ВИДАВНИЦТВО
РАНОК

Віктор Руденко, Наталія Речич,
Валентина Потієнко

Інтернет-
підтримка

Інформатика

З ПОГЛИБЛЕНИМ ВИВЧЕННЯМ ІНФОРМАТИКИ

8



Віктор Руденко, Наталія Речич,
Валентина Потієнко

Інформатика

Підручник для 8 класу
з поглибленим вивченням інформатики
закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України

Харків
Видавництво «Ранок»
2021

УДК 004:37.016 (075.3)
Р83

Рекомендовано
Міністерством освіти і науки України
(наказ Міністерства освіти і науки України
від 22.02.2021 № 243)

Видано за рахунок державних коштів.
Продаж заборонено

Руденко В. Д.
Р83 Інформатика : підруч. для 8 кл. з поглибл. вивч. інформатики закл. загал. серед. освіти / В. Д. Руденко, Н. В. Речич, В. О. Потієнко. — Харків : Вид-во «Ранок», 2021. — 256 с. : іл.

ISBN 978-617-09-6960-6

УДК 004:37.016 (075.3)



Інтернет-підтримка

© Руденко В. Д., Речич Н. В.,
Потієнко В. О., 2021
© ТОВ Видавництво «Ранок», 2021

ISBN 978-617-09-6960-6

Шановні учні та учениці!

Ви приступаєте до опанування шкільного курсу інформатики з поглибленим вивченням. Ви досягли певного рівня інформаційної культури і здатні самостійно оволодівати сучасними інформаційними технологіями. Та інформатика — дуже динамічна наука. Її подальші напрямки й темпи розвитку значною мірою визначатимуться рівнем підготовки людей, які мають ґрунтовні знання в цій галузі.

Цього року ви будете працювати з новими програмними засобами, реалізовувати базові алгоритми засобами мови програмування Python і середовища програмування IDLE та створювати й налаштовувати програми за розробленими алгоритмами, опануєте основні етапи та методологію розробки програмного забезпечення тощо.

Бажаємовам успіхів,
авторський колектив

Підручник, який ви тримаєте в руках, — ваш надійний помічник. У ньому ви знайдете завдання для самостійного виконання — виконуйте їх на комп'ютері з натхненням, повторюйте теоретичний матеріал і викладайте основні положення на папері. Ви маєте змогу пройти комп'ютерне тестування з автоматичною перевіркою результату.

Питання для перевірки знань і завдання для самостійного виконання відповідають таким рівням навчальних досягнень:

— початковий і середній

— достатній

— високий

У тексті також використано позначення:


— питання на повторення


— означення, висновок


— зверніть увагу


— цікаво знати


— завдання для виконання й обговорення в парах або групах


— вправи для домашнього виконання

Розділ 1
МАТЕМАТИЧНІ ОСНОВИ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

1.1. Поняття системи числення. Позиційні і непозиційні системи числення

З якими системами числення ви вже працювали? Чому, на вашу думку, десяткова система є домінуючою в побуті?

Числа виникли в глибоку давнину як засіб для лічби. Зі зростанням кількості предметів їх уже стало недостатньо, і для записування великих чисел потрібні були інші способи.

Умовні знаки (символи), використовувані для позначення чисел, називаються цифрами.

Система числення — це сукупність правил записування чисел за допомогою символів (цифрових знаків) і виконання операцій над ними.

Алфавіт системи числення — сукупність цифр для запису числа. Ми записуємо числа за допомогою алфавіту з 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. І розуміємо, що одна й та сама цифра на різних місцях у числі має різні значення (приклад 1).

Розрізняють позиційні та непозиційні системи числення.

Непозиційною системою числення називають систему числення, у якій кількісний еквівалент кожної цифри визначається тільки цифрою і не залежить від її позиції у записі числа. Саме число визначається як сума цифр, з яких воно складається.

Непозиційні системи числення в історії людства з'явилися першими. У різних куточках світу розвивалися власні системи. Наприклад, єгипетська, римська, китайська, грецька (рис. 1.1).

Приклад 1.
Кількісне значення цифри 7 у числі 75 дорівнює 70, у числі 39,75 — тільки 0,7. У числі 333 трійки означають 3 сотні — триста, 3 десятки — тридцять і просто 3.

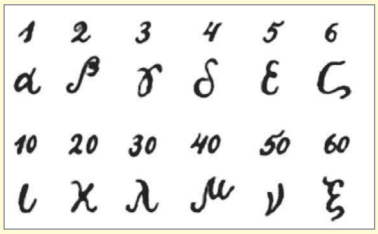


Рис. 1.1. Грецька система числення

Єгипетська система числення виникла в III тисячолітті до н. е. в Стародавньому Єгипті. Для позначення чисел у ній використовувалися спеціальні знаки — ієрогліфи. Ієрогліфами позначалися основні числа, а в записі інших чисел ієрогліфи просто повторювалися потрібну кількість разів. На рис. 1.2 відображено запис числа 345.



Рис. 1.2. Подання числа 345 у єгипетській системі числення

Римська система числення виникла понад 2,5 тисячі років тому в Стародавньому Римі. Числа в цій системі записуються за допомогою літер:

I	V	X	L	C	M
1	5	10	50	100	1000

Так, у римській системі числення рік 2021 буде записано як MMXXI, а MMM означає 3 тисячі.





Рис. 1.3. Приклади використання римських чисел

З непозиційних систем числення в наші дні збереглася лише римська система. Проте і вона майже не застосовується, окрім позначення століть, розділів, томів видань тощо (рис. 1.3).

Загалом, непозиційні системи числення досить складні для записування чисел і виконання арифметичних дій. Тому використовують позиційні системи числення. У час комп'ютерної техніки набули широкого розповсюдження десяткова, двійкова, вісімкова, шістнадцяткова та інші системи.

Позиційною системою числення називають систему числення, у якій кількісний еквівалент цифри в числі залежить не тільки від її значення, але й від позиції в записі числа.

У позиційних системах числення позицію цифри в числі називають розрядом. Кожний розряд має свій номер.

Для *цілої частини числа* використовується така нумерація розрядів: молодший розряд цілого числа має номер нуль, а кожний наступний збільшується на одиницю.

У *дробовій частині числа* старший розряд має номер –1, а кожний наступний номер зменшується на одиницю. Отже, якщо ціла частина числа має n розрядів, а дробова — m розрядів, то старший розряд цілої частини має номер $n–1$, а молодший розряд дробової частини — номер $–m$.

