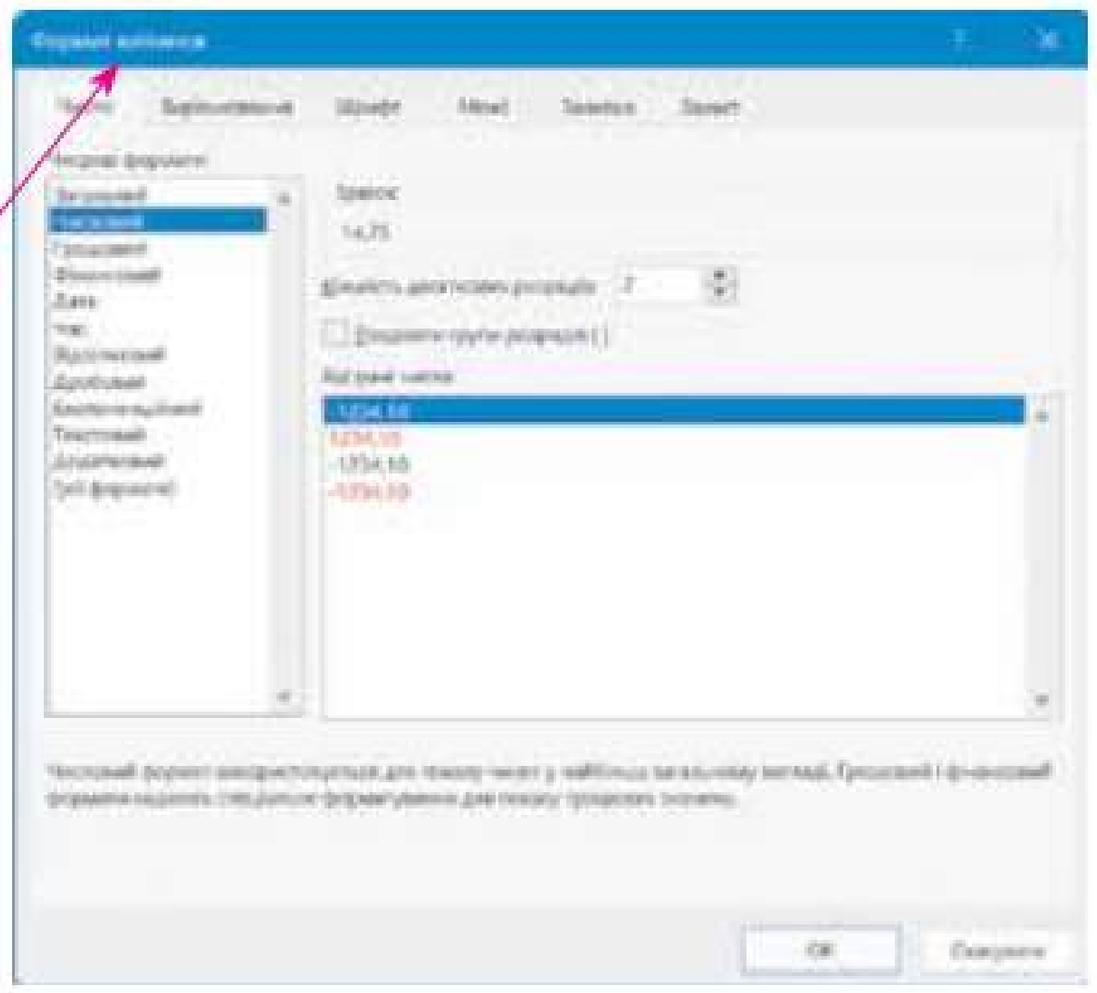
Установлення формату даних для поточної клітинки або для виділеного діапазону клітинок здійснюється з використанням елементів керування групи **Число** на вкладці **Основне** (мал. 6.4), списку поля **Числовий формат** (мал. 6.5), а також діалогового вікна **Формат клітинок** (мал. 6.6). Стрілками на малюнках указано спосіб відкриття зазначених елементів керування.

Формат **Загальний** 123 є форматом даних у клітинках за замовчуванням. Він використовується для подання чисел здебільшого так, як їх було введено (клітинки **B4:B6** на мал. 6.3). Під час уведення певних даних загальний формат може автоматично змінитися на інший.



Мал. 6.4. Група **Число**

Мал. 6.5. Список поля **Числовий формат**



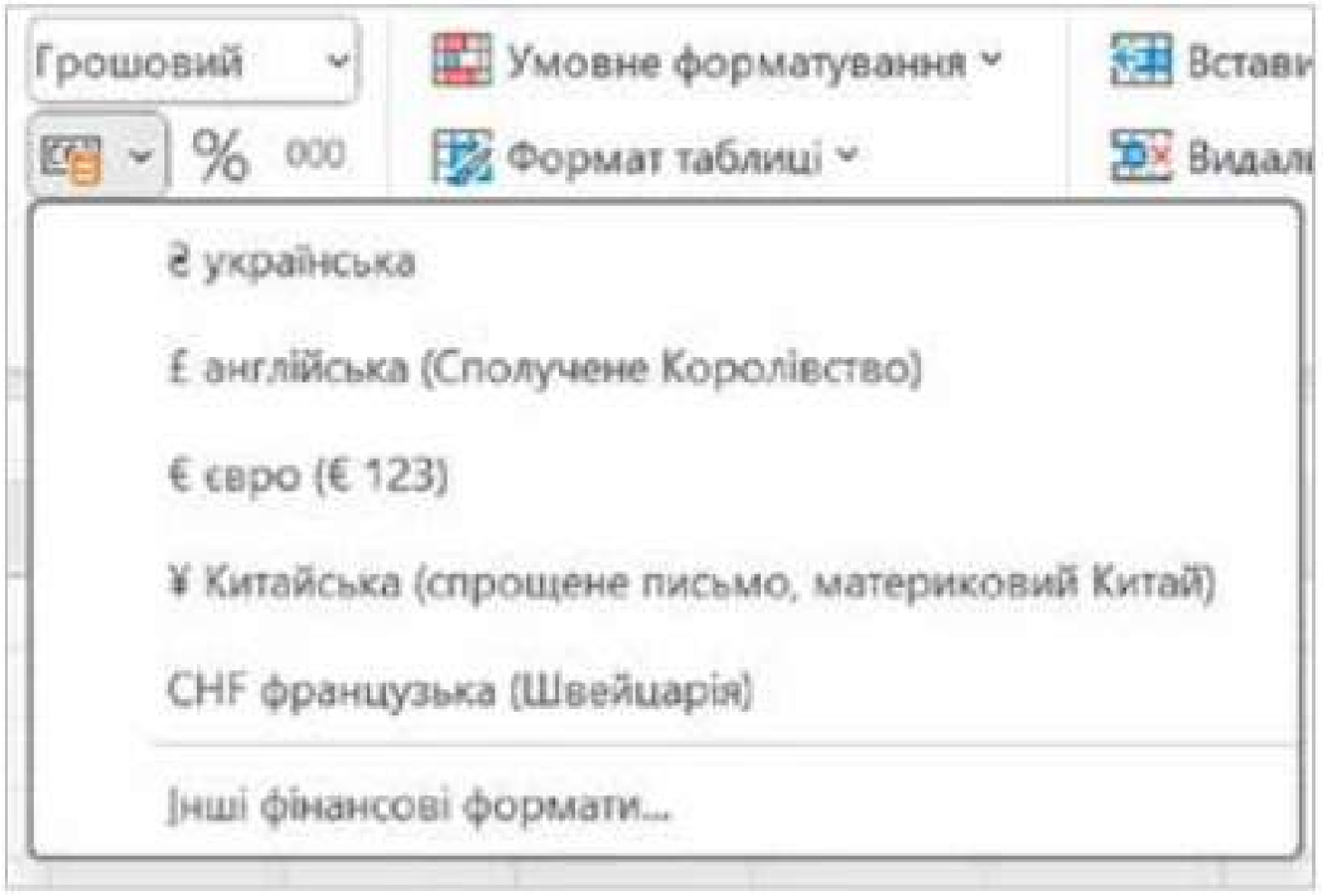
Мал. 6.6. Вікно **Формат клітинок**

ЧИСЛОВИЙ ФОРМАТ

Формат **Числовий** 12 (клітинки **C4:C6** на мал. 6.3) використовується для подання чисел у вигляді десяткового дробу із заданою кількістю десяткових розрядів. Ця кількість установлюється на лічильнику **Кількість десяткових розрядів** діалогового вікна **Формат клітинок** (мал. 6.6) і визначає, до якого десяткового розряду буде округлено число. Якщо число у клітинці має менше цифр після коми, ніж передбачено форматом, то під час виведення на екран воно буде доповнено нулями праворуч. Розрядність числа можна змінити, вибравши кнопки **Збільшити розрядність** ←,0 ,00 чи **Зменшити розрядність** ,00 →,0 у групі **Число** (мал. 6.4) на **Стрічці**.

Також можна встановити розділювач груп розрядів (класів) у вигляді пропуску між групами по три цифри у цілій частині числа, установивши прапорець **Розділяти групи розрядів** у діалоговому вікні **Формат клітинок** (мал. 6.6). Те саме відбудеться і за вибору кнопки **Формат з роздільниками** 000 у групі **Число** (мал. 6.4) на **Стрічці**.

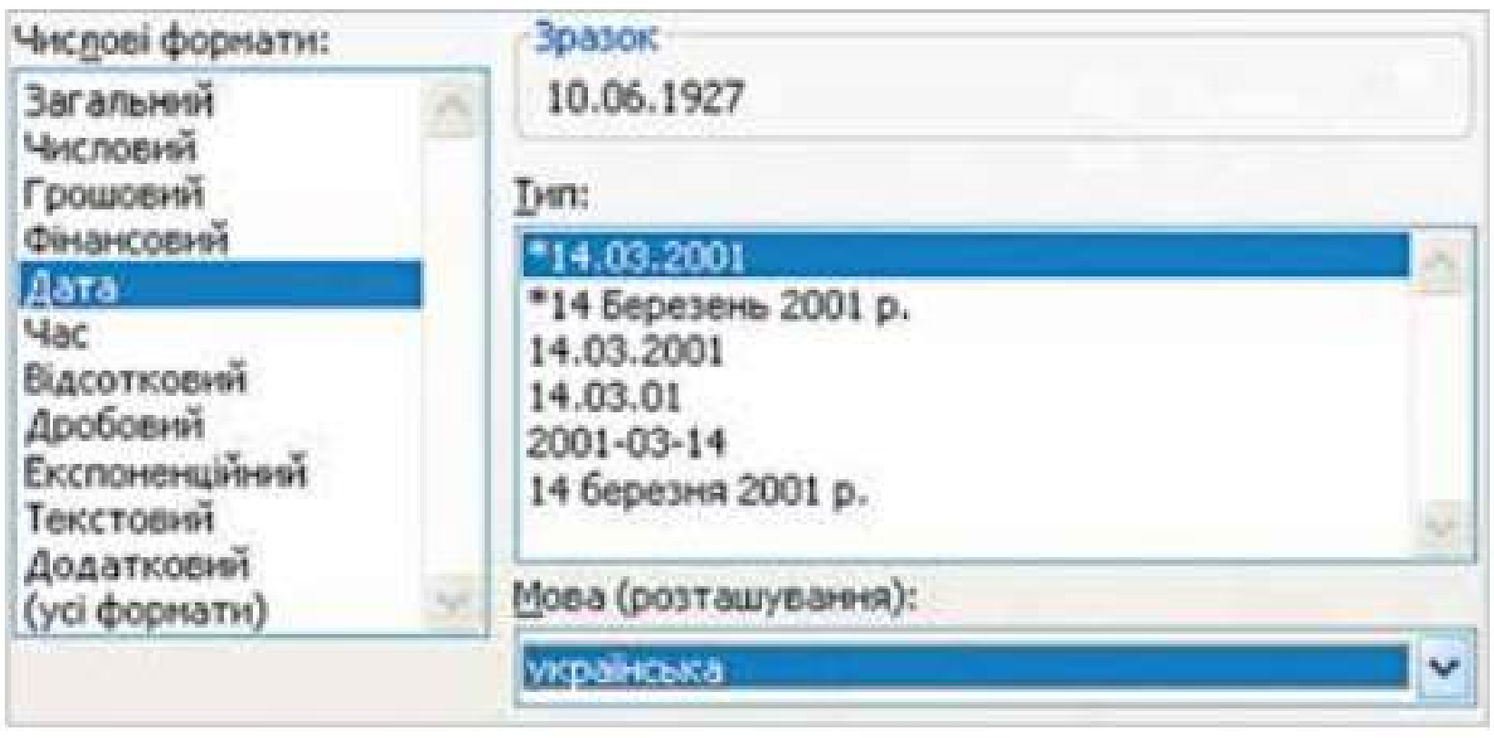
Формат **Грошовий** (клітинки **D4:D6** на мал. **6.3**) використовується для встановлення значень тих самих властивостей, що і для формату **Числовий**, з додаванням до числа позначення грошової одиниці, яке вибирається зі списку **Позначення** вікна **Формат клітинок**. Розділення груп розрядів (класів) у числі встановлюється автоматично. Вибір грошової одиниці можна зробити й у списку кнопки **Фінансовий формат чисел** у групі **Число** (мал. **6.7**) на **Стрічці**.



Мал. 6.7. Список кнопки **Фінансовий формат чисел**

Формат **Фінансовий** (клітинки **E4:E6** на мал. **6.3**) відрізняється від формату **Грошовий** тим, що значення у клітинках вирівнюються і за десятковою комою, і за знаком грошової одиниці.

Формат **Дата** (клітинки **F4:F6** на мал. **6.3**) використовується для подання числових даних у вигляді дат певного типу, які представлено на малюнку **6.8**. Дати зберігаються в **Excel** як натуральні числа. Відлік у **Excel** починається з **01.01.1900**, і цій даті відповідає число **1**. Кожній наступній даті відповідає наступне натуральне число:



Мал. 6.8. Формат **Дата**

02.01.1900 – 2,
03.01.1900 – 3, ... ,
06.06.2006 – 38 874, ... ,
01.09.2010 – 40 422.

Таке подання дат дає змогу виконувати обчислення над ними. Так, кількість днів між двома датами визначається різницею чисел, що відповідають цим датам. Наприклад, різниця 01.09.2025 – 01.01.2025 буде обчислюватися так: 45 901 – 45 658 = 243.

Формат **Час** (клітинки **G4:G6** на мал. 6.3) використовується для відображення даних як часу з певним виглядом подання. Час у **Excel** зберігається у вигляді десяткового дробу, який визначає частину доби. Наприклад,

24:00	24 год = 1
01:00	1 год = 1/24 = 0,041666666666667
00:01:00	1 хв = 1/(24*60) = 0,00069444444444444
00:00:01	1 с = 1/(24*3600) = 0,000011574074074074 1
13:35:00	відповідає число 0,565972222222222

У форматі **Відсотковий** % (клітинки **H4:H6** на мал. 6.3) дані подаються у вигляді числа, яке отримано автоматичним множенням умісту клітинки на 100, зі знаком % у кінці.

Формат **Дробовий** ½ використовують для подання числа у вигляді звичайного дробу. Наприклад, число **0,125** у клітинці **I4** подано як **1/8**, а число **2,6** у клітинці **I5** – дробовим числом **2 3/5** (мал. 6.3). Слід ураховувати, що більшість чисел у цьому форматі буде подано наближено. Наприклад, число 0,65 у форматі звичайного дробу буде подано як 2/3 (хоча 2/3 = 0,666666...).

Формат **Експоненційний** 10^2 використовують для подання числа у стандартному вигляді. Так, запис 2,80E-01 відповідає числу $2{,}80 \cdot 10^{-1}$, а 2,77E+03 відповідає числу $2{,}77 \cdot 10^{3}$ (клітинки **J4:J6** на мал. 6.3).

Стандартним виглядом додатного числа називають його представлення у вигляді $a \cdot 10^m$, де $1 \le a \le 10$, а m – натуральне число. Число m називають *порядком числа* a, при цьому a називають *значущою частиною числа*.

Формат **Текстовий** ab використовують для подання чисел у клітинках як текст (клітинки **K4:K6** на мал. 6.3). Значення відображаються так само, як і вводяться.

ВІДНОСНІ, АБСОЛЮТНІ ТА МІШАНІ ПОСИЛАННЯ



Пригадайте

● Як у формулах у табличному процесорі записують адреси клітинки? Для чого їх використовують? ● Що таке *модифікація формул*? Коли вона відбувається? За якими правилами здійснюється?

Як ви знаєте, в електронних таблицях для обчислень у формулах можна використовувати адреси клітинок чи діапазонів клітинок.

У такому разі під час змінення даних у цих клітинках відбувається автоматичне переобчислення значень за всіма формулами, які містять такі адреси. Адреси клітинок, які використовують у формулах, також називають *посиланнями на клітинки*.

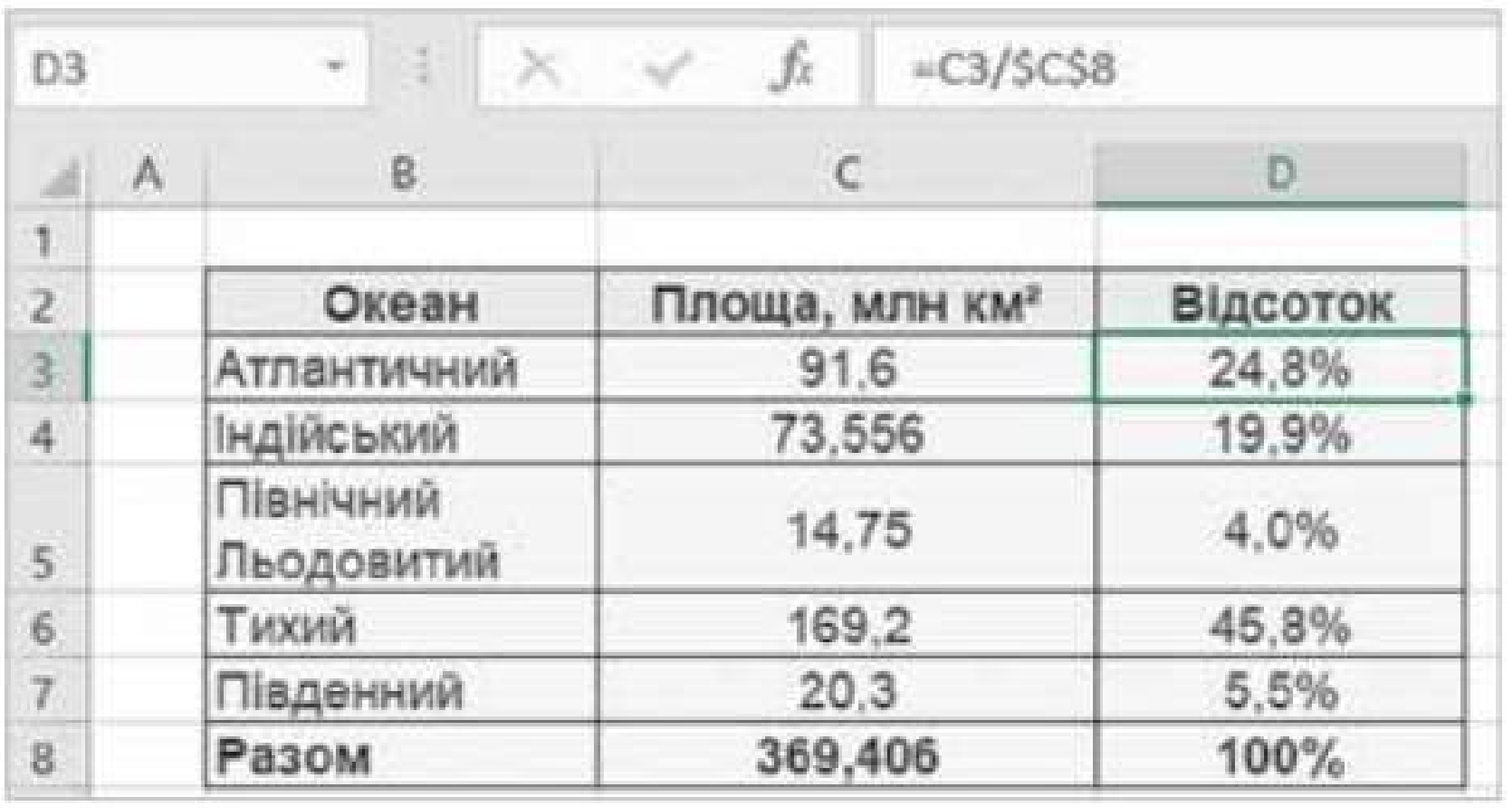
Під час копіювання формул посилання в них автоматично змінюються – **модифікуються**. Це відбувається за таким правилом: номери стовпців (рядків) у посиланнях змінюються на різницю номерів кінцевого і початкового стовпців (рядків). Під час переміщення формули не модифікуються.

Але інколи потрібно, щоб під час копіювання формул певні посилання не модифікувалися, тобто залишались незмінними.

Посилання, яке модифікується під час копіювання формули, називається **відносним**. Посилання, яке не модифікується під час копіювання формули, називається **абсолютним**. Щоб посилання було абсолютним, потрібно перед номером стовпця та перед номером рядка ввести символ **$**. Наприклад, посилання **B10** та **A1:A7** є відносними, а посилання **$B$10** та **$A$1:$A$7** – абсолютними.

Розглянемо приклад, який пояснює доцільність використання абсолютних посилань.

Нехай у клітинках діапазону **C3:C7** (мал. 6.9) міститься площа п'яти океанів, а у клітинці **C8** обчислено їхню загальну площу. У клітинках діапазону **D3:D7** потрібно обчислити, який відсоток загальної площі займає кожен океан.



Мал. 6.9. Використання відносних і абсолютних посилань

Для обчислення цих відсотків у клітинках діапазону **D3:D7** повинні розміщуватися такі формули з урахуванням установлення відсоткового формату чисел у клітинках:

Клітинка	*Формула у клітинці*
D3	=C3/C8
D4	=C4/C8
D5	=C5/C8
D6	=C6/C8
D7	=C7/C8

Якщо проаналізувати всі ці формули, то можна зробити висновок, що посилання в чисельнику змінюються, а посилання у знаменнику – ні. Якщо ми введемо у клітинку **D3** формулу **=C3/C8** і скопіюємо її у клітинки діапазону **D4:D7**, то всі посилання у формулі модифікуються. Тобто під час копіювання цієї формули отримаємо у відповідних клітинках такі формули:

Клітинка	*Формула у клітинці*
D3	=C3/C8
D4	=C4/C9
D5	=C5/C10
D6	=C6/C11
D7	=C7/C12

Щоб уникнути такої модифікації, потрібно посилання на клітинку **C8** у формулі **=C3/C8** зробити абсолютним, увівши у клітинку **D3** формулу **=C3/$C$8**. Посилання на клітинку **C3** залишається у формулі відносним і буде змінюватися під час копіювання відповідним чином.

Якщо в посиланні символ **$** додано тільки перед номером стовпця або тільки перед номером рядка, наприклад **$B10** або **B$10**, то під час копіювання такі посилання модифікуються частково: змінюється лише номер рядка або лише номер стовпця, біля якого не стоїть символ **$**. Такі посилання називаються **мішаними**.

Змінити у формулі вид посилання (абсолютне, відносне чи мішане) можна під час його введення або редагування послідовним натисненням клавіші **F4**. Тобто якщо у клітинку, наприклад, увести відносне посилання **C8**, то після першого натиснення клавіші **F4** посилання перетвориться на абсолютне посилання **$C$8**, після другого натиснення **F4** це посилання перетвориться на мішане посилання **C$8**, після третього натискання – на мішане посилання **$C8**, а після четвертого – знову стане відносним **C8**.

Якщо у формулі замість посилання на клітинку використати її ім'я, то під час копіювання цієї формули таке посилання модифікуватися не буде. Отже, **ім'я клітинки у формулі є абсолютним посиланням на цю клітинку**. Тому в розглянутому вище прикладі можна було клітинці **C8** надати ім'я, наприклад **Площа_разом**, і тоді формула у клітинці **D3** виглядала б так: **=C3/Площа_разом** і під час копіювання посилання **Площа_разом** не змінюватиметься.

Приклади абсолютних, відносних і мішаних посилань наведено в таблиці 6.2.

Таблиця 6.2

Типи посилань

Посилання	*Тип*	*Пояснення*
C3	Відносне	Під час копіювання формули може змінюватись і номер стовпця, і номер рядка

Продовження таблиці 6.2

Посилання	Тип	Пояснення
$A$1	Абсолютне	Під час копіювання формули номер стовпця і номер рядка незмінні
курс_валюти	Абсолютне	Під час копіювання формули номер стовпця і номер рядка незмінні
C$1	Мішане	Під час копіювання формули номер стовпця може змінюватись, а номер рядка незмінний
$A3	Мішане	Під час копіювання формули номер стовпця незмінний, а номер рядка може змінюватися



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Ee2cJpFj* або QR-кодом.



Найважливіше в цьому пункті

Основними об'єктами табличного процесора **Excel** є **електронна книга**, **аркуш**, **електронна таблиця**, **рядок**, **стовпець**, **клітинка**, **діапазон клітинок**.

Кожна клітинка електронної таблиці має **адресу**, яка складається з номера стовпця та рядка, на перетині яких вона розміщена. Деяка сукупність клітинок електронної таблиці утворює **діапазон клітинок**. Адреса діапазону клітинок задається адресами двох клітинок, розташованих у його протилежних кутах і розділених двокрапкою. На аркуші електронної книги можуть розміщуватися електронні таблиці, діаграми, малюнки та інші об'єкти. У клітинках електронної таблиці можуть зберігатися числа, тексти та формули.

Клітинці та діапазону клітинок можна надати **ім'я**. Його зручніше використовувати у формулах, оскільки ім'я клітинки не змінюється у формулах під час їх копіювання. Якщо клітинка має ім'я та її зробити поточною, то в полі **Ім'я** буде відображатися не адреса клітинки, а її ім'я.

Числові дані в **Excel** можна подати в кількох форматах: **Загальний**, **Числовий**, **Грошовий**, **Дата**, **Відсотковий**, **Дробовий**, **Текстовий** та інші. Форматування не змінює дані у клітинці таблиці, а лише встановлює певний вигляд їх подання. Реальне значення даних можна побачити в **Рядку формул**, зробивши відповідну клітинку поточною.

Установлення формату даних для поточної клітинки або для виділеного діапазону клітинок здійснюється з використанням елементів керування групи **Число** на вкладці **Основне**, списку поля **Числовий формат**, а також діалогового вікна **Формат клітинок**.

У формулах електронної таблиці можна використовувати різні види посилань клітинки: **відносні** (модифікуються під час копіювання формули), **абсолютні** (не модифікуються під час копіювання формули), **мішані** (модифікуються частково).

Щоб посилання було абсолютним, потрібно перед номером стовпця та номером рядка ввести символ **$**. У мішаних посиланнях символ **$** уводиться або перед номером стовпця, або перед номером рядка.

Під час копіювання формул адреси клітинок у них автоматично **модифікуються**. Це відбувається за таким правилом: номери стовпців (рядків) у посиланнях змінюються на різницю номерів кінцевого і початкового стовпців (рядків). Під час переміщення формули не модифікуються.

Якщо у формулі замість посилання на клітинку використати її **ім'я**, то під час копіювання цієї формули це посилання модифікуватися не буде, воно вважається абсолютним.

Дайте відповіді на запитання

1. Які об'єкти електронної таблиці ви знаєте? Назвіть їх властивості та наведіть приклади значень.

2. Які дії можна виконувати над об'єктами електронної таблиці?

3. Які формати числових даних можна встановити? Дайте їм стислу характеристику.

4. Які типи посилань (адрес) можна використовувати у формулах в електронних таблицях? Як вони записуються?

5. Чим відрізняється використання різних типів посилань на клітинки у формулах?

6. До якого типу посилань належить ім'я клітинки, яке надає користувач?

Обговоріть і зробіть висновки

1. Який тип посилання на клітинку **C5** потрібно використати у формулі, щоб під час її копіювання по рядку це посилання модифікувалось, а по стовпцю – ні? Не модифікувалося під час копіювання в будь-якому напрямку?

2. Чи однаковий результат буде під час копіювання формул, у яких використано мішані та абсолютні посилання?

3. З'ясуйте, які задачі з різних шкільних предметів під час їх розв'язування в **Excel** потребували б використання абсолютних посилань. Що спільного у знайдених прикладах?

4. Що спільного та відмінного у використанні абсолютних посилань та імен клітинок у формулах?

Виконайте завдання

1. У клітинці **E3** розміщено формулу **=$B$3+C4**. Запишіть, як виглядатиме ця формула, якщо її скопіювати у клітинку:

а) **E7**; б) **I8**;
в) **C3**; г) **A2**.

2. У клітинці **K8** розміщено формулу **=E5+$C3**. Запишіть, як виглядатиме ця формула, якщо її скопіювати у клітинку:

а) **C9**; б) **B4**;
в) **E9**; г) **D8**;
д) **I2**; е) **F5**.

3. Запишіть значення у клітинках **C2** і **D2**, якщо в них скопіювати формулу з клітинки **B2** (мал. 6.10).

	A	B	C	D
1	1	10	20	30
2		=B1+A$1		
3		100		

Мал. 6.10

4. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\пункт 6.1\зразок 6.1.4.xlsx**).

1. На аркуші **Аркуш1** обчисліть, який шлях пройшов кожний турист за день і залишок шляху, якщо весь маршрут змагань складає 400 км (це значення записано у клітинці **C10**). Які типи даних використано в таблиці?
2. На аркуші **Аркуш2** обчисліть ціни товарів у гривнях (курс євро розміщено у клітинці **B3**). Які типи даних використано в таблиці?
3. На аркуші **Аркуш3** обчисліть кількість учнів у кожному 8-му класі та на всій паралелі 8-х класів, використавши для клітинки **D10** ім'я **Учні_8**. Використайте це ім'я у формулі для обчислення відсотків кількості учнів кожного 8-го класу відносно загальної кількості учнів 8-х класів.

Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.4.xlsx**.

5. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\пункт 6.1\зразок 6.1.5.xlsx**).

1. На аркуші **Аркуш1** обчисліть прибуток кожної фірми та всього виробництва за рік, а також відсоток вкладу кожної фірми в річний прибуток.
2. На аркуші **Аркуш2** обчисліть щомісячний розмір депозитного вкладу з капіталізацією відсотків. Які типи даних використано в таблиці?
3. На аркуші **Аркуш3** обчисліть кількість змінних носіїв різних видів, потрібних для збереження даних, що містяться в 1 г молекули ДНК, – 455 млрд Гб (за дослідженнями американських учених). Використайте для клітинки з указаним значенням ім'я **ДНК**. Відшукайте самостійно потрібні відомості про сучасні види змінних носіїв та їх ємності. Які типи даних ви використали в таблиці? Які типи посилань на клітинки таблиці ви використали під час проведення обчислень?

Капіталізація відсотків – збільшення основної суми вкладу на суму накопичених за певний проміжок часу відсотків.

Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.5.xlsx**.

6. Створіть за наведеним зразком (мал. 6.11) електронну таблицю для обчислення значень 10 перших членів числових послідовностей різних видів:

Тип послідовності	*Різниця / знаменник*	*Номер члена послідовності*									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Арифметична прогресія	2	1									
Геометрична прогресія	2	1									
Числа Фібоначчі		1	1								

Мал. 6.11

- Кожен наступний член послідовності, починаючи з другого, дорівнює попередньому, до якого додано одне й те саме число (*різницю* послідовності). Така послідовність чисел називається *арифметичною прогресією*.
- Кожен наступний член послідовності, починаючи з другого, дорівнює попередньому, помноженому на одне й те саме число, відмінне від 0 (*знаменник* послідовності). Така послідовність чисел називається *геометричною прогресією*.
- Кожен наступний член послідовності, починаючи з третього, дорівнює сумі двох попередніх членів цієї послідовності. Така послідовність чисел називається *числа Фібоначчі*.

Цікаві факти з історії

Леонардо Фібоначчі, справжнє ім'я якого Леонардо Пізанський, був видатним математиком середньовічної Європи. Він народився приблизно 1170 року в Пізі, Італія. Вивчав математику в арабських учителів, познайомив європейців з арабською системою числення, яка тоді була невідомою у Європі. У 1202 році Леонардо Фібоначчі опублікував книгу «Liber Abaci» («Книга абака»), де пояснив нову систему числення на основі десяткової системи. Це значно полегшило обчислення і стало справжньою революцією в математиці того часу.

Виконайте обчислення. Які формули ви для цього використовували? Які типи посилань у них використовувались? Які типи даних використано в таблиці? Поясніть їх вибір. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.6.**

7. Створіть таблицю для визначення вартості різних українських грошових банкнот у доларах, євро та фунтах стерлінгів. Курси валют з'ясуйте на сайті Національного банку України. Передбачте зберігання значень курсів долара, євро та фунта стерлінгів в окремих клітинках. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.7.**

8. Створіть електронну таблицю для обчислення елементів таблиці множення (мал. 6.12), використавши одну формулу з мішаними посиланнями. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.8.** Надрукуйте створену таблицю множення.

	A	B	C	D	E	F	G	H	I	J
1	Таблиця множення									
2		1	2	3	4	5	6	7	8	9
3	1	1	2	3	4	5	6	7	8	9
4	2	2	4	6	8	10	12	14	16	18
5	3	3	6	9	12	15	18	21	24	27
6	4	4	8	12	16	20	24	28	32	36
7	5	5	10	15	20	25	30	35	40	45
8	6	6	12	18	24	30	36	42	48	54
9	7	7	14	21	28	35	42	49	56	63
10	8	8	16	24	32	40	48	56	64	72
11	9	9	18	27	36	45	54	63	72	81

Мал 6.12

9. Заповніть електронну таблицю натуральними числами від 1 до 90 по спіралі (мал. 6.13). Поміркуйте, яким чином можна автоматизувати цей процес. Які формули для цього ви використаєте? Який тип адресації буде присутній у формулах і чому? Збережіть книгу у вашій папці у файлі з іменем **завдання 6.1.9.** Надрукуйте створену таблицю.

1	2	3	4	5	6	7	8	9
34	35	36	37	38	39	40	41	10
33	60	61	62	63	64	65	42	11
32	59	78	79	80	81	66	43	12
31	58	77	88	89	82	67	44	13
30	57	76	87	90	83	68	45	14
29	56	75	86	85	84	69	46	15
28	55	74	73	72	71	70	47	16
27	54	53	52	51	50	49	48	17
26	25	24	23	22	21	20	19	18

Мал. 6.13

Готуємось до вивчення нового матеріалу

1. Що таке *функція*? Які математичні функції ви знаєте? Наведіть приклади.
2. Які вбудовані функції **Excel** ви використовували? Наведіть приклади.
3. Які правила запису функцій у формулах?

6.2. МАТЕМАТИЧНІ, СТАТИСТИЧНІ ТА ЛОГІЧНІ ФУНКЦІЇ

У цьому пункті йтиметься про:

- функції в табличному процесорі та їх вставлення у формули;
- використання математичних, статистичних і логічних функцій для обчислень.

ФУНКЦІЇ В ТАБЛИЧНОМУ ПРОЦЕСОРІ

Пригадайте

● Що таке *функція* в математиці? Що таке *аргумент функції*? Що таке *результат функції*? ● Для чого використовують у табличному процесорі вбудовані функції **SUM**, **MIN**, **MAX**, **AVERAGE**?

Ви вже знаєте, що у формулах у табличному процесорі **Excel** можна використовувати **вбудовані функції**. Табличний процесор **Excel** має бібліотеку таких функцій, до якої входить понад 500 різноманітних функцій. Для зручності їх розподілено за **категоріями**: *математичні*, *статистичні*, *текстові*, *логічні*, *фінансові* та інші.

У 6-му класі ви вже вивчали вбудовані функції в табличному процесорі та використовували такі з них: **SUM** – для знаходження суми

чисел, **AVERAGE** – для знаходження середнього арифметичного, **MAX** та **MIN** – для визначення максимального та мінімального значення серед указаних чисел. Перша з них належить до категорії математичних функцій, інші – до категорії статистичних.

Статистика (англ. *statistics*) – це галузь знань, яка вивчає методи та принципи, за якими проводиться збирання, аналіз, порівняння, подання та інтерпретація числових даних.

Функція в Excel має ім'я, аргументи та значення. Під час запису функції у формулі спочатку вказується її ім'я, а потім у дужках **аргументи** функції. Якщо аргументів кілька, то вони відділяються крапкою з комою.

Так, у наведеній на малюнку 6.14 формулі **=SUM(B3:B13;K3:K13)** використано функцію з іменем **SUM**, аргументами якої є два діапазони клітинок **B3:B13** та **K3:K13**, а значенням є сума чисел з указаних діапазонів клітинок.



Мал. 6.14. Приклад запису функції **SUM**

Кількість аргументів функції та їх тип залежать від конкретної функції. Аргументами функції можуть бути числа, тексти (їх потрібно брати в подвійні лапки), посилання на клітинки та діапазони, вирази та інші функції.

В **Excel** є функції з різною кількістю аргументів: з одним чи кількома, без аргументів. Приклади функцій наведено в таблицях 6.3 та 6.4 цього пункту підручника.

Результат однієї функції можна використати як аргумент іншої функції. Такі функції називають **вкладеними**. Наприклад, подана на малюнку 6.15 функція **SQRT** використовує вкладені функції **MIN** і **MAX**. Можна використовувати до семи рівнів вкладення функцій.



Мал. 6.15. Вкладені функції

Призначення кожної функції, наявність аргументів та їх кількість, типи аргументів можна подивитися в **Довідці** або в коментарях у ході вставлення функції у формулу, як це продемонстровано на малюнках 6.16 та 6.18.

Слід пам'ятати, що під час копіювання формул з функціями аргументи будуть модифікуватися.

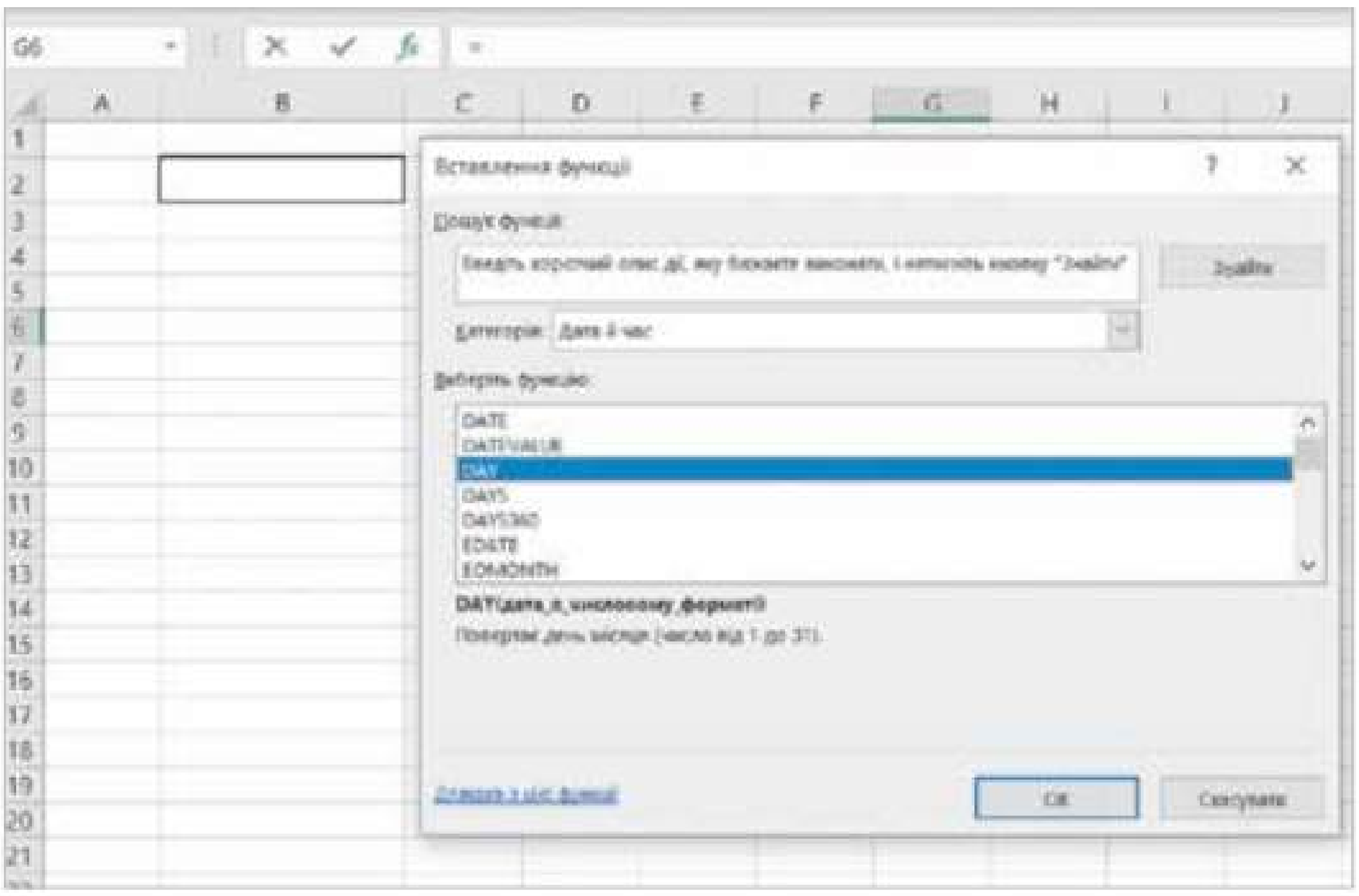
ВСТАВЛЕННЯ ФУНКЦІЙ У ФОРМУЛУ

Пригадайте

- Як уставити у формулу вбудовані функції **SUM**, **MIN**, **MAX**, **AVERAGE**?
- Що може бути аргументом функції в табличному процесорі?
- Як редагувати формулу в табличному процесорі?