У позиційних системах кожний розряд має свою вагу. Вона і визначає кількісний еквівалент цифри в цьому розряді. З урахуванням номера розряду та основи системи числення визначається кількісний еквівалент цифри (приклад 2).

Деякі правила римської системи числення

- Кілька однакових цифр, записані поспіль, додаються: III (три), XXX (тридцять)
- Молодша цифра, записана праворуч від старшої, додається: XIII (тринадцять)
- Молодша цифра, записана ліворуч від старшої, віднімається: IX (дев'ять), XL (сорок)

Приклад 2. У таблиці наведено кількісні еквіваленти цифр числа на прикладі з десятковим числом 326,75.

Цифра	3	2	6	7	5
Номер розряду	2	1	0	–1	–2
Вага розряду	$10^2=100$	$10^1=10$	$10^0=1$	$10^{-1}=0,1$	$10^{-2}=0,01$
Кількісний еквівалент цифри (кількість одиниць, яку позначає цифра)	$3\cdot100=300$	$2\cdot10=20$	$6\cdot1=6$	$7\cdot0,1=0,7$	$5\cdot0,01=0,05$

Появу сучасної десяткової позиційної системи числення пов'язують із працями індійських учених. Правила виконання арифметичних дій для неї описав великий перський математик Аль-Хорезмі. Немає точної дати появи індійської системи числення, часом появи її опису вважають 850 рік.

Цифри 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, які ми називаємо арабськими, були винайдені в Стародавній Індії. Саме звідти через арабські країни десяткова позиційна система числення прийшла в Європу.

Вага цифри — внесок, який вона вносить у значення числа.

Існують системи числення, основа яких більша за 10. У таких системах для запису цифр використовують літери, наприклад, *A* означає 10, B — 11, C — 12 тощо.

До основних характеристик позиційних систем числення належать основа і базис.

Основою позиційної системи числення є кількість цифр, які можуть використовуватися в кожному розряді числа.

Базисом позиційної системи числення є послідовність чисел, які визначають вагу розряду і можуть використовуватися в кожному розряді числа.

Вага розрядів у позиційних системах дорівнює основі системи числення у степені, що дорівнює номеру розряду. Для визначення кількісного значення цифри в позиційній системі числення необхідно помножити цю цифру на вагу розряду, так визначається її внесок у загальне кількісне значення числа (приклад 3).

Приклад 3.

У десятковому числі 947 четвірка стоїть у розряді з вагою 10, отже, вона має кількісне значення 40, а в числі 0,6483 четвірка стоїть у розряді з вагою $10^{-2} = 0{,}01$, тому її внесок у число 0,04. Розглянемо, як у десятковому числі 326 визначаються кількісні еквіваленти кожної цифри, які в сумі дають саме число (рис. 1.4).

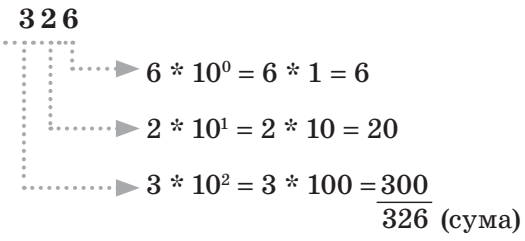


Рис. 1.4. Підрахунок внеску цифр у значення числа

Назву системам числення надають за значенням основи. Звична нам система з десятьма числами називається *десятковою*. Її базис складають числа — степені числа 10, як подано в таблиці (приклад 4). А *двійкова система числення* має основою 2 — дві цифри: 0 і 1. Базисом цієї системи є степені числа 2: 2^{n-1}, 2^{n-2}, ... 2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, ...

Приклад 4.

Базис десяткової системи числення:

..., 10^n, ..., 1000, 100, 10, 1, 0.1, 0.01, ...

Базис двійкової системи числення:

..., 2^n, ..., 16, 8, 4, 2, 1, 1/2, 1/4, ...

Зазвичай систему числення, у якій записане число, позначають нижнім індексом, наприклад, 326_{10}, 11001_{2}. Цей запис називають коротким (*згорнута форма*). *Розгорнутою формою* запису числа в позиційній системі числення є сума добутків цифр числа на вагу розряду, у якому стоїть цифра (приклад 5).

Приклад 5.

Дано число: а) 326_{10}; б) 11001_{2}. Запишемо числа в розгорнутій формі.

а) $326_{10} = 6 \cdot 10_{10}^{0} + 2 \cdot 10_{10}^{1} + 3 \cdot 10_{10}^{3}$;

б) $11001_{2} = 1 \cdot 2_{10}^{0} + 0 \cdot 2_{10}^{1} + 0 \cdot 2_{10}^{2} + 1 \cdot 2_{10}^{3} + 1 \cdot 2_{10}^{4}$.

За записом числа 11001_{2} можемо зробити висновок: значення степенів двійки 1, 8 і 16 у цьому числі є, а значення степенів 2 і 4 відсутні.

Прикладом нетрадиційних систем є *факторіальна система*, у якій вага розряду визначається послідовністю значень факторіалів чисел: $1! = 1$; $2! = 2 \cdot 1 = 2$; $3! = 3 \cdot 2 \cdot 1 = 6$; $4! = 24$,
або *Фібоначчева система*, числа якої і є вагою розрядів: 1, 1, 2, 3, 5, 8, 13, … .

Базисом систем є послідовність степенів основи системи — геометрична прогресія зі знаменником, який дорівнює основі. Такі позиційні системи називаються *традиційними*. У нетрадиційних базис складається з чисел, які її не утворюють.

Запитання для перевірки знань

1. Що називають системою числення?
2. Які системи називають позиційними; непозиційними?
3. Запишіть у римській системі числення числа 37, 19 і 43.
4. Що таке основа системи числення; базис?
5. Як визначається вага розряду в традиційній позиційній системі числення?
6. Скільки цифр має система числення, основа якої дорівнює p. Перелічіть їх, якщо $p = 6$.

Завдання для самостійного виконання

1. Запишіть числа 5346 і 11011.1112 у розгорнутій формі.
2. Запишіть у римській системі числення свій рік народження та поточний рік.

1.2. Переведення чисел із десяткової системи числення в довільну і навпаки

Як впливає розміщення цифри в числі позиційної системи числення? Які форми подання мають числа?

Ми опрацьовуватимемо числа з традиційних систем і щоразу не повторюватимемо, якій саме системі належить число.

Щоб визначити внесок кожної цифри в число, слід знайти добуток цифри і ваги розряду, у якому вона стоїть. Для здійснення аналізу внеску число зручно записувати в *розгорнутій формі*. Це стосується всіх чисел, записаних у системах з будь-якою основою (приклад 1).

Приклад 1.

Дано число: а) 1424; б) 2AB. Визначимо найменше значення основи системи числення, у якій записане число, та внесок кожної цифри у значення числа.

Алгоритм виконання: а) Основа першого числа 5. Чому? Найбільша цифра в числі 4, отже, можна використати цифри: 0, 1, 2, 3, 4. Усього 5 цифр. Зазначимо, що в цій системі числення $10_5 = 5_{10}$.

б) Аналогічно основа другого числа 12, бо найбільша цифра B=1. І $10_{12}=12_{10}$.

Запишемо числа в розгорнутій формі:

а) $1424_5 = 4\cdot 10_5{}^0 + 2\cdot 10_5{}^1 + 4\cdot 10_5{}^2 + 1\cdot 10_5{}^3 = 4\cdot 5_{10}{}^0 + 2\cdot 5_{10}{}^1 + 4\cdot 5_{10}{}^2 + 1\cdot 5_{10}{}^3 =$

(*перейдемо до десяткової системи числення*)

$= (4\cdot 1 + 2\cdot 5 + 4\cdot 25 + 1\cdot 125)_{10} = (4 + 10 + 100 + 125)_{10} = 239_{10}$;

б) $2AB_{12} = 11\cdot 12_{10}{}^0 + 10\cdot 12_{10}{}^1 + 2\cdot 12_{10}{}^2 = (11\cdot 1 + 10\cdot 12 + 2\cdot 144)_{10} = (11 + 120 + 288)_{10} = 319_{10}$.

У наведеному прикладі у записі числа в розгорнутій формі використовували вагу розряду у звичній для нас десятковій системі числення і розрахунки внеску цифр теж вели в ній, тому одразу отримували значення числа в десятковій системі.

Для переведення числа з будь-якої позиційної системи числення в десяткову необхідно обчислити суму добутків цифр числа на вагу розряду, у якому розміщена цифра.

Розглянемо приклад 2 — переведення числа з урахуванням дробової частини.

Приклад 2.

Дано число: а) $326{,}25_{10}$; б) 11001.111_2.

Переведемо число в десяткову систему числення з урахуванням дробової частини.

а) $326{,}25_{10} = 3\cdot 10^3 + 2\cdot 10^1 + 6\cdot 10^0 + 2\cdot 10^{-1} + 5\cdot 10^{-2} =$

$= 3\cdot 100 + 2\cdot 10 + 6\cdot 1 + \frac{2}{10} + \frac{5}{100} = 326{,}25_{10}$;

б) $11001.111_2 = 1\cdot 2^0 + 0\cdot 2^1 + 0\cdot 2^2 + 1\cdot 2^3 + 1\cdot 2^4 + 1\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3} =$

$= 1\cdot 1 + 0\cdot 2 + 0\cdot 4 + 1\cdot 8 + 1\cdot 16 + 0{,}5 + 0{,}25 + 0{,}125 = 25{,}875_{10}$.

У комп'ютерах основною є двійкова позиційна система числення із символами 0 і 1. Використовують також системи числення з основою 16 (зазвичай у цій системі записують нумерацію комірок пам'яті; коди кольорів).

Загалом, можна домовитись і позначати цифри, значення яких більше 10, будь-якими літерами. Але в основах інформатики зазвичай дотримуються таких правил математики: для запису чисел у системі з основою $p > 10$ використовують десяткові цифри та літери латинського алфавіту.

Таких цифр може бути 36, тому для запису числа з основою $p > 36$ загальноприйнятих правил не існує.

Розглянемо приклади переведення чисел (приклади 3 і 4).

Приклад 3.

Переведемо число $92C8_{16}$ із шістнадцяткової системи числення в десяткову:

$$92C8_{16} = 8\cdot 10_{16}{}^{0} + C\cdot 10_{16}{}^{1} + 2\cdot 10_{16}{}^{2} + 9\cdot 10_{16}{}^{3} =$$

$$= 8\cdot 16_{10}{}^{0} + C\cdot 16_{10}{}^{1} + 2\cdot 16_{10}{}^{2} + 9\cdot 16_{10}{}^{3} = 37576_{10}.$$

У тих системах числення, в яких основа p більша за 10, для позначення цифр використовують десять десяткових цифр і букви латинського алфавіту: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C і т. д.

Приклад 4.

Переведемо число 435_8 із вісімкової системи числення в десяткову:

$$435_8 = 5\cdot 10^{0}{}_{8} + 3\cdot 10^{1}{}_{8} + 7\cdot 10^{2}{}_{8} = 5\cdot 8^{0}{}_{10} + 3\cdot 8^{1}{}_{10} + 7\cdot 8^{2}{}_{10} = 477_{10}\cdot$$

Для переведення цілої частини чисел з p-кової системи числення в десяткову потрібно поділити цілу частину числа націло на p — основу тієї системи числення, у яку переводять число та окремо записати остачу від ділення.

Далі слід отриману частку цілочислового ділення послідовно поділити на p — повторюйте кроки 1 і 2, поки не отримаємо частку, що дорівнює нулю. Увесь час варто фіксувати остачу в порядку їх обчислення — якщо вони відсутні, то записуйте 0 обов'язково. Потім потрібно записати остачі в порядку, оберненому до порядку їх отримання (від останнього до першого), — вони і є цілою частиною числа в системі з основою p.

Будь-яке число можна подати у двійковій системі числення, іншими словами: будь-яке число можна записати як суму степенів числа 2.

Для переведення числа з десяткової системи числення в іншу позиційну діють правила для цілої і дробової частини числа.

Для переведення дробової частини числа потрібно помножити дробову частину числа на p — основу тієї системи, у яку переводять число та окремо записати отриману дробову частину добутку та цілу.

Далі слід послідовно помножити отриману дробову частину добутку на p — повторюйте кроки 1 і 2; у дробовій частині бажано отримати нуль. Увесь час варто фіксувати цілі значення добутку в порядку їх отримання — якщо такі відсутні, запишіть 0. Потім потрібно записати результат. Пряма послідовність цілих значень (записаних у порядку їх отримання) і є дробовою частиною числа в системі з основою p.

При переведенні в систему числення з основою $p > 10$ у записі числа необхідно записати числа, отримані як ціла частина добутку, а числа, більші за 10 у десятковій системі, цифрами нової системи числення.