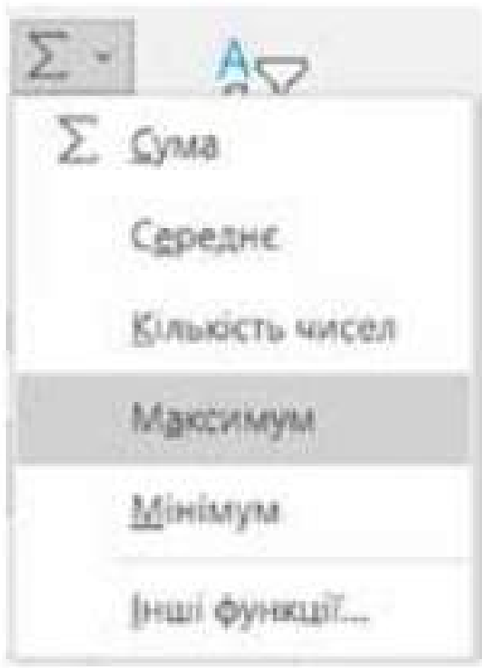
Уставити функцію у формулу можна кількома способами:

- Вибравши кнопку **Вставлення функції** fx ліворуч **Рядка формул** або виконавши ***Формули*** ⇒ ***Бібліотека функцій*** ⇒ ***Вставити функцію*** fx (мал. 6.16). Ці дії відкривають діалогове вікно **Вставлення функцій**, де всі функції розподілено за **Категоріями** та вміщено стислий опис призначення функцій.



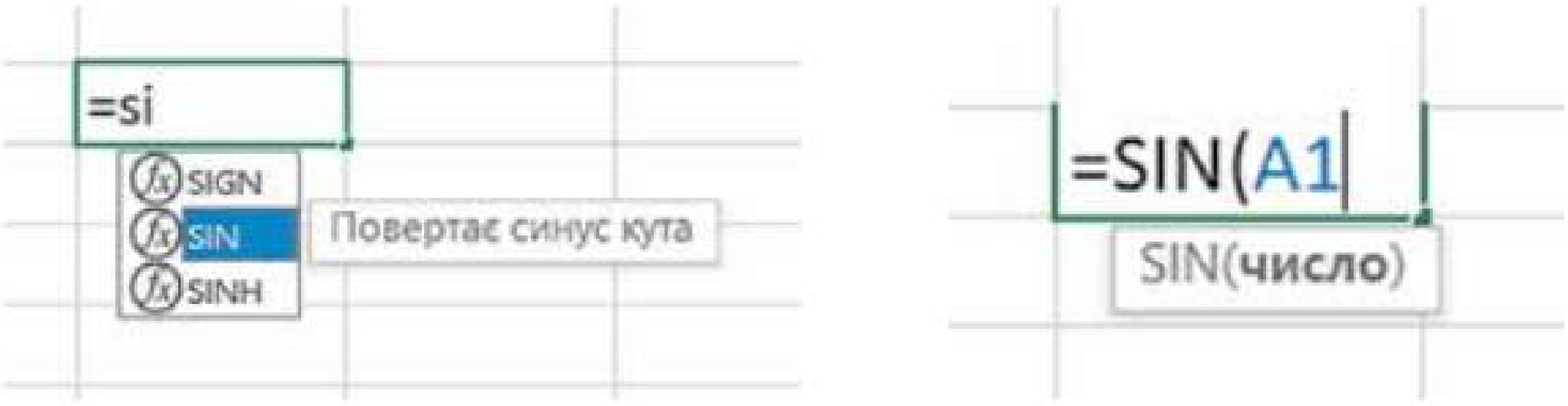
Мал. 6.16. Вставлення функції у формулу з використанням кнопки **Вставлення функцій**

- Вибравши зі списку **Автосума** Σ потрібну функцію **Сума** у групі **Редагування** вкладки **Основне** (мал. 6.17). Цим способом ви користувалися в попередніх класах під час обчислення суми та середньоарифметичних, мінімальних і максимальних значень у діапазоні клітинок. Команда **Інші функції** відкриває діалогове вікно **Вставлення функцій**, де можна вибрати будь-яку іншу функцію табличного процесора.
- Увівши функцію вручну безпосередньо у клітинку або в **Рядок формул** (мал. 6.18). Під час уведення з клавіатури перших літер імені функції поруч з клітинкою з'являються підказки, у яких послідовно демонструється список мож-



Мал. 6.17. Список кнопки **Автосума**

ливих функцій, зразок запису, порядок запису аргументів та їх типи. Вибравши потрібну функцію зі списку подвійним клацанням, далі можна вводити аргументи. Уводити з клавіатури імена функції та посилання на клітинки можна як малими, так і великими англійськими літерами. Цим способом зручніше користуватися під час уведення складних формул з кількома функціями.



Мал. 6.18. Введення функції у формулу в клітинку з клавіатури

Розглянемо детальніше алгоритм уставлення функції у формулу з використанням кнопки **Вставлення функції** fx. Наприклад, потрібно у клітинці **B1** обчислити добуток (функція **PRODUCT**) усіх чисел діапазону клітинок **A1:A10**.

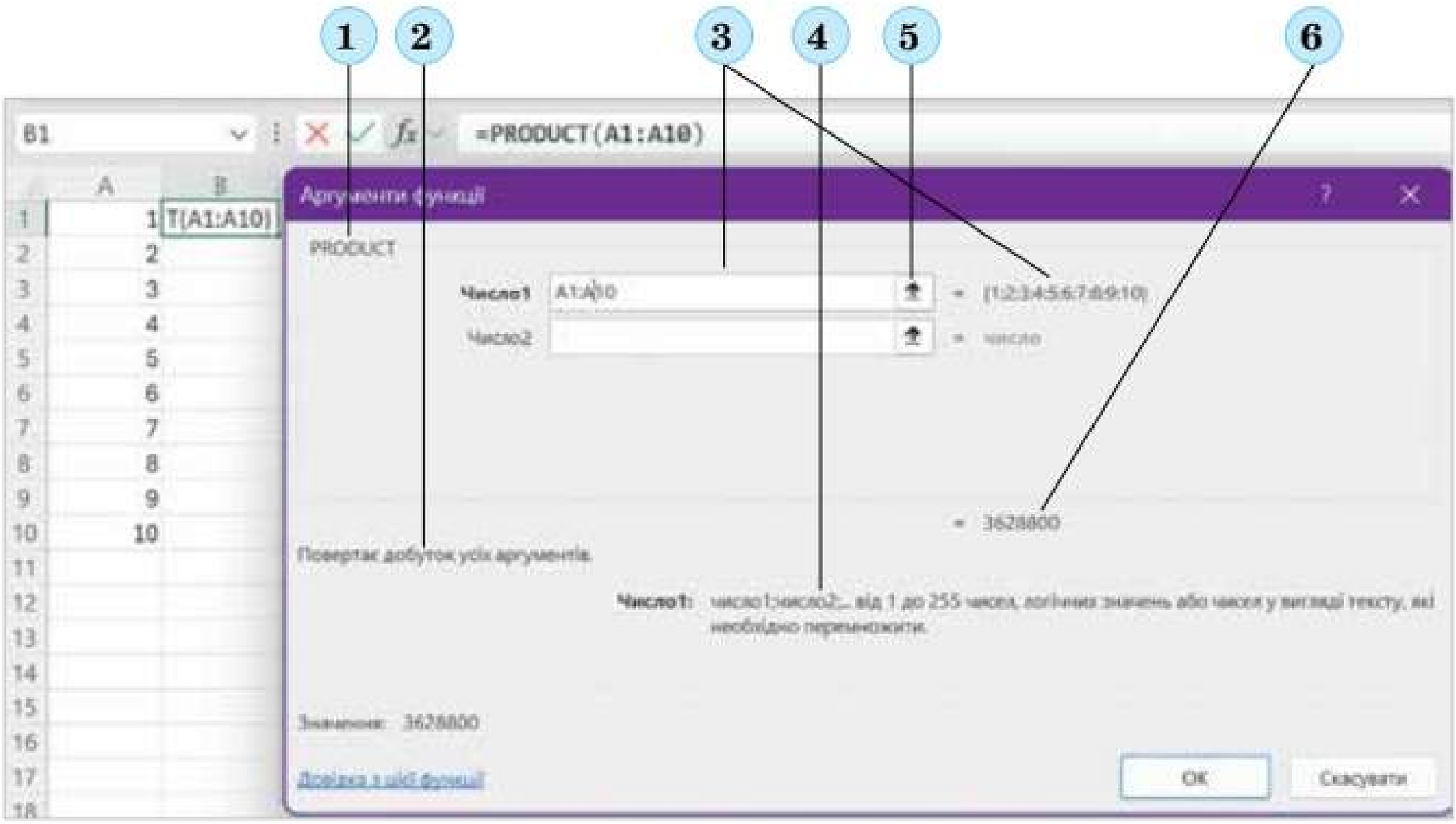
Для цього слід:

1. Вибрати клітинку електронної таблиці, у яку потрібно вставити формулу. Наприклад, **B1**.
2. Вибрати кнопку **Вставлення функції** fx ліворуч від **Рядка формул**.
3. Вибрати у вікні **Вставлення функції** у списку **Категорія** потрібну категорію функцій. Наприклад, **Математичні**.
4. Вибрати у списку потрібну функцію. Наприклад, **PRODUCT**.
5. Вибрати кнопку **ОК**.

   Після вибору виконання цих дій у поточну клітинку автоматично буде вставлено знак = (якщо в цій клітинці введення формули ще не розпочиналося), ім'я функції та пару круглих дужок, а також відкриється вікно **Аргументи функції** з полями для введення аргументів цієї функції.
6. Увести у вікні **Аргументи функції** в поле **Число1** і за потреби в інших полях аргументи функції – сталі значення, адреси клітинок і діапазонів (мал. 6.19). Наприклад, **A1:A10**.
7. Вибрати кнопку **ОК**.

   Після виконаних дій треба перевірити правильність уведення формули у **Рядку формул** та отриманий результат (у клітинці **B1**). І далі за потреби продовжити введення формули, якщо вона складається з кількох компонентів.

Вікно **Аргументи функції** (мал. 6.19) містить коментарі про призначення даної функції та її аргументи. Крім того, під час уведення аргументів праворуч від полів з'являються значення відповідних аргументів і нижче відображається поточний результат обчислення. На все це варто звертати увагу! Щоб детальніше дізнатися щодо використання функції, можна вибрати посилання **Довідка з цієї функції**.



Мал. 6.19. Введення даних у вікні **Аргументи функції**:
1 – ім'я функції; *2* – призначення функції; *3* – аргументи функції та їх значення;
4 – опис аргументів функції; *5* – кнопка **Розгорнути**; *6* – значення функції

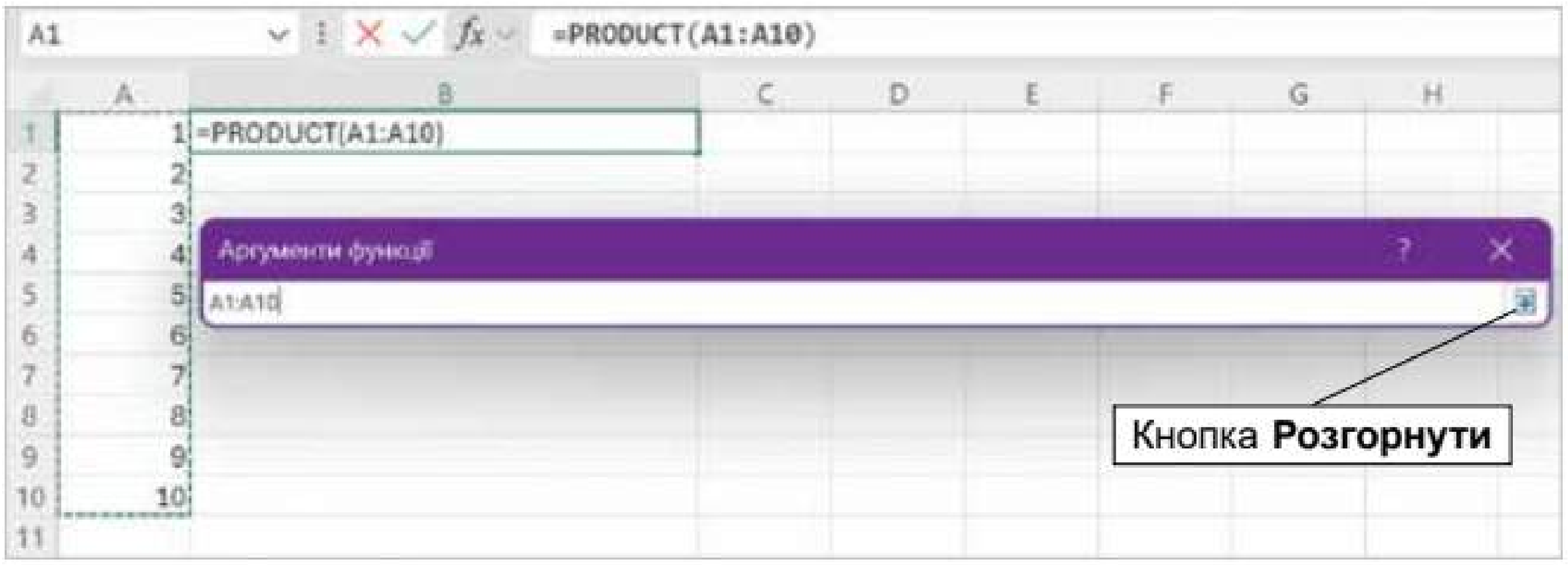
Якщо функція має фіксовану кількість аргументів, то вікно Аргументи функції одразу містить відповідну кількість полів для їх уведення. Якщо функція має змінну кількість аргументів, то у вікні спочатку з'являється кілька полів, а потім, у процесі введення аргументів, з'являються наступні поля.

Якщо аргументом є число або текст, то його можна вводити в поле з клавіатури або вставляти з Буфера обміну. Якщо аргументом є посилання на клітинки, то його можна або вводити з клавіатури, або виділити відповідні клітинки з використанням миші.

Для введення посилань на клітинки з використанням миші потрібно:

1. **Вибрати кнопку Згорнути**  **(мал. 6.19, *5*) відповідного поля для введення аргументів (після цього вікно Аргументи функції змінює вигляд: у ньому, крім рядка заголовка, залишається тільки це поле, а кнопка Згорнути змінюється на кнопку Розгорнути**  **).**
2. **Виділити потрібні клітинки (посилання на виділені клітинки автоматично вставляються у відповідне поле й у формулу (мал. 6.20)). Наприклад, клітинки A1:A10.**
3. **Вибрати кнопку Розгорнути**  **(після цього вікно Аргументи функції відновлює свій попередній вигляд).**

Для деяких функцій Excel автоматично пропонує перший аргумент. Наприклад, для функції SUM, PRODUCT пропонується знайти суму чи добуток чисел з діапазону клітинок, заповнених числовими



Мал. 6.20. Вставлення посилань на клітинки у функцію

даними, що знаходяться безпосередньо над клітинкою з формулою або безпосередньо зліва від неї, якщо верхній діапазон порожній. Користувач може прийняти цю пропозицію, натиснувши клавішу **Enter**, або ввести замість цього аргументу інший.

Коли формула введена, то її можна редагувати. Це можна зробити безпосередньо в **Рядку формул** або в самій клітинці, клацнувши на неї двічі.

МАТЕМАТИЧНІ ФУНКЦІЇ

Пригадайте

- Які функції ви вивчали на уроках математики?
- Як записуються арифметичні вирази в **Excel**?

До категорії математичних функцій належать усі функції, які ви вивчали наразі в математиці, а також функції **SUM** і **PRODUCT**, які ви вже знаєте.

Приклади деяких інших математичних функцій наведено в таблиці 6.3.

Таблиця 6.3

Деякі математичні функції

Функція	*Кількість аргументів*	*Результат*	*Приклад використання*
ABS(число) (англ. ***abs**olute* – абсолютний)	1	Модуль (абсолютна величина) **числа**	**ABS(C10)**
INT(число) (англ. ***int**eger* – цілий)	1	**Число**, округлене до найближчого меншого цілого (ціла частина числа)	**INT(B2)**
PI()	0	Значення числа **π = 3,14159265358979** з точністю до 15-го десяткового розряду	**PI()**

Продовження таблиці 6.3

Функція	***Кількість аргумен-тів***	***Результат***	***Приклад використання***
POWER(число; степінь) (англ. ***power*** – потужність, степінь)	2	**Число**, піднесене до степеня з показником **степінь** (показник степеня – будь-яке число)	**POWER(C5;5)**
ROUND(число; кількість_розря-дів) (англ. ***round*** – округлювати)	2	Округлене **число**. Якщо **кількість_роз-рядів** > 0, то в поданні числа залишається вказана кількість роз-рядів з округленням останнього після коми. Якщо **кількість_роз-рядів** < 0, то **число** округлюється до вказа-ної кількості розрядів перед комою. Якщо **кількість_роз-рядів** = 0, то **число** округлюється до цілого	**ROUND(C1;3)**
SIN(число) (англ. ***sin*** – синус)	1	Синус **числа** (кута, заданого в радіанах)	**SIN(B8)**
SQRT(число) (англ. ***square root*** – квадратний корінь)	1	Арифметичний квадратний корінь із **числа**	**SQRT(B10)**
DEGREES(кут) (англ. *degrees* – гра-дуси)	1	Градусна міра кута, поданого в радіанах	**DEGREES(A2)**
RADIANS(кут) (англ. *radians* – радіани)	1	Радіанна міра кута, поданого у градусах	**RADIANS(B2)**
RANDBETWEEN (нижн. межа; верх. межа) (англ. *random between* – випадкове між)	2	Випадкове ціле число з відрізка [нижн. межа; верх. межа]	**RANDBETWEEN(A5;10)**

Звертаємо вашу увагу! Якщо у клітинці, наприклад **A1**, міститься число – *градусна міра кута*, то для знаходження його синуса потрібно спочатку перевести градуси *в радіани*, а потім обчислити значення функції **SIN**. Тобто формула обчислення буде така: **=SIN(RADIANS(A1))**.

Радіан – одиниця вимірювання величини кута (позначається рад).

$$1 \text{ рад} = \left(\frac{180}{\pi}\right)^{\circ} \approx 57^{\circ}$$

$$1^{\circ} = \frac{\pi}{180} \text{ рад} \approx 0,017 \text{ рад}$$

СТАТИСТИЧНІ ФУНКЦІЇ



Пригадайте

• Чи знаєте ви, що таке *статистика*? Де вам траплявся цей термін? • Які статистичні функції ви вже використовували під час розв'язування завдань?

Ви вже використовували для обчислення статистичні функції **AVERAGE, MAX, MIN.** Інші функції цієї групи наведено в таблиці 6.4.

Таблиця 6.4

Деякі статистичні функції

Функція	*Кількість аргументів*	*Результат*	*Приклад використання*
COUNT(число1; [число2];…) (англ. ***count*** – підрахунок)	Від 1 до 255, усі, крім першого, необов'язкові	Кількість чисел у вказаних клітинках. **Число1** – обов'язковий аргумент: посилання на клітинку або діапазон клітинок. **Число2** та інші – необов'язкові аргументи	**COUNT(B2:C5;E1)**
MODE.SNGL (число1; [число2];...)	Від 1 до 255, усі, крім першого, необов'язкові	Найчастіше повторюване значення в указаних клітинках. **Число1** – обов'язковий аргумент: посилання на клітинку або діапазон клітинок. **Число2** і т. д. – необов'язковий аргумент. Для отримання результату потрібно, щоб хоч одне значення повторювалося хоча б 1 раз. В іншому разі результат обчислення буде відсутній	**MODE.SNGL (B1:A13)**
RANK(число; посилання; [порядок]) (англ. ***rank****ing* – рейтинг)	3	Ранг, або рейтинг (місце за величиною), числа серед інших значень діапазону. **Число** – обов'язковий аргумент: число, ранг якого потрібно визначити. **Посилання** – обов'язковий аргумент: діапазон чисел, серед яких визначається ранг указаного числа. **Порядок** – необов'язковий аргумент: число, що визначає, як розподіляються порядкові номери	**RANK(B5;B1:B10)**

ЛОГІЧНІ ФУНКЦІЇ



Пригадайте

- Що таке *логічні вирази*, *логічні операції*? Які значення вони можуть мати?
- Що таке *розгалуження*? Які види розгалуження ви знаєте? Як вони виконуються?
- Як записуються команди розгалуження в мові програмування, яку ви вивчаєте?

Крім формул, що містять числа, знаки дій, функції, в **Excel** можуть використовуватися формули, які містять **логічні вирази** зі знаками порівняння: > (більше), < (менше), = (дорівнює), <> (не дорівнює), >= (більше або дорівнює), <= (менше або дорівнює). Наприклад, **=A2+15=B4-1**; **=SUM(A2:C10)>100** та інші.

Ви вже знаєте, що такі логічні вирази можуть набувати значення **True** (істина) або **False** (хиба, хибність).

В **Excel** використовують і логічні функції. Функція, результат якої дорівнює **True** або **False**, називається **логічною**. До **логічних функцій** в **Excel** належать функції **IF** (англ. *if* – якщо), **AND** (англ. *and* – і), **OR** (англ. *or* – або), **NOT** (англ. *not* – не) та інші.

Їх використовують у формулах тоді, коли потрібно виконувати різні обчислення залежно від істинності або хибності певного логічного виразу. Такі задачі вам уже траплялися, зокрема у програмуванні. Наприклад, потрібно нарахувати працівнику премію у розмірі 20 % від окладу, якщо він перевиконав план, і не нараховувати премію у протилежному випадку. Або, наприклад, якщо термін продажу товару в магазині завершується, то його уцінюють на 50 %.

Загальний вигляд логічної функції **IF** такий:

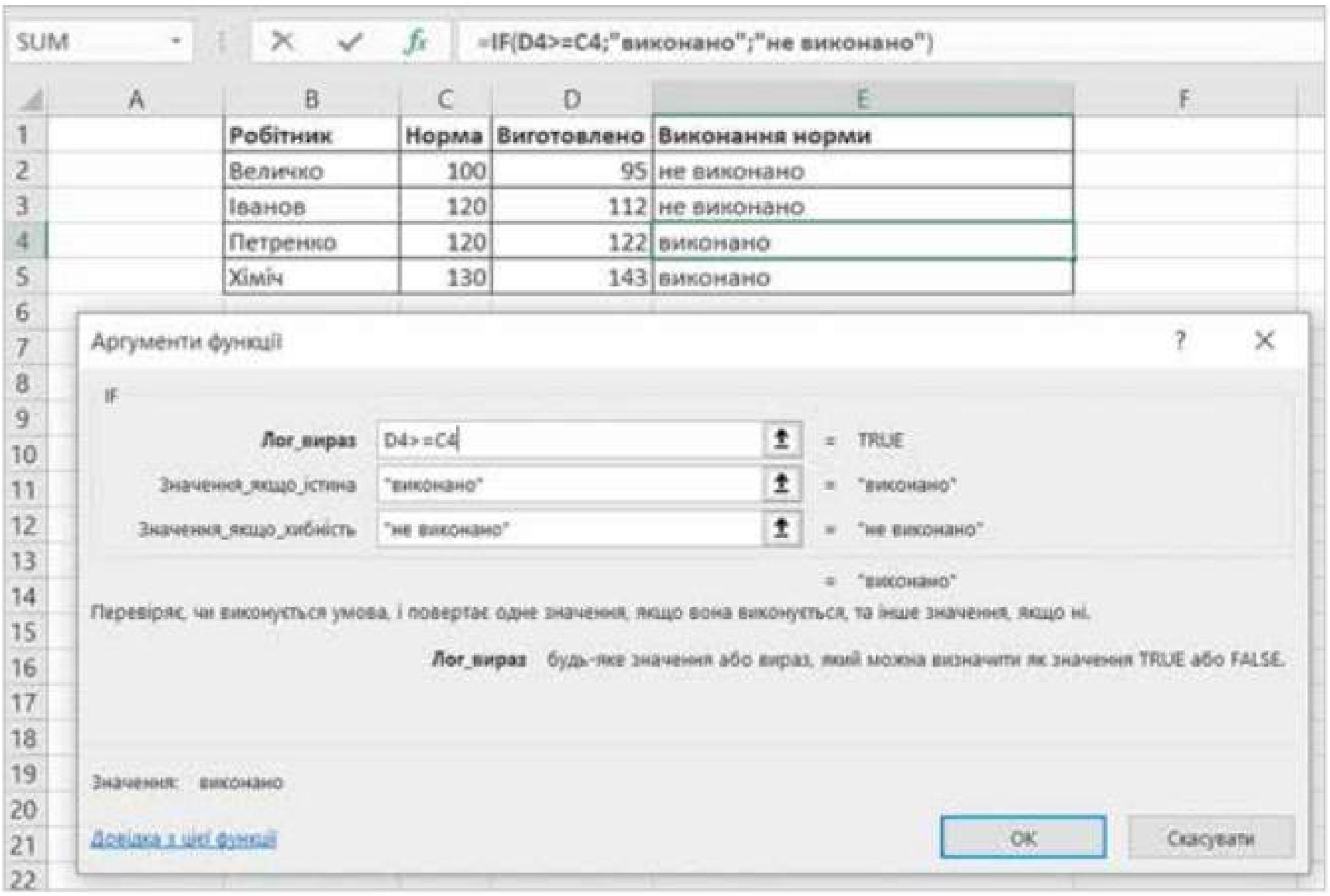
IF(логічний_вираз;значення_якщо_істина;значення_якщо_хибність).

Значення цієї функції визначається так:

1. Визначається значення **логічного_виразу** (**True** чи **False**).
2. Якщо **логічний_вираз** має значення **True**, то значення функції дорівнює значенню виразу **значення_якщо_істина**.
3. Якщо ж **логічний_вираз** має значення **False**, то значення функції дорівнює значенню виразу **значення_якщо_хибність**.

Для прикладу розглянемо таку задачу. *Обчислити премію продавцю-консультанту магазину. Якщо продавець за попередні два місяці обслуговував більше ніж 100 покупців, то йому нараховують премію 15 % від вартості відповідних покупок, у іншому разі премія складатиме 5 %.* Для розрахунків можна використати таку формулу: **=IF(A1+B1>100;C1*15%;C1*5%)**, обчислення значення за якою відбуватиметься так: спочатку **Excel** обчислить суму чисел з клітинок **A1** і **B1** та визначить значення логічного виразу **A1+B1>100**; якщо це значення **True**, то значенням функції буде значення виразу **C1*15%**; якщо значення логічного виразу **A1+B1>100** буде дорівнювати **False**, то значенням функції буде значення виразу **C1*5%**.

Розглянемо ще один приклад задачі, у розв'язку якої використано логічну функцію **IF**. *Кожен з робітників бригади має щоденну норму та реальну кількість виготовлених деталей за день. Потрібно з'ясувати для кожного робітника, чи виконав він свою денну норму.* На малюнку 6.21 наведено розв'язок цієї задачі.



Мал. 6.21. Приклад використання логічної функції **IF**

Як бачимо, правило обчислення значення логічної функції **IF** у табличному процесорі аналогічно до правила виконання команди повного розгалуження у програмуванні.

Аргумент **значення_якщо_хибність** – це необов'язковий аргумент логічної функції **IF**, тому в **Excel** можна використовувати і скорочений запис цієї функції, який відповідає команді неповного розгалуження:

IF(логічний_вираз;значення_якщо_істина).

Розглянемо логічні функції **AND, OR, NOT** (табл. 6.5).

Таблиця 6.5

Логічні функції AND, OR, NOT

Функція	*Кількість аргументів*	*Результат*	*Приклад запису функції*
AND(логіч1; [логіч2];…)	Від 1 до 255, усі, крім першого, необов'язкові	• **True**, якщо всі її аргументи мають значення **True**; • **False**, якщо хоча б один аргумент має значення **False**	**AND(A1>2;B1>10; B1<20;C1=5)**
OR(логіч1; [логіч2];…)	Від 1 до 255, усі, крім першого, необов'язкові	• **True**, якщо хоча б один з аргументів має значення **True**; • **False**, якщо всі аргументи мають значення **False**	**OR(A1>2;B1>10; B1<20;C1=5)**

Продовження таблиці 6.5

Функція	Кількість аргументів	Результат	Приклад запису функції
NOT(логічне)	1	• **True**, якщо аргумент має значення **False**; • **False**, якщо аргумент має значення **True**	**NOT(F1>25)**

Звертаємо увагу, результати наведених функцій в **Excel** збігаються з результатами аналогічних логічних операцій кон'юнкції, диз'юнкції, заперечення, які були розглянуті в п. 4.5.

Логічні функції **AND, OR, NOT** найчастіше використовують для запису складних логічних виразів як аргумент функції **IF**.

Наприклад, у функції **IF** не можна записувати подвійну нерівність. Тому замість подвійної нерівності **10<A3<20** потрібно використовувати логічний вираз **AND(A3>10;A3<20)**. Або, наприклад, функцію задано так:

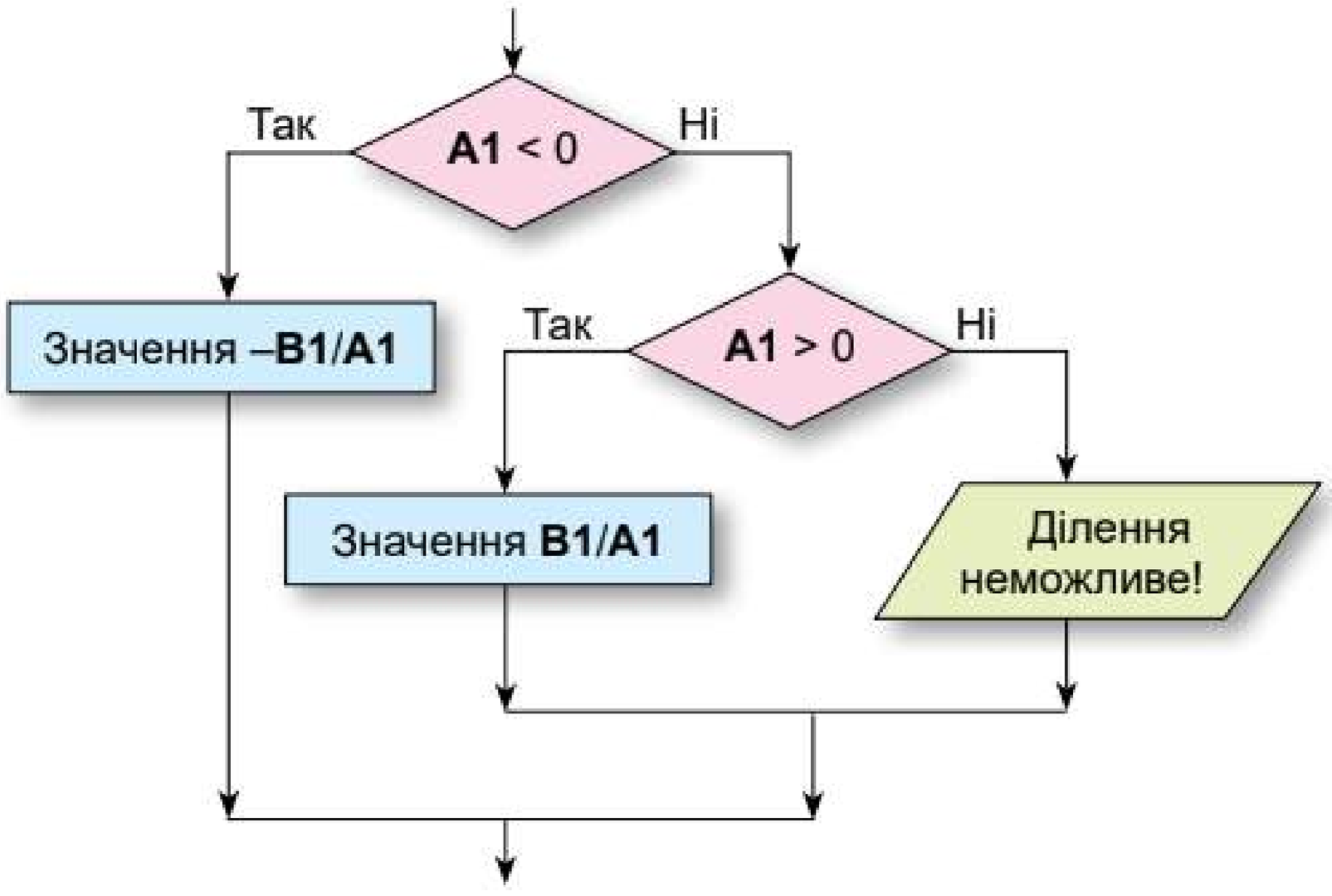
$$y = \begin{cases} 2x - 5, \text{ якщо } x < -2 \text{ або } x > 10, \\ \text{функція не визначена за інших значень } x. \end{cases}$$

Якщо, припустимо, значення x міститься у клітинці **A3**, то формулу для обчислення значення функції y записують так:

=IF(OR(A3<−2;A3>10);2*A3−5;"функція не визначена")

У логічній функції **IF** може бути до 7 рівнів вкладеності. Тобто другий і третій аргументи функції **IF** можуть містити вкладені арифметичні операції та функції, у тому числі й функцію **IF**. Наприклад, для обчислення значення за наведеною блок-схемою (мал. 6.22) слід використати таку функцію:

=IF(A1<0;−B1/A1; IF(A1>0;B1/A1;"Ділення неможливе!")).



Мал. 6.22. Блок-схема для обчислень



Для тих, хто хоче знати більше

Математичні й статистичні функції з умовами

Excel має функції, які обчислюють суму, середнє арифметичне, кількість не всіх значень з діапазонів клітинок, а лише тих, які задовольняють певній умові, – **SUMIF**, **AVERAGEIF**, **COUNTIF**.

Наведемо приклад використання цих функцій. Нехай у клітинках діапазону **A1:A22** розміщено кількості деталей, які виготовив робітник за кожний з 22 робочих днів місяця. Його щоденна норма зберігається у клітинці **C1** і, наприклад, дорівнює 15. За умовою контракту, якщо робітник перевиконує щоденну норму, то за ці деталі він одержує додаткову винагороду. Щоб обчислити, за скільки деталей робітнику потрібно заплатити винагороду, слід скористатися функцією **SUMIF**: **SUMIF(A1:A22;">15")**.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/1e2cJQbd* або QR-кодом.





Найважливіше в цьому пункті

Табличний процесор **Excel** має бібліотеку вбудованих функцій, до якої входить понад 500 різноманітних функцій. Усі вони для зручності розподілені на категорії: *математичні*, *статистичні*, *текстові*, *логічні*, *фінансові* та інші.

Функція в **Excel** має **ім'я**, **аргументи** і **значення**. Під час запису функції у формулі спочатку вказується її ім'я, а потім у дужках **аргументи**, розділені крапкою з комою. Кількість аргументів функції та їх тип залежать від конкретної функції, це можуть бути числа, текст (його потрібно брати в подвійні лапки), посилання на клітинки та діапазони, математичні та логічні вирази, інші функції. Результат однієї функції можна використати як аргумент іншої функції. Такі функції називають **вкладеними**.

Уставити функцію у формулу можна кількома способами:

- виконавши ***Формули*** ⇒ ***Бібліотека функцій*** ⇒ ***Вставити функцію*** або вибравши кнопку **Вставлення функції** ліворуч у **Рядку формул**;
- відкривши список кнопки **Сума** у групі **Редагування** вкладки **Основне**;
- увівши ім'я функції та її аргументи безпосередньо з клавіатури у клітинку або в **Рядок формул**.

Математичні та статистичні функції призначено для автоматизації рутинних обчислень, їх використання підвищує точність обчислень і дає змогу ефективніше аналізувати великі обсяги даних. Приклади математичних функцій: модуль числа **ABS**, арифметичний квадратний корінь із числа **SQRT**, піднесення числа до степеня **POWER**, сума чисел **SUM**. Приклади статистичних функцій: середнє арифметичне **AVERAGE**, максимальне число **MAX**, мінімальне число **MIN**, кількість чисел **COUNT**.

До логічних функцій в **Excel** належать функції **IF**, **AND**, **OR**, **NOT** та інші. Їх використовують у формулах тоді, коли табличний процесор повинен виконувати різні операції залежно від істинності або хибності певного логічного виразу.

Загальний вигляд логічної функції **IF**:

IF(логічний_вираз;значення_якщо_істина;значення_якщо_хибність).

Значення цієї функції визначається так само, як і виконання команди розгалуження у програмуванні.



Дайте відповіді на запитання

1. Чим характеризуються функції в табличному процесорі **Excel**? Наведіть кілька прикладів.

2. Які ви знаєте категорії функцій в **Excel**? Наведіть кілька прикладів.

3. Що таке *логічні вирази* та *функції* в **Excel**? Яких значень вони можуть набувати?

4. Як визначається значення логічної функції **IF**?

5. Як визначається значення логічних функцій **AND, OR, NOT**?

6. Що таке *вкладеність функцій*? Як обчислюється значення функції з вкладенням?



Обговоріть і зробіть висновки

1. Які типи даних можуть мати аргументи функцій? Поясніть це на прикладах.

2. Перегляньте список функцій у табличному процесорі **Excel** і знайдіть функції, у яких немає аргументів. Яке їх призначення? Як ці функції записують?

3. Перегляньте підручник з алгебри та виберіть, які функції там розглянуто та які задачі можна розв'язати в табличному процесорі **Excel**. Запропонуйте кілька прикладів.

4. Ви проводили опитування учнів / учениць вашої школи щодо дотримання ними правильного розпорядку дня, де були питання про вік респондентів / респонденток, їх стать, клас навчання, кількість калорій споживання на день, тривалість нічного сну, фізичні навантаження протягом дня. Запропонуйте функції **Excel**, які ви можете використати для аналізу отриманих даних в електронних таблицях.



Виконайте завдання

1. Відкрийте табличний процесор **Excel**. Створіть таблицю для обчислення довжин діагоналей 5 прямокутників за даними довжинами їх сторін. Значення довжин сторін визначте випадковими значеннями в діапазоні від 5 до 25. Установіть формат результатів обчислення з двома десятковими розрядами. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.1**.

2. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\Пункт 6.2\зразок 6.2.2.xlsx**). На аркуші **Аркуш1** наведено таблицю очок, набраних кожним з учасників змагань з п'ятиборства в кожному виді змагань. Додайте до таблиці формули для обчислення суми очок і максимальної кількості очок, набраних кожним з учасників, а також середньої кількості очок, набраних учасниками в кожному виді змагань. Середню кількість очок округліть до цілих, установивши відповідний формат клітинок. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.2**.

3. У клітинці **A5** міститься число *10*, у клітинці **B5** – число *−7*, а у клітинці **C5** – число *0*. Визначте та запишіть у зошит значення наведених логічних функцій:

а) **AND(A5>5;A5<20)**;
б) **OR(C5<10;C5>=20)**;
в) **OR(C5<10;A5>20)**;
г) **AND(A5>−2;B5>0)**;
д) **NOT(B5<20)**;

е) **AND(OR(B5>5;B5<−5);NOT(B5<>10));**
ж) **OR(AND(A5>2;A5<=10);B5<0);**
и) **NOT(AND(A5<100;C5=0)).**

4. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\Пункт 6.2\зразок 6.2.4.xlsx**). На аркуші **Аркуш1** наведено кількості деталей, зроблених кожним з робітників протягом дня. У клітинці **A1** міститься денна норма. Додайте до таблиці формули для визначення, виконав кожний з робітників норму чи ні. Результат подайте у вигляді **True** – **False**. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.4.**

5. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\Пункт 6.2\зразок 6.2.5.xlsx**). На аркуші **Аркуш1** наведено результати виступів учасників / учасниць олімпіади у кожному з двох турів. За умовами проведення олімпіади, **Дипломом I ступеня** нагороджуються учасники, що набрали не менше ніж 85 % очок, **Дипломом II ступеня** – учасники, що набрали не менше ніж 75 %, але менше від 85 % очок, **Дипломом III ступеня** – учасники, що набрали не менше ніж 50 %, але менше від 75 % очок, усі інші одержують **Диплом учасника / учасниці**. Додайте до таблиці формули для визначення, який **Диплом** буде вручено кожному / кожній з учасників / учасниць. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.5.**

6. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\Пункт 6.2\зразок 6.2.6.xlsx**). На аркуші **Аркуш1** наведено таблицю значень змінної x. Додайте до таблиці формули для визначення відповідних значень функції y:

$$y = \begin{cases} 2x - 5, \text{ якщо } x < -2 \text{ або } x > 10, \\ 3x + 1, \text{ якщо } 2 \leqslant x \leqslant 3, \\ 5 - 4x^3 \text{ за інших значень } x. \end{cases}$$

Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.6.**

7. Складіть електронну таблицю, за якою визначте, які умови вкладу в банк вигідніші клієнту: 9 % річних з нарахуванням відсотків щомісячно чи 10 % річних з нарахуванням відсотків кожного півріччя. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.7.**

8. Складіть електронну таблицю для визначення коренів квадратного рівняння, значення коефіцієнтів обчисліть як випадкові цілі числа в діапазоні від −5 до 5. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.8.**

9. Знайдіть в інтернеті дані про кількість населення в 10 країнах Європи. Складіть електронну таблицю і визначте місце кожної із цих країн за кількістю населення в їх загальному списку, використавши функцію **RANK**. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.9.** *Порада*: щоб функція дала правильні результати для всіх країн, потрібно діапазон, у якому визначається рейтинг, зробити з абсолютною адресацією.

10. Використовуючи дані з інтернету, складіть електронну таблицю для визначення індексу маси тіла людини та порівняння її з нормою. Перевірте виконання обчислень для себе, а вдома – для членів вашої родини. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.10.**

11. Відомі дати народження п'яти ваших друзів. Визначте, у який день тижня вони народилися та у який день тижня святкуватимуть свій день народження цього року. Самостійно знайдіть відповідні функції в переліку вбудованих функцій у категорії **Дата та час.** Збережіть книгу у вашій папці у файлі з іменем **завдання 6.2.11.**

Практична робота № 8. «Використання функцій у табличному процесорі»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Відкрийте вказаний учителем / учителькою файл (наприклад, **Розділ 6\Пункт 6.2\практична 8.xlsx**).
2. На аркуші **Аркуш1** наведено таблицю показників лічильника електроенергії за кожний місяць минулого року. У клітинці **A1** розміщено вартість *1 кВт · год*. Додайте до таблиці формули для обчислення щомісячної плати за електроенергію, загальної плати за рік, середніх щомісячних витрат електроенергії.
3. На аркуші **Аркуш2** наведено прізвища учнів / учениць класу та їх зріст. Додайте до таблиці формули для визначення рейтингу учнів / учениць у класі за спаданням зросту та визначення для кожного / кожної учня / учениці, нижче чи вище його / її зріст від середнього значення.
4. На аркуші **Аркуш3** побудуйте таблицю значень функції $y = \sqrt{x + 9} + x^2$ на відрізку $[-3; 3]$.
5. Збережіть книгу у вашій папці у файлі з іменем **практична 8.xlsx.**

Готуємось до вивчення нового матеріалу

1. Як відсортувати елементи списку в текстовому документі?
2. Як швидко в таблиці в текстовому документі знайти потрібний запис?