Перевести числа з однієї системи числення в іншу можна за допомогою **онлайн-калькулятора**.

Для цього потрібно:

- у формі калькулятора ввести число
- зазначити, у якій системі числення число подано
- вибрати систему числення, в яку потрібно перевести число
- натиснути **Порахувати**

Розглянемо приклади переведення чисел (приклади 5 і 6). Як бачимо, ділити можна усно й одразу записувати результати, або письмово (у стовпчик).

Приклад 5.

Переведемо число $25{,}875_{10}$ у двійкову систему числення.

Здійснюємо цілочислове ділення 25 на 2		Здійснюємо множення 0,875 на 2	
остача	частка	ціла частина	дробова частина
1	12	1	750
0	6	1	500
0	3	1	000
1	1		
1	0		
$25_{10} = 11001_2$		$0{,}875_{10} = 0.111_2$	

Приклад 6.

Переведемо число 37576_{10} з десяткової системи числення у шістнадцяткову (рис. 1.5).

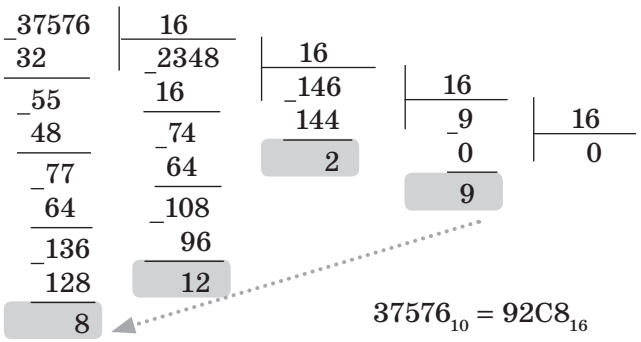


Рис. 1.5. Переведення числа з десяткової системи числення у шістнадцяткову

Запитання для перевірки знань

1. Як записують цифри в системах, основа яких $p > 10$?
2. Як отримати значення числа будь-якої позиційної системи в десятковій системі?
3. Опишіть алгоритм переведення цілих чисел з однієї системи числення в іншу.
4. Опишіть алгоритм переведення дробових чисел з однієї системи числення в іншу.
5. Як вплине на значення числа в системі числення з основою $p > 10$ помилковий запис у розряді двоцифрового значення замість символу латинського алфавіту? Відповідь обґрунтуйте. Наведіть приклади.
6. Як вплине на значення числа в системі числення з основою p додавання 0 в кінець цілої частини? Відповідь обґрунтуйте. Наведіть приклади.

Завдання для самостійного виконання

1. Запишіть число 488 (десяткова система числення) у двійковій, вісімковій і шістнадцятковій системах числення. Переведіть у десяткову систему числення число A(2) = 1011011,101.
2. Переведіть число A(10) = 49,375 у двійкову систему числення.
3. Знайдіть десяткове значення числа: а) 8.2516; б) 1356.
4. Переведіть число A(8) = 47,125 у десяткову систему числення, а з десяткової — у двійкову.

1.3. Двійкова та шістнадцяткова системи числення. Опрацювання систем числення, основою яких є степінь числа 2

У десятковій системі числення відбувається перенесення з нульового в перший розряд, якщо є перебільшення значення 9. При перебільшенні якого значення відбувається перенесення в наступний розряд у вісімковій системі; у шістнадцятковій?



В обчислювальній техніці для опрацювання даних використовують двійкову, вісімкову і шістнадцяткову системи числення. Розглянемо порівняльну таблицю запису десяткових чисел у цих системах, наведену праворуч. Із таблиці видно, що запис однакових десяткових чисел у різних системах числення містить різну кількість розрядів. Оскільки $16 = 2^4$, а $8 = 2^3$, то для запису одного шістнадцяткового розряду необхідно чотири двійкові розряди, а для одного вісімкового розряду — три двійкові.

Для четвіркової системи числення необхідно два розряди двійкової, бо $4 = 2^2$. Степінь при основі 2 визначає кількість двійкових розрядів, необхідних для запису однієї цифри.

Якщо основа однієї системи числення є цілим степенем основи іншої системи числення, то правила переведення запису чисел між такими системами спрощуються. Для переведення числа із системи з основою, значення якої є степенем числа 2, у двійкову необхідно записати значення кожної цифри у двійковій системі числення — отримаємо групи символів.

Застосуємо наведене правило на прикладах 1–3.

З прикладу 1 видно, що кожну цифру вісімкової системи числення замінили групами з трьох розрядів (щоб отримати 8, двійку необхідно піднести до степеня 3) і записали значення цифр у двійковій системі.

Зверніть увагу на другу цифру: $3_8 = 11_2$, але в групі має бути три розряди, тому 3_8 записали як 011_2. Наприклад, є дробове число $0{,}2_{16}$. Цифру 2 необхідно записати в групі з чотирма розрядами як 0010, і тоді $0{,}2_{16} = 0{,}0010_2$ або $0{,}001_2$.

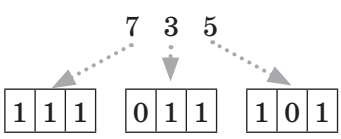
Порівняльна таблиця запису десяткових чисел у різних системах числення

10	16	8	2
0	0	0	0
1	1	1	1
2	2	2	10
3	3	3	11
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
10	A	12	1010
11	B	13	1011
12	C	14	1100
13	D	15	1101
14	E	16	1110
15	F	17	1111

Для переведення числа з двійкової системи числення у число в системі з основою, що є степенем двійки ($p = 2^n$), необхідно в цілій частині цього числа справа наліво, а у дробовій — зліва направо створити групи розрядів (кількість символів у кожній групі дорівнюватиме степеню n) і записати значення кожної групи в новій системі числення (приклад 2).

Під час переведення числа з двійкової системи слід уважно визначати групи. З прикладу 3 видно, що в крайній правій

Приклад 1.
Порозрядно переведемо число 735_8 з вісімкової системи числення у двійкову.



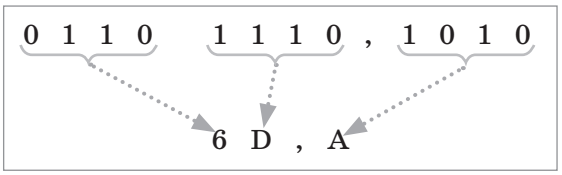
$735_8 = 111011101_2$

групі цілої частини необхідно додавати 0 *зліва*, а в крайній лівій групі дробової частини 0 необхідно додавати *справа*. За цим правилом зручно перетворювати числа в системах з основами, що є степенями числа 2, використовуючи двійкову систему для отримання проміжного результату.

Приклад 2.
Число $7CA,5E_{16}$ у двійковій системі має такий запис: $11111001010,01011110_2$.

Якщо запис чисел позначити літерою A, то отримаємо такі ланцюжки: $A_{16} \to A_2 \to A_8$ або $A_8 \to A_2 \to A_{16}$.

Приклад 3.
У крайніх групах для відповідності необхідній кількості розрядів слід дописати нулі: у цілій частині зліва, у дробовій частині справа (див. рис. 1.7).



$1101110,1012 = 6D,A$

Рис. 1.7. Порозрядне переведення числа з двійкової системи числення у вісімкову

Алгоритм виконання дій для ланцюжка $A_{16} \to A_2 \to A_8$

Крок 1	Для кожної цифри шістнадцяткового подання числа запишіть її двійковий еквівалент, створюючи групи по 4 розряди
Крок 2	Отримане двійкове число розподіліть на групи з трьох розрядів. Не забувайте про крайні 0 за потреби
Крок 3	Кожну отриману групу двійкового числа замініть цифрою у вісімковій системі. Отримайте вісімкове число

Запитання для перевірки знань

1. Поясніть правила переведення чисел системи з основою 2^n у двійкове число.
2. Поясніть правила переведення двійкових чисел у числа системи з основою 2^n.
3. Переведіть число $1110101,011_2$ у вісімкову систему числення. Запишіть кількість розрядів.
4. Переведіть число $1110101,011_2$ у шістнадцяткову систему числення. Запишіть кількість розрядів.
5. Поясніть різницю в кількості розрядів у числах — результатах виконання п. 3 і 4.
6. Переведіть число — результат виконання п. 4 у четвіркову систему числення.

Завдання для самостійного виконання

1. Переведіть число A(16) = D1,6 у двійкову систему, а з двійкової — у вісімкову. Обчисліть числа в шістнадцятковій і вісімковій системах в десятковій системі — результати мають збігатися.
2. Переведіть число A(10) = 140,75 у шістнадцяткову систему числення, а в шістнадцятковій — у двійкову, а далі із двійкової — знов у десяткову.

1.4. Арифметичні операції у двійковій і шістнадцятковій системах числення

Пригадайте правила додавання і віднімання чисел у стовпчик у десятковій системі числення.

Ви знаєте, що основою системи числення є розмірність алфавіту цієї системи. Значення, на одиницю більше розмірності алфавіту в будь-якій системі, вже буде записано як 10_p, де p — основа. Із цього випливає *нове тлумачення поняття* основи: це те мінімальне значення в одному розряді, при перевищенні якого відбувається перенесення в наступний розряд.

Дійсно, у двійковій системі числення для запису чисел використовуються символи 0 і 1. Якщо внаслідок виконання будь-якої операції буде отримано значення 2, то відбувається перенесення одиниці в перший наступний розряд і $2_{10} = 10_2$.

Правила арифметичних дій із числами в усіх системах числення однакові. Додавання цифр здійснюється послідовно, розряд за розрядом, починаючи з молодшого, з урахуванням одиниць перенесення з меншого до більшого.

Приклад 1.

Збільшимо $3_{10} = 11_2$ на 1: додамо порозрядно — у нульовому розряді отримаємо $1+1=2_{10}$. Але в алфавіті двійкової системи числення відсутня 2, її значення 10_2 — відбулось перенесення в інший (перший) розряд, як у звичній нам десятковій системі при додаванні: $9+1$. Надалі вже в першому розряді до наявної 1 додамо перенесену 1.

Знову отримаємо $2_{10} = 10_2$ і загальний результат, 4_{10} уже буде записано як 100_2.

Так відбувається з усіма позиційними системами числення. Наприклад, додамо у вісімковій системі $6+3$. У десятковій системі це буде 9_{10}, а у вісімковій — 8 одиниць із 9 перенесеться в наступний розряд як 10_8, а в попередньому залишиться 1: $6_8 + 3_8 = 11_8$.

Повторимо виконання арифметичних дії у десятковій системі числення. На рис. 1.8 одноразово показано перенесення у старший розряд у десятковій системі.

Розглянемо інші приклади додавання (приклади 2 і 3).

Приклад 2.

Виконаємо додавання у двійковій системі числення.

Пригадаємо: $0_2 + 1_2 = 1_2$; $1_2 + 0_2 = 1_2$; $0_2 + 0_2 = 0_2$; $1_2 + 1_2 = 10_2$.

Маємо:
$$\begin{array}{r} 1011{,}11 \\ +\ 101{,}101 \\ \hline 10001{,}011 \end{array}.$$

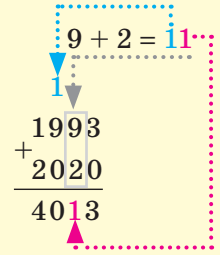


Рис. 1.8. Додавання чисел у десятковій системі числення

Результат додавання цифр a і b в одному розряді в системі числення з основою q визначається за таким правилом:

$$a+b = \begin{cases} a+b, \text{ якщо } a+b<q, \\ a+b-q, \text{ якщо } a+b \geqslant q, \end{cases}$$

де a і b — цифри, що додаються; q — основа системи числення. Якщо сума двох цифр більша або дорівнює системі числення, з'являється одиниця перенесення у найближчий старший розряд.

Приклад 3.

Виконаємо додавання шістнадцяткових чисел $7AB,5_{16}$ і $C1D,F_{16}$. Для зручності в операціях із цифрами будемо використовувати звичну нам десяткову систему, адже знаємо еквівалент шістнадцяткової цифри в десятковій системі.

Розряд	Результат додавання в розряді	Залишається в розряді	Переноситься в старший розряд
–1	$5_{16}+F_{16}=5_{10}+15_{10}=20_{10}=14_{16}$	4	1
0	$1+B_{16}+D_{16}=1+11_{10}+13_{10}=25_{10}=19_{16}$	9	1
1	$1+A_{16}+1_{16}=1+10_{10}+1_{10}=12_{10}=C_{16}$	C	—
2	$7_{16}+C_{16}=7_{10}+12_{10}=19_{10}=13_{16}$	3	1
3	1	1	

Таким чином, $7AB,5_{16}+C1D,F_{16}=13C9,4_{16}$ або $\begin{array}{r} 7AB,5 \\ +\ C1D,F \\ \hline 13C9,4 \end{array}$.