6.3. УПОРЯДКУВАННЯ ТА ФІЛЬТРУВАННЯ ДАНИХ У ТАБЛИЧНОМУ ПРОЦЕСОРІ

У цьому пункті йтиметься про:

- упорядкування даних у електронній таблиці;
- фільтрування даних у електронній таблиці.

УПОРЯДКУВАННЯ ДАНИХ У ЕЛЕКТРОННІЙ ТАБЛИЦІ

Пригадайте

● Що таке *таблиці кодів символів*? Для чого їх використовують? Які таблиці кодів символів ви знаєте? ● Яким чином здійснюється сортування елементів списку в текстовому процесорі?

Дані в електронній таблиці можна впорядковувати (*сортувати*) – змінювати порядок розташування даних у рядках або стовпцях за їх значеннями.

У табличному процесорі можна виконувати сортування **за зростанням** або **за спаданням**. Дані різних типів у таблиці впорядковуються за різними правилами.

В **Excel** сортування за зростанням відбувається за такими правилами:

- числа упорядковуються від найменшого значення до найбільшого, тобто так, як вони розміщуються на числовій прямій;
- дати упорядковуються від старішої до новішої, тобто у хронологічній послідовності;
- тексти впорядковуються в алфавітному порядку, в порядку розміщення символів тексту в кодовій таблиці **Unicode**; тобто так, як це відбувається у словниках;
- спочатку розташовуються малі літери, потім великі літери;
- спочатку розташовуються тексти, написані латиницею, потім – кирилицею;
- логічне значення **FALSE** розміщується перед значенням **TRUE**;
- порожні клітинки розташовуються останніми;
- якщо у стовпці таблиці містяться дані різних типів, то після сортування вони будуть розміщуватися в такому порядку: числові, дата / час, текстові, логічні.

Під час сортування за спаданням порядок розташування буде зворотний. Приклади розміщення значень різних типів після сортування за зростанням і за спаданням наведено на малюнку 6.23.

Уведені дані	Тип даних
09.03.1964	Дата
Тетяна	Текстові
FALSE	Логічні
134	Числові
Victor	Текстові
06.07.1985	Дата
10%	Числові
Oleg	Текстові
0,78	Числові
TRUE	Логічні
-23	Числові
Людмила	Текстові

Сортування за зростанням	Тип даних
-23	Числові
10%	Числові
0,78	Числові
134	Числові
09.03.1964	Дата
06.07.1985	Дата
Oleg	Текстові
Victor	Текстові
Людмила	Текстові
Тетяна	Текстові
FALSE	Логічні
TRUE	Логічні

Сортування за спаданням	Тип даних
TRUE	Логічні
FALSE	Логічні
Тетяна	Текстові
Людмила	Текстові
Victor	Текстові
Oleg	Текстові
06.07.1985	Дата
09.03.1964	Дата
134	Числові
0,78	Числові
10%	Числові
-23	Числові

Мал. 6.23. Приклади сортування даних

Щоб швидко виконати сортування даних в одному зі стовпців таблиці, потрібно:

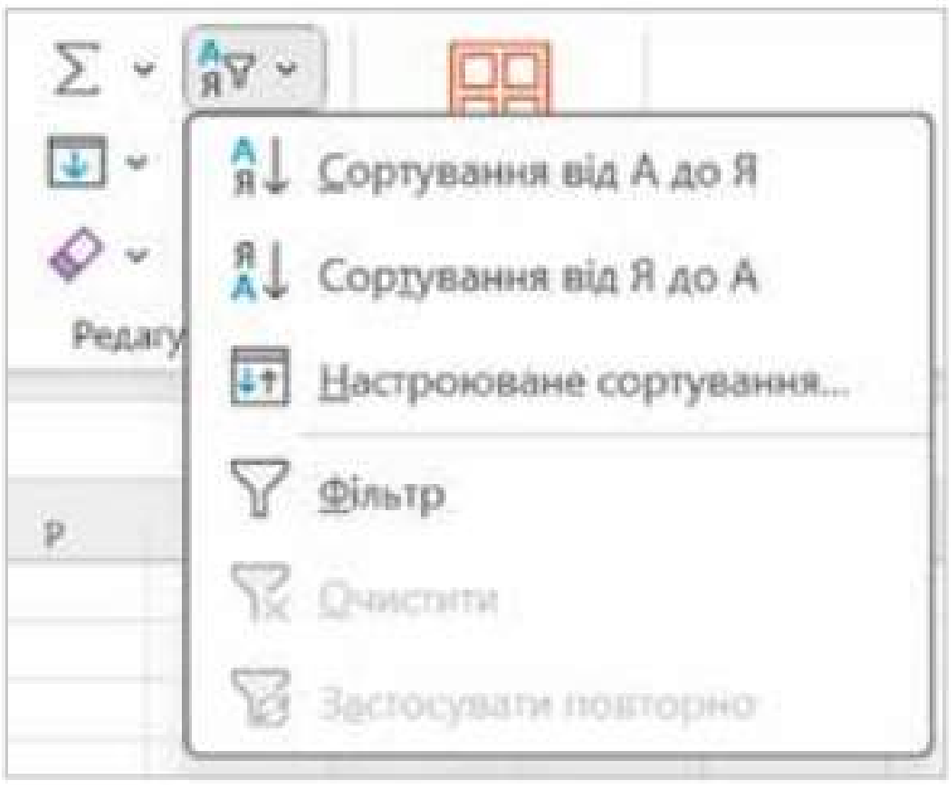
1. Виділити відповідний стовпець.
2. Виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Сортування від найменшого до найбільшого*** або

Сортування від найбільшого до найменшого (мал. 6.24).

У результаті цих дій дані у стовпці змінять своє розташування відповідно до вибраного порядку.

Звертаємо вашу увагу! Назва команди **Сортування від найменшого до найбільшого** і **Сортування від найбільшого до найменшого** автоматично змінюється залежно від типу даних у перших клітинках виділеного діапазону, які потрібно відсортувати:

- для текстових даних – **Сортування від А до Я** та **Сортування від Я до А;**
- для дат і часу – **Сортування від найстаршого до наймолодшого** та **Сортування від наймолодшого до найстаршого.**

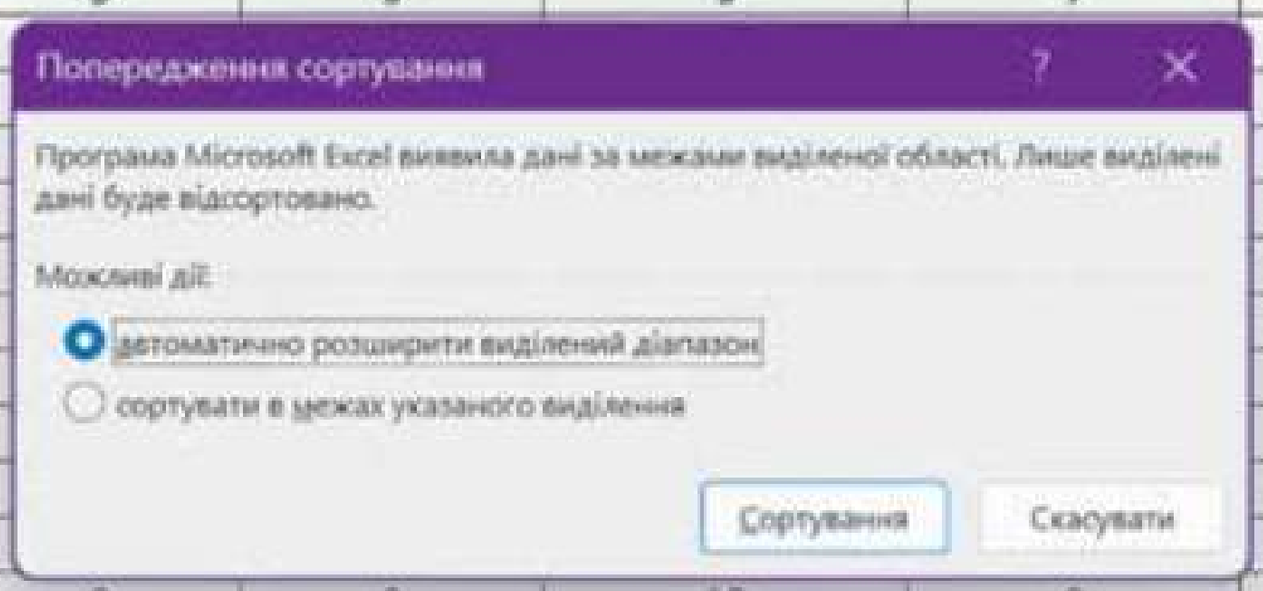


Мал. 6.24. Список команд кнопки **Сортувати й фільтрувати**

Якщо під час сортування виділити діапазон клітинок у кількох стовпцях, то значення в першому стовпці цього діапазону будуть упорядковані у вибраному порядку, а дані в інших стовпцях будуть переміщені разом з відповідним рядком.

Якщо виконувати сортування в таблиці з кількома стовпцями, виділивши дані тільки в одному стовпці, відкривається вікно **Попередження сортування** (мал. 6.25), у якому пропонується вибрати: розширити виділений діапазон клітинок на всю заповнену даними таблицю або сортувати в межах виділення.

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійський	Сума
2	Бурчак	Андрій	9	9	9	7	34
3	Бурчак	Сергій					31
4	Калина	Юлія					35
5	Кручина	Петро					35
6	Левчук	Олексій					38
7	Левчук	Олексій					33
8	Петренко	Катерина					33
9	Пройда	Іван					36
10	Стеценко	Іван					31
11	Туркевич	Сергій					30
12	Штейн	Ілля					40
13	Штейн	Ірина	9	9	10	9	37



Мал. 6.25. Вікно **Попередження сортування**

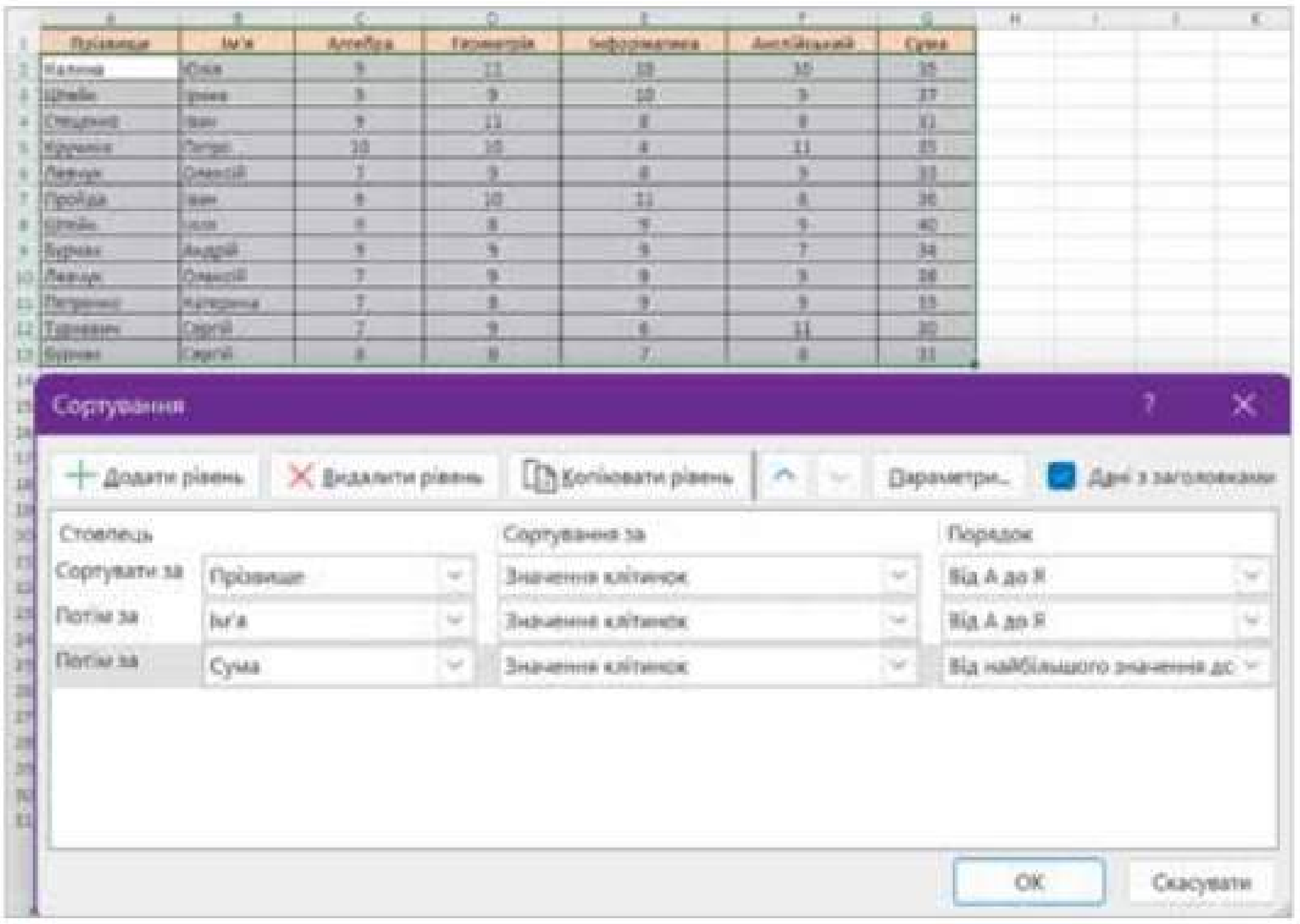
Сортування у виділеному діапазоні можна здійснювати і за значеннями в кількох стовпцях (мал. 6.26). Це відбувається так:

- спочатку рядки переставляються за значеннями в першому з указаних стовпців;
- потім сортування в кожному наступному з вибраних стовпців відбувається лише для тих рядків, у яких значення в усіх попередніх указаних для сортування стовпцях збігаються.

Кожний з указаних для сортування стовпців називається *рівнем сортування*.

Щоб відсортувати дані у виділеному діапазоні клітинок за значеннями в кількох стовпцях, потрібно:

1. Виділити потрібний діапазон клітинок (бажано, щоб до нього увійшли заголовки стовпців, але до нього не повинні входити об'єднані клітинки).
2. Виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Настроюване сортування*** (або ***Дані*** ⇒ ***Сортування й фільтр*** ⇒ ***Сортувати***).



Мал. 6.26. Електронна таблиця та вікно **Сортування**

3. Установити або змінити у вікні **Сортування** значення параметрів сортування: у списку поля **Стовпець** вибрати заголовок першого стовпця, за значеннями в якому потрібно відсортувати дані у стовпці, у списку поля **Сортування за** – *Значення клітинок*, а у списку поля **Порядок** – вид сортування (мал. 6.26). Зверніть увагу в таблиці на однофамільців.
4. За потреби сортування за значеннями іншого стовпця вибрати кнопку **Додати рівень** і встановити умови сортування за цим стовпцем, вибравши в полях рядка **Потім за** значення потрібних параметрів сортування.
5. За потреби повторити попередній крок кілька разів для інших стовпців, за якими слід провести сортування.
6. Якщо у виділений діапазон увійшов рядок заголовка з підписами стовпців, а позначка прапорця **Дані з заголовками** не встановлена, потрібно встановити її (тоді рядок заголовка не буде враховуватися під час сортування і залишиться на своєму місці).
7. Вибрати кнопку **OK**.

Пояснимо таке сортування на конкретному прикладі. На малюнку 6.26 наведено приклад невідсортованої таблиці річних оцінок учнів / учениць, а на малюнку 6.27 цю саму таблицю відсортовано за значеннями у трьох стовпцях: **Прізвище** (від А до Я), **Ім'я** (від А до Я), **Сума** (від найбільшого значення до найменшого).

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
2	Бурчак	Андрій	9	9	9	7	34
3	Бурчак	Сергій	8	8	7	8	31
4	Калина	Юлія	9	11	10	10	35
5	Кручина	Петро	10	10	4	11	35
6	Левчук	Олексій	7	9	9	9	38
7	Левчук	Олексій	7	9	8	9	33
8	Петренко	Катерина	7	8	9	9	33
9	Пройда	Іван	6	10	11	8	36
10	Стеценко	Іван	9	11	8	8	31
11	Туркевич	Сергій	7	9	6	11	30
12	Штейн	Ілля	9	8	9	9	40
13	Штейн	Ірина	9	9	10	9	37

Мал. 6.27. Електронна таблиця, відсортована за умовами, наведеними на малюнку 6.26

Процес упорядкування відбувався так. Спочатку рядки переставлялися так, щоб у стовпці **Прізвище** значення були розташовані в алфавітному порядку (**від А до Я**). Подальше сортування відбувалося вже за значеннями у другому з указаних стовпців (**Ім'я**). Оскільки таблиця містить три групи однофамільців: Бурчак, Левчук, Штейн, – тому за другим стовпцем сортування відбувалося тільки в межах кожної із цих груп рядків, по кожній окремо. У результаті цього кроку сортування рядки переставилися так, щоб імена були розташовані в алфавітному порядку (**від А до Я**).

Сортування за значеннями у третьому з вибраних стовпців (**Сума**) відбувалося тільки в межах рядків, у яких у перших з двох вибраних стовпців збіглися значення (*Левчук Олексій*). Рядки переставились так, щоб у стовпці **Сума** значення були відсортовані за спаданням (*38* та *33*).

Для тих, хто хоче знати більше

Сортувати дані в таблиці можна не тільки за значеннями у стовпцях, а й за значеннями в рядках. Для зміненння режиму сортування потрібно у вікні **Сортування** вибрати кнопку **Параметри**, а потім у вікні вибрати відповідний перемикач. Там само можна визначити, чи враховувати регістр літер під час сортування, тобто чи розрізняти великі й малі літери.

ФІЛЬТРУВАННЯ ДАНИХ

Пригадайте

● Що таке *фільтрування*? Де вам траплялася така дія? ● Яких значень можуть набувати логічні функції **AND** та **OR** залежно від значень аргументів?

Фільтрування **– це автоматизований відбір значень у клітинках електронної таблиці, що відповідають певним умовам.**

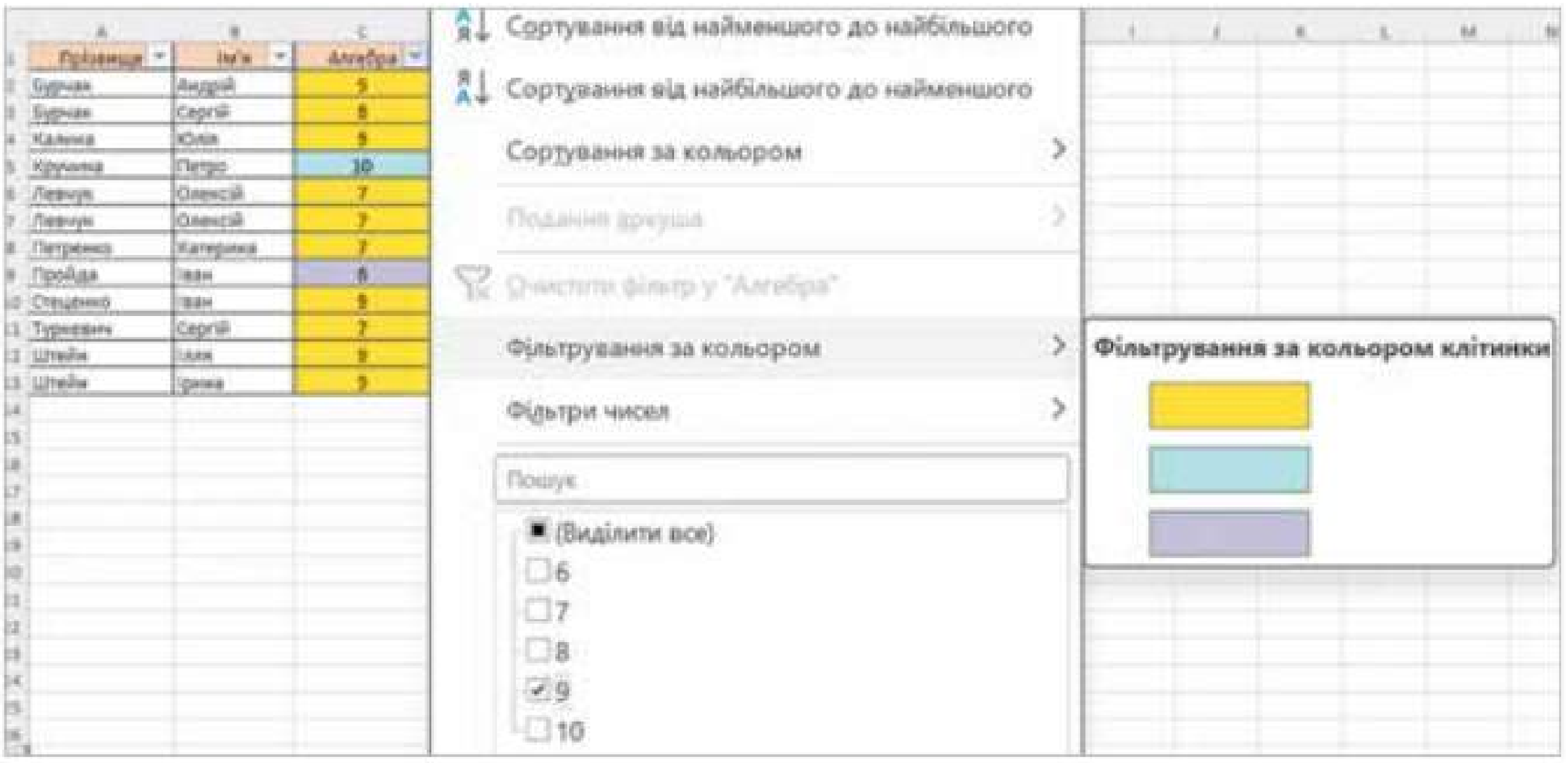
Після виконання фільтрування в таблиці відображаються лише ті рядки, дані в яких відповідають **умовам фільтрування**. Усі інші рядки – тимчасово приховуються.

У **Excel** є два режими фільтрування: простий та розширений фільтри.

Щоб установити режим простого фільтрування даних, потрібно:

1. Установити табличний курсор у довільну клітинку заповненої даними електронної таблиці або виділити потрібний діапазон клітинок.
2. Виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Фільтр*** (або ***Дані*** ⇒ ***Сортування й фільтрування*** ⇒ ⇒ ***Фільтр***).

Після виконання цих дій біля правої межі заголовків стовпців усієї заповненої даними таблиці або виділеного діапазону клітинок з'являться кнопки – відкриття списку команд фільтрування. У цьому списку (мал. 6.28) містяться команди сортування та фільтрування за значеннями (числовими, текстовими або інших типів), а також за кольором.



Мал. 6.28. Застосування фільтра за значенням у стовпці **Алгебра**

Розглянемо деякі приклади виконання фільтрування.

Найчастіше виконують фільтрування за значеннями в одному або кількох стовпцях даних. Наприклад, потрібно за наведеною на малюнку 6.28 таблицею оцінок відобразити список учнів / учениць, які мають з ***алгебри*** оцінку ***9***. Для цього потрібно:

1. Застосувати фільтр до діапазону з даними, виконавши ***Основне*** ⇒ ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Фільтр***.
2. Відкрити список команд фільтрування у стовпці **Алгебра**.
3. Зняти позначку прапорця **Виділити все**.
4. Установити позначку прапорця біля значення **9**.
5. Вибрати кнопку **ОК**.

Після виконання цих дій у таблиці будуть відображені лише ті рядки, у яких значення у стовпці **Алгебра** дорівнюють **9**. Результат описаного прикладу наведено на малюнку 6.29.

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
2	Бурчак	Андрій	9	9	9	7	34
4	Калина	Юлія	9	11	10	10	35
10	Стеценко	Іван	9	11	8	8	31
12	Штейн	Ілля	9	8	9	9	40
13	Штейн	Ірина	9	9	10	9	37

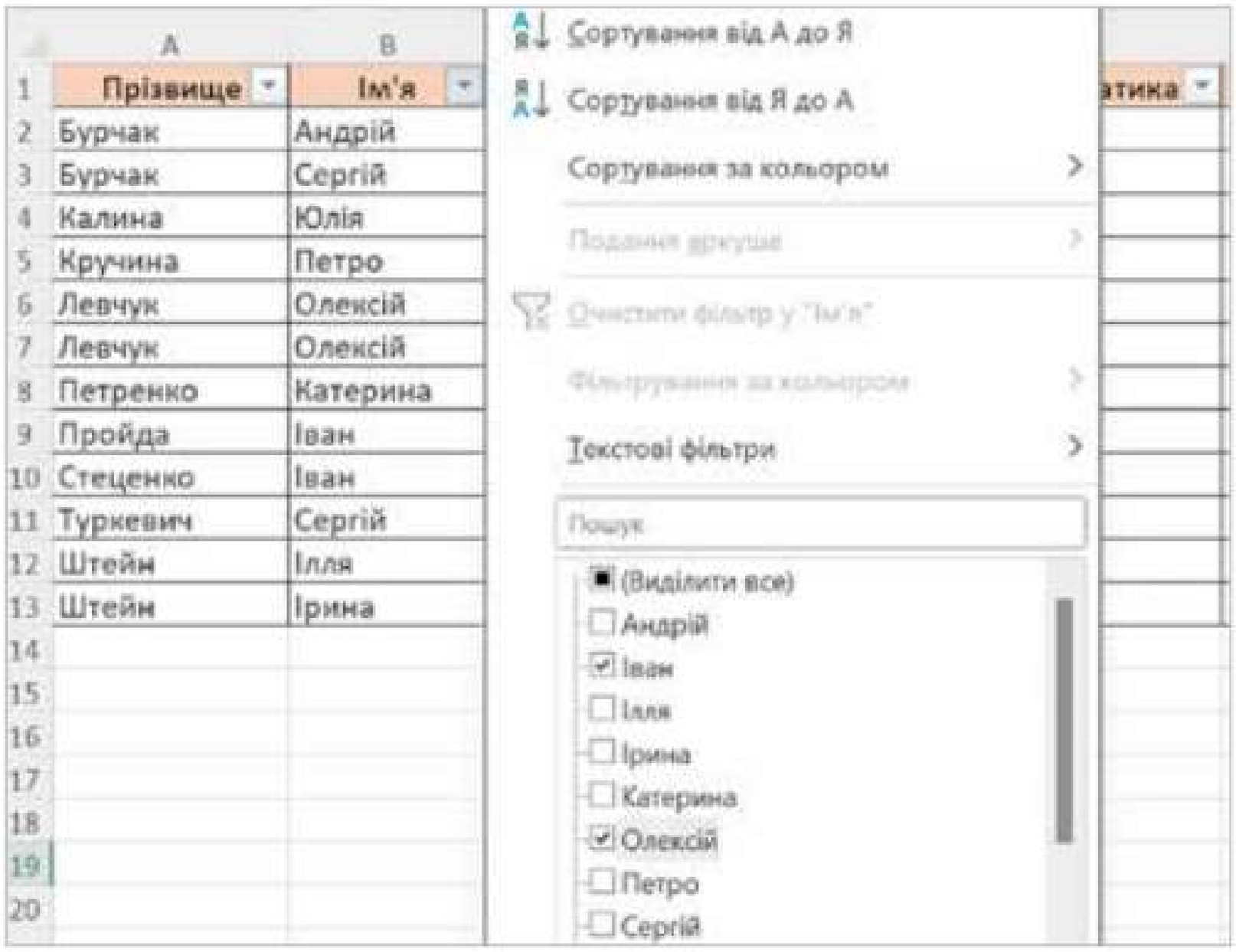
Мал. 6.29. Результат застосування фільтра до таблиці на малюнку 6.28

Зверніть увагу, вигляд кнопки фільтра у стовпці **Алгебра** на малюнку 6.29 змінився на такий: , це означає наявність установленого фільтра для даних у цьому стовпці таблиці. Також колір номерів рядків та їх нумерація з пропусками говорить про те, що фільтр застосовано.

Фільтрування можна здійснювати і за кількома значеннями. Наприклад, потрібно відобразити дані про учнів з іменами ***Іван*** або ***Олексій***. Для цього потрібно:

1. Застосувати фільтр до діапазону з даними.
2. Відкрити список команд фільтрування у стовпці **Ім'я**.
3. Зняти позначку прапорця **Виділити все**.
4. Установити позначку прапорців біля значень ***Іван*** та ***Олексій*** (мал. 6.30).
5. Вибрати кнопку **ОК**.

У результаті отримаємо таблицю, яку наведено на малюнку 6.31.

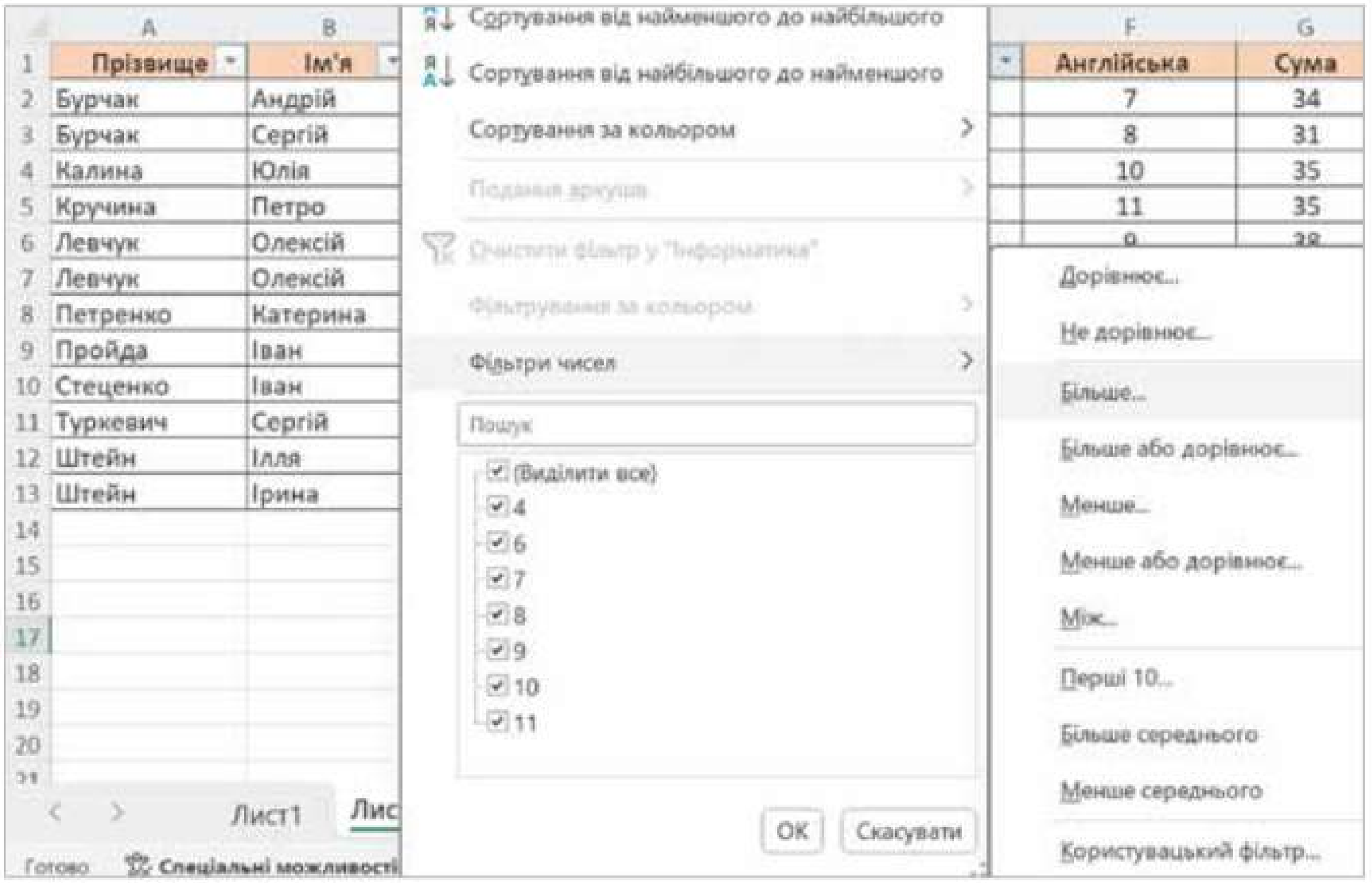


Мал. 6.30. Фільтрування за двома значеннями стовпця **Ім'я**

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
6	Левчук	Олексій	7	9	9	9	38
7	Левчук	Олексій	7	9	8	9	33
9	Пройда	Іван	6	10	11	8	36
10	Стеценко	Іван	9	11	8	8	31

Мал. 6.31. Результат застосування фільтра, наведеного на малюнку 6.30

Якщо у стовпці міститься більше **числових даних,** то за даними цього стовпця можна виконувати фільтрування з використанням **Фільтрів чисел,** застосовуючи команди **Дорівнює, Не дорівнює, Більше, Менше, Між** тощо (мал. 6.32).



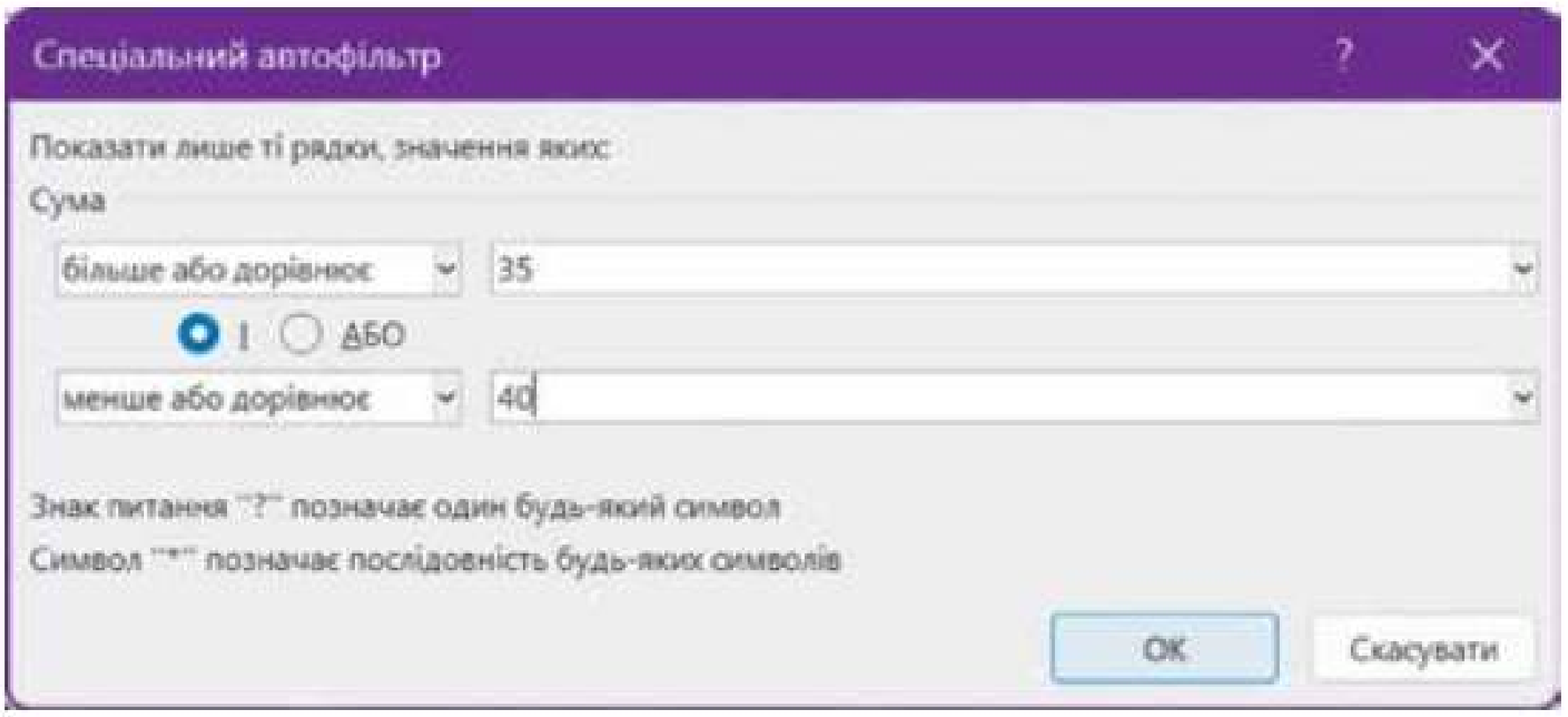
Мал. 6.32. Команди встановлення умов фільтрування числових даних

Наприклад, потрібно відобразити дані тих учнів, які мають сумарний бал за предметами ≥ 35 і ≤ 40. Для цього слід виконати:

1. Застосувати фільтр до діапазону з даними.
2. Відкрити список команд фільтрування у стовпці **Сума**.
3. Вибрати ***Фільтри чисел*** $\Rightarrow$ ***Між***.
4. Увести у відповідні поля вікна потрібні дані ***35*** та ***40*** (мал. 6.33).
5. Вибрати кнопку **ОК**.

Застосувавши фільтр відповідно до малюнка 6.33, отримаємо таблицю, яку наведено на малюнку 6.34.

Команда **Перші 10** зі списку команд **Фільтри чисел** відкриває вікно **Автофільтр для добору найкращої десятки** (мал. 6.35), у якому можна встановити умову фільтрування, щоб у таблиці відображалися тільки ті рядки таблиці, які містять у даному стовпці 10 найбільших / найменших (або іншу встановлену кількість) значень.

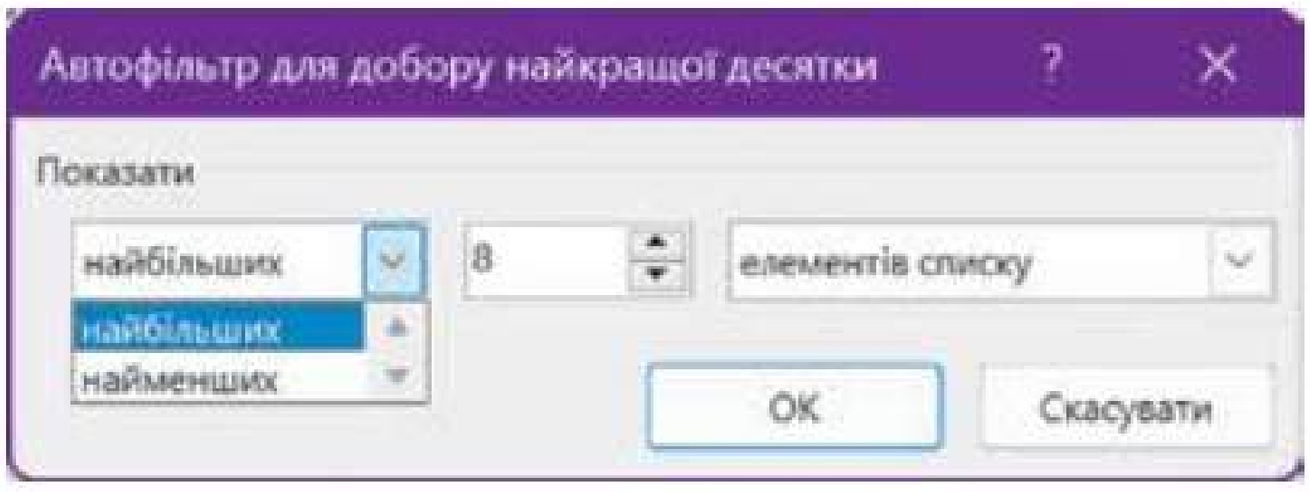


Мал. 6.33. Установлення числового фільтра

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
4	Калина	Юлія	9	11	10	10	35
5	Кручина	Петро	10	10	4	11	35
6	Левчук	Олексій	7	9	9	9	38
9	Пройда	Іван	6	10	11	8	36
12	Штейн	Ілля	9	8	9	9	40
13	Штейн	Ірина	9	9	10	9	37

Мал. 6.34. Результат застосування фільтра для таблиці з малюнка 6.33

Команда **Перші 10** зі списку команд **Фільтри чисел** відкриває вікно **Автофільтр для добору найкращої десятки** (мал. 6.35), у якому можна встановити умову фільтрування, щоб у таблиці відображалися тільки ті рядки таблиці, які містять у даному стовпці 10 найбільших / найменших (або іншу встановлену кількість) значень.



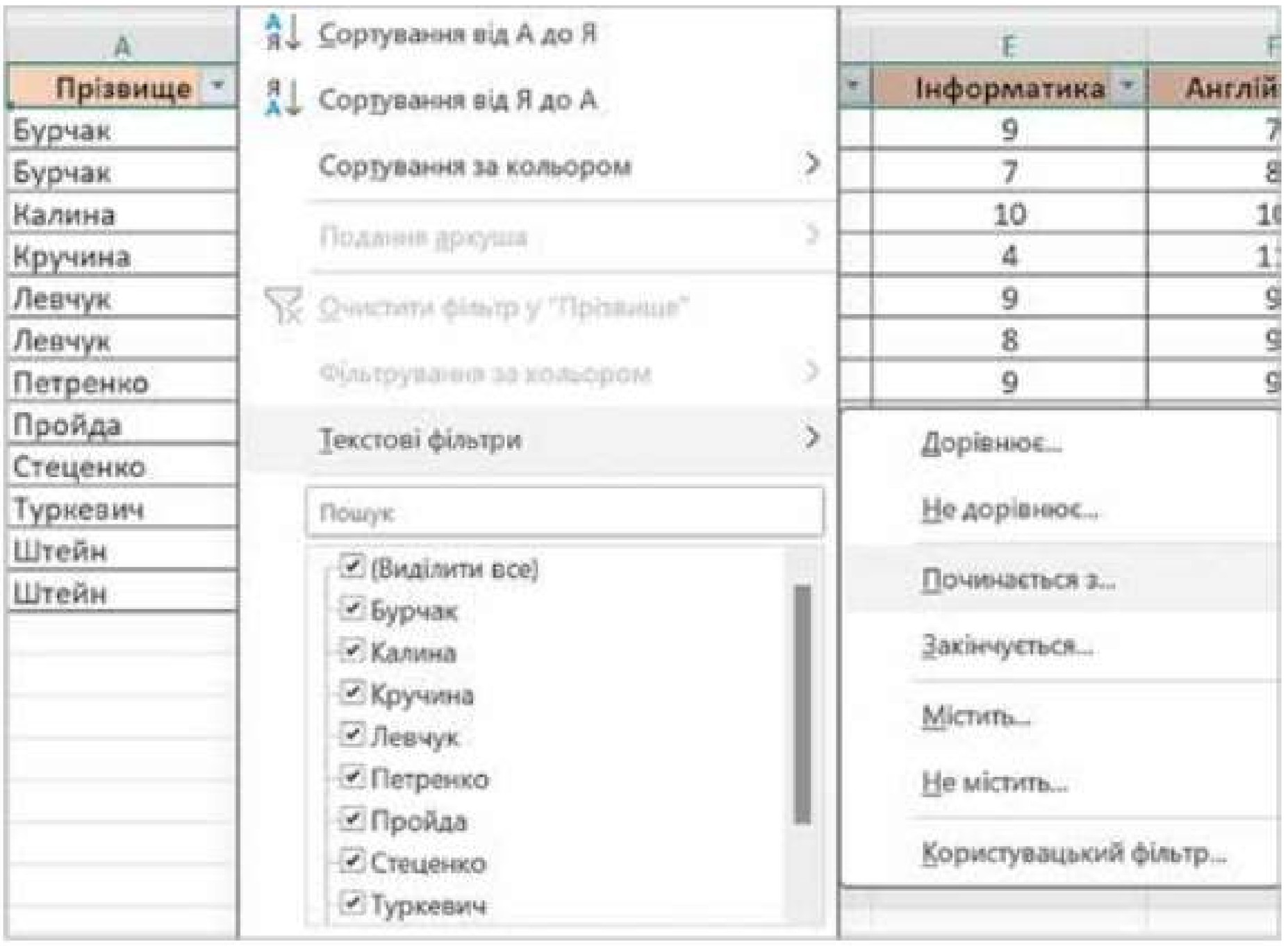
Мал. 6.35. Вікно **Автофільтр для добору найкращої десятки**

Команда **Більше середнього (Менше середнього)** виконує фільтрування, за яким відображаються тільки ті рядки таблиці, які містять у даному стовпці значення більші (менші), ніж середнє арифметичне значення в даному стовпці.

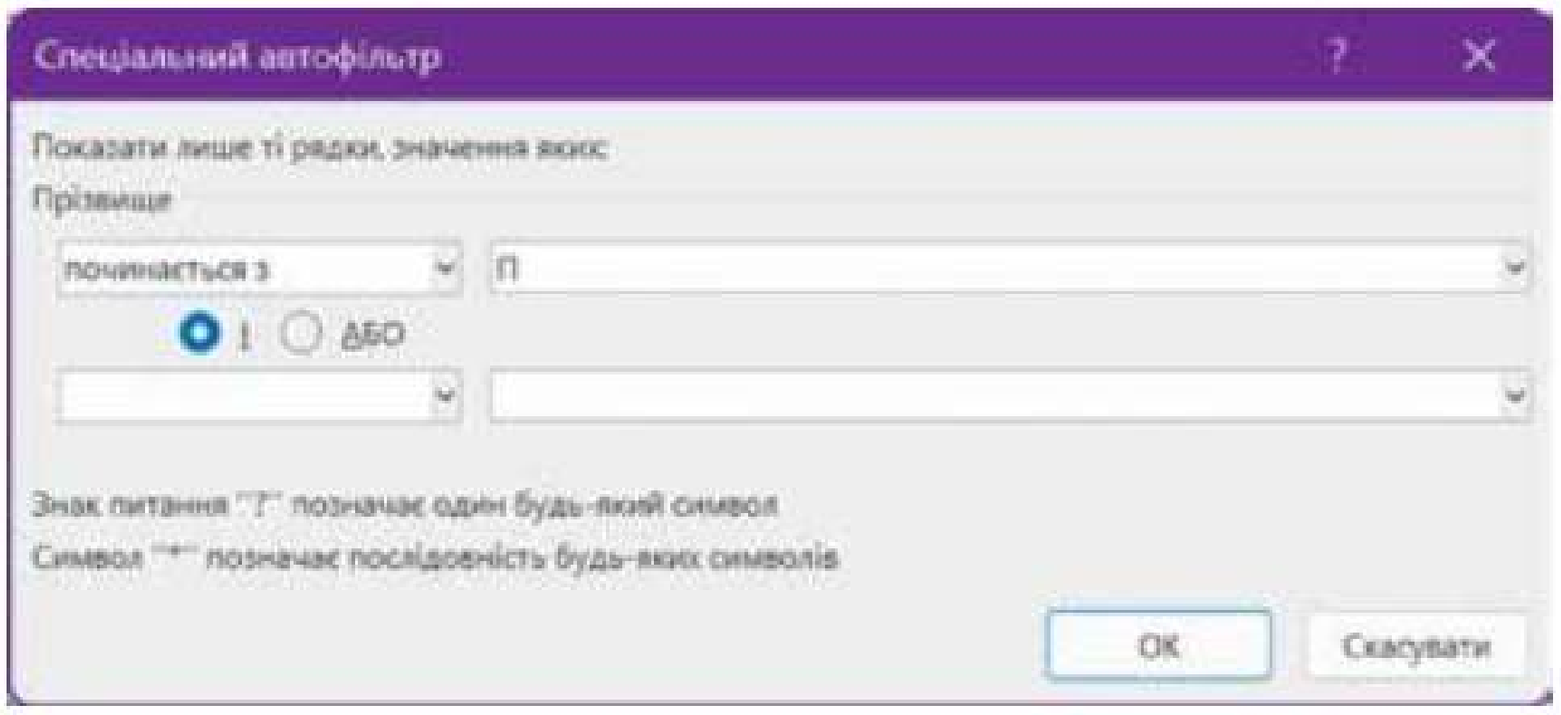
Для **текстових даних** застосовують дещо інші команди фільтрування (мал. 6.36): **Дорівнює, Не дорівнює, Починається з, Закінчується, Містить** тощо. Наприклад, потрібно відобразити дані про тих учнів / учениць, прізвища яких починаються з літери ***П***. Для цього слід:

1. Застосувати фільтр до діапазону з даними.
2. Відкрити список команд фільтрування у стовпці **Прізвище**.

3. Вибрати ***Текстові фільтри*** ⇒ ***Починається з...*** (мал. 6.36).
4. Увести у відповідне поле літеру ***П*** (мал. 6.37).
5. Вибрати кнопку **ОК**.



Мал. 6.36. Команди встановлення умов фільтрування текстових даних



Мал. 6.37. Умова для фільтрування текстових даних

Фільтрування також можна здійснювати послідовно для кількох стовпців таблиці. Тоді кожний наступний фільтр буде застосований вже тільки для тих рядків таблиці, які відображаються після застосування всіх попередніх фільтрів. У таких випадках загальна умова фільтрування – це умови фільтрування для кожного стовпця, об'єднані логічною операцією **І**. Наприклад, якщо провести фільтрування за даними стовпця **Сума** з умовою **більша за 35**, а потім за даними стовпця **Інформатика** з умовою **більший за 8**, то загальне фільтрування буде виконано за умовою **(сума балів більша за 35) І (бал з інформатики більший за 8)**. Результат такого фільтрування наведено на малюнку 6.38.

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
6	Левчук	Олексій	7	9	9	9	38
9	Пройда	Іван	6	10	11	8	36
12	Штейн	Ілля	9	8	9	9	40
13	Штейн	Ірина	9	9	10	9	37

Мал. 6.38. Результат послідовного фільтрування у стовпцях **Сума** та **Інформатика**

Скасувати фільтрування даних можна кількома способами:

- виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Очистити***;
- виконати команду **Очистити фільтр** у списку кнопки фільтрування стовпця, за даними якого було проведено фільтрування;
- установити позначку прапорця **Виділити все** у списку кнопки фільтрування стовпця, за даними якого було проведено фільтрування, після чого вибрати кнопку **ОК**.

Щоб узагалі скасувати режим фільтрування, потрібно повторно виконати одну з команд, яка встановлює цей режим. Наприклад, виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ⇒ ***Фільтр***.

Для тих, хто хоче знати більше

Розширений фільтр

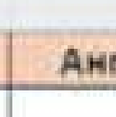
Використання простих фільтрів дає змогу виконати фільтрування не в усіх випадках. Так, наприклад, розглянутими способами не можна виконати фільтрування за умовою, яка є об'єднанням умов фільтрування двох стовпців логічною операцією **АБО**, наприклад: (**сума балів більша за 35**) **АБО** (**бал з інформатики більший за 8**).

Виконати таке фільтрування в табличному процесорі **Excel** можна з використанням **розширених фільтрів**. Для цього слід:

1. Скопіювати у вільні клітинки електронної таблиці (зручніше це зробити під заповненою таблицею) назви тих стовпців, за даними яких буде здійснюватися фільтрування (наприклад, на малюнку 6.39 назви стовпців *Сума*, *Інформатика*).



Мал. 6.39. Умови фільтрування та вікно **Розширений фільтр**

2. Увести у клітинки під назвами стовпців умови фільтрування.
   Якщо ці умови повинні об'єднуватися логічною операцією **І**, то вони мають розташовуватися в одному рядку. Якщо логічною операцією **АБО** – то в різних (на малюнку 6.39 умова *Сума > 35 або Інформатика > 8*).
3. Виконати ***Дані*** ⇒ ***Сортування й фільтр*** ⇒ ***Додатково***  .
4. У вікні **Розширений фільтр**:
   1. Вибрати один з перемикачів для визначення області розташування результату фільтрування (на малюнку 6.39 – *фільтрувати список на місці*).
   2. Увести в поле **Вихідний діапазон** діапазон клітинок, дані в яких повинні фільтруватися (на малюнку 6.39 – **A1:G13**).
   3. Увести в поле **Діапазон умов** діапазон клітинок, у яких розташовано назви стовпців і умови фільтрування (наприклад, **D15:E17**).
   4. Якщо був вибраний перемикач **скопіювати результат до іншого розташування**, то ввести в поле **Діапазон для результатів** діапазон клітинок, де має розміститися результат фільтрування.
   5. Вибрати кнопку **ОК**.

На малюнку 6.40 наведено результат фільтрування, виконаного за умовами, наведеними на малюнку 6.39.

	A	B	C	D	E	F	G
1	Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
2	Бурчак	Андрій	9	9	9	7	34
4	Калина	Юлія	9	11	10	10	35
6	Левчук	Олексій	7	9	9	9	38
8	Петренко	Катерина	7	8	9	9	33
9	Пройда	Іван	6	10	11	8	36
12	Штейн	Ілля	9	8	9	9	40
13	Штейн	Ірина	9	9	10	9	37

Мал. 6.40. Результат фільтрування за умовами, наведеними на малюнку 6.39



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ve2cJBaY* або QR-кодом.





Найважливіше в цьому пункті

Дані в електронній таблиці можна впорядковувати (*сортувати*) – змінювати порядок розташування даних у рядках або стовпцях за їх значеннями.

У табличному процесорі можна виконувати сортування **за зростанням** або **за спаданням**. Дані різних типів у таблиці впорядковуються за різними правилами.

Щоб виконати сортування виділеного діапазону клітинок за значеннями одного стовпця цього діапазону, потрібно виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Сортування від найменшого до найбільшого*** або ***Сортування від найбільшого до найменшого***.

Сортування у виділеному діапазоні можна здійснювати і за значеннями кількох стовпців. Для цього слід виконати ***Дані*** ⇒ ***Сортування й фільтр*** ⇒ ***Сортувати*** та встановити в діалоговому вікні параметри та умови сортування. Кожний з указаних для сортування стовпців називається *рівнем сортування*. При цьому спочатку рядки сортуються за першою умовою, а потім на наступному рівні сортування відбувається лише для тих рядків, у яких значення в усіх попередніх стовпцях сортування збігаються.

***Фільтрування* – це автоматизований відбір значень у клітинках електронної таблиці, що відповідають певним умовам**. Після виконання фільтрування в таблиці

відображаються лише ті рядки, дані в яких відповідають **умовам фільтрування**. Усі інші – тимчасово приховуються.

Для встановлення режиму *простого* фільтрування потрібно встановити табличний курсор у довільну клітинку заповненої даними електронної таблиці або виділити потрібний діапазон клітинок і виконати ***Основне*** ⇒ ***Редагування*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Фільтр*** або ***Дані*** ⇒ ***Сортування й фільтр*** ⇒ ***Фільтр***.

Після виконання цих дій біля правої межі заголовків стовпців усієї заповненої даними таблиці або виділеного діапазону клітинок з'являться кнопки – відкриття списку команд фільтрування. У цьому списку містяться команди сортування та фільтрування за значеннями (числовими, текстовими або інших типів), а також за кольором.

Фільтрування можна здійснювати послідовно для кількох стовпців таблиці. Тоді кожний наступний фільтр буде застосовано тільки для тих рядків таблиці, які відображаються після застосування всіх попередніх фільтрів. У таких випадках загальна умова фільтрування – це умови фільтрування для кожного стовпця, об'єднані логічною операцією **І**. Для виконання фільтрування за умовою, яка є об'єднанням умов фільтрування двох стовпців логічною операцією **АБО**, використовують *розширені фільтри*.

Для скасування фільтрування потрібно виконати ***Дані*** ⇒ ***Сортувати й фільтрувати*** ⇒ ***Очистити***.

Дайте відповіді на запитання

1. У чому полягає сортування даних у **Excel**?

2. Які правила сортування даних за зростанням у **Excel**?

3. Як здійснюється сортування даних у таблицях за значеннями в кількох стовпцях?

4. Коли команда **Сортування від найменшого до найбільшого** змінюється на команду **Сортування від А до Я**?

5. Що таке *фільтрування даних*? Які види фільтрування можна здійснювати для числових даних? Для текстових даних?

6. Для чого використовують простий фільтр?

Обговоріть і зробіть висновки

1. Під час сортування таблиці на екрані з'явилося повідомлення про розширення діапазону сортування. У якому випадку це може статися? Які дії користувача в такому разі? Які наслідки ігнорування цього повідомлення?

2. Що спільного та відмінного в операціях сортування в електронних таблицях і текстових документах?

3. Якщо виконувати сортування за кольором, то яким чином розміститься кольори у відсортованій таблиці? Як це перевірити практично?

4. Як використовувати кольорові фільтри для організації даних? Наведіть кілька прикладів.

Виконайте завдання

1. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.1xlsx**. На аркуші **Аркуш1** відсортуйте таблицю за зростанням часу відправлення поїздів. На аркуші **Аркуш2** відсортуйте таблицю так, щоб дані про поїзди, що відправляються з одного пункту, йшли за алфавітом, дані про кінцевий пункт слідування

розміщувались у порядку **від Я до А**, а час відправлення – зростав. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.1.**

2. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.2.xlsx**. На аркуші **Аркуш1** відсортуйте таблицю за значеннями у трьох стовпцях: **Бали** (за спаданням), **Час** (за зростанням), **Висота** (за спаданням). На аркуші **Аркуш2** впорядкуйте дані у стовпці *Прізвище* за зростанням, а у стовпці *Продано* – за спаданням. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.2.**

3. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.3.xlsx**. На аркуші **Аркуш1** відсортуйте дані про пристрої в порядку спадання ціни та зростання ємності пристроїв. На аркуші **Аркуш2** відсортуйте дані про пристрої в алфавітному порядку фірми-виробника, за спаданням діагоналі та спаданням ціни. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.3.**

4. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.4.xlsx**. На аркуші **Аркуш1** відобразіть дані лише про поїзди, які слідують до Львова. На аркуші **Аркуш2** відобразіть 5 найкоротших за часом поїздки маршрутів. На аркуші **Аркуш3** відобразіть дані про поїзди, які відправляються до Харкова з 9 год до 15 год. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.4.**

5. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.5.xlsx**. На аркуші **Аркуш1** відобразіть дані про учасників / учасниць, які набрали у фехтуванні не менше ніж 40 очок і стрибнули у висоту від 2,20 м до 2,25 м. На аркуші **Аркуш2** виберіть книжки, прибуток від продажу яких більший за 1000 грн та ім'я авторів яких *Дмитро* або *Микола*. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.5.**

6. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.3\зразок 6.3.6.xlsx**. На аркуші **Аркуш1** задайте умову фільтрування для добору даних про пристрої, у яких ємність більша ніж 1000 ГБ і ціна менша ніж 2000 грн. На аркуші **Аркуш2** задайте умову фільтрування для добору даних про пристрої, ціна яких менша ніж середня. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.3.6.**

Готуємось до вивчення нового матеріалу

1. Що розуміють під форматуванням текстових об'єктів? Яким чином це можна виконати?
2. Як відформатувати символи у клітинках електронної таблиці?
3. Як надрукувати текстовий документ? Які властивості сторінки документа потрібно визначити?

6.4. УМОВНЕ ФОРМАТУВАННЯ. ДРУКУВАННЯ ЕЛЕКТРОННОЇ ТАБЛИЦІ

У цьому пункті йтиметься про:

- умовне форматування даних в електронній таблиці;
- друкування електронної таблиці.

УМОВНЕ ФОРМАТУВАННЯ

Пригадайте

● Що таке *сортування даних електронної таблиці*? Якими способами його можна виконати? ● Що розуміють під форматуванням клітин і даних у електронній таблиці? Як це можна виконати?

Ще одним способом виділити в таблиці дані, які задовольняють певним умовам, є так зване ***умовне форматування***.

Умовне форматування змінює формат клітинки або її вмісту на заданий, якщо дані в цій клітинці задовольняють певній умові.

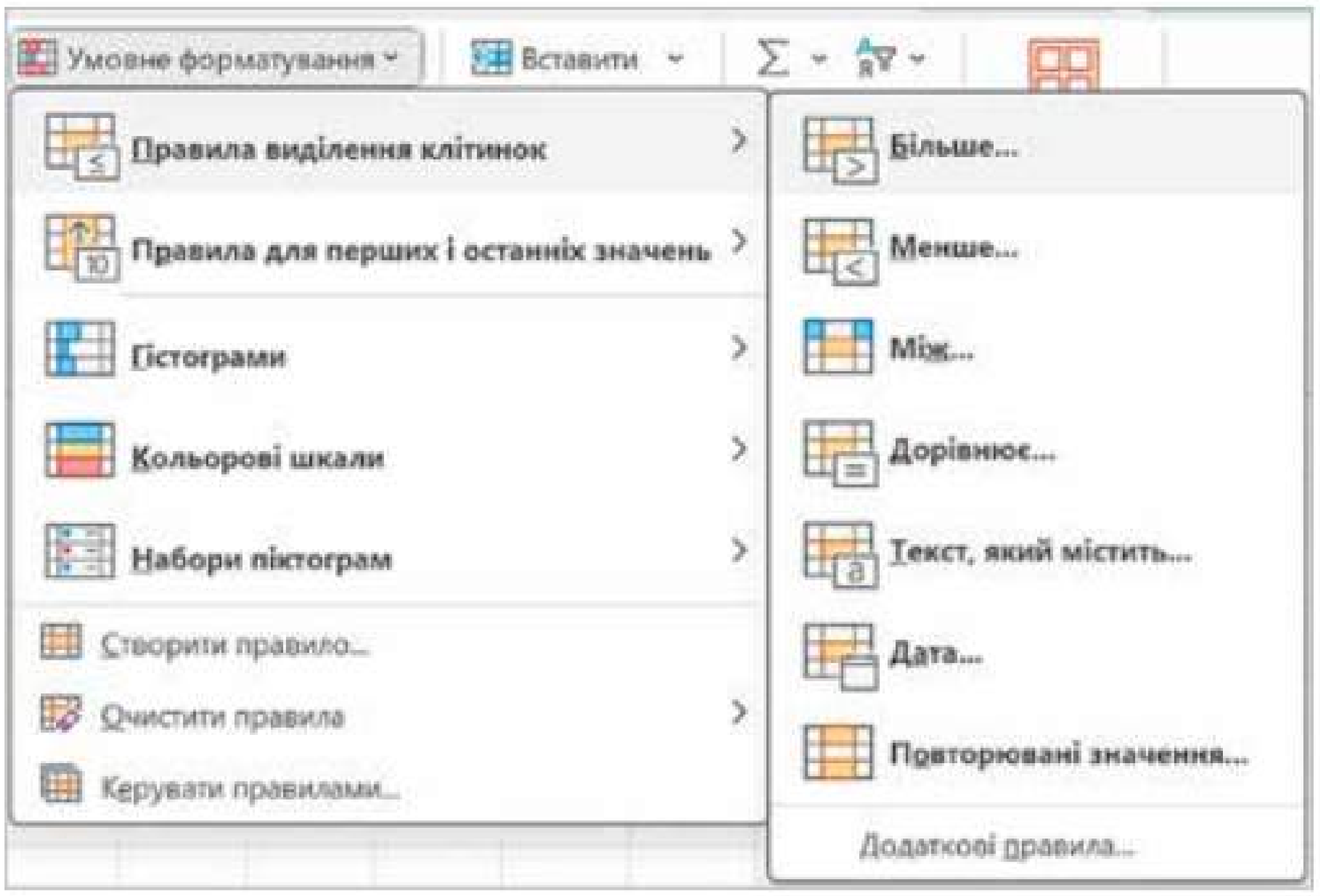
Умовне форматування застосовується до виділеного діапазону клітинок і виконується на основі правил, які складаються з **умови**, що стосується значення у клітинці, та **значень властивостей** форматування вмісту клітинки.

Наприклад, у електронному журналі класу зручно позначати різними кольорами дані про учнів, які мають оцінки низького, середнього, достатнього та високого рівнів. Тобто клітинки, значення яких знаходяться в діапазоні від 1 до 3, наприклад, виділити червоним кольором, у діапазоні від 4 до 6 – зеленим, у діапазоні від 7 до 9 – фіолетовим, у діапазоні від 10 до 12 – синім кольором (мал. 6.41).

	A	B	C
1	Прізвище	Ім'я	Інформатика
2	Бурчак	Андрій	9
3	Бурчак	Сергій	7
4	Калина	Юлія	10
5	Кручина	Петро	4
6	Левчук	Олексій	3
7	Левчук	Олексій	8
8	Петренко	Катерина	3
9	Пройда	Іван	11
10	Стеценко	Іван	8
11	Туркевич	Сергій	6
12	Штейн	Ілля	5
13	Штейн	Ірина	10

Мал. 6.41. Форматування вмісту клітинок відповідно до умов

В **Excel** існує 5 **типів правил** виділення клітинок під час умовного форматування (мал. 6.42):



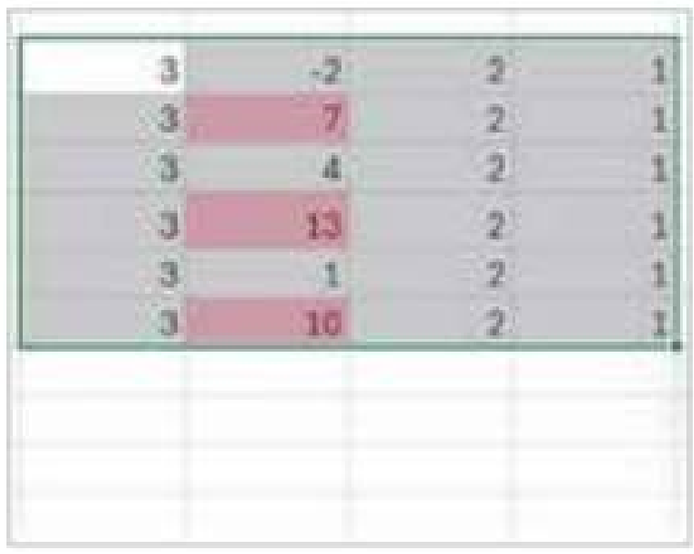
Мал. 6.42. Кнопка **Умовне форматування**, список типів правил, а також список правил типу **Правила виділення клітинок**

- **Правила виділення клітинок;**
- **Правила для перших і останніх значень;**
- **Гістограми;**
- **Кольорові шкали;**
- **Набори піктограм.**

Для встановлення перших двох правил умовного форматування, а саме **Правила виділення клітинок** і **Правила для перших і останніх значень**, слід:

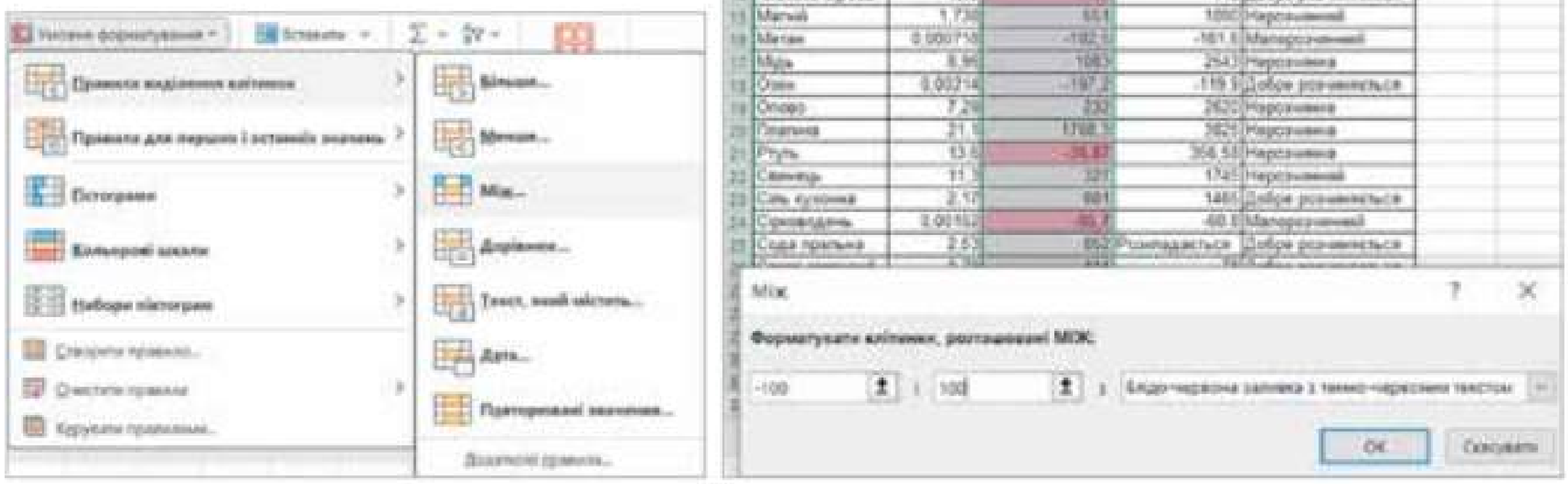
1. Виділити потрібний діапазон клітинок, у якому буде застосовуватись умовне форматування.
2. Виконати ***Основне*** ⇒ ***Стилі*** ⇒ ***Умовне форматування*** 

.
3. Вибрати у списках кнопки **Умовне форматування** потрібний тип правил (мал. 6.42), наприклад **Правила виділення клітинок.**
4. Вибрати у списку обраного типу потрібне правило форматування, наприклад **Більше...**
5. Задати у вікні, що відкриється, умову форматування та формат, який буде встановлено, якщо умова буде виконуватися, а саме значення властивостей фону і символів тексту (мал. 6.43). Формат можна вибрати з наведених або, вибравши в кінці списку команду **Настроюваний формат**, установити власний.
6. Вибрати кнопку **ОК.**



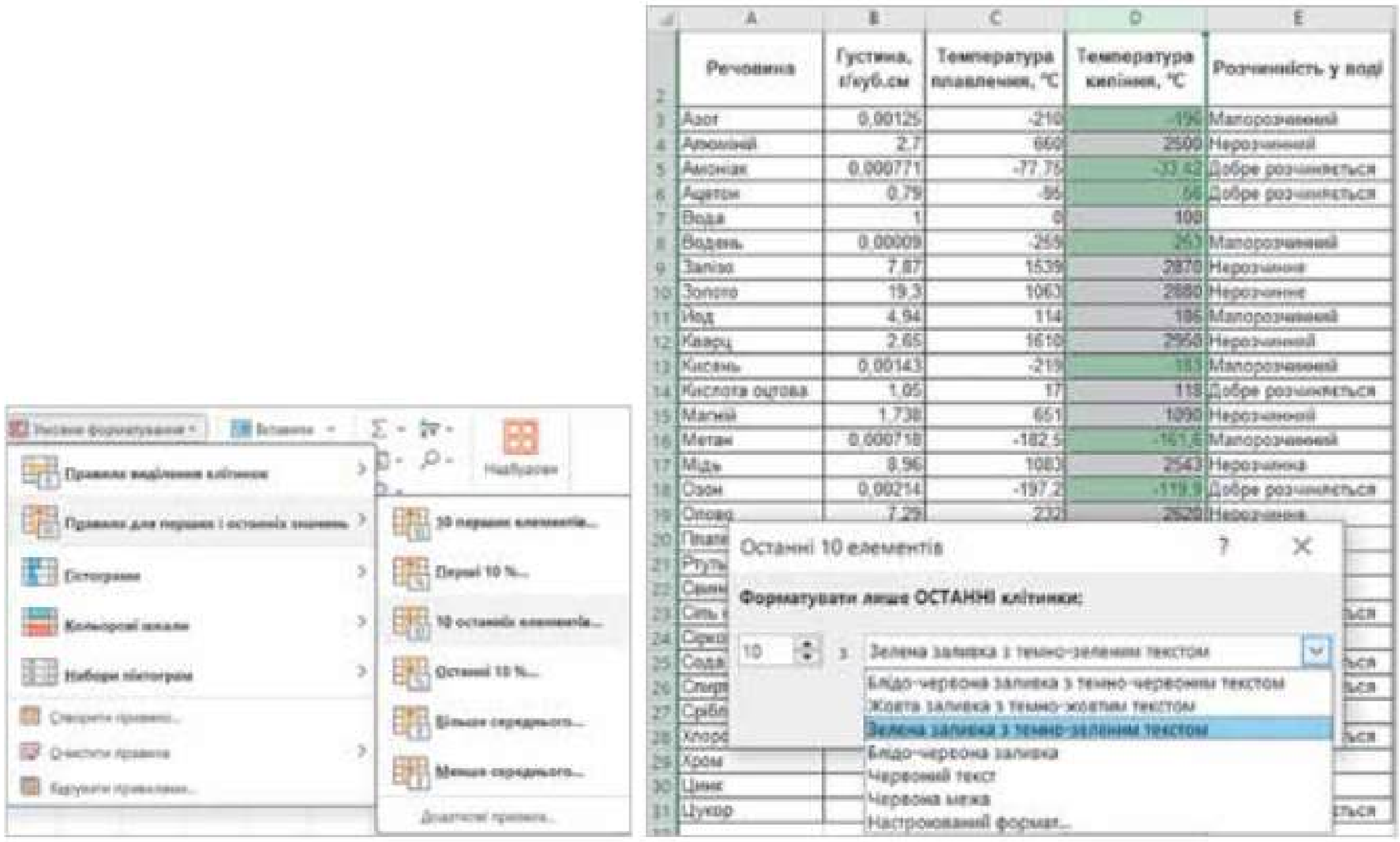
Мал. 6.43. Установлення умови та формату для умовного форматування

На малюнку 6.44 наведено як приклад вікно умови **Між**, у якому встановлено умову форматування **Між −100 і 100** градусів Цельсія як значення температури плавлення речовин, з вибраним стандартним форматом *Блідо-червона заливка з темно-червоним текстом*, а також попередній перегляд результатів вибраного умовного форматування.



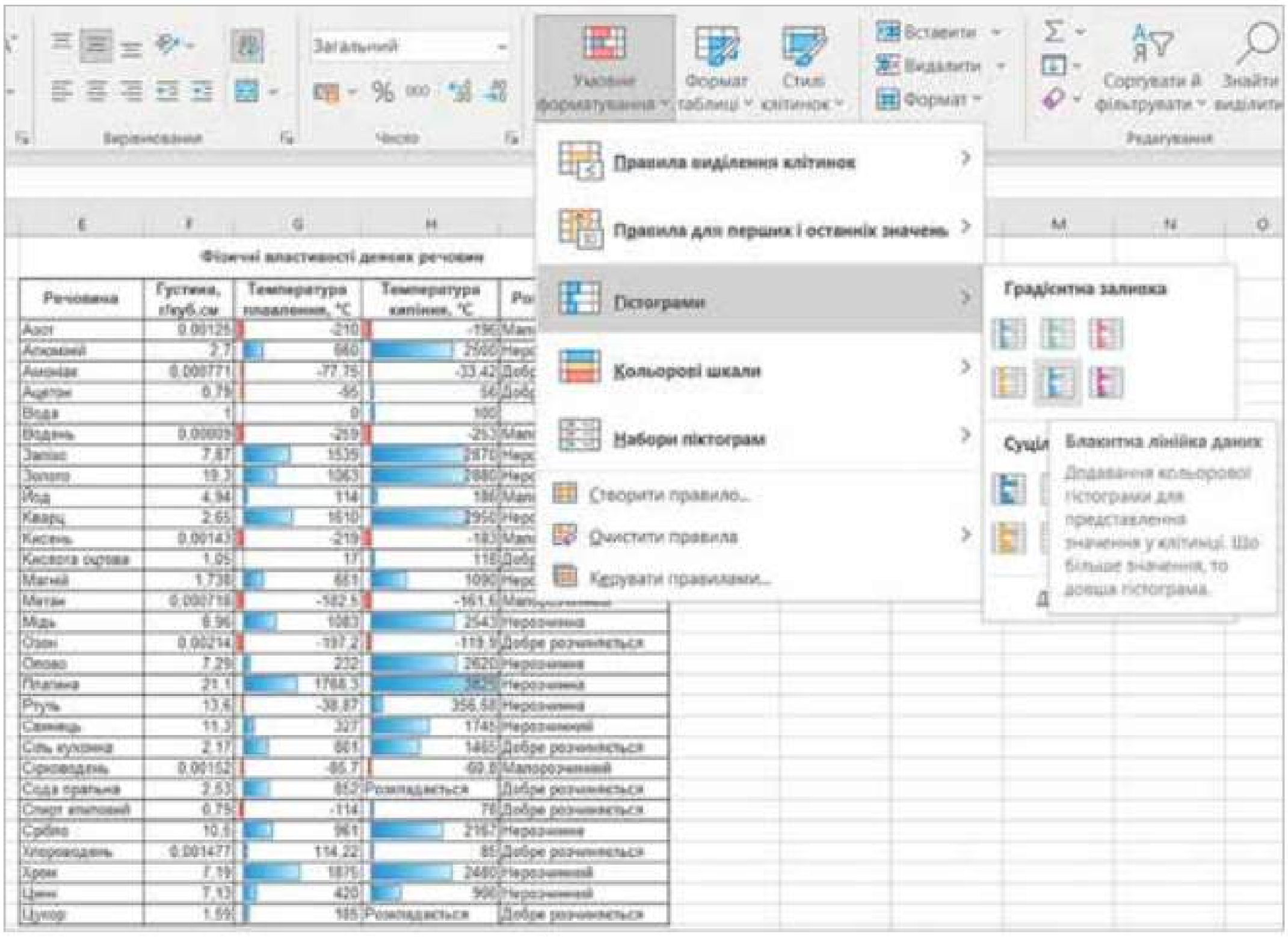
Мал. 6.44. Виділений діапазон клітинок і вікно **Між** з умовою форматування

А на малюнку 6.45 наведено установлення та попередній перегляд умовного форматування **10 останніх елементів** (10 найменших значень) з вибраним форматом *зелена заливка з темно-зеленим текстом*.



Мал. 6.45. Виділений діапазон клітинок, вікно **Останні 10 елементів** з умовою і списком стандартних форматів

Використання іншого правила умовного форматування **Гістограми** приводить до вставлення у клітинки виділеного діапазону гістограм, довжина кольорового рядка яких пропорційна значенню у клітинці, а колір позначає додатні та від'ємні значення (мал. 6.46).



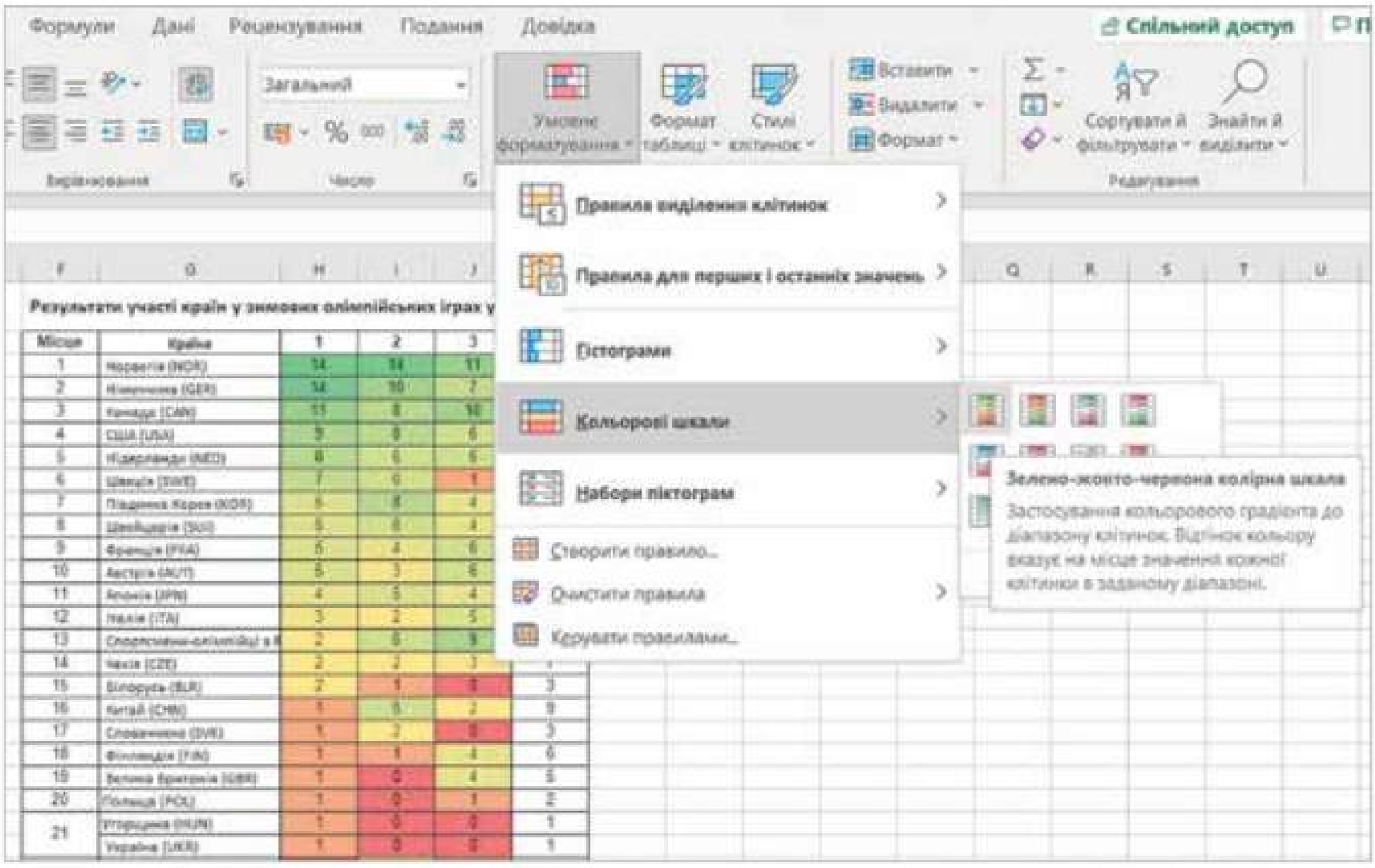
Речовина	Густина, г/куб.см	Температура плавлення, °С	Температура кипіння, °С	Ро
Азот	0.00125	-210	-196	Мал
Алюміній	2.7	660	2500	Нер
Аміак	0.000771	-77.75	-33.42	Доб
Ацетон	0.79	-95	56	Доб
Вода	1	0	100	
Водень	0.00009	-259	-253	Мал
Залізо	7.87	1535	2870	Нер
Золото	19.3	1063	2660	Нер
Йод	4.94	114	186	Мал
Кварц	2.65	1610	2950	Нер
Кисень	0.00143	-219	-183	Мал
Кислота оцтова	1.05	17	118	Доб
Магній	1.738	651	1090	Нер
Метан	0.000716	-182.5	-161.6	Мал
Мідь	8.96	1083	2543	Неро
Озон	0.00214	-197.2	-119.9	Добре розчиняється
Олово	7.29	232	2620	Неро
Платина	21.1	1768.3	3825	Неро
Ртуть	13.6	-38.87	356.58	Неро
Свинець	11.3	327	1745	Неро
Сіль кухонна	2.17	801	1465	Добре розчиняється
Сірководень	0.00152	-85.7	-59.6	Малорозчинний
Сода кальцинована	2.53	851	Розкладається	Добре розчиняється
Спирт етиловий	0.79	-114	78	Добре розчиняється
Срібло	10.5	961	2167	Неро
Углекислий газ	0.001977	114.22	-85	Добре розчиняється
Хром	7.19	1875	2460	Нерозчинний
Цинк	7.13	420	906	Нерозчинний
Цукор	1.59	185	Розкладається	Добре розчиняється

Мал. 6.46. Умовне форматування з уставленням у клітинки гістограм

Використання умовного форматування **Кольорові шкали** приводить до градієнтної заливки клітинок виділеного діапазону встановленими кольорами. Клітинки з однаковими значеннями мають один і той самий колір, а насиченість чи відтінок кольорів для клітинок з різними значеннями змінюється в межах кольорів вибраного формату (мал. 6.47).