В операції додавання відбувається перенесення одиниці до старшого розряду, а при відніманні, навпаки, зі старшого розряду береться одиниця в молодший, якщо в ньому цифра зменшуваного менша за цифру у від'ємнику.

Перед здійсненням операції доцільно проаналізувати значення чисел, потім відняти від більшого числа менше і визначити знак результату залежно від значень зменшуваного і від'ємника. Якщо зменшуване більше за від'ємник, отримують додатний результат; якщо менше за від'ємник — від'ємний.

Результат віднімання двох цифр a і b в одному розряді в системі числення з основою q визначається за таким правилом:

$$a-b=\begin{cases} a-b, \text{ якщо } a\geqslant b, \\ q+a-b, \text{ якщо } a<b, \end{cases}$$

де a — зменшуване і b від'ємник; q — основа системи числення. Якщо цифра зменшуваного менша за цифру від'ємника, запозичується одиниця з найближчого старшого розряду.

У комп'ютері операція віднімання здійснюється інакше: через обчислення додаткового коду двійкового подання числа і подальше додавання цього коду.

Для унаочнення операції віднімання розглянемо приклад 4.

Аналогічно додаванню перенесення між розрядами здійснюється і в операціях множення. Множити двійкові числа просто, адже в результаті отримуємо або 0, або 1, і ніяких перенесень в інший розряд не відбувається:

$$0_2\cdot 1_2=0_2;\ 1_2\cdot 0_2=0_2;\ 0_2\cdot 0_2=0_2;\ 1_2\cdot 1_2=1_2.$$

Приклад 4.

Виконаємо віднімання $EA8,D_{16}-7BC,A_{16}$. Скористаємося таблицею.

Розряд	Результат віднімання в розряді	Чи є запозичення в старшому розряді	Береться зі старшого розряду
–1	$D_{16}-A_{16}=13_{10}-10_{10}=3_{10}=3_{16}$	—	—
0	$8_{16}-C_{16}\Rightarrow 10_{16}+8_{16}-C_{16}=18_{16}-C_{16}=$ $=24_{10}-12_{10}=12_{10}=C_{16}$	є	1
1	$A_{16}-1-B_{16}\Rightarrow 10_{16}+A_{16}-1-B_{16}=1A_{16}-1-B_{16}=$ $=26_{10}-1-11_{10}=14_{10}=E_{16}$	є	1
2	$E_{16}-1-7_{16}=14_{10}-1-7_{10}=6_{10}=6_{16}$	—	—

Отримаємо результат: $EA8,D_{16}-7BC,A_{16}=6EC,3_{16}$.

У прикладі 5 множення здійснювали у стовпчик із зсувом кодового наступного добутку вліво. У комп'ютерах множення виконується, починаючи зі старшого розряду множника, при цьому часткові добутки зсуваються праворуч (приклад 6).

З іншими системами числення виникають труднощі в обчисленнях: якщо при додаванні в старший розряд переноситься 1, то в результаті множення це число значно збільшується і варто мати таблицю добутків цифр системи числення.

В обох наведених таблицях ліва верхня клітинка містить значення основи:

16	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Приклад 5. Виконаємо множення $110,1_2\cdot 11,1_2$.

$$\begin{array}{r} \times 110,1 \\ 11,1 \\ \hline 1101 \\ + \quad 1101 \\ \hline 1101 \\ \hline 10110,11 \end{array}$$

Приклад 6. Ті самі числа можна помножити і так:

$$\begin{array}{r} \times 110,1 \\ 11,1 \\ \hline 1101 \\ + \quad 1101 \\ \hline 1101 \\ \hline 10110,11 \end{array}$$

2	0	1
0	0	0
1	0	1

Приклад 7. Виконаємо множення $2F9_{16} \cdot A_{16}$. Скористаємося таблицею.

Розряд	Результат множення в розряді	Залишається в розряді	Переноситься у старший розряд
1	$A_{16} \cdot 9_{16} = 5A_{16}$	A	5
2	$5_{16} + A_{16} \cdot F_{16} = 5_{16} + 96_{16} = 9B_{16}$	B	9
3	$9_{16} + A_{16} \cdot 2_{16} = 9_{16} + 14_{16} = 1D_{16}$	D	1
4	1	1	

Отримали результат: $2F9_{16} \cdot A_{16} = 1DBA_{16}$.

У програмуванні вказують на те, скільки пам'яті необхідно виділити під значення змінних. Тож неправильне прогнозування може призвести до перенесення в розряд, якого в поданні числа вже не існує.

Множення багаторозрядних чисел здійснюється аналогічно прикладу 5 за правилами множення у стовпчик.

Перевірка результату здійснюється з використанням десяткової системи. Для цього кожний множник і добуток переводять у десяткову систему числення; здійснюють множення десяткових чисел, отримують добуток; порівнюють добутки після переведення і після множення. Якщо добутки збігаються, то результат множення правильний.

Таким чином, із розглянутих прикладів ми бачимо, що може відбуватися перенесення значення в старший розряд.

Запитання для перевірки знань

1. Наведіть правило додавання двох цифр у позиційній системі числення.
2. Як виконується множення двійкових символів?
3. Як зручно перевірити правильність операцій у недесяткових системах числення?
4. Який результат суми та добутку двійкових чисел 110111,011 і 1101,001?
5. Який результат суми та добутку шістнадцяткових чисел F7A,A і 9D,E?
6. Який результат віднімання шістнадцяткових чисел CB,E і DAF5,8?

Завдання для самостійного виконання

1. Виконайте арифметичні дії у двійковій системі числення.

   а) $10010_2 + 1010_2$; в) $11011_2 \cdot 1101_2$;

   б) $11101_2 + 11011_2$; г) $11010101_2 - 1110_2$.

2. Знайдіть значення суми чисел $A_{(2)} = 1110101{,}011$ і $B_{(2)} = 1101{,}01$. Для перевірки обчисліть числа A, B і їхню суму в десятковій системі — сума A_{10} і B_{10} має збігатися з результатом переведеної суми.
3. Знайдіть двійкове значення суми чисел $A_{(10)} = 37{,}375$ і $B_{(8)} = 23{,}3$.
4. Знайдіть добуток чисел:

   а) $A_{(16)} = C7$ і $B_{(16)} = 23AF$;

   б) $A_{(8)} = 5347$ і $B_{(2)} = 11101$.