Можна також вибрати правило умовного форматування зі списку **Набори піктограм**. За такого форматування у клітинках виділеного діапазону з'являтимуться піктограми з вибраного набору. Поява конкретної піктограми з набору в клітинці означає, що значення в цій клітинці задовольняє умові, встановленій для кожної піктограми набору.

Звертаємо вашу увагу! До одного діапазону клітинок може бути застосовано кілька правил умовного форматування, установлених *послідовно*. У такому разі умови цих правил перевіряються по черзі. І якщо вони не суперечать одна одній, то умовне форматування відбувається послідовно. Наприклад, можливо установити червоний колір символів для чисел, які *більше за 10*, та застосувати для того самого діапазону форматування у вигляді *гістограм*.



Мал. 6.47. Умовне форматування зелено-жовто-червоною колірною шкалою

Для видалення умовного форматування потрібно виконати ***Умовне форматування*** ⇒ ***Очистити правила*** ⇒ ***Очистити правила у виділених клітинках*** або ***Очистити правила у всьому аркуші.***

ДРУКУВАННЯ ЕЛЕКТРОННОЇ ТАБЛИЦІ

Пригадайте

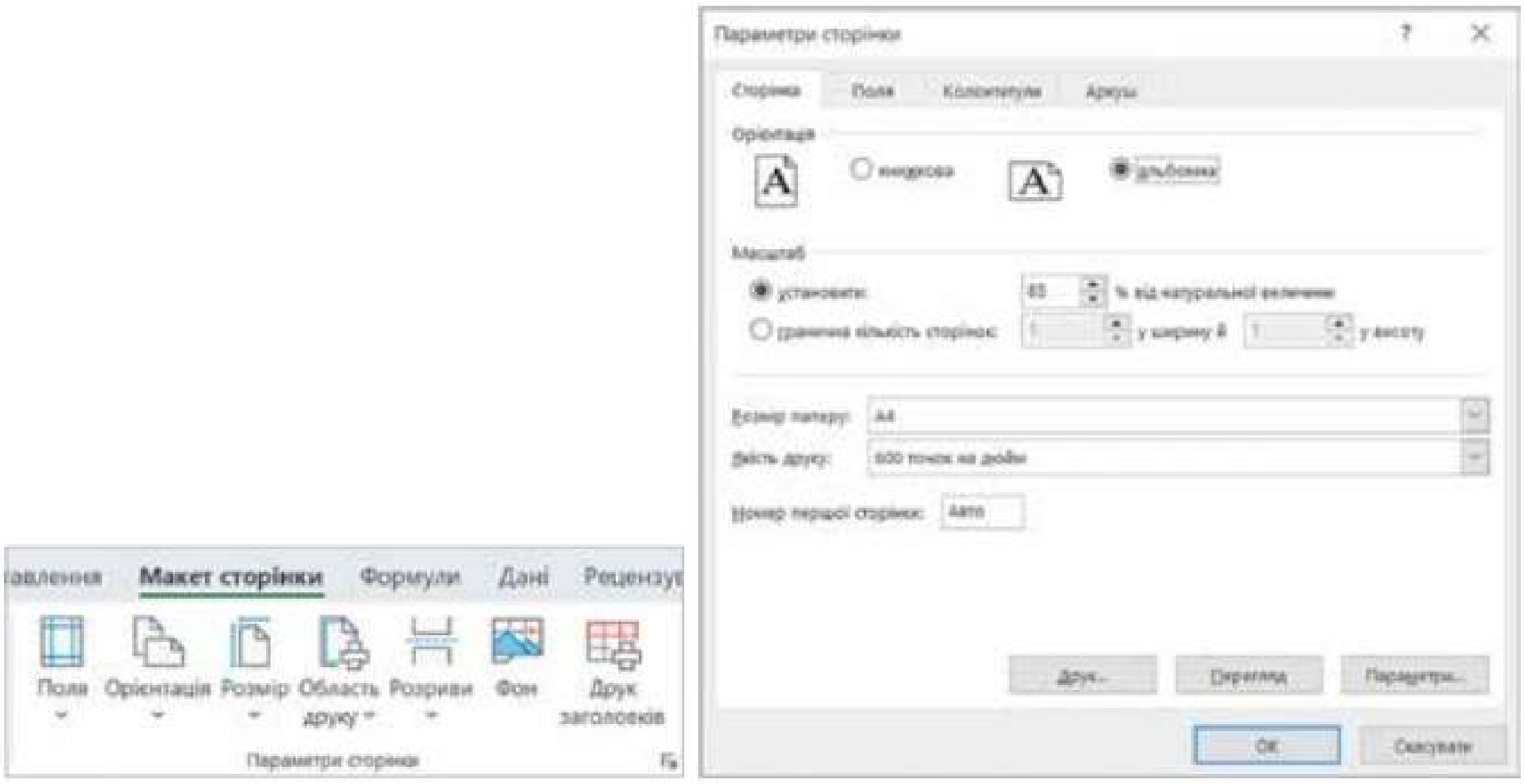
● Які властивості має сторінка текстового документа? Які значення цих властивостей ви можете навести? ● Як надрукувати текстовий документ у текстовому процесорі **Word**?

Ви вже вмієте друкувати текстові документи і знаєте, що перед тим, як надрукувати підготовлений документ, потрібно встановити значення властивостей його сторінок (розмір аркуша, орієнтацію сторінки, ширину полів тощо), потім у режимі **Попереднього перегляду** переглянути зовнішній вигляд документа і надрукувати його.

Схожим є й алгоритм друкування електронної таблиці, але з певними відмінностями.

Установлення значень властивостей сторінки документа, який планується друкувати, здійснюється з використанням елементів керування групи **Параметри сторінки** вкладки **Макет сторінки** або на вкладках діалогового вікна **Параметри сторінки** (мал. 6.48), яке відкривається вибором кнопки **Параметри сторінки** у нижньому правому куті групи **Параметри сторінки.**

На вкладці **Сторінка** можна встановити значення виду орієнтації аркуша паперу під час друкування (*альбомна* або *книжкова*); масштаб



Мал. 6.48. Група **Параметри сторінки** та діалогове вікно **Параметри сторінки**

таблиці у відсотках від натуральної величини, розмір аркуша паперу для друкування; якість друку, номер першої сторінки в надрукованому документі та інші.

На вкладці **Поля** можна встановити значення таких властивостей друкування: розміри полів, висота колонтитулів, спосіб центрування таблиці на аркуші та інші.

На вкладці **Колонтитули** можна створити верхній і нижній колонтитули, вибравши відповідну кнопку, або вибрати потрібний колонтитул у полях зі списком **Верхній колонтитул** або **Нижній колонтитул** тощо.

На вкладці **Аркуш** можна встановити значення таких властивостей друкування:

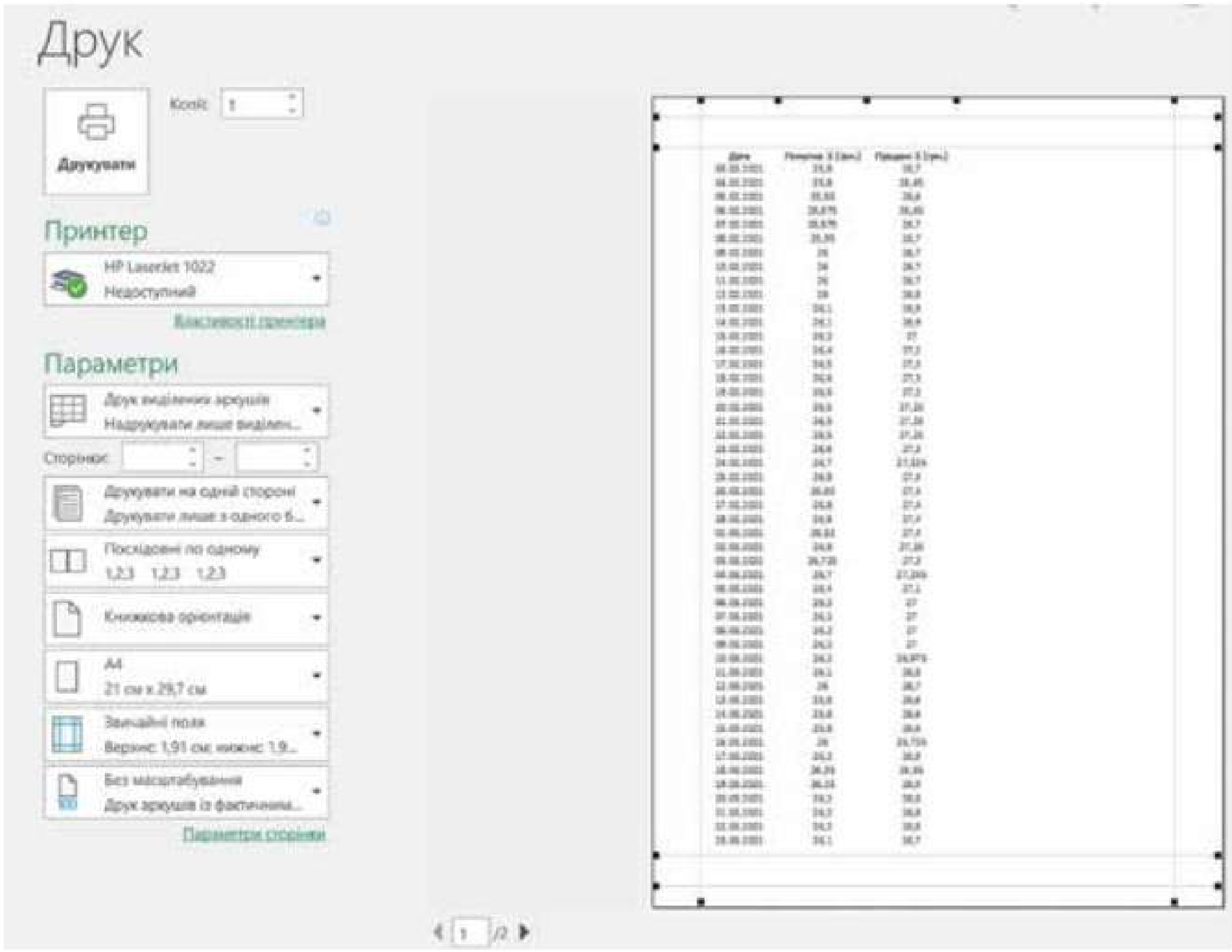
- область друкування – діапазон клітинок аркуша електронної таблиці, який буде виводитися на друк;
- друк наскрізних рядків і стовпців на кожній сторінці – діапазон клітинок, уміст якого друкуватиметься як підписи рядків і стовпців на кожній сторінці, коли надрукована таблиця на папері займає більше ніж одну сторінку;
- якість друку (чернетка, чорно-білий, сітка тощо);
- порядок друку сторінок – вниз, потім управо чи вправо, потім униз

тощо.

Вибором кнопки **Перегляд** у вікні **Параметри сторінки** можна встановити режим **Попереднього перегляду**, щоб побачити, як виглядатиме надрукована електронна таблиця на папері. Це можна зробити також, виконавши ***Файл*** ⇒ ***Друк***.

У цьому режимі буде відображено таблицю, підготовлену до друку, яка розділена на сторінки. Якщо область друкування не вміщується на одному аркуші паперу заданого розміру, табличний процесор **Excel** ав-

томатично розділяє цю область на кілька аркушів (такий випадок наведено на мал. 6.49). Уміст на окремих аркушах паперу можна переглянути, обираючи потрібну сторінку в **Області перегляду** ◀ 1 /2 ▶.



Мал. 6.49. Вкладка **Друк** режиму **Попередній перегляд**

На вкладці **Друк** також можна змінити значення різних властивостей друкування сторінок електронної таблиці – розміри полів, розмір аркуша паперу, орієнтацію аркуша тощо, а також налаштувати параметри друкування – кількість копій, поточний принтер, об'єкт для друкування (виділений діапазон, або виділені аркуші чи всю книгу, або окремі сторінки з указаними номерами), послідовність друку частин таблиці електронної книги, масштаб тощо.

Після того як всі параметри друкування електронної таблиці визначені та попередній вигляд задовольняє користувача, потрібно вибрати на вкладці **Друк** кнопку **Друкувати** .

Для повернення з вкладки **Друк** в основне вікно програми слід вибрати кнопку у верхній частині бічного меню. При цьому на аркуші електронної таблиці з'являються вертикальні та горизонтальні штрихові лінії, які демонструють розбиття таблиці на частини за розмірами аркушів для друку (мал. 6.50).

D	E	F	G	H
Дата	**Покупка $ (грн)**	**Продаж $ (грн)**		
03.02.2021	25,8	26,7		
04.02.2021	25,8	26,65		
05.02.2021	25,85	26,6		
06.02.2021	25,875	26,65		
07.02.2021	25,875	26,7		

Мал. 6.50. Розбиття електронної таблиці на сторінки



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/ae2cKeOo* або QR-кодом.



Найважливіше в цьому пункті

Одним зі способів виділити в таблиці дані, які задовольняють певним умовам, є так зване ***умовне форматування***. Умовне форматування змінює формат клітинки або її вмісту на заданий, якщо дані в цій клітинці задовольняють певній умові.

Умовне форматування застосовується до виділеного діапазону клітинок і виконується на основі правил, які складаються з **умови**, що стосується значення у клітинці, та **значень властивостей** форматування вмісту клітинки.

В **Excel** існує **5 типів** виділення клітинок під час умовного форматування: *виділення клітинок за певними правилами*, *виділення перших або останніх значень діапазону*, *форматування клітинок у вигляді гістограм*, *кольорових шкал* чи *наборів піктограм*.

Для встановлення цих видів правил форматування для виділеного фрагмента електронної таблиці використовуються відповідні команди з меню кнопки **Умовне форматування** 

. До одного діапазону клітинок може бути застосовано кілька правил умовного форматування, заданих послідовно.

Установлення значень параметрів сторінки електронної таблиці для друкування здійснюється на **Стрічці** елементами керування групи **Параметри сторінки** на вкладці **Макет сторінки** або на вкладках діалогового вікна **Параметри сторінки** цієї групи.

Щоб переглянути, як буде виглядати електронна таблиця на папері, потрібно перейти до режиму **Попередній перегляд**, який установлюється ***Файл*** ⇒ ***Друк***. У цьому режимі на вкладці **Друк** можна додатково налаштувати параметри друкування – кількість копій, вибір принтера, об'єкт для друкування (виділений діапазон, або виділені аркуші чи всю книгу, або окремі сторінки з указаними номерами), послідовність друку сторінок електронної книги, масштаб тощо.

Щоб надрукувати електронну таблицю після узгодження параметрів друку, потрібно вибрати кнопку **Друкувати** на вкладці **Друк**.



Дайте відповіді на запитання

1. Що таке *умовне форматування*? Із чого складаються правила умовного форматування?

2. Які типи умовного форматування існують в **Excel**?

3. Як виконати умовне форматування? Як очистити умовне форматування на всьому аркуші?

4. Як надрукувати електронну таблицю?

5. Для чого використовують попередній перегляд сторінок електронної таблиці? Які дії можна виконувати в цьому режимі?

6. Які властивості друкування електронної таблиці можна встановити?



Обговоріть і зробіть висновки

1. Що спільного та відмінного між умовним форматуванням та фільтрацією?

2. Як буде відформатовано діапазон клітинок, якщо до нього застосовують два різних правила, які між собою несумісні? Перевірте це на практиці і сформулюйте загальне правило виконання умовного форматування.

3. Як ви вважаєте, що потрібно зробити, щоб надрукувати всю електронну таблицю на одному аркуші паперу?

4. Що спільного та відмінного у друці текстового документа та електронної таблиці?



Виконайте завдання

1. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.4\зразок 6.4.1.xlsx**. На аркуші **Аркуш1** наведено таблицю з даними про монітори. Установіть рожевий колір тла для клітинок, у яких діагональ монітора менша ніж 21″. Установіть зелену заливку з темно-зеленим текстом у клітинках семи найдорожчих моніторів. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.4.1**.

2. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.4\зразок 6.4.2.xlsx**. На аркуші **Аркуш1** наведено таблицю з даними про зовнішні накопичувачі. Установіть *синій* колір фону для клітинок, у яких ціна *більша за 2000 грн*. Установіть *блідо-рожеву* заливку клітинок, ємність у яких *менша від середньої*. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.4.2**.

3. Знайдіть в інтернеті прогноз погоди на 10 днів для м. Києва. Створіть електронну таблицю та вставте знайдені дані на **Аркуш1**. Установіть умовне форматування (*жовта заливка з темно-жовтим текстом*) для *повторюваних* значень температури вдень. Установіть умовне форматування **Гістограми** *(фіолетова лінійка даних)* для значень температури вночі. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.4.3**.

4. Знайдіть в інтернеті дані про 10 найбільших гір України. Створіть електронну таблицю та вставте знайдені дані на **Аркуш1**. Установіть умовне форматування з використанням **Наборів піктограм** *(4 світлофори)* для числових даних таблиці. Виділіть *червоним кольором* назви гір, у яких міститься літера Г. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.4.4**.

5. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.4\зразок 6.4.5.xlsx**. Виконайте попередній перегляд сторінок перед друкуванням електронної таблиці з аркуша **Аркуш1**. Виконайте друкування першої сторінки цієї електронної таблиці з такими значеннями властивостей друкування: кількість копій – 2, орієнтація аркуша – книжкова, поля – по 1 см, сторінки за замовчуванням. Збережіть книгу у вашій папці у файлі з іменем **завдання 6.4.5**.



Практична робота № 9. «Вибирання даних в електронних таблицях»

Увага! *Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.*

1. Відкрийте вказаний учителем / учителькою файл, наприклад **Розділ 6\Пункт 6.4\зразок практична 9.xlsx.**
2. На аркуші **Аркуш1** виконайте сортування таблиці за зростанням значень у стовпці **Фірма, що взяла кредит.**
3. На аркуші **Аркуш2** виконайте сортування таблиці за значеннями в кількох стовпцях в указаному порядку: **Банк-кредитор** (за зростанням), **Сума кредиту** (за зростанням), **Річний відсоток** (за спаданням).
4. На аркуші **Аркуш3**, використавши фільтрування, відобразіть дані про фірми, сума кредиту яких більша за 750 000, а річний відсоток менший ніж 15 %.
5. На аркуші **Аркуш4**, використавши фільтрування, відобразіть дані про фірми, які взяли кредит на 6 місяців або рік в банку «Ажіо».
6. На аркуші **Аркуш5** установіть умовне форматування для клітинок стовпця **Сума кредиту**, якщо їх значення більше за середнє у стовпці, вибравши для них зелений колір заливки.
7. Збережіть електронну книгу у вашій папці у файлі з іменем **практична 9.xlsx.**



Готуємось до вивчення нового матеріалу

1. Що таке *комп'ютерна мережа*? Які її складові? Для чого вони призначені?
2. Скільки пристроїв підключено до інтернету у вас удома? Поясніть, навіщо це зроблено.
3. Чи чули ви термін *«розумний» дім*? Як ви його розумієте?



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Опрацювання даних в електронних таблицях»** *(початковий, середній, достатній, високий рівень).*

- Я розумію призначення та функції табличного процесора для опрацювання числових даних.
- Я знаю типи об'єктів електронної таблиці, їх властивості та можливі операції над ними.
- Я знаю формати числових даних і використовую їх під час обчислень.
- Я вмію добирати та аргументувати вибір засобів табличного процесора для розв'язування певного кола навчальних і життєвих задач.
- Я знаю різні типи посилань на клітинки та використовую їх для обчислень.
- Я вмію обрати спосіб візуального подання даних засобами табличного процесора та застосовувати його.
- Я вмію створювати електронні таблиці для реалізації різних моделей об'єктів, явищ і процесів.
- Я вмію використовувати табличний процесор для опрацювання числових даних, отримання результатів з використанням вбудованих функцій (математичних, статистичних, логічних тощо).
- Я вмію виконувати сортування та фільтрацію даних.
- Я вмію застосовувати умовне форматування.
- Я знаю значення властивостей друкування та вмію їх налаштовувати під час друку.

SMART–ТЕХНОЛОГІЇ. ШТУЧНИЙ ІНТЕЛЕКТ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- «розумні» речі та Smart-технології;
- інтернет речей та основні принципи його функціонування;
- штучний інтелект;
- використання штучного інтелекту в різних галузях.

7.1. ІНТЕРНЕТ РЕЧЕЙ І ЙОГО ВИКОРИСТАННЯ. SMART-ТЕХНОЛОГІЇ

У цьому пункті йтиметься про:

- «розумні» речі та Smart-технології;
- інтернет речей;
- основні принципи функціонування мережі «Інтернет речей».

«РОЗУМНІ» РЕЧІ ТА SMART-ТЕХНОЛОГІЇ

Пригадайте

● Що ви знаєте про «розумні» пристрої, «розумний» дім? Як ви розумієте ці терміни? ● Чи є у вас «розумні» пристрої? Для чого ви їх використовуєте?

Раніше на уроках інформатики ви вже ознайомилися з поняттям **«розумні»** пристрої, або **Smart-речі** (англ. *Smart* – розумний, енергійний, кмітливий). Наприклад, «розумний» годинник з підтримкою функції відслідковування серцевого ритму, фітнес-трекер, Smart-окуляри, окуляри віртуальної реальності тощо (мал. 7.1).



Мал. 7.1. Приклади «розумних» речей

Ці гаджети мають невеликі розміри та малу вагу, їх зручно носити із собою. Вони можуть мати камеру, термометр, барометр, компас, GPS-навігатор. Також їх можна під'єднати до локальної мережі чи інтернету для збирання й передавання даних, які вони отримують. Через інтернет за допомогою спеціальних програм ними можна керувати віддалено (дистанційно), що сприяє прийняттю

оперативних рішень, організації особистого життя та спілкування тощо.

Наприклад, «розумна» годівниця призначена для віддаленого керування годуванням тварин і керується через додаток у смартфоні. Годівниця видає встановлену їжу в певний час і повідомляє хазяїна про її закінчення. Пластир-градусник, який через певні інтервали часу визначає температуру тіла людини, передає дані на смартфон для аналізу лікарем або пацієнтом у спеціальній програмі.

З використанням «розумних» пристроїв створюють «розумні» будинки, оснащуючи оселю різноманітними датчиками та сенсорами. Пристрої збирають дані та передають їх через інтернет у спеціальну програму керування «розумним» домом на сервері, смартфоні господарів або пульті керування (мал. 7.2).



Мал. 7.2. Функціонування системи «розумний» дім

Ця система, наприклад, аналізує дані навколишнього середовища і, залежно від показників, регулює температуру в приміщенні. У зимовий період регулює інтенсивність опалення, а в разі спекотної погоди – вмикає кондиціонер. За допомогою розумних пристроїв у домі може працювати охоронна та пожежна сигналізація, налаштовуватися використання електроенергії із сонячних батарей.

Також власник «розумного» будинку може з мобільного телефона через інтернет керувати різними пристроями у своєму домі. Наприклад, включити полив трави, відчинити / зачинити двері, включити / виключити пральну машинку тощо.

Водночас украй важливим у світі «розумних» пристроїв постає питання **інтернет-безпеки.** Якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, замість

користі вони можуть завдати шкоди. Оскільки такі пристрої можуть зберігати багато інформації про їх власника, зокрема можуть знати його точне місцезнаходження. Доступ кіберзлочинців до такої інформації і до комп'ютерної системи керування може призвести до порушення інформаційної безпеки та скоєння злочину, а в гірших випадках і нести загрозу життю та здоров'ю людей.

Усі такі технології, які забезпечують взаємодію цифрових пристроїв з навколишнім середовищем і наділяють цю систему здатністю до прийняття рішень, віддаленого керування задля ефективного виконання щоденних справ, побутових і виробничих завдань, називають **Smart-технологіями.**

На основі розвитку цих технологій виникають нові поняття: *Smart-міста, Smart-країни, Smart-освіта, Smart-економіка, Smart-суспільство* (мал. 7.3). В основі цього «розумного» суспільства лежить розвиток «суспільства знань», цифрових технологій, усього того, що приведе до цифрової ери розвитку нашої цивілізації.



Мал. 7.3. «Розумний» світ

ІНТЕРНЕТ РЕЧЕЙ

Пригадайте

● Що таке *комп'ютерна мережа*? Які пристрої можуть бути приєднані до комп'ютерної мережі? ● Що таке *інтернет*? Яким чином комп'ютерні пристрої підключаються до інтернету?

«Розумні» пристрої, «розумні» будинки є прикладами реалізації найпопулярнішої наукової ідеї сучасної інформатики, яка зараз активно втілюється в життя, – **інтернету речей**. Ці технології кардинально впливають на розвиток сучасного суспільства, оскільки забезпечують можливість багатьом процесам відбуватися автоматично, без безпосередньої участі людини, але під її контролем.

Інтернет речей (англ. *Internet of Things*, скорочено ***IoT***) – це сукупність пристроїв, оснащених датчиками, сенсорами, засобами опрацювання, передавання та прийому сигналів, підключених до глобальної мережі «Інтернет» з метою передавання власних даних та отримання команд.

У такому цифровому середовищі створюються нові умови для бізнесу, охорони здоров'я, забезпечення екологічної безпеки, змінюються особисті та соціальні аспекти життя (мал. 7.4).



Мал. 7.4. Інтернет речей і його використання

Так, уже зараз лікарі з використанням переносних датчиків, які встановлено на пацієнті, можуть віддалено відслідковувати його стан

здоров'я та реагувати на зміни в режимі реального часу. Автодорожні служби встановлюють спеціальні датчики на автомобільних шляхах для відслідковування швидкості автомобілів, автоматичної фіксації порушень і відправлення інформації про цей факт у патрульну службу та водієві.

Для запобігання лісовим пожежам використовують мережу «Лісовий дозор». У лісах установлюють систему датчиків температури та відеокамери, дані з яких передаються у програму для аналізу ситуації та сповіщають чергових у разі пожежі.

Наразі широкої популярності набули моніторинг стану водного середовища, моніторинг атмосферного повітря, а також сейсмічної активності. Використовуються датчики, які вимірюють уміст вуглекислого газу, вологість повітря, кількість радіоактивних елементів і передають ці дані на обробку в центри керування. Такі пристрої можуть знаходитися в різних куточках планети, у тому числі у важкодоступних місцях, і працювати через Wi-Fi, Bluetooth, GPS.

Підключення пристроїв до інтернету дає людям більше можливостей для раціонального управління ресурсами: витрачання газу, води, світла, видобутку різних видів енергії тощо. Наприклад, стало холодніше – вмикається опалення в усіх будинках міста із центральним опаленням, стало тепліше – вимикається. Увечері зменшився рівень освітленості на вулиці, то відповідно вмикається вуличне освітлення; світлофори переходять у режим «увага» в певний час доби, коли зменшується потік автомобілів на дорозі.

Автоматизація вирішення в такий спосіб багатьох життєвих потреб і виробничих завдань, оперативного ухвалення рішень надає можливість людині більше часу приділяти сім'ї, творчості, хобі, професійному та особистому розвитку.

Цікаві факти з історії

Термін *інтернет речей* уперше використав американський фахівець з телекомунікацій Пітер Льюїс у 1985 році. Він описав **IoT** як «інтеграцію людей, процесів і технологій з пристроями для підключення і датчиками для забезпечення віддаленого моніторингу стану, маніпулювання та оцінки тенденцій таких пристроїв».

Пізніше, у 1999 році, термін *інтернет речей* підтримав і засновник дослідницького центру **Auto-ID Center** у Массачусетському технологічному інституті Кевін Ештон (нар. 1968 р.) (мал. 7.5). Він висловив припущення, що згодом у кожної з речей реального фізичного світу в **IoT** буде цифровий двійник, її віртуальна копія.



Мал. 7.5. Кевін Ештон

Напрям **IoT** став активно розвиватися, коли на початку 2000-х років кількість пристроїв, підключених до мережі «Інтернет», перевищила кількість користувачів інтернету. Тобто *інтернет речей* перевищив *інтернет людей*.

Період з 2008 по 2009 роки аналітики корпорації **Cisco** вважають «справжнім народженням інтернету речей», тому що саме в цей період кількість пристроїв, підключених до глобальної мережі, перевищила чисельність населення Землі. А за даними компанії **Ericsson**, у 2018 році кількість таких пристроїв перевищила кількість мобільних телефонів у світі. У 2021 році кількість **IoT**-пристроїв перевищила 12 млрд одиниць, а до 2025 року може досягти 41 млрд. І надалі їх кількість буде зростати.

У зв'язку з такими глобальними масштабами використання **IoT** набуває поширення новий термін – **всеохопний інтернет**, **інтернет усього** (англ. ***Internet of Everything, IoE***), який із часом може прийти на зміну **IoT**.

ОСНОВНІ ПРИНЦИПИ ФУНКЦІОНУВАННЯ МЕРЕЖІ «ІНТЕРНЕТ РЕЧЕЙ»

Пригадайте

- Що таке *IP-адреса комп'ютера*? Із чого вона складається?
- Що таке *інформаційна система*? Які її складові?

Інтернет речей ґрунтується на таких базових **принципах:**

- наявність глобальної комп'ютерної мережі з високою швидкістю;
- до мережі під'єднуються різноманітні пристрої, які мають конкретну IP-адресу в мережі;
- пристрої мають можливість отримувати дані та відправляти їх по мережі;
- робота пристроїв у автоматизованому режимі відбувається за спеціальними програмами.

На малюнку 7.6 наведено загальну схему функціонування мережі «Інтернет речей».



Мал. 7.6. Функціонування мережі «Інтернет речей»

Для відслідковування стану навколишнього середовища пристрої оснащують ***датчиками (сенсорами)***, які отримують інформацію про зовнішні умови в реальному часі. Наприклад, датчики для вимірювання температури, тиску, швидкості руху, місця розташування, вологості та інші. У більшості випадків вони мають автономні джерела живлення.

Датчик вимірює фізичні параметри деякого об'єкта / явища та перетворює їх у цифровий сигнал, який може бути переданий і прийнятий відповідним комп'ютерним пристроєм. Для опрацювання та накопичення даних з датчиків використовують ***вбудовані комп'ютерні пристрої***, до яких датчики приєднуються через *контролери*. ***Контролери*** – це спеціальні пристрої, що з'єднують елементи системи один з одним і зв'язують систему із зовнішнім світом.

Якщо ж опрацювання отриманих даних планується на локальному комп'ютері, смартфоні, сервері, то для обміну даними між пристроями використовують різні ***канали передавання даних*** (*бездротові*

або *дротові мережі*). Водночас важливим є забезпечення надійності та конфіденційності передавання даних.

Для аналізу даних і прийняття рішення комп'ютерні пристрої використовують відповідне ***програмне забезпечення***, яке за певними алгоритмами опрацьовує отримані дані. Ці програми можуть установлюватися на мобільні пристрої, у вбудовані комп'ютери або використовувати хмарні технології.

Результати обробки отриманих даних у вигляді відповідних сигналів поступають на так звані ***виконуючі пристрої*** (*актуатори, кінцеві пристрої*). Це цифрові пристрої, які безпосередньо виконують команди, наприклад автоматичні вимикачі, підйомники, розетки, сирени, клапани, двигуни тощо.

Більшість операцій в інтернеті речей відбувається без участі людини, але при цьому **користувач** може взаємодіяти з будь-яким пристроєм: віддавати команди, інструкції, отримувати доступ до інформації.

Зручність, економічність, різноманітність варіантів використання **IoT** робить інтернет речей найважливішою технологією XXI століття, яка буде продовжувати набирати обертів. У міру того, як дедалі більше компаній усвідомлюють потенціал інтернету речей, збільшуватиметься кількість підключених пристроїв і продовжиться розвиток функціональних можливостей.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Re2cKlsk* або QR-кодом.



Найважливіше в цьому пункті

«**Розумні**» пристрої, або **Smart-речі**, – це цифрові пристрої, які під'єднують до локальної мережі чи інтернету для збирання й передавання даних з використанням різноманітних датчиків. Вони мають невеликі розміри й незначну масу, їх зручно носити із собою.

З використанням «розумних» пристроїв створюють «**розумні**» будинки. Датчики та сенсори цих пристроїв збирають дані та передають через інтернет у спеціальну програму керування «розумним» домом на сервері, смартфоні господарів або пульті керування.

Під час використання «розумних» пристроїв важливим є питання **інтернет-безпеки**. Якщо елементи такої системи не будуть належним чином захищені від несанкціонованого втручання, замість користі вони завдадуть шкоди.

Технології, які забезпечують взаємодію цифрових пристроїв з навколишнім середовищем і наділяють цю систему здатністю до прийняття рішень, віддаленого керування задля ефективного виконання щоденних справ, побутових і виробничих завдань, називаються **Smart-технологіями**.

На основі розвитку цих технологій виникають нові поняття: *Smart-міста*, *Smart-країни*, *Smart-освіта*, *Smart-економіка*, *Smart-суспільство*. В основі цього «розумного» суспільства лежить розвиток «суспільства знань», цифрових технологій, усього того, що приведе до цифрової ери розвитку нашої цивілізації.

Інтернет речей (англ. *Internet of Things*, скорочено ***IoT***) – це сукупність пристроїв, оснащених датчиками, сенсорами, засобами опрацювання, передавання та прийому сигналів, підключених до глобальної мережі «Інтернет» з метою передавання власних даних і отримання команд.

У такому цифровому середовищі створюються нові умови для бізнесу, охорони здоров'я, забезпечення екологічної безпеки, змінюються особисті та соціальні аспекти життя. Виникають можливості автоматизації вирішення багатьох життєвих потреб і виробничих завдань, оперативного ухвалення рішень.

Інтернет речей ґрунтується на таких базових **принципах**:

- наявність глобальної комп'ютерної мережі;
- до мережі під'єднуються різноманітні пристрої (датчики, контролери, актуатори та інші), які мають конкретну IP-адресу в мережі;
- пристрої мають можливість отримувати дані та відправляти їх по мережі;
- робота пристроїв у автоматизованому режимі відбувається за спеціальними програмами.

Зручність, економічність, різноманітність варіантів використання **IoT** робить інтернет речей найважливішою технологією XXI століття, яка буде продовжувати набирати обертів.

Дайте відповіді на запитання

1. Які ви можете навести приклади використання «розумних» речей? Поясніть їх функціонування.

2. Що розуміють під *Smart-технологіями*? Чи використовують такі технології у вашому населеному пункті?

3. У чому сутність питання *інтернет-безпеки* під час використання «розумних» пристроїв?

4. Що таке *інтернет речей*? Із чого він складається?

5. Які базові принципи функціонування інтернету речей?

6. Як функціонує мережа «Інтернет речей»?

Обговоріть і зробіть висновки

1. Яке значення має інтернет речей у житті людей?

2. Які негативні аспекти інтернету речей ви можете назвати? Чим це може нашкодити суспільству; людям; природі?

3. Яким чином інтернет речей змінює спосіб життя людей?

4. Чи можна вважати інтернет речей, «розумний» будинок інформаційними системами? Відповідь поясніть.

Виконайте завдання

1. Перегляньте в інтернеті відеоролик про інтернет речей за адресою *https://cutt.ly/aeLdRwVE* або QR-кодом. Підготуйте на основі відео презентацію про використання інтернету речей і «розумних» пристроїв. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.1.1**.

2. Знайдіть в інтернеті інформацію про використання інтернету речей у різних сферах економіки. Підготуйте групове повідомлення у вигляді відеоролика. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.1.2**.

3. Опишіть функціонування «розумних» систем на наведених прикладах у текстовому документі. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.1.3**.

а) Фітнес-браслет



б) «Розумний» завод



4. Знайдіть в інтернеті інформацію про використання Smart-технологій в освіті та навчанні. Підготуйте колективну презентацію на тему «Smart-освіта сьогодні», використовуючи онлайн-редактор презентацій **Google**. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.1.4.**

5. Знайдіть в інтернеті інформацію про створення в різних країнах світу сучасних Smart-міст. Оформте результати роботи в текстовому документі у вигляді буклета та збережіть його у вашій папці у файлі з іменем **завдання 7.1.5.**

6. Ознайомтесь з відео щодо підключення в домі «розумної» розетки за посиланням *https://cutt.ly/EeLdTKU3*. За можливості спробуйте встановити її у вашому домі спільно з батьками чи рідними. Зніміть відео, як ви це робили. Які зручності та переваги принесе цей пристрій вашій сім'ї? Поясніть це у своєму відеозвіті.

7. Об'єднайтесь у групи по 4–5 учасників / учасниць. Виконайте міждисциплінарний проєкт «Розумна теплиця» за прикладом завдання № 1 у рубриці «Працюємо з комп'ютером». Для цього використайте свої знання з фізики, технологій, інформатики тощо.

8. Виконайте проєкт «Наша Smart-школа», у якому передбачте підключення різноманітних датчиків для автоматизації функціонування школи. Складіть схему реалізації вашого проєкту. Розрахуйте орієнтовний кошторис реалізації такого проєкту, використавши ціни на обладнання з інтернет-магазинів. Збережіть результати роботи у вашій папці у файлі з іменем **завдання 7.1.8**.

9. Виконайте проєкт «Мій "розумний" дім», у якому передбачте підключення різноманітних датчиків для автоматизації функціонування будинку / квартири. Складіть схему реалізації вашого проєкту. Розрахуйте орієнтовний кошторис реалізації такого проєкту. Збережіть результати роботи у вашій папці у файлі з іменем **завдання 7.1.9**. Виконайте завдання разом з батьками чи рідними. Під час виконання проєкту скористайтесь інформацією із сайту за адресою *https://220lviv.top/rozumnyj-budynok/*.

10. Перегляньте статтю за посиланням *http://surl.li/jiqily* та ознайомтеся з прикладами використання інтернету речей у майбутньому. Визначте, які перспективи розвитку суспільства та життя людей у новому цифровому світі. Створіть презентацію. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.1.10**.

11. Відкрийте сторінку онлайн-курсу **Вступ до IoT та цифрової трансформації** від мережної академії **Cisco** за адресою *http://surl.li/usbmjf*. Ознайомтеся зі змістом курсу, зареєструйтеся в ньому та пройдіть навчання. Після завершення покажіть свій сертифікат учителю / учительці.

Готуємось до вивчення нового матеріалу

1. Що таке, на вашу думку, *штучний інтелект*? Чи використовували ви його? Для яких цілей?
2. Які ви бачите переваги та недоліки використання штучного інтелекту в школі?

7.2. ПОНЯТТЯ ШТУЧНОГО ІНТЕЛЕКТУ. ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В РІЗНИХ ГАЛУЗЯХ

У цьому пункті йтиметься про:

- штучний інтелект і напрями розробок систем штучного інтелекту;
- використання штучного інтелекту в різних галузях;
- переваги та недоліки використання штучного інтелекту;
- штучний інтелект у навчанні.

ШТУЧНИЙ ІНТЕЛЕКТ



Поміркуйте

● Що, на вашу думку, означає слово *розум*? Яку людину вважають *розумною*? ● Що таке *навчання*? Як навчаєтеся ви? ● Що таке *алгоритм*; *математична модель*? Де й коли ви використовуєте ці терміни?

Одним з найсучасніших, найпрогресивніших і найнеоднозначніших напрямів розвитку інформатики як науки є штучний інтелект (ШІ).

Про це свідчить хоча б той факт, що у 2024 році дві з трьох наукових нагород Нобелівської премії отримали за відкриття в галузі штучного інтелекту – у сфері фізики та хімії.

Перші згадування про штучний інтелект можна знайти ще у стародавньому світі в міфах про створення штучних істот, яких майстри наділили розумом чи свідомістю.

Інтелект (лат. *intellectus* – відчуття, сприйняття, розуміння, розум) – здатність до пізнання і подолання труднощів, яка об'єднує всі пізнавальні здібності людини: відчуття, сприйняття, пам'ять, уявлення, мислення.

Сьогодні використання ШІ стало невід'ємною частиною життя сучасної людини. Сканер відбитків пальців, розпізнавання обличчя в телефоні, введення тексту за допомогою автонабору, спілкування із чат-ботом у онлайн-магазині, голосові помічники у смартфоні, система «Розумний будинок», автопілоти в машинах, сервіс **Google Maps**, електронний додаток таксі Uber – усе це функціонує на основі штучного інтелекту.

Штучний інтелект (англ. *artificial intelligence*) – це область інформатики, яка здійснює розробку інтелектуальних комп'ютерних систем і програм, що імітують роботу людського мозку.

Такі інтелектуальні системи та програми запрограмовані виконувати певні дії, як люди: розуміти мову, навчатися, міркувати, робити висновки та передбачати, розв'язувати проблеми, ухвалювати рішення тощо. Інакше кажучи, у рамках штучного інтелекту вивчають способи розв'язування комп'ютером задач, що не мають явного алгоритмічного розв'язку.

Сьогодні створення систем штучного інтелекту ведеться за такими напрямами:

1. ***Створення комп'ютерних систем, що імітують діяльність людини*** (наприклад, емоції, мовлення, жести, відчуття, творчість тощо). Це так звані програмно-апаратні асистенти, помічники, що заміняють людей під час виконання деяких робіт.

Наприклад, людиноподібного робота Софію (мал. 7.7) було створено в 2015 році гонконгською компанією Hanson Robotics у вигляді жінки. Вона наділена штучним інтелектом, здатна імітувати 62 вирази обличчя, установлювати зоровий контакт, розпізнавати обличчя, запам'ятовувати людей



Мал. 7.7. Людиноподібний робот Софія

і підтримувати з ними діалог. Вона постійно вдосконалюється, стаючи «розумнішою».

2. *Створення комп'ютерних систем на основі використання біологічних елементів* (наприклад, *нейрокомп'ютер, біокомп'ютер*). Нейрокомп'ютер розробляється для прямого зв'язку між мозком людини та зовнішніми пристроями.

Наприклад, керування пристроями силою думки через уживлені в мозок датчики можна використати для людей з обмеженими фізичними можливостями. Датчики, які при цьому використовуються, реагують на сигнали мозку, декодують їх і перетворюють у спеціальні команди для комп'ютерного пристрою (мал. 7.8).



Мал. 7.8. Елементи нейрокомп'ютера

3. *Створення комп'ютерних систем, які імітують логічне мислення людини, на основі використання систем логічного програмування* (наприклад, мов програмування **Пролог, Ліспа** та інших). Такі системи ухвалення рішень (*експертні системи*) використовують для опрацювання великих неструктурованих даних з урахуванням багатьох факторів (діагностування).