Практична робота № 1



ТЕМА. Опрацювання чисел у різних системах числення

ЗАВДАННЯ: виконати арифметичні дії в різних системах числення і переведення чисел з однієї системи числення в іншу.

ОБЛАДНАННЯ: комп'ютер з ОС Windows 7 і старше, під'єднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Запишіть у двійковій і шістнадцятковій системах числення число $A_{(10)} = 235{,}125$.
2. Числа $A_{(10)} = 19{,}5$ і $B_{(10)} = 23{,}125$ переведіть у двійкову систему числення. Додайте числа у двійковій системі числення.
3. Помножте число $A_{(2)} = 1101{,}01$ на число $B_{(2)} = 1011{,}1$. Отриманий результат переведіть у десяткову систему числення.
4. Переведіть число $A_{(10)} = 96{,}75$ у шістнадцяткову систему числення.
5. Переведіть число $A_{(16)} = 5\text{A,D}$ у десяткову систему числення.
6. Складіть шістнадцяткові числа DBC7,A і EF9,5.
7. Від шістнадцяткового числа 7BA9,4 відніміть шістнадцяткове число FEC,2.
8. Здійсніть множення шістнадцяткових чисел 7BA9,4 і FEC,2.
9. Переведіть числа $A_{(16)} = 5\text{B}$ і $B_{(16)} = \text{FA5}$ у двійкову і десяткову системи числення.
10. Переведіть числа $A_{(8)} = 76$ і $B_{(8)} = 123$ у двійкову, десяткову і шістнадцяткову системи числення.
11. Переведіть числа $A_{(2)} = 0{,}1011$ і $B_{(2)} = 0{,}11$ у десяткову систему числення.
12. Знайдіть в інтернеті онлайн-калькулятори для переведення чисел з однієї системи числення в іншу та для виконання арифметичних операцій над числами в різних системах числення.

Зробіть висновок про ефективність методів перетворення чисел з однієї системи числення в іншу.

1–3. Доведіть правильність отриманого результату.

Скористайтеся **онлайн-калькулятором** для переведення чисел з однієї системи числення в іншу, а також для перевірки результатів, які ви отримали, виконуючи наведені вище завдання практичної роботи.

Тест 1

Виконайте тестове завдання з автоматичною перевіркою результату

Розділ 2
КОДУВАННЯ ДАНИХ

2.1. Кодування та декодування повідомлень

Пригадайте означення таких понять, як дані, повідомлення, інформація, інформаційний процес.

Ви вже познайомилися з різними системами числення, зокрема з двійковою. Це дуже важливо, адже люди нині живуть в інформаційному суспільстві, і дані, що опрацьовуються та зберігаються комп'ютерами, подані у вигляді двійкового коду.

Пригадаємо, що є різні типи даних: числа, текст, звук, графічне зображення, відео. У процесі передавання даних бере участь передавач і приймач. Ними може бути як жива істота — людина, тварина, так і технічний прилад (приклад 1).

Спілкування людей за допомогою звуку або жестів є прикладом *передавання* та *приймання* даних. У ході передавання дані між людиною та пристроями перетворюються в сигнали, які пристрої можуть опрацьовувати.

Усі типи даних подаються в однаковому вигляді. Потім над даними здійснюються операції — відбувається такий інформаційний процес, як *опрацювання* даних. Під час передавання даних у комп'ютер за допомогою мікрофона, вебкамери чи шляхом уведення тексту з клавіатури відбувається перетворення цих даних. Усі типи даних перекодовуються пристроями комп'ютера в однотипні сигнали для подальшої роботи з ними.

Приклад 1.
Люди спілкуються між собою безпосередньо або, скажімо, за допомогою телефону. В цьому випадку передавач — особа 1, а приймач — особа 2. Зв'язок між ними здійснюється через пристрої: пристрій 1 є приймачем від особи 1 і передавачем на пристрій 2, а пристрій 2 є приймачем від пристрою 1 та передавачем особі 2.



Рис. 2.1. Схема обміну даними

Код — система правил для перетворення форми даних.
Кодування — процес замінення однієї форми даних на іншу, зручну для передавання, опрацювання та зберігання за допомогою деякого коду.
Декодування — процес відновлення змісту закодованої інформації.

Процеси кодування та декодування можуть повторюватися багато разів. У прикладі 1 обмін даними можна відтворити таким ланцюжком (рис. 2.1). У пристрої 1 дані, отримані від особи 1, кодуються, а в пристрої 2, навпаки, декодуються.

Дані в комп'ютері зберігаються та опрацьовуються у вигляді комбінації електричних сигналів (сигнал є, сигнал відсутній), які для зручності описують двійковим кодом — послідовністю цифр 0 і 1: відсутність сигналу — 0, наявність — 1. Для кожного сигналу є канал зв'язку, так, цифра 3 буде передана як 0111.

Біт (від англ. *binary digit* — двійкова цифра, скорочено **bit**) — це двійковий знак із значенням 0 або 1.

Біт є найменшою одиницею кодування, а байт — послідовність із 8 бітів — найменшою одиницею зберігання даних у комп'ютері.

Сучасні комп'ютери оперують об'ємами пам'яті, які дозволяють опрацьовувати значно більшу кількість даних. Одиниці вимірювання довжини двійкового коду:

1 кілобайт (Кб) = 1024 (2^{10}) байти
1 мегабайт (Мб) = 1024 Кб = 2^{20} байти
1 гігабайт (Гб) = 1024 Мб = 2^{30} байти
1 терабайт (Тб) = 1024 Гб = 2^{40} байти

З прикладами кодування ми зустрічаємося не лише під час роботи з технікою, а й у повсякденному житті. Всім знайомі дорожні знаки, що є повідомленнями для водіїв та пішоходів, нотні знаки — для відтворення звуків за допомогою музичних інструментів, які розуміє музикант. Для спілкування між людьми різних країн запрошують перекладача, а для розуміння тексту, написаного незнайомою мовою, його перекладають — здійснюється кодування — декодування.

У кодування давня історія, ми ознайомимося лише з його найвідомішими досягненнями.

Для нашого сьогодення характерне створення автоматизованих систем кодування — декодування. Так, система штрих-кодів однозначно ідентифікує товари в магазинах; на стрічці або чіп-елементі банківських карток закодовані дані власника та його рахунку в банку.