Однією з перших логічних мов програмування був **Пролог** (1972 рік), який і залишається на сьогодні найпопулярнішою серед мов для розробки так званих експертних систем. Програма на **Пролозі** не є такою у традиційному розумінні, оскільки не містить базових алгоритмічних конструкцій. Основне призначення програми – не виконати послідовність дій і отримати потрібний результат, а знайти рішення на основі аналізу набору фактів за заданими правилами побудови логічного висновку (мал. 7.9).

4. *Створення комп'ютерних систем, які будуть так званими інтелектуальними агентами*, що сприймають навколишній світ за допомогою датчиків і впливають на об'єкти в навколишньому середовищі за допомогою деяких механізмів.

Наприклад, автоматизовані блоки (мал. 7.10) керування в літаку, автомобілі чи БпЛА, які можуть в автономному режимі без участі людини керувати рухомим об'єктом, виконувати певні дії, аналізуючи при цьому швидкість, траєкторію, перешкоди та інші зовнішні фактори. Або це може бути пошуковий робот, який збирає для користувача



Мал. 7.9. Схема отримання висновку на мові **Пролог**

інформацію за потрібним запитом, аналізуючи дані в мережі «Інтернет» або на визначених носіях, обирає найбільш доречні та зберігає відібрані дані в указаному місці або повідомляє користувачу.





Мал. 7.10. Автоматизовані блоки керування та пристрої – автопілот метро, літальний апарат

Для тих, хто хоче знати більше

Існує кілька основних типів алгоритмів, за якими працює штучний інтелект: *машинне навчання*, *глибинне навчання*, *генетичні алгоритми* тощо. Одним з ключових алгоритмів є **машинне навчання**, яке надає комп'ютерній системі алгоритми, як на основі великих обсягів даних робити висновки, ймовірнісні умовиводи, знаходити закономірності, без їх явного програмування на такий вид завдань. Це так званий *етап навчання ШІ*. І ці ж отримані результати в подальшому система ШІ використовує вже як свої знання та отриманий досвід. Отримуючи нові вхідні дані, система застосовує до них свої знання та досвід і може сформулювати на цій основі свої висновки, свої рішення. Це другий етап – *використання ШІ*. Таким чином інтелектуальна система отримує нові знання, тобто вчиться так, як це робить людина. Унаслідок цього система стає продуктивнішою, ефективнішою та точніше опрацьовує дані (мал. 7.11).



Мал. 7.11. Як працює машинне навчання

Основою систем штучного інтелекту є **нейронні мережі**, які імітують роботу людського мозку в цифровому вигляді. Вони складаються із цифрових нейронів, які обробляють інформацію та передають її далі через свої зв'язки по цій мережі.

У реалізації алгоритмів роботи систем штучного інтелекту лежать певні математичні моделі, кожна з яких має свої особливості та застосування:

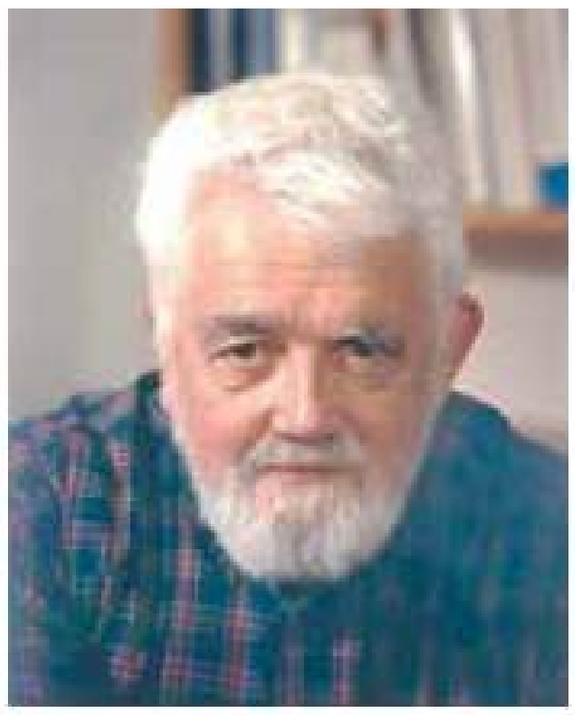
- *мовні моделі* – використовують для обробки та генерації тексту;
- *генеративні моделі* – здатні створювати новий контент (зображення, текст або музику);
- *дискримінативні моделі* – використовують для класифікації та розпізнавання об'єктів.

Цікаві факти з історії

Активні розробки зі створення ШІ розпочалися в середині 50-х років XX ст. Влітку 1956 року в Університеті Дартмута в США пройшла перша робоча конференція науковців з проблем штучного інтелекту. Саме тоді і з'явився сам термін «*штучний інтелект*», який увів Джон Маккарті (американський інформатик, 1927–2011) на тій самій конференції.

У цей період почали створювати перші інтелектуальні системи, розробляти алгоритми їх роботи, але відсутність потужних комп'ютерів не давала можливості отримати вагомих результатів. Потім фінансування розробок у галузі ШІ було призупинено на довгий період часу, настала «Зима ШІ».

Інвестиції та інтерес до штучного інтелекту відродилися на початку XX ст., коли для опрацювання величезних обсягів даних у багатьох задачах науки та промисловості стали успішно застосовувати потужні та надпотужні комп'ютери. Це, своєю чергою, забезпечувало розробку методології роботи інтелектуальних систем, створювалися, удосконалювалися та розповсюджувалися моделі нейронних мереж для різних сфер застосування, накопичувалися датасети для навчання мереж, розроблялися нові алгоритми ШІ.



Мал. 7.12.
Джон Маккарті

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В РІЗНИХ ГАЛУЗЯХ ДІЯЛЬНОСТІ ЛЮДИНИ

Поміркуйте

● Чи використовували ви онлайн-перекладачі для перекладів тексту? Онлайн-карти для побудови маршруту переміщення? ● Скільки разів ви сьогодні використовували «розумні» пристрої, штучний інтелект? Наведіть приклади.

2 грудня 2020 року в Україні Розпорядженням Кабінету Міністрів України № 1556-р було схвалено Концепцію розвитку штучного інтелекту в Україні *http://surl.li/orqejn*, яка передбачає визначення основних напрямів і пріоритетних завдань розвитку технологій штучного інтелекту з метою забезпечення конкурентоспроможності національної економіки та захисту технологічних інформаційно-комунікаційних систем.

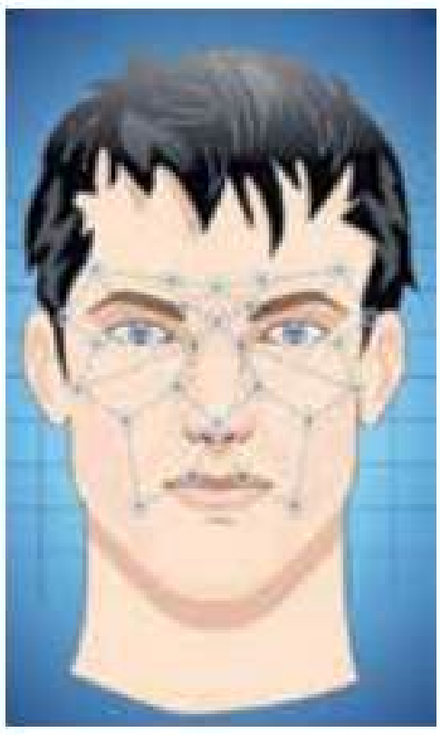
Штучний інтелект знаходить широке застосування в різних галузях – на виробництві, у побуті, науці, медицині, освіті, банківській сфері та торгівлі, управлінні та менеджменті, логістиці, на транспорті, у сільському господарстві тощо. Його використовують для автоматизації виконання рутинних завдань, оптимізації бізнес-процесів, аналізу великих обсягів даних, розпізнавання образів, управління системами та створення інтелектуальних помічників.

В Україні розробками в галузі штучного інтелекту активно займаються співробітники Інституту проблем штучного інтелекту, Інституту кібернетики імені В. М. Глушкова НАН України, інших наукових установ та ІТ-компаній.

Більш детально з прикладами використання ШІ в різних сферах діяльності людини можна ознайомитися, переглянувши відеоролик за посиланням *https://cutt.ly/Srq6YGQb*.

Узагальнюючи, можна виокремити, що сьогодні штучний інтелект у світі використовують у таких напрямах:

- **Розпізнавання об'єктів** (текстів, мови, графічних об'єктів на зображеннях, емоцій, запахів, шумів тощо): розпізнавання облич людей на смартфонах і з камер відеоспостережень (мал. 7.13), аналіз аерокосмічних фотографій і геодезичних карт, перетворення графічних зображень сканованих текстів у текстовий документ тощо. Наприклад, розпізнавання тексту успішно виконують програми *ABBYY FineReader*, *Office Lens для смартфонів*, *Online OCR*.
- **Машинний переклад** текстів різними мовами, наприклад з використанням комп'ютерних програм *Amazon Translate*, *DeepL Pro*, *SYSTRAN Translate Pro*, *Google Перекладач* (мал. 7.14) тощо.



Мал. 7.13. Розпізнавання обличчя

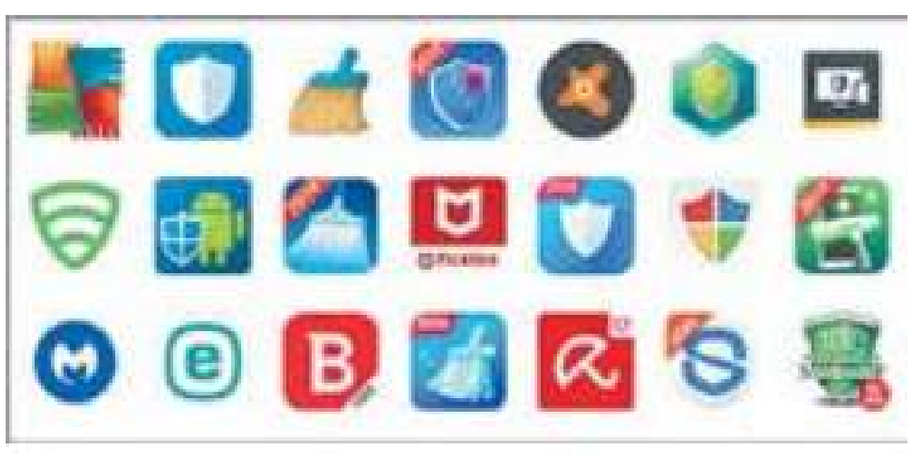


Мал. 7.14. Переклад тексту в **Google Перекладач**

- **Аналітична діяльність, експертні системи** – наприклад, підбір квитків на транспорт з пересадками, прокладання оптимального маршруту по карті (мал. 7.15), діагностика захворювань, створювання нових ліків, автопілот літака та автотранспорту, управління ядерним реактором тощо.
- **Інтелектуальні системи** інформаційної безпеки – наприклад, розпізнавання та захист від комп'ютерних вірусів (мал. 7.16), кібератак, програми інтелектуального захисту банківських систем тощо.
- **Робототехніка** – створення та застосування *роботів* (технічних пристроїв для автоматизації людської праці), наприклад, на конвеєрних лініях автомобілів, у важкодоступних місцях вугільних шахт, у небезпечних для лю-



Мал. 7.15. Маршрут, побудований у **Google Карти**



Мал. 7.16. Антивірусні засоби захисту інформаційних систем

дини місцях атомного виробництва, військовій справі (мал. 7.17) тощо.

- **Творчість та ігри** – наприклад, створення комп'ютерної музики та малювання картин (мал. 7.18), комп'ютерних програм гри у шахи, розробка інтелектуальних пристроїв-іграшок.



Мал. 7.17. Робот для дистанційного розмінування



Мал. 7.18. **Ukraine Heroica** – фільм-лауреат премії MEGOGO **AI Film Festival** 2024 року
http://surl.li/acexks

Цікаві факти з історії

Значний внесок у розвиток теорії штучного інтелекту зробив український учений Глушков Віктор Михайлович (1923–1982) – математик і кібернетик, засновник Інституту кібернетики АН України, академік і віцепрезидент АН України (мал. 7.19). Суть поглядів Глушкова полягала в тому, що він бачив у машині не замінник людського мозку, а спеціальний інструмент, який його посилює, як молоток підсилює руку, а мікроскоп – очі. Він уважав, що машина – це не конкурент людини, а його знаряддя, яке багаторазово збільшує можливості людини.



Мал. 7.19. Глушков В. М.

ПЕРЕВАГИ ТА НЕДОЛІКИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ

Поміркуйте

● Як ви розумієте такі поняття, як *авторське право*, *академічна доброчесність*, *плагіат*? ● З якого віку можна самостійно створювати власні акаунти в різних онлайн-сервісах? ● Які правила безпечного використання інтернету ви можете назвати?

Дедалі частіше в новинах з'являються повідомлення про те, як людина і робот змагаються в різних сферах діяльності. І не завжди результати виявляються на користь людини. Так, наприклад, у січні 2018 року компанії **Alibaba Group** і **Microsoft** провели порівняння рівня розуміння прочитаного тексту людьми та інтелектуальними системами цих компаній. Виявилося, що показник штучного інтелекту склав 82,4 бала, а найбільш високий показник серед людей – 82,3 бала. Людина, наприклад, програє системам штучного інтелекту в грі у шахи. У довготривалому керуванні літаком штучний інтелект «не втомлюється», на відміну від людини, і точно слідкує за показниками на автопілоті. А влітку 2024 року штучний інтелект отримав срібну медаль на міжнародній учнівській олімпіаді з математики.

У зв'язку із цим у багатьох виникає етичне питання – чи зможе штучний інтелект повністю замінити людину? Із цього приводу дискусії ведуться вже майже сто років. І однозначної відповіді на сьогодні немає.

Одним з головних викликів використання систем ШІ сьогодні є те, чи завжди ШІ ухвалює правильні рішення на основі отриманих даних. Наприклад, у таких критичних сферах, як охорона здоров'я або правосуддя, використання ШІ може призвести до непередбачуваних наслідків або упереджених рішень через недосконалість алгоритмів та обмеженість бази знань ШІ.

Також ШІ уже сьогодні автоматизує багато виробничих процесів, через що певні професії можуть стати непотрібними. Це викликає занепокоєння щодо майбутнього ринку праці та того, як забезпечити нові можливості для працівників, чиї професії зникають.

Оскільки ШІ вимагає великих обсягів даних для своєї ефективної роботи, виникає питання захисту конфіденційної інформації. Хакери можуть використовувати вразливості в алгоритмах опрацювання даних для зловживань, що загрожує як приватним особам, так і компаніям.

Крім того, системи ШІ чудово справляються з аналітичними задачами, але їм все ж бракує емоційної складової, яка є важливою в таких сферах, як обслуговування клієнтів або ухвалення рішень, що вимагають людської інтуїції.

Важливим і складним на сьогодні є і питання плагіату з використанням штучного інтелекту. Це відбувається через те, що генеративні моделі ШІ навчаються на великих наборах даних, які можуть бути взяті з інтернету, в тому числі включати захищений авторським правом матеріал. І спираючись на них, інтелектуальні системи можуть ненавмисно відтворювати великі уривки тексту із цих матеріалів без посилання на джерела, що призводить до порушення авторських прав і плагіату.

Але існує і зворотна ситуація – створено інструменти на основі ШІ, які допомагають виявляти плагіат у текстах. Вони порівнюють наданий текст з великою базою даних вебсторінок, наукових робіт, книжок та інших джерел, щоб знайти збіги або схожість. Це допомагає забезпечити академічну доброчесність і захистити авторські права.

21 травня 2024 року Рада ЄС схвалила закон про штучний інтелект, установивши обмеження у використанні цих сучасних технологій (*http://surl.li/porhhe*). Нові правила забороняють певні інструменти ШІ, які порушують права громадян: використання біометрії, збирання світлин людей з мережі або записів камер відеоспостереження для створення баз даних розпізнавання облич, створення фейків і пропагандистської інформації тощо.

Крім того, існують вікові обмеження для використання деяких систем штучного інтелекту. Наприклад, для використання ChatGPT користувачам повинно бути щонайменше 13 років, а для користувачів віком від 13 до 18 років потрібна згода батьків або законних представників для використання ШІ.

Проте у будь-якому разі використання штучного інтелекту має свої переваги, зокрема:

- **Ефективність та автоматизація.** Рутинні завдання, які раніше займали години, тепер можуть виконуватися автоматично. ШІ звільняє наш час для більш важливих і креативних речей.
- **Швидкість обробки даних.** Завдяки складним алгоритмам і моделям машинного навчання штучний інтелект може швидко аналізувати величезні масиви даних, виявляти певні закономірності та тенденції, отримувати оперативні висновки. Це особливо важливо у сферах, де від швидкості залежать критичні рішення.
- **Персоналізація інформації.** ШІ аналізує ваші останні запити та пропонує за результатами пошуку те, що вас найбільше цікавило. Усі чат-боти та голосові помічники, які завжди готові допомогти, також працюють на основі ШІ. Вони відповідають на наші запити швидко та точно.
- **Витривалість.** Штучному інтелекту не потрібен сон і перерва на обід, він не допускає помилок через перевтому.
- **Інновації та творчість.** Штучний інтелект є каталізатором інновацій і творчості в різних галузях. Автоматизуючи рутинні та повсякденні завдання, ШІ вивільняє людський талант, щоб зосередитися на більш творчих починаннях. Спільна взаємодія між ШІ і людською творчістю сприяє інноваціям, розширюючи межі можливого в різних сферах – від дизайну продуктів до розв'язування проблем і генерування ідей.
- **Скорочення витрат і оптимізація ресурсів.** Завдяки автоматизації процесів та оптимізації розподілу ресурсів, які визначають системи ШІ на виробництві, бізнес може значно скоротити витрати на робочу силу, пов'язані з виконанням рутинних завдань, мінімізувати втрати та досягти кращих фінансових результатів.
- **Посилення безпеки та виявлення шахрайства.** У сфері кібербезпеки штучний інтелект є потужним союзником у виявленні потенційних загроз і реагуванні на них як у сфері персональних даних, так і на рівні установ, держави та світу.
- **Мовний переклад і комунікація.** Інструменти машинного перекладу на основі штучного інтелекту сприяють безперешкодному спілкуванню людей у глобальному масштабі, це значно розширює культурні, освітні та бізнес-можливості.

Для тих, хто хоче знати більше

Уважається, що штучний інтелект здатний проявляти поведінку, яка не відрізняється від людської. Так, один з основоположників-розробників теорії штучного інтелекту Алан Тьюрінг (1912–1954) (мал. 7.20) у своїй книзі «Чи може машина думати?» у 1950 році висловив думку, що машина може вважатися розумною тоді, коли буде здатна підтримувати спілкування з людиною, і та не зможе зрозуміти, що спілкується з машиною. Такий своєрідний тест (так званий *тест Тьюрінга*) вважається пройденим штучним інтелектом, якщо 30 % експертів, які братимуть участь у розмові, не розпізнають штучний інтелект.

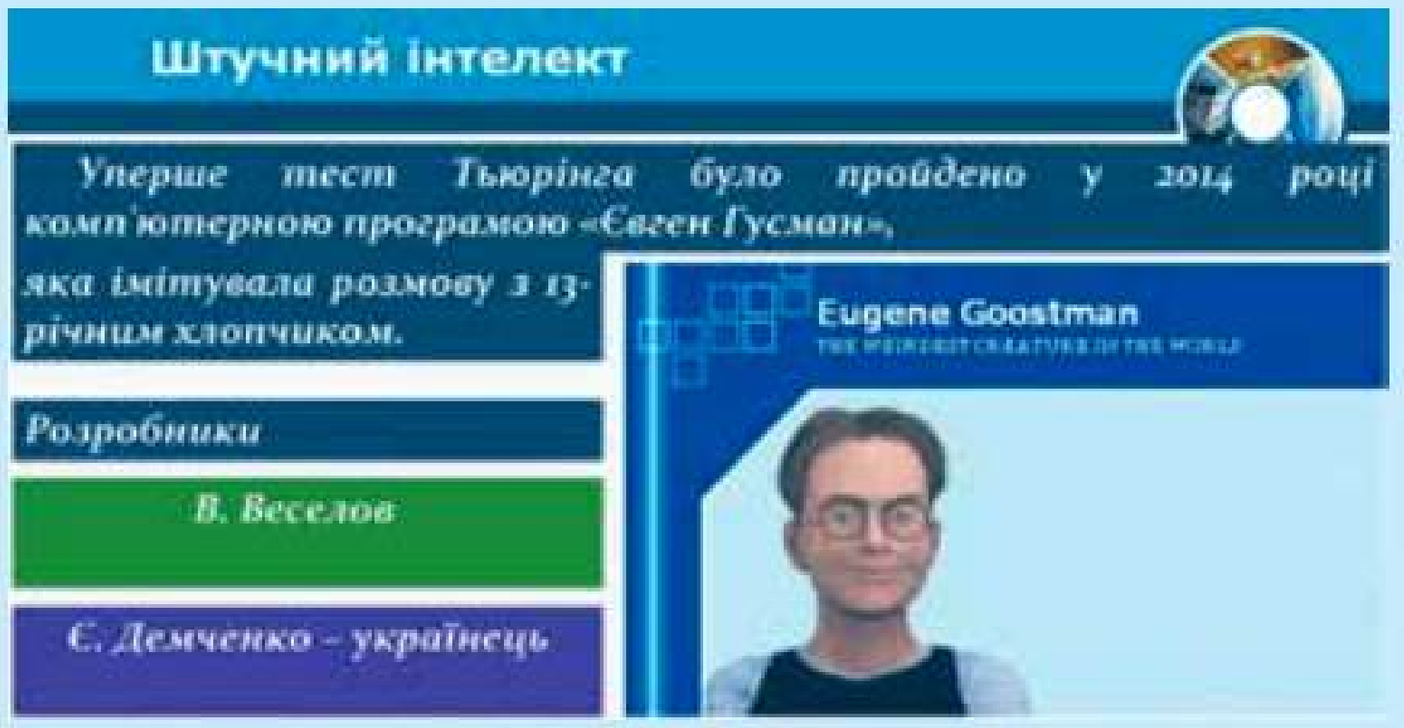


Мал. 7.20. Алан Тьюрінг

Однією з перших комп'ютерних програм (2014 рік), що пройшла тест Тьюрінга, була програма «Eugene Goostman»

(мал. 7.21). Вона вела розмову з людиною від імені 13-річного хлопчика з Одеси Євгена Гусмана (один з розробників програми Є. Демченко родом з України).

У 2024 році дослідники з Каліфорнійського університету провели експеримент, у якому 500 експертів по 5 хвилин спілкувалися в чаті з трьома моделями штучного інтелекту (ChatGPT-4, ChatGPT-3.5 і Eliza 1966 рік) та однією людиною. За результатами, у 54 % бесід з програмою ChatGPT-4 експерт уважав, що спілкується з людиною; з ChatGPT-3.5 – у 50 % випадків. Ці результати показують, що штучна система успішно пройшла тест Тьюрінга, але водночас дослідники наголошують, що тест може відображати здатність штучного інтелекту наслідувати або обманювати людей, а не демонструвати справжній інтелект.



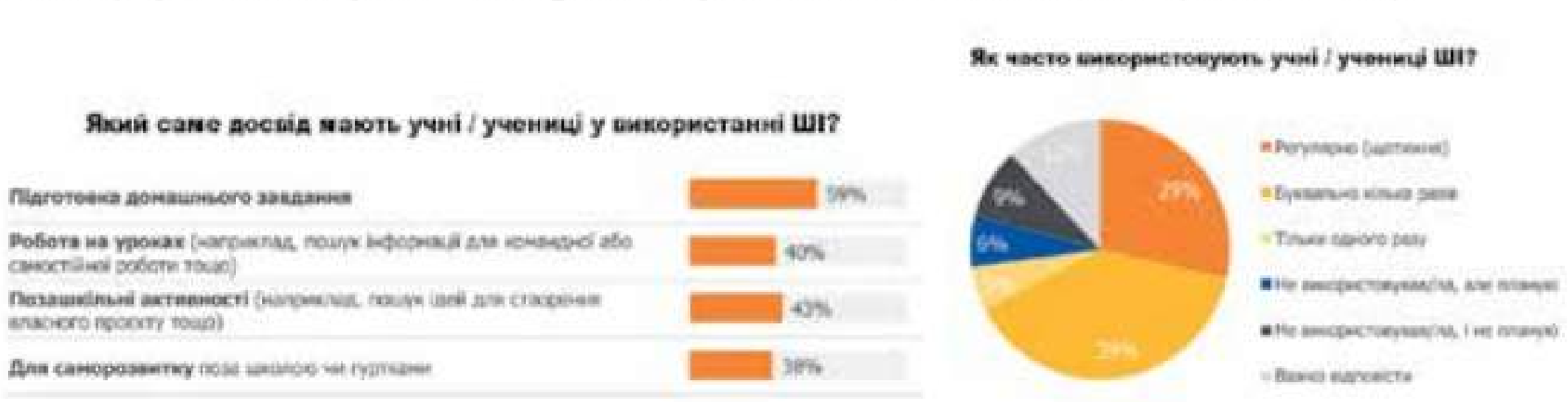
Мал. 7.21. Заставка програми **Eugene Goostman**

ШТУЧНИЙ ІНТЕЛЕКТ У НАВЧАННІ

Поміркуйте

• Як здійснюється пошук потрібної інформації в пошукових системах? • Чи використовували ви на уроках системи штучного інтелекту? Якщо так, то для вирішення яких навчальних завдань?

У вересні – жовтні 2024 року в закладах освіти України проводилося дослідження щодо використання систем штучного інтелекту в освіті, у якому взяли участь 3190 учителів / учительок та учні / учениці 8–11 класів. Виявилося, що 76 % опитаних учителів / учительок мають успішний досвід використання ШІ у своїй діяльності та 54 % уважають, що ШІ змінить шкільне навчання в найближчі роки. І 85 % учнів / учениць уже використовували ШІ в навчанні (мал. 7.22).



Мал. 7.22. Використання учнями / ученицями систем штучного інтелекту

Штучний інтелект відкриває перед учнівством і вчителями безліч нових можливостей і значно полегшує виконання рутинних завдань.

Наведемо кілька напрямів використання систем ШІ в навчанні:

- Швидко знаходити потрібну інформацію, відповіді на різноманітні запитання, вести розмови на цікаві для вас теми, уточнюючи та деталізуючи зміст відповідей.
- Отримувати від систем ШІ візуальне подання інформації для кращого розуміння складних навчальних тем (графіки, схеми, діаграми, інфографіка).
- Перевірити грамотність написання тексту будь-якою мовою, поліпшити стиль написання тексту за допомогою інструментів ШІ.
- Отримувати за запитом згенеровані ШІ ідеї для виконання власних проєктів, конкурсів, творів, сценаріїв тощо.
- Створювати за допомогою систем ШІ заготовки презентацій, текстів, зображень, резюме, аудіо- та відеоматеріалів до уроків, позакласних заходів, життєвих потреб.
- Досліджувати за допомогою різних віртуальних симуляторів складні питання шкільних предметів, проводити віртуальні дослідження з використанням ШІ.
- Генерувати з використанням ШІ різні однотипні навчальні завдання з різних предметів і тем, розв'язувати їх з перевіркою ШІ, використовуючи як навчальний тренажер, або тренуватися в розмові будь-якою іноземною мовою зі штучним інтелектом.
- Використовувати ШІ-інструменти не тільки для розв'язування завдань, а й для пояснення кроків для отримання результатів, інструкції для виконання такого завдання.
- Повноцінно навчатися на онлайн-курсах з різних предметів, де ШІ може спланувати персональну траєкторію навчання, залежно від рівня успішності, потреб і бажаного темпу навчання.

Сьогодні існує велика кількість різноманітних інструментів, які працюють на основі штучного інтелекту, вони постійно вдосконалюються, змінюються умови їх використання, деякі з часом зникають і з'являються нові. У зв'язку із цим якихось виключних списків таких систем не існує, їх приклади та способи використання можна знайти в інтернеті самостійно.

Деякі системи ШІ безкоштовні, але більшість платні або з безкоштовною пробною версією. Майже всі ці сервіси потребують попередньої реєстрації на цьому ресурсі, деякі можуть використовувати раніше створені акаунти в **Google** чи **Microsoft**. При цьому потрібно пам'ятати, що самостійне використання більшості ШІ-інструментів дозволено з 13 років, але за погодження батьків.

Вибір системи штучного інтелекту залежить від ваших потреб: тематики запиту, цільової аудиторії, фінансових і вікових обмежень, технічних можливостей тощо.

Наведемо кілька прикладів безкоштовних систем штучного інтелекту, які можна використовувати в навчальній діяльності (табл. 7.1). Більш розширений список ШІ-інструментів наведено в ***Додатку 3*** до підручника.

Таблиця 7.1

Назва ШІ-інструмента, тип моделі, посилання на ресурс	***Приклади використання***
Gemini Google (мовна модель) *https://gemini.google.com/app?hl=uk*	Пошук інформації за запитом, переклад текстів різними мовами, створення різних текстів за сюжетом, генерування ідей за тематикою, складання планів, розв'язування задач з математики, логіки, мов; аналіз текстів, надання рекомендацій, ведення розмов та інше
Copilot Дизайнер (генеративна модель) *https://www.bing.com/create*	Генератор графічних зображень, які створюються за текстовим описом, покращення готових зображень, створення макетів і шаблонів
Prezo (генеративна модель) *https://prezo.ai/*	Генератор і редактор презентацій за текстовим описом, скорочує довгі документи до стилю презентацій, додає до створених слайдів згенеровані графічні зображення, надає можливість поділитися презентацією

Ефективність використання систем ШІ залежить не тільки від розуміння їх можливостей і вміння користуватися їх інструментами, а й від того, як правильно сформульовано до них **запити (промпти,** англ. *prompt* – підказка). Точні та зрозумілі запити до систем забезпечують отримання корисних і релевантних відповідей.

Рекомендуємо дотримуватися таких правил:

- Сформулюйте чітко дію, яку повинен зробити ШІ (обговорити, сформулювати, запропонувати, класифікувати тощо).
- Чітко формулюйте тематику запиту, використовуйте ключові слова, уникайте двозначних слів, сленгу, загальних фраз.
- Укажіть результат, який бажаєте отримати, і його вигляд.
- Укажіть роль, від імені якої потрібно отримати результат, або цільову аудиторію, для якої буде використовуватися.
- Уникайте зайвих слів і складних речень, складне запитання краще розбити на кілька простих.
- Пишіть граматично правильно, дотримуйтеся правил граматики.
- Указуйте максимальні розміри потрібного результата в кількісних чи якісних показниках.
- Надайте за потреби ШІ приклад виконання, зразок, шаблон оформлення.
- Отримавши результат виконання завдання, визначте його відповідність вашим потребам.
- Якщо результат з першого разу вас не задовольнив, змініть запит, надавши додаткову інформацію, використавши інші ключові слова.
- Системи ШІ постійно вчаться на ваших запитах, тому тему одного завдання бажано продовжувати в тому самому чаті, щоб ШІ зміг отримати з вашого діалогу більш системні знання.

Але ж звертаємо увагу, що не слід на 100 % довіряти результатам роботи ШІ. Можливість технічних збоїв, нерозуміння контексту,

відсутність емоційного сприйняття, обмеженість даних – усе це може призвести до неправильних висновків і некоректних результатів.

Тому, користуючись ШІ, слід завжди критично оцінювати отриману інформацію, перевіряти її з інших джерел. І пам'ятати про обмеженість можливостей ШІ, що він усього лише потужний інструмент, який може допомогти вам у навчанні. Використовуйте його, щоб генерувати ідеї, створювати чернетки контенту або долати творчий ступор. Але пам'ятайте, що останній штрих завжди має залишатися за людиною – додавайте ваше особисте бачення, творчі ідеї, власні висновки.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/Se2cKNSC* або QR-кодом.



Найважливіше в цьому пункті

Штучний інтелект – це область інформатики, яка здійснює розробку інтелектуальних комп'ютерних систем і програм, що імітують роботу людського мозку: розуміти мову, навчатися, міркувати, робити висновки та передбачати, розв'язувати проблеми, ухвалювати рішення тощо.

Сьогодні створення систем штучного інтелекту ведеться за напрямами створення комп'ютерних систем, що імітують діяльність людини; імітують логічне мислення людини; імітують функціонування людини; будуть інтелектуальними агентами.

Штучний інтелект набуває широкого застосування в різних галузях діяльності. Його використовують для розпізнавання різних об'єктів, машинного перекладу текстів, аналізу даних, створення експертних систем та інтелектуальних систем інформаційної безпеки, у робототехніці, для творчості та ігор тощо.

Основними недоліками використання результатів роботи систем ШІ сьогодні є: можливість отримання недостовірних даних і неправильних рішень, слабкий захист персональних даних і збереження конфіденційності використовуваної інформації, відсутність емоційної складової у спілкуванні та інтуїції, недотримання академічної доброчесності та допущення плагіату користувачами ШІ. Існують і певні обмеження його використання, зокрема заборонено використовувати інструменти ШІ, які порушують права громадян, або вікові обмеження.

У використанні штучного інтелекту є й свої переваги: ефективність та автоматизація деяких рутинних видів діяльності, швидкість обробки даних, персоналізація отримуваної інформації, витривалість і стійкість системи протягом тривалого часу, скорочення витрат і оптимізація ресурсів, посилення безпеки та виявлення шахрайства, скасування мовних бар'єрів тощо.

Штучний інтелект відкриває нові можливості й перед учнями / ученицями та вчителями / вчительками, значно полегшує виконання рутинних завдань, таких як пошук потрібної інформації, візуалізація текстового матеріалу, перевірка правопису, генерування ідей, створення інформаційних продуктів різного типу та інше.

Існує велика кількість різноманітних інструментів, які працюють на основі штучного інтелекту, вони постійно вдосконалюються, змінюються умови їх використання, деякі з часом зникають і з'являються нові. Вибір системи штучного інтелекту залежить від ваших потреб: тематики запиту, цільової аудиторії, фінансових і вікових обмежень, технічних можливостей тощо.

Значною мірою ефективність використання систем ШІ залежить від точності та зрозумілості сформульованих **запитів (промптів)**.

Користуючись ШІ, слід завжди критично оцінювати отриману інформацію, перевіряти її з інших джерел. Усвідомлювати, що це лише потужний інструмент, який може допомогти вам у навчанні, на роботі, у житті.

Дайте відповіді на запитання

1. Що таке *штучний інтелект*?

2. У яких напрямах ведеться створення систем штучного інтелекту? Наведіть приклади.

3. Які напрями використання систем штучного інтелекту в різних галузях діяльності людини? Наведіть приклади.

4. Що таке *промпт*? Які правила формулювання промптів?

5. Які існують етичні проблеми створення систем штучного інтелекту?

Обговоріть і зробіть висновки

1. Що спільного між машинним і людським інтелектом?

2. Як ви розумієте слова Норберта Вінера (засновника кібернетики): «Обчислювальна машина цінна рівно настільки, наскільки цінна людина, що її використовує»?

3. Які переваги та недоліки використання систем штучного інтелекту ви можете назвати в таких прикладах:

а) штучний інтелект визначає діагноз хворій людині;

б) штучний інтелект створює казку для маленьких дітей;

в) штучний інтелект веде урок у школі;

г) штучний інтелект проєктує ро́бота.

4. Які ризики несуть системи штучного інтелекту під час їх використання? Поспілкуйтеся про це із самим штучним інтелектом. Чи розділяєте ви висловлені зауваження? Проранжуйте висловлені ризики за ступенем важливості для вас.

5. Як ви вважаєте, чого не може зробити штучний інтелект? Які професії він не зможе замінити?

6. Чи може штучний інтелект отримати авторське право на згенеровані ним матеріали? Як потрібно діяти авторам, щоб уникнути плагіату під час використання систем ШІ? Проаналізуйте та обговоріть із цього приводу статті Закону України «Про авторське право та суміжні права» та проєкту Закону України «Про академічну доброчесність».

Виконайте завдання

Увага! *Виконання завдань з використанням ШІ дозволяється за згодою батьків / рідних. За відсутності дозволу батьків / рідних виконайте завдання без використання ШІ.*

1. Відкрийте інтернет-майданчик для інтелектуального малювання **AutoDraw** *https://www.autodraw.com*. Ви можете намалювати будь-яку криву лінію, і програма домалює за вас малюнок, який можна обрати з певного переліку зображень. Ознайомтеся з роботою програми. Намалюйте кілька зображень з використанням можливостей цього ресурсу. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 7.2.1.** Чи можна вважати цю програму прикладом системи штучного інтелекту?

2. Відкрийте сайт **Акінатор** *https://www.silvergames.com/uk/akinator*. Налаштуйте переклад інтерфейсу україн-

ською і пограйте у віртуальну гру «Вгадай слово». Чи вдалося програмі у вас виграти? Чи можна вважати цю програму прикладом системи штучного інтелекту?

3. Відкрийте онлайн-додаток для озвучування тексту *https://qudata.com/uk/text-to-speech/* та ознайомтеся з його роботою. Уведіть кілька текстових фраз і прослухайте їх озвучування. Налаштуйте різні режими роботи програми, збережіть отримані результати у вашій папці. Поясніть, чи можна вважати цю програму прикладом штучного інтелекту. Запишіть свої аргументи в зошит.

4. Відкрийте сайт **Quick,Draw!** *https://quickdraw.withgoogle.com/?locale=ua#*. Ознайомтеся з роботою нейронної мережі **Google**, для якої потрібно намалювати зображення за вказаною темою. Опишіть у зошиті, як ця система штучного інтелекту вчиться.

5. Ознайомтеся зі статтею «Забута «Машина логічного мислення» професора Щукарьова» за посиланням *https://uacomputing.com/stories/professor-shchukarevs-forgotten-logical-thinking-machine/*. Визначте та запишіть у зошит кілька ідей, які були сформульовані вченим та які знайшли відображення в сучасних розробках штучного інтелекту.

6. Перегляньте освітній телесеріал для учнів / учениць *Штучний інтелект для школярів https://osvita.diia.gov.ua/courses/artificial-intelligence-for-schoolchildren*. Запишіть у зошит по три приклади: які відомості ви вже знали; що нового ви дізналися; про що хотіли б дізнатися більше?

Готуємось до вивчення нового матеріалу

1. Що таке *авторське право*? Які вимоги до його дотримання?
2. Як цифрові технології використовуються в освіті?
3. Які ресурси зі штучним інтелектом ви вже використовували?

Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Smart-технології. Штучний інтелект»** *(початковий, середній, достатній, високий рівень).*

- Я можу пояснити, що таке Smart-пристрій і чим він відрізняється від традиційних.
- Я розумію, які переваги та недоліки пов'язані з використанням Smart-пристроїв та інтернету речей.
- Я можу навести приклади пристроїв, які належать до інтернету речей.
- Я розумію, як пристрої інтернету речей під'єднуються до мережі.
- Я можу описати, як Smart-технології та інтернет речей змінюють наше життя.
- Я можу пояснити, що таке штучний інтелект і які його види існують.
- Я знаю основні напрями розробок систем ШІ та їх використання в різних сферах діяльності людини.
- Я вмію створювати запити до систем ШІ та передбачати результат їх виконання.
- Я можу оцінити переваги та недоліки використання штучного інтелекту.
- Я розумію етичні аспекти застосування штучного інтелекту.

Повторіть той матеріал, який ви знаєте недостатньо.

РОЗДІЛ 8

ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

У цьому розділі ви отримаєте нові, а також поглибите та розширите свої знання й удосконалите навички з таких тем:

- навчальний проєкт та етапи його виконання;
- використання систем штучного інтелекту для виконання проєкту.

8.1. НАВЧАЛЬНИЙ ПРОЄКТ. ВИКОРИСТАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИКОНАННЯ ПРОЄКТУ

НАВЧАЛЬНИЙ ПРОЄКТ ТА ЕТАПИ ЙОГО ВИКОНАННЯ



Пригадайте

● Що таке *навчальний проєкт*? Наведіть приклади проєктів, учасником виконання яких ви були. ● Що таке п'ять «П» проєкту? У чому їх сутність? ● Які методи збирання відомостей для проєкту ви використовували?

Навчальні проєкти ви почали виконувати ще в початковій школі та продовжували брати в них участь і в 5–7 класах.

Під час роботи над проєктом учасники / учасниці досліджують цікаві теми, шукають відповіді на складні запитання, вчаться працювати в команді та презентувати результати своєї роботи.

Нагадаємо етапи реалізації навчального проєкту (мал. 8.1):



Мал. 8.1. Етапи виконання проєкту

1. ***Постановка завдання.*** Спочатку потрібно чітко сформулювати, чого ми хочемо досягти. Яка проблема нас цікавить? Які цілі та завдання ми ставимо перед собою?
2. ***Планування проєкту.*** Далі створюємо детальний план реалізації проєкту та встановлюємо терміни виконання. Розподіляємо завдання між учасниками проєкту, визначаємо ролі та завдання кожного.

3. ***Збирання відомостей.*** Шукаємо потрібні матеріали в різних джерелах: книжках, журналах, інтернеті, проводимо інтерв'ю та опитування.
4. ***Опрацювання даних.*** Аналізуємо та опрацьовуємо зібрані відомості, систематизуємо їх та робимо висновки.
5. ***Оформлення результатів.*** Оформляємо результати проєкту у вигляді презентації, відеоролика, вебсторінки або іншого інформаційного продукту.
6. ***Презентація результатів проєкту.*** Представляємо свою роботу однокласникам / однокласницям, учителю / учительці або батькам / рідним.

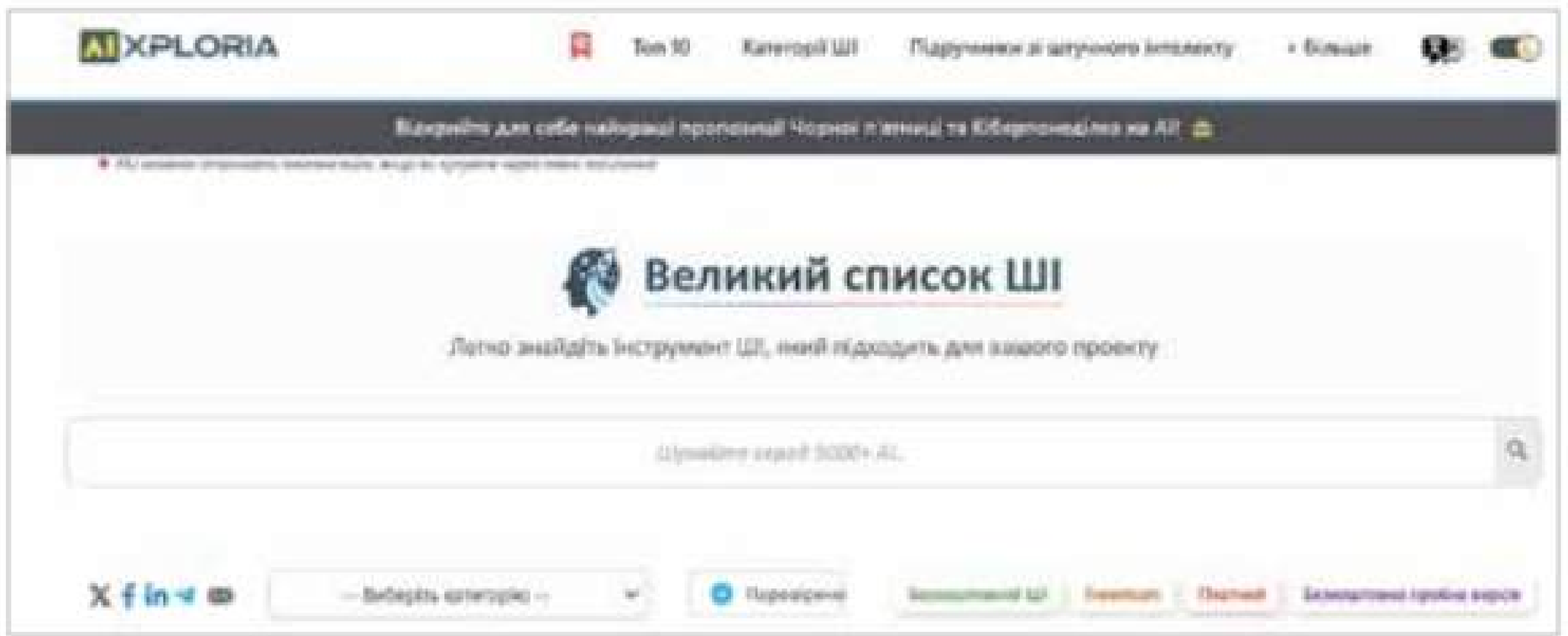


Мал. 8.2. Учні виконують груповий проєкт (згенеровано ШІ)

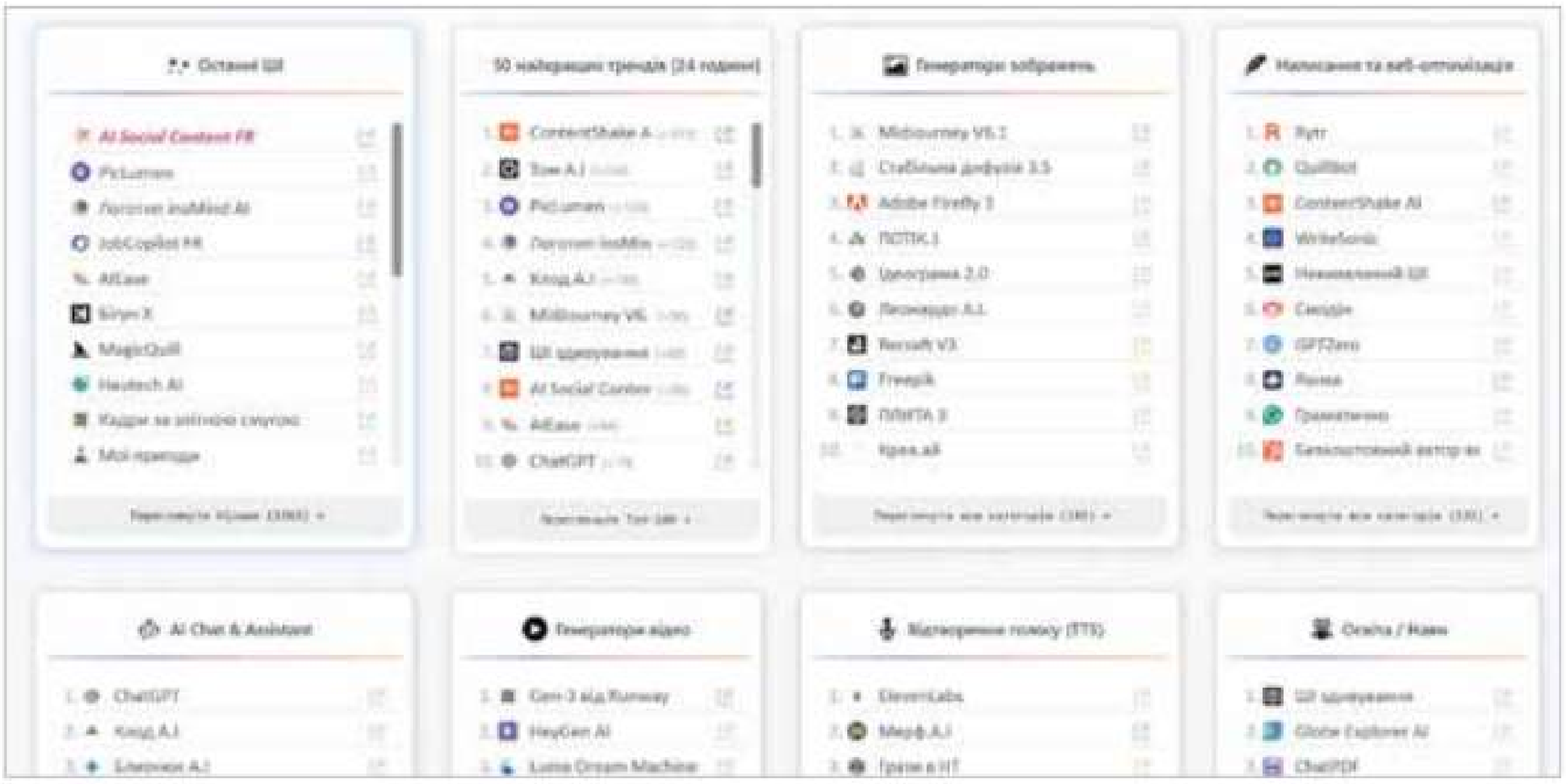
Участь у виконанні проєктів допомагає розвинути творчі, дослідницькі та комунікативні навички. Завдяки проєктам можна детально вивчити навчальну тему, над якою працюєте. Робота в команді вчить взаємодіяти з іншими людьми, ділитися ідеями та досягати спільних цілей. Проєкти дають змогу використовувати різноманітні інструменти та сервіси, які роблять роботу цікавішою та ефективнішою.

Цього року пропонуємо вам виконати проєкти з використанням систем штучного інтелекту. Ви зможете обрати тему, яка вас найбільше цікавить, і створити проєкт разом з вашими однокласниками / однокласницями. Результатом дослідження може бути вебсайт, збірка фотоальбомів, колаж малюнків, презентація, відеоролик або щось інше, за вашим бажанням. А для більш ефективної роботи скористайтеся ШІ-інструментами, які допоможуть вам на всіх етапах реалізації проєкту.

Для вибору інструментів ШІ ви можете скористатися ***Додатком 3*** підручника. Або переглянути майже повний перелік актуальних на сьогодні систем ШІ на сайті **Aixploria** *https://www.aixploria.com/* (за потреби виберіть переклад сторінки українською мовою) (мал. 8.3). На ньому представлено добірку ШІ-інструментів, згрупованих за категоріями, щоб легко знайти ресурс, який відповідає вашим потребам. До речі, сам перелік створено також штучним інтелектом.



Мал. 8.3 (а). Сайт **Aixploria**



Мал. 8.3 (б) . Перелік ШІ-інструментів на сайті **Aixploria**

За потреби під час реалізації проєкту рекомендуємо скористатися текстовими та відеоінструкціями для роботи з різними ШІ-інструментами *http://surl.li/qdjwwp*.

ВИКОРИСТАННЯ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ВИКОНАННЯ ПРОЄКТУ

Пригадайте

• Які комп'ютерні програми можуть бути використані для оформлення результатів проєкту? • Які можливості використання систем штучного інтелекту в навчанні?

Розглянемо виконання проєкту з використанням систем штучного інтелекту на прикладі проєкту з інформатики *Штучний інтелект у нашому житті*.

На кожному етапі реалізації проєкту використовуватимемо відповідні ШІ-інструменти для прискорення нашої роботи. Їх перелік досить великий, і ви можете добирати для реалізації своїх задумів різні ресурси.

1. Постановка завдання.

Виконання цього етапу рекомендуємо провести вашою групою спільно, для того щоб визначити ідею проєкту та обрати тему, цікаву для всіх. Для формулювання мети та завдань проєкту скористайтеся системою штучного інтелекту **Gemini** (*https://gemini.google.com/app*).

Надамо системі такий запит: *«Ми, учні 8 класу, плануємо брати участь у навчальному проєкті з інформатики за темою "Штучний інтелект у нашому житті"». Тривалість проєкту 2–3 уроки. Сформулюй 10 ідей для проєктів за цією темою та їх мету».*

У результаті ШІ запропонував 10 ідей, наприклад «Штучний інтелект у кіно», «Штучний інтелект і здоров'я», «Штучний інтелект у смартфоні», «"Розумний" дім: зручно чи небезпечно?» тощо.

Далі спільним обговоренням у групі потрібно проаналізувати запропоновані варіанти, обрати цікаву для всіх ідею проєкту, спільно уточнити назву. Для визначення інших складових проєкту можна сформулювати додаткові запити ШІ, орієнтовно такі: *«Наша команда обрала назву проєкту "Штучний інтелект у смартфоні". Сформулюй завдання такого проєкту. Які ресурси для цього потрібні? Що може бути результатом такої роботи?»*. Результати роботи ШІ потрібно скопіювати в текстовий документ, обговорити та уточнити у групі, зберегти у спільній папці групи у файлі з іменем *Завдання та ресурси проєкту*.

Таким чином маємо орієнтовно такий опис проєкту:

Назва проєкту: «Штучний інтелект у моєму смартфоні».

Мета проєкту: розглянути, які додатки смартфона використовують штучний інтелект і для чого ми їх застосовуємо.

Завдання проєкту:

- Створити список додатків на своїх смартфонах, які, на вашу думку, використовують штучний інтелект (наприклад, камера, голосовий помічник, GPS-навігатор, перекладач). *Результат* – текстовий документ.
- Знайти відомості про те, як працюють голосові помічники (Siri, Google Assistant, Alexa тощо). *Результат* – збірка текстових і графічних матеріалів у спільній папці на **Google Диску**.
- Дослідити, як штучний інтелект використовують у додатках камер смартфонів (розпізнавання облич, поліпшення якості зображень, переклад текстів на зображеннях). *Результат* – відеоролик проведення дослідів.
- Створити опитувальник для однолітків, щоб з'ясувати, які додатки смартфона, що використовують штучний інтелект, вони застосовують найчастіше і чому. *Результат* – **Google Форма** з п'яти питаннями.
- Створити презентацію, яка демонструє результати роботи групи у проєкті. *Результат* – презентація на **10** слайдів.

Час виконання проєкту – **10** днів.

Для подальшої роботи придумайте назву нашої групи, наприклад «Мобільні дослідники». Запросіть ідеї логотипу групи в системи штучного інтелекту **Copilot** (*https://copilot.microsoft.com/*), надавши запит *«Намалюй логотип для нашої групи "Мобільні дослідники", які виконують навчальний проєкт "Штучний інтелект у смартфоні". Це повинен бути схематичний кольоровий малюнок невеликого розміру»*. Отриманий логотип (мал. 8.4) збережіть у спільній папці проєкту у файлі з іменем *Логотип*.



Мал. 8.4. Логотип групи

2. Планування реалізації проєкту.

Складіть детальний план реалізації проєкту, указавши послідовність виконання завдань, виконавців, терміни роботи, використавши систему штучного інтелекту **Gemini**. Укажіть у запиті перелік ваших завдань, терміни роботи та подання плану у вигляді таблиці. Отриманий текст

у додатку **Gemini** потрібно **Експортувати в Таблиці**, проаналізувати у групі та відредагувати за потреби, зазначити прізвища виконавців і конкретні дати виконання, зберегти у спільній папці групи (мал. 8.5).

План реалізації проєкту. План роботи групи «Мобільні дослідники»

Дні	*Дати*	*Завдання*	*Деталі*	*Відповідальний*
1–2	05–06.05.2025	Постановка завдання та планування проєкту	Розподіл завдань, створення спільної папки на **Google Диску**, розробка ппану досліджень	Усі члени групи
3–4	07–08.05.2025	Збирання відомостей	Пошук в інтернеті, створення списків додатків у мобільних телефонах учасників груп та їх батьків	Іван
			Дослідження голосових помічників, пошук в інтернеті принципів їх роботи	Катя
			Дослідження додатків камер, запис відео проведення досліджень	Петро
			Складання опитувальника, створення **Google Форми**	Олена
			Керівництво проєктом, допомога учасникам	Тарас
5–7	09–11.05.2025	Опрацювання даних	Упорядкування списків додатків, їх узагальнення, створення текстового документа з описом додатків	Іван
			Створення порівняльного опису роботи голосових помічників	Катя
			Обробка відео, монтаж	Петро
			Проведення опитування, створення діаграм та їх опису	Олена
			Керівництво проєктом, допомога учасникам	Тарас
8–9	12–13.05.2024	Оформлення результатів	Обговорення результатів, створення презентації (кожен по 2 слайди за своєю роботою)	Усі члени групи
10	14.05.2025	Презентація проєкту	Презентація результатів перед класом	Тарас

Мал. 8.5. План реалізації проєкту

3. Збирання відомостей.

Далі учасники / учасниці групи виконують частину спільної роботи за своїм завданням. Щоб визначити ресурси для пошуку відомостей, а також поставити змістові питання щодо виконання окремих дій, можна знову ж таки використати системи штучного інтелекту, наприклад **Gemini**, **Copilot**, **Perplexity AI** (*https://www.perplexity.ai/*). Приклади запитів до сервісів ШІ:

- Склади список десяти додатків у мобільних телефонах, які використовують штучний інтелект.
- Склади список десяти пристроїв у мобільних телефонах, додатки яких використовують штучний інтелект.
- Які операції може виконувати камера мобільного телефона? Які з них застосовують ШІ?
- Склади список з десяти запитань для опитування однокласників / однокласниць щодо користування ними додатками смартфона, які використовують ШІ.

Далі отримані відповіді аналізуються й обов'язково перевіряються в інших джерелах. Після чого учасники / учасниці групи проводять практичні дослідження додатків зі ШІ (аудіозаписи спілкування з голосовими помічниками, відео щодо роботі з камерами тощо,

скріншоти додатків) і створюють опитувальник у **Google Формах**. Усі матеріали зберігаються у спільній папці групи.

4. Опрацювання даних.

На цьому етапі виконання проєкту кожен учасник / учасниця групи аналізує отримані результати своїх досліджень додатків смартфона, створює узагальнений документ за підсумками роботи над своїм завданням і візуалізує результати.

Усе це можна зробити в текстовому документі, електронній таблиці, презентації. А можна доручити узагальнення цих даних системам ШІ, наприклад **Gemini**, **Copilot**, **Perplexity AI** або іншим. Для цього потрібно ввести в поле для запитів дані, які потрібно узагальнити, та сформулювати завдання – *Створити порівняльну таблицю, узагальнений список* тощо.

5. Оформлення результатів.

На наступному етапі за результатами власного дослідження учасник / учасниця створює по 2 слайди у спільній презентації з використанням хмарного сервісу **Google Презентації**, а лідер / лідерка проєкту – загальні слайди: титульний слайд, список учасників, мета та завдання проєкту тощо. Для підготовки окремих об'єктів презентації можна використати системи штучного інтелекту. Наприклад, створити аватарки учасників / учасниць групи та інфографіку за результатами з використанням сервісів **Copilot**, **DALLE-3** (*https://www.bing.com/images/create*), ***DreamStudio*** (*https://dreamstudio.com/start/*). Або створити всю підсумкову презентацію проєкту з використанням сервісу для створення електронних презентацій **Gamma** (*https://gamma.app/*).

6. Захист проєкту.

На заключному етапі проєкту лідер / лідерка групи представляє на спільній конференції вашого класу мету та завдання проєкту, хід його реалізації, вклад кожного члена групи, отримані результати, створені інформаційні продукти, основні висновки.



Працюємо з комп'ютером

Завдання та алгоритми їх виконання ви зможете знайти за адресою *https://cutt.ly/5e2cLqze* або QR-кодом.



Найважливіше в цьому пункті

Реалізація навчальних проєктів передбачає виконання певних етапів, а саме: постановка завдання, планування роботи, збирання відомостей, опрацювання даних, оформлення та презентація результатів проєкту. На кожному етапі виконання проєкту для більш ефективної та творчої його реалізації можна використовувати системи штучного інтелекту: для формулювання мети та завдань проєкту, планування проєкту та його візуалізації, генерування зображень, діаграм, інфографіки, відеороликів та інших інформаційних продуктів. Для кожної з таких задач сьогодні існує значна кількість ШІ-інструментів.

Виконання проєктів з використанням ШІ сприяє ознайомленню із сучасними ІТ, освоєнню нових інструментів для опрацювання даних різних типів і створенню за їх допомогою різноманітних інформаційних продуктів.

Важливим моментом використання таких систем під час виконання проєкту є питання доброчесності та творчої адаптації ідей, запропонованих ШІ. У всіх випадках штучний інтелект використовується для полегшення роботи у проєкті, виконання рутинних завдань.

Дайте відповіді на запитання

1. Які ресурси штучного інтелекту ви використовували для збирання відомостей у навчальних проєктах? Наведіть приклади.

2. Як технології штучного інтелекту можуть допомогти в пошуку джерел інформації? Наведіть приклади.

3. Як штучний інтелект можна використати для аналізу даних, зібраних під час дослідження? Наведіть приклади.

4. Які програми штучного інтелекту можна використовувати для створення інформаційних продуктів під час реалізації навчальних проєктів? Наведіть приклади.

Обговоріть і зробіть висновки

1. Які основні переваги та ризики використання штучного інтелекту в навчальних проєктах? Наведіть приклади.

2. Які етичні питання виникають під час використання штучного інтелекту в проєктах? Наведіть приклади.

3. Які ШІ-інструменти використовувалися різними групами під час виконання проєктів? Які інструменти вам стали до вподоби? Поясняти чому.

4. Обговоріть найбільш вдалі моменти роботи в навчальному проєкті та складності, з якими зіштовхнулися в ході роботи. Чи сприяло цьому використання систем ШІ? Яким чином?

5. Поясніть, які нові знання та навички ви отримали в ході виконання проєкту з використанням ШІ.

6. Чи може штучний інтелект замінити вчителя / вчительку? Поясніть вашу думку.

Виконайте завдання

Увага! *Виконання завдань з використанням ШІ дозволяється за згодою батьків / рідних. За відсутності дозволу батьків / рідних виконайте завдання без використання ШІ.*

1. Виконайте мініпроєкт, створивши інформаційний продукт з використанням системи штучного інтелекту, за одним з наведених завдань. Збережіть результат роботи у вашій папці у файлі з іменем **завдання 8.1**.

Орієнтовна тематика завдань:

1) Створити інфографіку або анімоване пояснення складного фізичного явища (наприклад, рух планет, будова атома) за допомогою інструмента для генерації зображень.
2) Створити навчальне відео, де штучний інтелект озвучує розв'язування математичної задачі або доводить теорему.
3) Створити короткий історичний огляд певної події чи постаті, використовуючи чат зі штучним інтелектом як помічника в дослідженні.
4) Створити алгоритм для розв'язування певної логічної задачі (наприклад, про козу, вовка та капусту) і візуалізувати його роботу за допомогою інструмента для створення блок-схем.
5) Створити презентацію про вплив технологій на суспільство, використовуючи генератор зображень для створення візуальних елементів.

6) Написати короткий науково-фантастичний твір, де штучний інтелект відіграє ключову роль. Використати ШІ для отримання ідей.
7) Проаналізувати достовірність інформації, згенерованої штучним інтелектом про деяких героїв літературних творів, порівнявши її з даними з інших джерел.
8) Створити казку для молодших учнів про «розумні» пристрої в нашому житті з використанням ШІ-інструментів.

2. Реалізуйте навчальний проєкт у групі однокласників / однокласниць за однією з наведених тем. Виберіть об'єкти дослідження відповідно до мети, засоби опрацювання даних і форму подання результатів. Використайте в реалізації проєкту системи штучного інтелекту на різних етапах його виконання. Проаналізуйте результати роботи ШІ та відредагуйте відповідно до мети завдання. Збережіть підсумковий результат роботи вашої групи у вашій папці у файлі з іменем **завдання 8.2**.

Орієнтовна тематика проєктів:

1) *Професія в ІТ. Мета*: скласти довідник професій, які можна набути в галузі інформаційних технологій; з'ясувати, навички роботи з якими програмними засобами повинні опанувати представники цих професій.
2) *Історія штучного інтелекту. Мета*: дослідити етапи створення штучного інтелекту у світі та в Україні в тому числі.
3) *Людина чи машина? Мета*: дослідити ставлення оточення до етичних аспектів використання штучного інтелекту.
4) *Штучний інтелект на службі людині. Мета*: дослідити використання ШІ в різних галузях діяльності людини.
5) *Школа майбутнього. Мета*: дослідити думку учнів / учениць, учителів / учительок, батьків / рідних щодо використання штучного інтелекту в навчанні.



Оцініть свої знання та вміння

Оцініть свої навчальні досягнення з розділу **«Практикум з використання інформаційних технологій. Навчальні проєкти»** *(початковий, середній, достатній, високий рівень).*

- Я вмію формулювати мету та завдання виконання проєкту з використанням ІТ для вирішення навчальної чи життєвої ситуації.
- Я вмію спланувати свої дії під час виконання групового проєкту, під час взаємодії з іншими, відповідально ставлюсь до виконання своїх завдань.
- Я знаю та застосовую різні методи збирання відомостей для проєкту, можу оцінити надійність і доцільність джерел.
- Я вмію добирати різні типи ресурсів до виконання проєкту, у тому числі з використанням систем ШІ.
- Я вмію обирати оптимальний спосіб подання результатів проєкту та обґрунтовувати свій вибір.
- Я вмію створювати інформаційні продукти різних видів як результати виконання проєктів, у тому числі з використанням ШІ-інструментів.
- Я вмію зберігати результати виконання проєкту у файлах різних типів на різних носіях даних, у тому числі в мережі та хмарних середовищах.
- Я вмію працювати у групі з іншими учасниками в ході реалізації спільного проєкту, враховувати погляди й емоційний стан членів групи.

Повторіть той матеріал, який ви знаєте недостатньо.

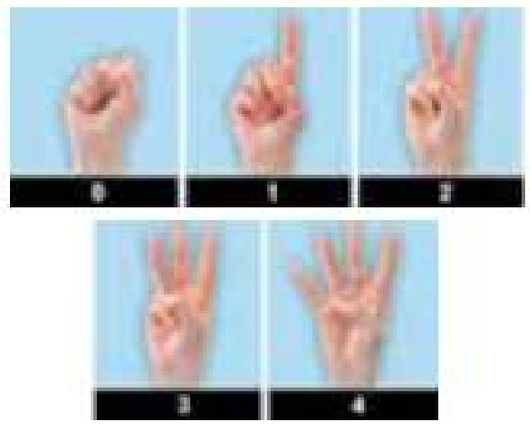
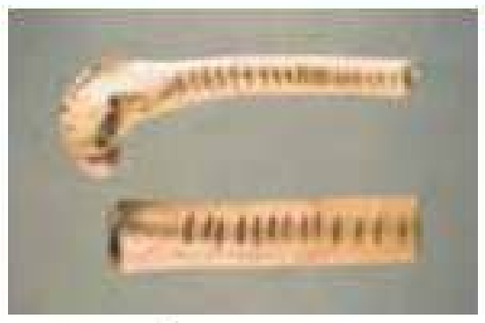
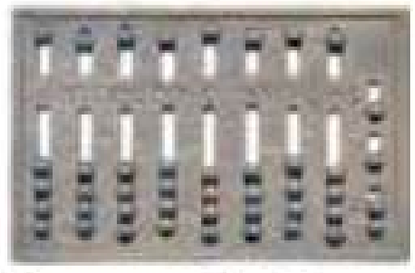
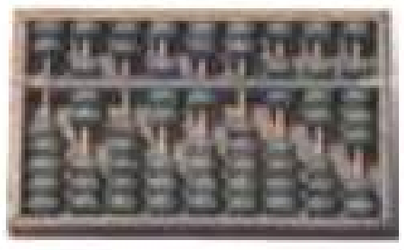
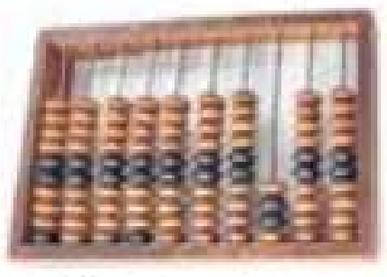
Додаток 1

Таблиця кодів символів ASCII

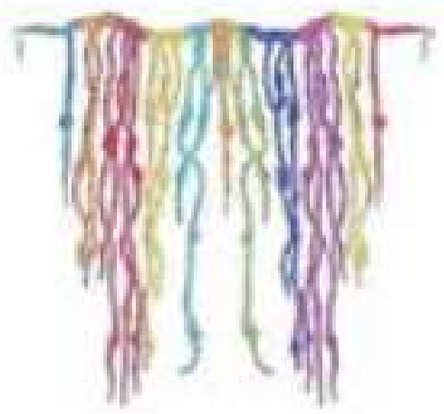
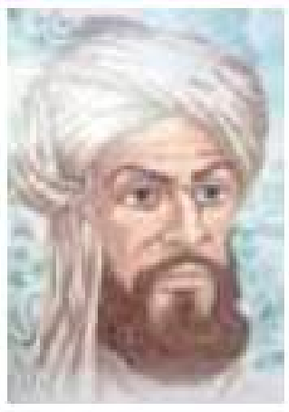
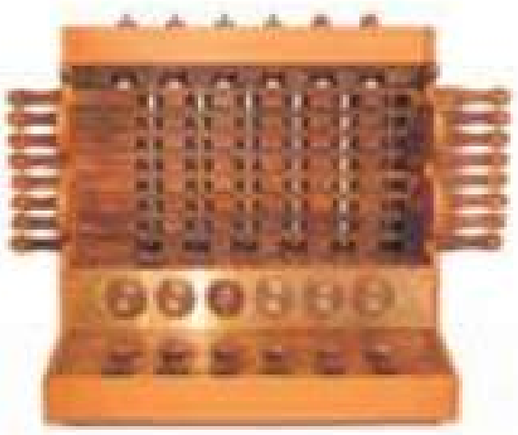
Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ	Код	Символ
32	пропуск	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	«	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	‘	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	i	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	□

Додаток 2

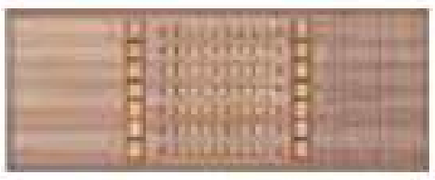
Хронологічна таблиця розвитку обчислювальної техніки

Роки	***Опис обчислювального пристрою або події***	***Ілюстрації***
40–30 тис. років до н. е.	Одними з перших засобів опрацювання числових повідомлень були пальці. Про це свідчать, наприклад, римські цифри (I, V, X). У стародавній слов'янській нумерації одиниці називалися «перстами», тобто пальцями, а назва цифри «5» походить від слова *п'ясть* (кисть руки)	
30 тис. років до н. е.	Для підрахунку людина почала використовувати зарубки на різних предметах – паличках, кістках тощо. Одна з найстаріших кісток із зарубками (датується 30 тис. років до н. е.) була знайдена біля чеського села Дольні Вестоніце. Це кістка вовка з 55 зарубками для рахунку. Подібні кістки були знайдені й у багатьох інших місцях	 Рахункові палички, знайдені в Abri Cellier (Франція, 28 тис. років до н. е.)
3000 років до н. е.	Людина для спрощення підрахунків почала складати з камінців невеличкі купки, зазвичай з десяти камінців. Замість камінців використовували також кістки, боби, мушлі тощо. Пізніше камінці або інші предмети почали викладати вздовж ліній на піску або на камені. Так був винайдений абак. Історики вважають, що вперше його почали використовувати у Вавилоні	 Мармурова дошка для підрахунків з острова Саламін (Греція), 300 років до н. е.
V ст. до н. е. і пізніше	У різних країнах з'являються конструкції абаків, що набули широкого розповсюдження та використовувались до XX ст. включно	 Римський абакус  Китайський суаньпань  Японський соробан  Рахівниця

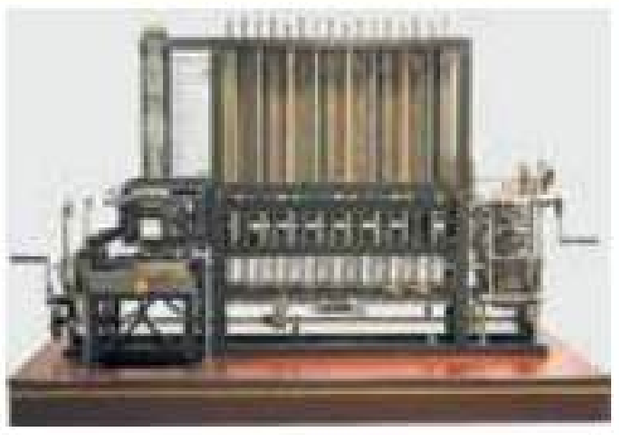
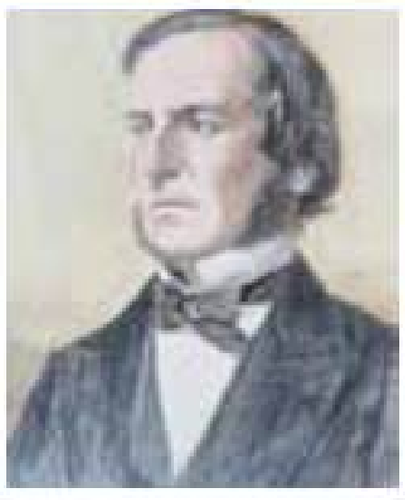
Продовження таблиці

Роки	**Опис обчислювального пристрою або події**	**Ілюстрації**
V ст. до н. е. і пізніше	Різні народи – китайці, перси, індійці, перуанці – використовували для обрахунків і запам'ятовування чисел ремінці та мотузки з вузликами. Інки такі мотузки ще й фарбували в різні кольори та називали квіпі	 Квіпі
100–150 років до н. е.	«Антикитерський механізм» – пристрій, що використовувався як календар, а також для автоматичного визначення положення небесних тіл – Сонця, Місяця – та відомих грекам планет – Меркурія, Венери, Марса, Юпітера, Сатурна	
Приблизно 850 р.	Аль-Хорезмі видає книжку «Арифметика», у якій викладає порядок здійснення обчислень. У подальшому чітку послідовність дій для отримання певного результату стали називати «алгоритмом» відповідно до звучання імені вченого	 Аль-Хорезмі
1614–1617 роки	Джон Непер винайшов логарифми. Він зауважив, що множення та ділення чисел може бути виконано додаванням і відніманням логарифмів цих чисел. Для спрощення обчислень Непер запропонував використовувати спеціальні палички	 Джон Непер
1623 р.	Вільгельм Шикард, професор університету Тюбінгену, розробив пристрій на основі зубчастих коліс для додавання та віднімання шестирозрядних десяткових чисел	 Відновлений за описом автора «Годинник для обчислень» В. Шикарда

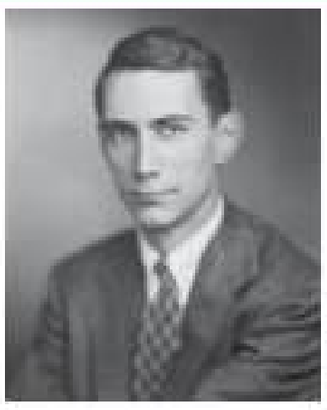
Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1630 р.	Річард Деламейн створив кругову логарифмічну лінійку	 Кругова логарифмічна лінійка, створена в кінці XX ст.
1642 р.	Французький математик, фізик, інженер і філософ Блез Паскаль подав на розгляд королівської ради свій механічний обчислювальний пристрій. У подальшому він створив кілька десятків таких пристроїв	 Одна з «Паскалін»
1654 р.	Англійці Р. Бісакар і незалежно від нього в 1657 р. С. Патрідж розробили прямокутну логарифмічну лінійку, конструкція якої майже без змін використовувалася до кінця XX ст.	 Фрагмент логарифмічної лінійки XX ст.
1666 р.	Англійський учений Самуел Морланд розробив просту у використанні машину для проведення обчислень	 Машина С. Морланда для обчислень
1673 р.	Готфрід Лейбніц створив перший у світі арифмометр – механічний обчислювальний пристрій, що виконував усі чотири арифметичні дії	 Арифмометр Лейбніца
1700 р.	Французький учений Клод Перо розробляє оригінальний компактний пристрій для проведення обчислень – рабдологічний абак	 Абак Клода Перо
1709 р.	Італійський учений Джованні Полені запропонував конструкцію дерев'яної обчислювальної машини. У цій машині вперше замість ручного приводу була використана вага додаткового вантажу, як у годинниках з гирями	 Арифмометр Д. Полені

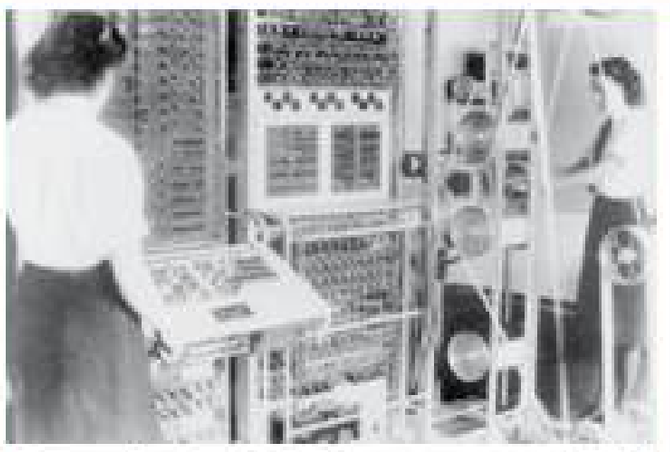
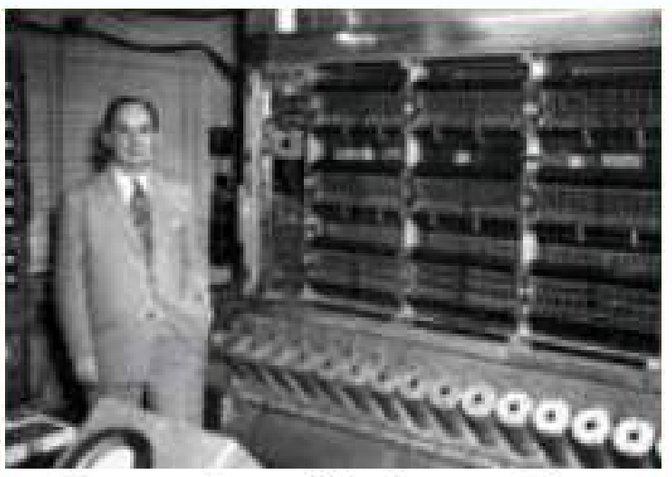
Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1774 р.	Німецький винахідник Філіп Матеус Ган створює арифмометр круглої форми. Пізніше його конструкцію удосконалює ще один німецький винахідник Й. Мюллер	 Арифмометр Й. Мюллера
1801 р.	Французький винахідник Жозеф Марі Жаккар збудував ткацький верстат з програмним керуванням, програма роботи якого задавалася комплектом перфокарт	 Перфокарти до верстата Ж. Жаккара
1820 р.	Перший промисловий випуск арифмометрів налагоджує французький винахідник Томас де Кальмар. Механічні калькулятори використовувалися до 1970-х років, коли були витіснені електронними пристроями	 Арифмометр Т. Кальмара
1822 р.	Англійський математик Чарлз Беббідж розробив конструкцію універсальної обчислювальної машини. За життя вченого її створення завершити не вдалося, але у XX ст. її було побудовано й підтверджено правильність ідей ученого	 Відтворена за кресленнями одна з машин Ч. Беббіджа
1854 р.	Англійський математик і логік Джордж Буль сформулював закономірності символьної логіки (так звана *Булева алгебра*), що становить основу сучасної комп'ютерної техніки	 Джордж Буль

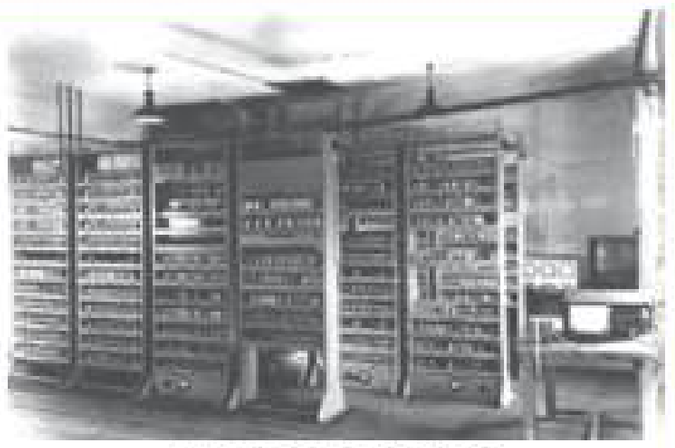
Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1884–1887 рр.	Герман Голлеріт (1860–1929) запропонував новий пристрій, робота якого базувалася на використанні електричного струму, – табулятор. Він був призначений для опрацювання даних перепису населення. Дані про конкретну людину не записувалися на аркуші паперу, а відмічались отворами у строго визначених місцях персональної карти – перфокарти	 Табулятор
1936 р.	Американський учений Клод Шенон створює теорію передачі даних різними каналами, уводить термін для обчислення кількості інформації – біт	 Клод Шенон
1937 р.	Англійський математик Алан Тюрінг створює концепцію «Машини Тюрінга». Ця концепція використовувалася для перевірки правильності (можливості) побудови алгоритмів розв'язку певних задач	 Алан Тюрінг
1937–1944 рр.	Говард Ейкен створює одну з перших електромеханічних обчислювальних машин Mark-1. Її активно використовували понад 15 років для розрахунків траєкторії артилерійських снарядів, прогнозування погоди тощо	 **Mark-1**
1938–1941 рр.	Німецький інженер Конрад Цузе створює цикл машин від механічної Z1 до електромеханічної Z3. Програмування останньої здійснювалося з використанням отворів у кінострічці	 Конрад Цузе
1938–1941 рр.	У США Джон Вінсент Атанасов і його асистент Кліфорд Беррі створили перший комп'ютер на електронних лампах з використанням двійкового кодування – ABC (англ. *Atanasoff Berry Computer*)	 Відтворена в кінці XX ст. ЕОМ **ABC**

Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1943 р.	Колектив англійських учених (М. Ньюман, Т. Флауерс та ін.) за участі А. Тюрінга створює першу електронну обчислювальну машину «Colossus», що використовувалася для розшифровування німецьких військових кодів «Енігма» та «Лоренц»	 За обслуговуванням ЕОМ **Colossus**
1943–1946 рр.	У США Джон Моклі та Преспер Еккерт створили ENIAC – електронну обчислювальну машину, що містила 18 000 електронних ламп, важила 30 тонн і виконувала 5000 операцій за секунду	 ЕОМ **ENIAC**
1946 р.	Американський математик Джон фон Нейман у статті «Попередня доповідь про машину EDVAC» сформулював основні принципи побудови універсальної обчислювальної машини. Незважаючи на те, що ці ідеї були розроблені Еккертом і Моклі, вони стали відомі всьому світові як «принципи фон Неймана»	 Джон фон Нейман біля ЕОМ **MANIAC**
1947 р.	Грейс Мюрей Хопер, американська програмістка та контрадмірал флоту США, в ході усунення недоліків у роботі ЕОМ Mark II знайшла метелика, який застряг в одному з компонентів і замкнув контакти реле. Це був перший «баг» (англ. *bug* – комаха) – перша комп'ютерна помилка. Хопер належала ідея використання автоматичної компіляції програм. Вона стояла у витоків створення перших мов програмування високого рівня	 Запис у щоденнику про знайденого метелика  Грейс Мюрей Хопер

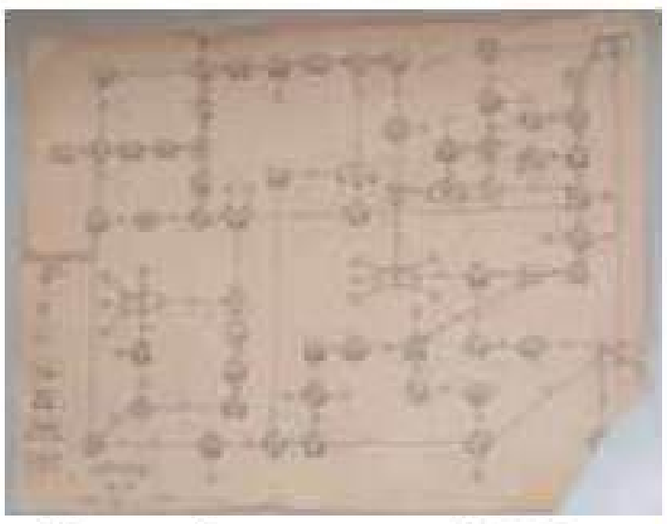
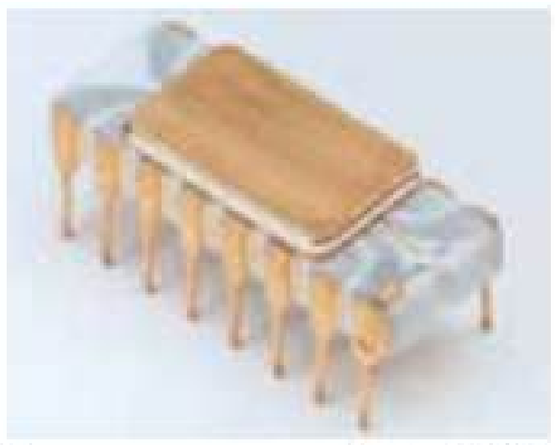
Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1949 р.	У Кембридзькому університеті (Велика Британія) під керівництвом Моріса Вілкіса створюється ЕОМ з двійковим кодуванням і програмами, що зберігаються в пам'яті, – EDSAC	 ЕОМ **EDSAC**
1945–1950 рр.	Конрад Цузе створює комп'ютер на електронних лампах – Z4. Цей комп'ютер був проданий швейцарському університету й успішно працював там протягом тривалого часу	 ЕОМ **Z4**
1945–1951 рр.	На замовлення військового відомства США для контролю за радарними станціями створюється ЕОМ Whirlwind (англ. *Whirlwind* – вихор, ураган). Уперше в ЕОМ використовуються аналоги сучасних пристроїв уведення та виведення даних – клавіатура та монітор	 ЕОМ **Whirlwind**
1947 р.	Американські вчені Джон Бардін, Вільям Шоклі та Волтер Браттейн створюють перший транзистор. Сьогодні транзистори у складі електронних мікросхем є основними елементами комп'ютерної техніки	 Лабораторний зразок першого транзистора
1950 р.	Алан Тюрінг в одній зі своїх робіт пропонує програму перевірки комп'ютерів на можливість імітації людського розуму, яка отримала назву «Тест Тюрінга»	
1951 р.	У Києві під керівництвом Сергія Олексійовича Лебедєва введено в дію універсальну ЕОМ з програмою, що зберігалася в пам'яті, – МЭСМ. Це була перша ЕОМ у Радянському Союзі	 За пультом **МЭСМ**

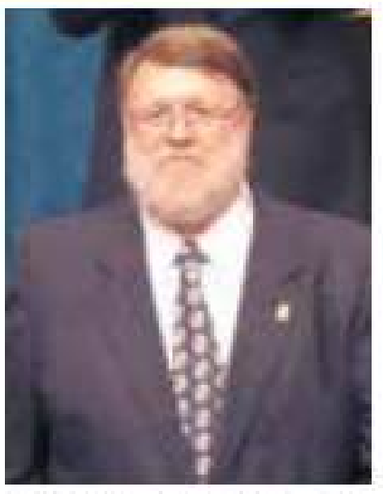
Продовження таблиці

Роки	**Опис обчислювального пристрою або події**	**Ілюстрації**
1954–1957 рр.	З'являються перші матричні принтери і прототипи перших вінчестерів (IBM 305 RAMAC)	
1956 р.	У Київському університеті імені Т. Г. Шевченка і Київському політехнічному інституті В. С. Королюк і К. Л. Ющенко розпочинають читання лекцій з програмування	 К. Л. Ющенко
1958 р.	Американські вчені Джек Кілбі та Роберт Нойс незалежно один від одного винаходять інтегральну схему. Уперше більш ніж один транзистор був розміщений на одному напівпровідниковому кристалі. Зараз на одному кристалі розміщується кілька мільярдів транзисторів	 Перша мікросхема
1959 р.	Під керівництвом В. М. Глушкова завершено розробку ЕОМ «Київ» (безпосередні розробники Б. В. Гнеденко, Л. М. Дашевський, К. Л. Ющенко)	 ЕОМ **«Київ»**
1961 р.	Під керівництвом В. М. Глушкова завершено розробку універсальних ЕОМ «Дніпро» (головний конструктор Б. М. Малиновський). Це перші в СРСР комп'ютери на основі напівпровідникових елементів. Обчислювальний центр Академії наук України реорганізовано в Інститут кібернетики. Директором призначено В. М. Глушкова	 ЕОМ **«Дніпро-2»**
1963 р.	Під керівництвом В. М. Глушкова започатковано серію ЕОМ для інженерних розрахунків «Промінь». Захищена перша в СРСР докторська дисертація з програмування. Її захистила К. Л. Ющенко. Тема дисертації – «Адресне програмування»	 В. М. Глушков і розробники за пультом ЕОМ **«Промінь»**
1963 р.	Дуглас Енгельбарт (США) отримує патент на пристрій введення, який у подальшому дістав назву «миша»	 Один з перших екземплярів миші

Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1964 р.	Корпорація IBM анонсує створення серії машин – «Система 360» (System 360 або IBM/360). Число 360 указувало на «всеспрямність» застосувань машин цієї серії. Тобто машини були універсальними та могли застосовуватись у різних галузях. Крім того, деталі машин були взаємозамінними, а програмне забезпечення сумісним	 ЕОМ **IBM/360**
1965 р.	Інститутом кібернетики Академії наук України під керівництвом академіка В. М. Глушкова створено ЕОМ МІР. Одна з перших у світі персональних ЕОМ. Випускалася серійно та призначалася для використання в навчальних закладах, невеликих інженерних бюро та наукових відділах	 ЕОМ **МІР**
1965 р.	Фірма Texas Instruments випустила перший кишеньковий напівпровідниковий калькулятор. Він міг здійснювати чотири арифметичні операції і результат роздруковував на паперовій стрічці	 Калькулятор **Cal Tech**
1969 р.	У структурі Агентства з перспективних досліджень Міністерства оборони США розпочато розробку і впровадження глобальної мережі. Спочатку це була військова мережа ARPAnet, яка в подальшому переросла в інтернет. 2 січня – день початку цих робіт – вважається днем народження інтернету. Перша мережа об'єднала чотири комп'ютери в університетських центрах у Лос-Анджелесі, Санта-Барбарі, Солт-Лейк-Сіті та у Стенфордському дослідницькому інституті (Менло-Парк, Каліфорнія)	 Карта інтернету, 1979 р.
1971 р.	Фірма Intel (США) випускає перший у світі мікропроцесор Intel-4004, який містить 2250 транзисторів	 Мікропроцесор **Intel-4004**

Продовження таблиці

Роки	Опис обчислювального пристрою або події	Ілюстрації
1971 р.	Рей Томлінсон розробляє основи електронної пошти та вводить для позначення її адрес символ @	 Рей Томлінсон
1975 р.	Розпочато випуск першого персонального комп'ютера, доступного широким верствам населення, ALTAIR 8800 фірми MITS на основі мікропроцесора Intel 8080	 **ALTAIR 8800**
1975 р.	Рік створення корпорації Apple Computer. Початок випуску персональних комп'ютерів Apple I	 ПК **Apple I**
1978 р.	Корпорація Intel випустила процесор 8086, який став першим у великому сімействі x86 процесорів	
1981 р.	Корпорація IBM розробила свій перший персональний комп'ютер, що отримав назву IBM 5150 Personal Computer, або скорочено IBM PC. Комп'ютер мав відкриту архітектуру, що дало можливість багатьом фірмам не тільки розробляти пристрої для нього, але й створювати подібні комп'ютери самостійно. Зародилося поняття «IBM-сумісний комп'ютер». Для цього комп'ютера фірма Microsoft розробила операційну систему MS DOS	 **IBM PC**
1984 р.	Apple Computer Corporation випускає перший масовий персональний комп'ютер з графічним інтерфейсом операційної системи Macintosh. У ньому в одному з перших почали використовуватися дискети 3,5-дюймового формату	 **Macintosh**

Додаток 3

ПРИКЛАДИ ІНСТРУМЕНТІВ ГЕНЕРАТИВНОГО ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ОСВІТИ

За матеріалами ІНСТРУКТИВНО-МЕТОДИЧНИХ РЕКОМЕНДАЦІЙ щодо запровадження та використання технологій штучного інтелекту в закладах

Назва, покликання	*Опис*	*Умови безоплатного використання*
Чат-боти на основі великих мовних моделей Універсальні інструменти роботи з текстом для різних цілей: пошуку ідей, планування, розробки дидактичних матеріалів, аналітики даних, оцінювання та ін.		
Chat GPT 3.5 *https://chat.openai.com/*	Робота з текстом	Без помітних обмежень, без реєстрації
Googe Gemini *https://gemini.google.com/*	Робота з текстом і зображеннями	Без помітних обмежень
Microsoft Copilot *https://copilot.microsoft.com/*	Робота з текстом, генерація зображень, можливість установлення додаткових плагінів	Без помітних обмежень, можна використовувати без реєстрації
Claude *https://claude.ai/*	Робота з текстом	Без помітних обмежень
Генерація та редагування зображень Інструменти для підготовки дидактичної наочності й творчих учнівських проєктів		
Ideogram *https://ideogram.ai/*	Генератор зображень	25 генерацій по 4 зображення на добу
Copilot Дизайнер *https://www.bing.com/create*	Генератор зображень	15 генерацій по 4 зображення на добу швидко, далі — повільно
Freepik Pikaso *https://www.freepik.com/pikaso/sketch*	Генератор зображень за текстовим промптом і за мальованим ескізом	Кілька використань на добу
Vizcom *https://app.vizcom.ai/*	Генератор зображень за текстовим промптом і за мальованим ескізом	Без помітних обмежень
ClipDrop Hotpot *https://hotpot.ai/*	Видалення фону, заміна фону, видалення об'єктів на фото, переосвітлення, домальовування та ін.	Деякі з інструментів по декілька використань з обмеженням роздільної здатності зображень на добу

Продовження таблиці

Назва, покликання	Опис	Умови безоплатного використання
Restore Photos *https://www.restorephotos.io/restore*	Реставрація старих фото: затертості, подряпини, згини	До 5 генерацій на добу
Palette *https://palette.fm/*	Розфарбовування чорно-білих фото	Обмеження роздільної здатності зображень
Дизайн і схематична візуалізація Інструменти для створення колажів, ментальних карт, інфографіки, діаграм та ін.		
Microsoft Designer *https://designer.microsoft.com/*	Створення колажів, обкладинок та інших візуальних матеріалів за текстовим промптом	Без помітних обмежень
Hue hive *https://huehive.co/*	Добір кольорової палітри за текстовим промптом	Без помітних обмежень
AutoDraw *https://www.autodraw.com/*	Перетворення власних простих малюнків і піктограми	Без помітних обмежень
MyLens *https://mylens.ai/*	Генератор ментальних карт і стрічок часу	До 3 генерацій на добу
HoverCode *https://hovercode.com/*	Генератор унікальних і художніх QR-кодів на основі текстових описів	До 10 000 сканувань на місяць, 10 динамічних QR-кодів
Miro *https://miro.com/*	Візуальне робоче місце для співпраці та спільного планування із застосуванням ШІ для автоматизації завдань, пропонування шаблонів і візуалізації даних	До 3 спільних дощок
Відео, анімація Інструменти для підготовки дидактичної наочності у відеоформаті й творчих учнівських проєктів		
Vidnoz AI *https://www.vidnoz.com/*	Створення відеоуроків з віртуальними дикторами, «оживлення» й озвучення фотопортретів	Вихідне відео тривалістю до 3 хв на добу
Pixverse *https://app.pixverse.ai/*	Генератор коротких відео за текстовим промптом, «оживлення» фото	До 5 генерацій на добу
Pika *https://pika.art/*	Генератор коротких відео за текстовим промптом, «оживлення» фото	Декілька генерацій на добу

Продовження таблиці

Назва, покликання	***Опис***	***Умови безоплатного використання***
ClipChamp *https://clipchamp.com/*	Автоматична генерація слайдшоу з власних фото, відеоредактор	Без помітних обмежень
Animated Drawings *https://sketch.metademolab.com/*	Анімування власного малюнка	Без помітних обмежень
Adobe Express *https://www.adobe.com/express/feature/video/animate/audio*	Оживлення персонажа зі своєю озвучкою	Без помітних обмежень
Filmot *https://filmot.com/*	Пошук **YouTube**-роликів з певною фразою	Без помітних обмежень
Аудіо Інструменти для генерації вимови та музики, обробки голосових записів, пошуку музики		
Suno AI *https://app.suno.ai/*	Генератор пісень будь-якою мовою	До 10 генерацій на добу
Eleven Labs *https://elevenlabs.io/*	Синтезатор вимови різними голосами, клонування власного голосу	До 10 тис. символів на місяць
Voice Changer *https://voicechanger.io/*	Зміна голосу в різних стилях	Без помітних обмежень
Audioatlas *https://www.audioatlas.com/*	Пошук реальної музики за детальним текстовим промптом (наприклад, музика для навчального відео)	Без помітних обмежень
Тривимірна графіка Інструменти для генерації 3D-моделей і віртуальних аватарів		
Luma AI *https://lumalabs.ai/dashboard/captures*	Оцифровування реальних об'єктів (скануємо смартфоном і отримуємо його 3D-модель)	Без помітних обмежень
Meshy *https://www.meshy.ai/*	Генерація 3D-моделей за текстовим промптом	Декілька генерацій
in3D (Android) *https://play.google.com/store/apps/details?id=gsize.Avatar*	Оцифровування свого тіла та обличчя (скануємо смартфоном і отримуємо 3D-модель)	Декілька аватарів
Avaturn *https://avaturn.me/*	Створення власного 3D-обличчя зі штучним тілом	Без помітних обмежень

Продовження таблиці

Назва, покликання	***Опис***	***Умови безоплатного використання***
Боти-помічники та пошук інформації Інструменти для створення власних асистентів, агенти для виконання конкретних завдань		
Ghola *https://www.ghola.ai/*	Створення власного ШІ-бота персонажа, історичного діяча, митця тощо	Без помітних обмежень
ChatTube *https://chattube.io/*	Чат з ШІ-ботом за змістом будь-якого **YouTube**-відео, резюмування вмісту відео	До 3 відео на добу
Tutor AI *https://www.tutorai.me/*	Генератор онлайн-курсів з будь-якої теми (англ.)	3 курси
Chat D-ID *https://chat.d-id.com/*	Чат з «оживленим» будь-яким обличчям	6 діалогів
Poe *https://poe.com/*	Доступ одночасно до різних мовних моделей, створення власних ШІ-ботів	По-різному, залежно від обраної мовної моделі
Mizou *https://mizou.com/*	Створення навчальних ботів і різних видів завдань. Передбачена робота без створення облікових записів учнів	50 учнівських сесій на день
SchoolAI *https://schoolai.com/*	Створення навчальних ботів і різних видів завдань	Без помітних обмежень
MagicSchoolAI *https://www.magicschool.ai/*	Створення планів занять, навчальних ботів і різних видів завдань. Доступний моніторинг роботи учнів із сервісом	Без помітних обмежень
Harpa *https://harpa.ai/*	Автоматизація вебзавдань, генерація тексту, мовний помічник у браузері	До 10 повідомлень на добу
Learn anything *https://learn-anything.xyz/*	Організація світових знань, персоналізація рекомендацій і створення маршрутів навчання із заданої теми	Обмеження на кількість закладок і створених маршрутів навчання
Perplexity *https://www.perplexity.ai/*	Пошукова система на основі штучного інтелекту	Без помітних обмежень
Consensus *https://consensus.app/*	Пошук та узагальнення наукових статей	Пошук до 3 статей на день

Продовження таблиці

Назва, покликання	Опис	Умови безоплатного використання
Презентації, робота з текстом Інструменти для спеціалізованої генерації текстових матеріалів		
Wepik *https://wepik.com/ai-presentations*	Генератор і редактор презентацій за текстовим промптом	Без помітних обмежень
Prezo *https://prezo.ai/*	Генератор і редактор презентацій за текстовим промптом	Без помітних обмежень
AhaSlides *https://ahaslides.com/uk/*	Генератор і редактор презентацій за текстовим промптом, інтерактивне залучення аудиторії	2 презентації на місяць, до 7 учасників, обмежені функції
Kazka fun *https://kazka.fun/*	Генератор казок	Без помітних обмежень
2txt *https://2txt.vercel.app/*	Розпізнає зображення й виводить текст з нього	5 безоплатних описів зображень на день
Grammarly *https://www.grammarly.com/*	Редактор тексту для покращення граматики, правопису, стилю та чіткості письма	Базова перевірка граматики, правопису та стилю
Гумор Інструменти для нестандартних творчих проєктів, дидактичних матеріалів тощо		
Super Meme *https://www.supermeme.ai/*	Генератор мемів за текстовим промптом	До 10 генерацій на добу
MemeCam *https://www.memecam.io/*	Генератор мемів за зображенням	Без помітних обмежень, добровільний донат
Ігри Інструменти для ознайомлення з можливостями ШІ в ігровій формі		
Emoji scavenger hunt *https://emojiscavengerhunt.withgoogle.com/*	Рухливий квест зі смартфоном: знайти у кімнаті предмет, схожий на той чи інший емоджі, й навести на нього камеру смартфона	Без помітних обмежень
Quick Draw *https://quickdraw.withgoogle.com/*	Гра з механікою: намалювати за 20 секунд той чи інший об'єкт, ШІ має вгадати	Без помітних обмежень

Продовження таблиці

Назва, покликання	***Опис***	***Умови безоплатного використання***
Odd One Out *https://artsandculture.google.com/experiment/odd-one-out/wAHNn4JsVTFOiw*	Гра з механікою: знайти з поданих зображень справжні об'єкти мистецтва та ШІ-фейки	Без помітних обмежень
За навчальними предметами Спеціалізовані інструменти з окремих галузей знань		
Мови		
DeepL *https://www.deepl.com/translator*	Перекладач на основі ШІ	Декілька десятків використань на добу
Gliglish *https://gliglish.com/free*	Тренажер з іноземних мов: генерація діалогів, розпізнавання вимови	Без помітних обмежень
Twee *https://twee.com/*	AI-помічник для викладачів англійської мови: створення завдань, діалогів, історій, листів, генерування питань до відео та ін.	До 50 завдань на день, 2 діалоги, 2 історії, 2 листи, 10 питань до відео
LanguageTool *https://chromewebstore.google.com/detail/grammar-checker-paraphras/oldceeleldhonbafppcapldpdifcinji*	Браузерне розширення для перевірки орфографії	Без помітних обмежень
Математика		
PhotoMath (Android) *https://play.google.com/store/apps/details?id=com.microblink.photomath&hl=uk&gl=US*	Покрокове розв'язування математичних завдань за фото	Розв'язування завдань без додаткових навчальних функцій
Microsoft Math Solver (Android) *https://play.google.com/store/apps/details?id=com.microsoft.math*	Покрокове розв'язування математичних завдань за фото, ігровий режим	Без помітних обмежень
Hissab *https://hissab.io/*	Просунутий багатофункціональний калькулятор	Без помітних обмежень

Назва, покликання	*Опис*	*Умови безоплатного використання*
Природничі науки		
Experiments with Google *https://experiments.withgoogle.com/search?q=physics*	Фізичні експерименти	Без помітних обмежень
HyperWrite AI Chemistry Assistant *https://www.hyperwriteai.com/aitools/chemistry-assistant*	Помічник у вивченні хімії та розв'язуванні задач	Пояснення термінів, пошук інформації, деякі навчальні матеріали. Для розв'язування задач потрібна платна підписка
HyperWrite AI Biology Assistant *https://www.hyperwriteai.com/aitools/ai-biology-learning-assistant*	Помічник у вивченні біології та розв'язуванні задач	Пояснення термінів, пошук інформації, деякі навчальні матеріали. Для розв'язування задач потрібна платна підписка
Інформатика, робототехніка		
Tabnine *https://www.tabnine.com/*	Кодування / програмування, дописування коду за користувачем, модулювання та візуалізація, аналіз даних	Обмежена кількість рядків коду на день
Codepal *https://codepal.ai/*	Генерування коду з природної мови, пояснення, рецензування, виявлення помилок, спрощення	До 1000 рядків коду на місяць
CodeSnippets *https://codesnippets.ai/*	Бібліотека кодових фрагментів з підтримкою штучного інтелекту (генерація коду, перевірка) для команд	До 100 фрагментів коду, співпраця з 1 користувачем
Google Colab *https://colab.research.google.com/*	Хмарний сервіс для виконання коду **Python** та машинного навчання за допомогою TensorFlow, PyTorch, Keras та ін.	12 ГБ TPU v4-32 на 24 години, 10 ГБ GPU Tesla K80 на 24 години, безоплатне зберігання 15 ГБ
The Construct *https://www.theconstruct.ai/*	Платформа для онлайн-навчання робототехніки	Безоплатна пробна версія з обмеженими можливостями
Teachablemachine *https://teachablemachine.withgoogle.com/*	Розпізнавання об'єктів чи зображень, звуків і положення тіла перед вебкамерою. Можна використовувати самостійно або з інструментами для з'єднання ШІ з micro:bit (наприклад, через інструмент **Makeairobots**)	Без помітних обмежень

СЛОВНИК

Абсолютне посилання – посилання на клітинку електронної таблиці, яке не модифікується під час копіювання формули, с. 199.

Адаптивний дизайн – дизайн вебсторінки, при забезпеченні якого одна й та сама сторінка, можливо, по-різному, але в одному стилі, коректно відображається і на великих, і на маленьких екранах комп'ютерних пристроїв, с. 167.

Адреса клітинки – це унікальне позначення, яке вказує на точне розташування цієї клітинки в електронній таблиці. Вона складається з номера стовпця та рядка, на перетині яких вона розміщена, с. 193.

Аргументи функції – дані, над якими виконується обчислення у функціях (*числа, тексти, вирази, посилання, інша функція*), с. 206.

Арифметично-логічний пристрій – складова процесора, яка забезпечує здійснення операцій над даними, с. 52.

Аркуш електронної книги – об'єкт електронної книги, на якому розміщується електронна таблиця, діаграма тощо, с. 192.

Архів (комп'ютерний) – копії даних, як правило, упакованих в один файл, с. 31.

Архівний файл – файл, що містить копії даних у стисненому або нестисненому вигляді, с. 31.

Архіватор – програма, з використанням якої створюються архіви даних, с. 42.

Архівування – процес створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, с. 42.

Байт – послідовність з восьми бітів, с. 21.

Біт – цифра 0 або 1 у двійковому коді повідомлення, с. 20.

Вбудовані функції – функції, які наявні в бібліотеці функцій табличного процесора (*математичні, статистичні, текстові, логічні, фінансові та ін.*), с. 205.

Висловлювання – речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами і про яке можна однозначно сказати, **істинне** воно чи **хибне**, с. 127.

Висота тону – одна з основних властивостей звуку, що використовується для його кодування; визначається частотою коливань звукової хвилі, с. 16.

Відносне посилання – посилання на клітинку електронної таблиці, яке модифікується під час копіювання формули, с. 199.

Відсотковий формат – формат даних в електронній таблиці, який використовується для подання числових даних в перерахунку на відсотки (%), с. 198.

Вкладені функції – функції табличного процесора, у яких результат однієї функції використовується як аргумент іншої, с. 206.

Внутрішня пам'ять – складова пам'яті комп'ютера, що призначена для забезпечення роботи процесора. Складається внутрішня пам'ять з оперативної, постійної і кеш-пам'яті, с. 51.

Генеративна модель – алгоритм машинного навчання, здатний створювати нові дані, імітуючи стиль та структуру наявних даних, с. 256.

Глибина кольору – кількості бітів, вибраних для кодування кольорів на зображенні, с. 23.

Гучність – одна з основних властивостей звуку, що використовується для його кодування; визначається амплітудою звукової хвилі, с. 16.

Графічний планшет – пристрій уведення графічних даних. Використовується властивість робочої поверхні планшета реагувати на дотик стилуса. Користувач створює малюнок, дотикаючись стилусом до поверхні графічного планшета, с. 66.

Грошовий формат – формат даних в електронній таблиці, який використовується для подання числових даних з додаванням позначення грошової одиниці, с. 197.

Датчики (сенсори) – об'єкти для відслідковування стану навколишнього середовища, реалізують з'єднання фізичного і віртуального світів, забезпечуючи збирання й опрацювання даних в реальному часі, с. 248.

Двійкова система числення – система числення, у якій для запису чисел використовують лише цифри 0 та 1, с. 22.

Двійкове кодування – кодування повідомлень з використанням сигналів лише двох видів, с. 20.

Двійковий код – повідомлення, отримане в результаті двійкового кодування, с. 20.

Декодування – процес отримання початкового повідомлення із закодованого повідомлення, с. 7.

Диз'юнкція двох логічних виразів x і y – логічна операція, результат якої дорівнює **True**, якщо значення хоча б одного з логічних виразів x або y дорівнює **True**, і дорівнює **False**, якщо значення кожного з логічних виразів x і y дорівнює **False**, с. 130.

Діапазон клітинок – деяка сукупність клітинок електронної таблиці, с. 193.

Довжина двійкового коду повідомлення (**обсяг даних**) – це кількість бітів у двійковому коді цього повідомлення, с. 21.

Дробовий формат – формат даних в електронній таблиці, який використовується для подання числа у вигляді звичайного дробу, с. 198.

Друкування таблиці – перенесення електронної таблиці, створеної в табличному процесорі, на папір за допомогою принтера, с. 237.

Експоненційний формат – формат даних в електронній таблиці, який використовується для подання числа в стандартному вигляді, с. 198.

Електронна книга – документ табличного процесора, який є сукупністю кількох електронних таблиць, розміщених на окремих аркушах, с. 192.

Електронна таблиця – основний об'єкт табличного процесора, який складається з рядків і стовпців, с. 193.

Ергономіка – складова дизайну, формулює вимоги до зручності та зрозумілості під час користування сайтом, с. 164.

Загальний формат – формат даних в електронній таблиці за замовчуванням, с. 196.

Заперечення логічного виразу x – логічна операція, результат якої дорівнює **True**, якщо значення логічного виразу x дорівнює **False**, і дорівнює **False**, якщо значення логічного виразу x дорівнює **True**, с. 129.

Запит до системи штучного інтелекту (prompt) – текстове формулювання, яке задає конкретне завдання моделі штучного інтелекту, с. 264.

Захист проєкту – публічне представлення результатів проєкту , с. 268.

Збір інформації в проєкті – пошук потрібних матеріалів у різних джерелах, с. 268.

Змінна логічного типу – змінна, яка може набувати одного з двох значень: **True** або **False**, с. 128.

Зовнішня пам'ять – складова пам'яті комп'ютера, що призначена для тривалого зберігання даних. До її складу входять пристрої на магнітних та оптичних дисках, флешнакопичувачі тощо, с. 51.

Інтернет речей (IoT) – концепція, яка передбачає об'єднання різних фізичних пристроїв через інтернет, забезпечуючи їм взаємодію та обмін даними, с. 246.

Інтернет-безпека – система захисту онлайн-даних від несанкціонованого втручання, с. 244.

Ім'я клітинки – персоналізована адреса клітинки в електронній таблиці, яка не модифікується під час копіювання у формулі, с. 193.

Ім'я функції – унікальне словосполучення, яке ідентифікує функцію в бібліотеці функцій табличного процесора, с. 206.

Карта (мапа) сайту – структура, яка містить перелік тематичних розділів сайту та сторінок у них, визначає зв'язки між сторінками, пояснює логічну структуру сайту, с. 163.

Код – система умовних сигналів для передавання, опрацювання та зберігання повідомлень, с. 5.

Кодування – процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю сигналів, с. 5.

Колірна модель – спосіб кодування різних кольорів спектра впорядкованим набором числових значень певних базових компонентів, с. 14.

Колористика – складова дизайну, формулює вимоги до гармонійності поєднання кольорів на сторінках сайту, с. 164.

Композиція – складова дизайну, формулює вимоги до узгодженості розмірів об'єктів і їх розташування на сторінці сайту, с. 164.

Контент – текстові, графічні, мультимедійні та інтерактивні елементи на сторінках сайту, с. 165.

Кон'юнкція двох логічних виразів x і y – логічна операція, результат якої дорівнює **True**, якщо значення кожного з логічних виразів x і y дорівнюють **True**, і дорівнює **False**, якщо значення хоча б одного з логічних виразів x або y дорівнює **False**, с. 129.

Клавіатура – пристрій введення, що призначений для введення символьних даних і команд, с. 61.

Клітинка – об'єкт електронної таблиці, який утворився на перетині рядків і стовпців таблиці, с. 193.

Контролери – спеціальні пристрої, що призначені для управління зовнішніми пристроями комп'ютера або датчиками, с. 248.

Контрольні точки відновлення системи – копії (архіви) системних налаштувань, с. 37.

Кратні одиниці вимірювання довжини двійкового коду – утворюються з використанням префіксів *кіло-*, *мега-*, *гіга-*, *тера-* та інших, с. 24.

1 кБ (кілобайт) = 2^{10} Б = 1024 Б

1 МБ (мегабайт) = 2^{10} кБ = 2^{20} Б = = 1 048 576 Б

1 ГБ (гігабайт) = 2^{10} МБ = 2^{20} кБ = = 2^{30} Б

1 ТБ (терабайт) = 2^{10} ГБ = 2^{20} МБ = = 2^{30} кБ = 2^{40} Б

Логічна функція (IF, AND, OR, NOT) – функція, результат якої дорівнює **True** або **False**, с. 128, 213.

Логічна помилка – причина неспівпадіння результатів виконання проєкту з очікуваними результатами, с. 105.

Логічний вираз – вираз, який може набувати одного з двох значень: **True** або **False**, с. 128, 213.

Мапа сайту – див. Карта сайту, с. 163.

Материнська, або системна, плата – основна складова сучасних персональних комп'ютерів, яка забезпечує передавання даних між пристроями комп'ютера, с. 53.

Машинне навчання – підгалузь штучного інтелекту, яка визначає спосіб самостійного виконання комп'ютерними системами завдань та ухвалення рішень, аналізуючи великі обсяги даних і виявляючи в них закономірності, без явного їх програмування, с. 256.

Методи збирання даних – спостереження, опитування, опрацювання друкованих джерел та історичних знахідок, пошук відомостей в інтернеті, фіксація даних з датчиків, результатів експериментів та інші, с. 268.

Мікрофон – пристрій, що здійснює перетворення звукових коливань в електричні, с. 67.

Мішане посилання – посилання на клітинку електронної таблиці, яке частково модифікується під час копіювання формули (або номер рядка, або номер стовпця), с. 199.

Модифікація формули – автоматична зміна адрес клітинок у формулах під час їх копіювання в табличному процесорі, с. 199.

Монітор – основний пристрій для виведення даних у персональних комп'ютерах. Монітори поділяють за системою створення зображення на монітори на рідинних кристалах або LCD-монітори, плазмові, OLED, на електронних чорнилах – e-ink тощо, с. 71.

Навчальний проєкт – це вид та результат діяльності, яка спрямована на досягнення певної навчальної мети, розв'язування деякої проблемної задачі, с. 268.

Напис – елемент керування, який використовується для виведення у вікні текстових повідомлень, с. 114.

Нейронні мережі – математичні моделі, натхнені біологічними нейронними мережами, які складаються із взаємопов'язаних вузлів, що обробляють інформацію та адаптуються до даних, імітуючи роботу біологічних нейронів мозку людини, с. 256.

Обсяг даних – див. Довжина двійкового коду, с. 21.

Обробка даних у проєкті – аналіз та опрацювання зібраної інформації, її систематизація та формулювання висновків, с. 269.

Оперативна пам'ять – основна пам'ять комп'ютера, у якій розміщуються програми та дані, що в подальшому опрацьовуються процесором, с. 53.

Оформлення результатів проєкту – створення інформаційних продуктів за підсумками виконання проєкту, с. 269.

Перемикачі – елементи керування, які використовують для вибору тільки одного з кількох можливих варіантів. З кількох перемикачів однієї групи може бути вибраний тільки один, с. 146.

Планування роботи в проєкті – складання детального плану вико-

нання проєкту, встановлення термінів, розподіл завдань та ролей між учасниками проєкту, с. 268.

Плотер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір А2 і більше, с. 75.

Поле – елемент керування, який використовується для введення значень вхідних даних і виведення значень результатів, с. 119.

Постановка завдання проєкту – чітке формулювання проблеми, цілі та завдань проєкту, с. 268.

Постійна пам'ять – пам'ять комп'ютера, яка містить програми та дані, які потрібні для виконання дій, пов'язаних з початком роботи комп'ютера, с. 53.

Прапорці – елементи керування, які використовують для вибору одного або кількох варіантів з набору можливих чи не вибору жодного. З кількох прапорців може бути вибраний або один, або кілька, або всі, або жодний, с. 144.

Принтер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір зазвичай не більше ніж А3, с. 73.

Пристрій керування – складова процесора, яка забезпечує виконання команд комп'ютерних програм, с. 52.

Пристрої керування роботою програм – пристрої, що використовуються для передавання команд від користувача комп'ютерним програмам, с. 62.

Проєкційна клавіатура – тип пристрою введення даних, у якому зображення клавіатури проєктується на плоску поверхню і введення даних здійснюється «натисненням» на зображення клавіш, с. 62.

Простий фільтр – це інструмент у табличному процесорі, який дозволяє швидко відфільтрувати дані в таблиці за одним критерієм в одному стовпці, с. 224.

Процесор – пристрій, який забезпечує опрацювання даних у комп'ютері. Він містить пристрій керування, який забезпечує виконання команд комп'ютерної програми, та арифметично-логічний пристрій, який і здійснює операції над даними, с. 52.

Режим попереднього перегляду – це відображення в табличному процесорі електронної таблиці в тому вигляді, як вона буде надрукована на папері, с. 238.

Результат функції – значення, яке повертається функцією після виконання обчислень, с. 206.

Розгалуження – фрагмент алгоритму, який розпочинається з команди визначення значення логічного виразу і в якому виконуються або не виконуються певні команди залежно від значення цього виразу, с. 135.

Резервне копіювання – процес створення копій даних та налаштувань операційної системи, с. 32.

«Розумний» (Smart) будинок – будинок, у якому використовуються цифрові технології для автоматизації побутових операцій, с. 244.

«Розумний» пристрій – електронний пристрій, здатний самостійно виконувати завдання, підключатися до інтернету та взаємодіяти з користувачем або іншими пристроями, с. 243.

Рядок – об'єкт електронної таблиці, який містить опис одного об'єкта за всіма його властивостями, с. 193.

Сенсорний екран – пристрій уведення і виведення даних. Для введення даних використовується властивість поверхні екрана

реагувати на дотик пальця або стилуса, с. 64.

Сканер – пристрій для введення графічних даних. За принципом дії виділяють планшетні, з протяжним механізмом, ручні та сканкамери, с. 64.

Сортування – змінення порядку розташування даних в рядках або стовпцях таблиці за деякими ознаками, с. 220.

Стиснення даних – процес кодування даних, який здійснюється з метою зменшення розмірів файлів, с. 30.

Стовпець електронної таблиці – об'єкт електронної таблиці, який містить опис однієї властивості для всіх об'єктів таблиці, с. 193.

Сублімаційний друк – друк, що використовує явище сублімації – фізичне явище переходу речовини з твердого стану в газоподібний, минаючи рідкий стан, с. 74.

Табличний процесор – це прикладна програма, яка призначена для опрацювання даних, поданих в електронних таблицях, с. 192.

Таблиця істинності – таблиця, у якій обчислюється значення логічного виразу за всіх можливих значень змінних, які входять у цей вираз, с. 129.

Таблиця кодів символів – таблиця для кодування текстів, у якій кожному символу, що може бути використаний у текстовому повідомленні, поставлено у відповідність деяке число, с. 11.

Тачпед – пристрій уведення, що використовується в ноутбуках і нетбуках, для керування роботою комп'ютера, с. 63.

Тег – елемент мови **HTML**, що вказує браузеру, який об'єкт розмістити на сторінці та як його відобразити, с. 172.

Текстовий формат – формат даних у електронній таблиці, який використовується для подання чисел як тексту, с. 198.

Тестові набори вхідних даних – набори значень вхідних даних, які використовуються для визначення правильності роботи проєкту, с. 105.

Типи даних в електронних таблицях – числа, тексти і формули, с. 193.

Тривимірні, або **3D-принтери**, – принтери, що використовуються для створення об'ємних об'єктів складної форми. Розрізняють тривимірні принтери за типом речовини, з якої «будується» об'єкт, та способом її отвердіння, с. 75.

Умови фільтрування – це набір критеріїв у табличному процесорі, що визначають, які саме дані з таблиці будуть відображені, а які – ні, с. 224.

Умовне форматування – вид форматування даних, який автоматично змінює формат клітинки електронної таблиці на заданий, якщо виконується певна умова, с. 233.

Фільтрування – це автоматизований відбір значень у клітинках електронної таблиці, що відповідають певним умовам, с. 223.

Флешпам'ять – один з видів зовнішньої пам'яті, який використовує як носій даних електронні мікросхеми. Розрізняють такі пристрої флешпам'яті: USB-флешнакопичувач («флешка»), флешкарта (карта пам'яті), твердотілий, або SSD-накопичувач, с. 55.

Формат даних – спосіб відображення даних у клітинках електронної таблиці, с. 195.

Формат Дата – формат даних у електронній таблиці, який використовується для подання числових даних у вигляді дати певного типу, с. 197.

Формат Час – формат даних у електронній таблиці, який використовується для подання числових даних у вигляді часу певного типу, с. 198.

Формула в електронній таблиці – це вираз, який задає операції над даними в клітинках електронної таблиці та порядок їх виконання, с. 193.

Функція в табличному процесорі – це попередньо визначена формула, яка виконує певні обчислення над заданими даними (аргументами) і занесена в бібліотеку функцій табличного процесора, с. 206.

Цикл в алгоритмі – фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз, с. 151.

Числовий формат – формат даних у електронній таблиці, який використовується для подання чисел у вигляді десяткового дробу із заданою кількістю десяткових розрядів, с. 196.

Штучний інтелект – область інформатики, яка здійснює розробку інтелектуальних комп'ютерних систем та програм, що імітують роботу людського мозку, с. 253.

Юнікод – таблиця кодів символів, що містить 1 114 112 символів та їх коди, з використанням якої можуть бути закодовані символи усіх алфавітів світу, с. 12.

ASCII – таблиця кодів символів, у якій літерам англійського алфавіту, цифрам, розділовим знакам, символам редагування та форматування тексту поставлено у відповідність числа від 0 до 127, с. 11.

CMYK – колірна модель, у якій використовують чотири базові компоненти: блакитний, пурпурний, жовтий, чорний кольори, с. 14.

EXCEL – табличний процесор пакета прикладних програм **Microsoft Office**, с. 192.

False – один з можливих результатів логічної функції, с. 128.

HSV (HSB, HSL) – колірна модель, у якій використовують три базові компоненти: відтінок, насиченість і значення (яскравість, світлість), с. 14.

HTML – мова, якою описують структуру та контент вебсторінки для відображення цієї сторінки у вікні браузера, с. 172.

RGB – колірна модель, у якій базовими компонентами є три кольори спектра – червоний, зелений і синій, с. 14.

True – один з можливих результатів логічної функції, с. 128.

Smart-технології – технології, що базуються на використанні датчиків, інтернету речей і штучного інтелекту для створення «розумних» систем, які можуть самонавчатися та адаптуватися до змінних умов, с. 245.

XLSX – стандартне розширення імені файлу електронної книги в **Microsoft Office Excel**, с. 192.

ЗМІСТ

Дорогі восьмикласниці та восьмикласники!
Шановні вчительки та вчителі! 3

Розділ 1. Кодування даних

1.1. Кодування та декодування повідомлень 5
1.2. Кодування текстових, графічних і мультимедійних даних . . 10
1.3. Двійкове кодування 19
Практична робота № 1. «Розв'язування задач на визначення довжини двійкового коду повідомлень» 29

Розділ 2. Архіви даних

2.1. Стиснення даних. Архівування даних. Резервне копіювання 30
2.2. Архіватори. Операції над архівами файлів 42
Практична робота № 2. «Архівування та розархівування даних» 50

Розділ 3. Апаратне забезпечення персонального комп'ютера

3.1. Персональний комп'ютер і його основні складові 51
3.2. Пристрої введення даних 60
3.3. Пристрої виведення даних 70
3.4. Історія обчислювальних і комп'ютерних пристроїв 79
3.5. Види сучасних комп'ютерів та їх застосування 91
Практична робота № 3. «Добір складових персонального комп'ютера залежно від його призначення» 103

Розділ 4. Алгоритми та програми

4.1. Етапи розв'язування задач з використанням алгоритмів і проєктів 104
4.2. Кнопка 108
4.3. Напис 114
4.4. Поле. Проєкти з вхідними даними та результатами 119
Практична робота № 4. «Проєкти з полями, написами і кнопками, з уведенням даних і виведенням результатів» 126
4.5. Логічні вирази. Змінні логічного типу. Логічні операції . . . 127
4.6. Логічні вирази в розгалуженнях. Віконні проєкти з розгалуженнями 134
4.7. Прапорці та перемикачі у віконних проєктах 143
Практична робота № 5. «Проєкти з розгалуженнями» 150
4.8. Цикли з лічильником 150
4.9. Цикли з передумовою 156
Практична робота № 6. «Проєкти із циклами» 162

Розділ 5. Створення та публікація вебресурсів

5.1. Етапи створення сайту. Дизайн вебсторінок. Тренди вебдизайну 163
5.2. Поняття про мову **HTML** 171
5.3. Автоматизовані засоби створення та публікації вебсайтів . . 179
Практична робота № 7. «Створення сайту з використанням онлайн-системи конструювання сайтів» 191

Розділ 6. Опрацювання даних в електронних таблицях

6.1. Об'єкти електронної таблиці, їх властивості. Формати даних в електронних таблицях. Адресація в електронних таблицях 192
6.2. Математичні, статистичні та логічні функції 205
Практична робота № 8. «Використання функцій у табличному процесорі» 219
6.3. Упорядкування та фільтрування даних у табличному процесорі 219
6.4. Умовне форматування. Друкування електронної таблиці . . 232
Практична робота № 9. «Вибирання даних в електронних таблицях» 242

Розділ 7. Smart-технології. Штучний інтелект

7.1. Інтернет речей і його використання. **Smart**-технології 243
7.2. Поняття штучного інтелекту. Використання штучного інтелекту в різних галузях 252

Розділ 8. Практикум з використання інформаційних технологій

8.1. Навчальний проєкт. Використання систем штучного інтелекту для виконання проєкту 268

Додатки 276

Словник 293