Шифр Цезаря названо на честь римського імператора Гая Юлія Цезаря і призначено для кодування літер алфавіту. Кожна літера заміняється на ту, що віддалена від неї в алфавіті на певну кількість позицій (такий зсув по алфавіту називається *ключем*). При цьому алфавіт може зчитуватися по замкненому колу, наприклад, літера «Я» з ключем 3 закодується як «В». Юлій Цезар використовував для листування шифр зсуву з ключем 3.

Слово «біт» із ключем 2 буде закодовано як «гйф» (рис. 2.2). Процес декодування виконується у зворотному порядку.

а	б	в	г	...	і	ї	й	...	т	у	ф	...
а	б	в	г	...	і	ї	й	...	т	у	ф	...

Рис. 2.2. Демонстрація зсуву за правилами шифру Цезаря

Шифр Віженера відрізняється від шифру Цезаря змінним ключем на основі літер у ключовому слові. Тобто це вже відомий нам шифр Цезаря, але з різними значеннями зсуву.

Шрифт Брайля був створений у 1928 році французом Луї Брайлем для читання тексту сліпими й слабозорими людьми. В основі шрифту лежить шість об'ємних крапок. Зчитування відбувається дотиками пальців до крапок і розпізнаванням їх комбінацій. Першою книгою, надрукованою за системою Брайля, була «Історія Франції».

Для різних мов шрифт Брайля різний.

На рис. 2.3 подано фрагмент абетки Брайля українською мовою.

Символ	а	б	в	г	ґ	д	е	є	ж	з	и
Шрифт Брайля	●○ ○○ ○○	●○ ●○ ○○	○● ●● ○○	●● ●● ○○	●● ●● ○●	●● ○● ○○	●○ ○● ○○	○● ○● ○○	○● ●● ○○	●○ ○● ●○	○● ○● ○○
Символ	і	ї	й	к	л	м	н	о	п	р	с
Шрифт Брайля	●● ○● ●○	●● ○● ●●	●● ●○ ●●	●○ ○○ ●○	●○ ●○ ●○	●● ○○ ●○	●● ○● ●○	●○ ○● ●○	●● ●○ ●○	●○ ●● ●○	○● ●● ●○
Символ	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
Шрифт Брайля	○● ●● ●○	●○ ○○ ●●	●● ●○ ○○	●○ ●● ○○	●● ○○ ○○	●● ●● ●○	●○ ○● ●○	●● ○○ ●●	○● ●● ●●	●○ ●● ○●	●● ○○ ○●

Рис. 2.3. Українські літери абетки Брайля

Азбука Морзе містить літери, закодовані послідовністю крапок і тире, різною за довжиною для різних літер. Її створив американський винахідник і художник Семюел Морзе у 1838 році. Принцип кодування полягає в тому, що найчастіше вживані в англійській мові літери кодуються простішими сполученнями крапок і тире. Так, літера «Е» кодується крапкою, а «Т» — тире. На рис. 2.4 наведено варіант азбуки Морзе українською мовою.

Метод кодування Морзе, заснований на двох знаках, нагадує двійковий код, але на відміну від кодування у комп'ютерах літери алфавіту Морзе мають різну довжину.

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

Рис. 2.4. Українська абетка Морзе

У 1890 році під час підбиття результатів перепису населення було застосовано табуляційну машину Голлеріта (рис. 2.5). Її розробив американський інженер і винахідник Герман Голлеріт. Машина використовувала закодовані дані на перфокартах. Вони завантажувались у з'єднаний із машиною пристрій, де нанизувалися на голки: коли голка потрапляла в отвір, то замикала електричний контакт і спрацьовував лічильник.

Перфокарти тривалий час були основним носієм комп'ютерних даних під час збереження й опрацювання. У 1980-х роках їх замінили гнучкі магнітні дискети.

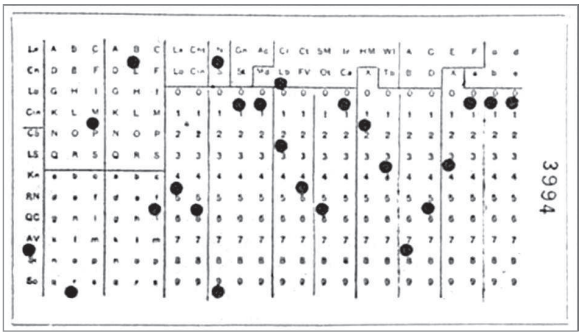


Рис. 2.5. Табулятор Голлеріта

Запитання для перевірки знань

1. Дайте означення коду; двійкового коду; біта.
2. Назвіть одиниці зберігання даних.
3. Наведіть приклади з історії кодування.
4. Яка різниця між кодуванням і декодуванням?
5. Наведіть приклади символьного та графічного кодування даних.
6. Поясніть, у чому полягає різниця між поняттями «кілобайт» і «кібібайт». Наведіть приклади.

Завдання для самостійного виконання

Дослідіть назви та призначення одиниць вимірювання кількості інформації.

1. Увійдіть на сайт Вікіпедії, у пошукове поле введіть слово «кібібайт».
2. Перейдіть на сторінку, що містить це слово.
3. Занотуйте в робочий зошит значення слова «кібібайт».
4. Занотуйте скорочений запис цього слова.
5. Запишіть означення та значення інших подібних понять.

2.2. Кодування чисел

Що таке кодування? Наведіть приклади кодування даних із повсякденного життя. Навіщо здійснюється кодування в пристроях?



Ви вже познайомились із різними способами кодування даних. Заміна початкових даних на числа, інші символи або графічні зображення відбувається за певними правилами.

У комп'ютерах всі дані подаються у вигляді двійкового коду, що пояснюється особливістю апаратної будови комп'ютера. Будь-які дані зберігаються в пам'яті комп'ютера в комірках пам'яті. Комірка складається з однотипних фізичних елементів, стан яких реагує на електричний сигнал: за нульової напруги стан один, за наявності напруги — стан інший. Математично такий стан елемента можна описати за допомогою цифр 0 і 1, знайомих вам як значення біта.

Існують такі способи кодування:
- за допомогою чисел (наприклад, двійкове)
- символьне (шифр Цезаря)
- графічне (дорожні знаки)

Розглянемо способи кодування чисел.

Для цілих чисел задіяні всі розряди коду. Це визначає діапазон чисел. За двобітового кодування отримуємо у двійковій системі коди таких чисел: 00, 01, 10 і 11. Отже, можна закодувати числа від 0 до 3 — чотири числа.

Одним байтом можна закодувати числа від 0 до 255 — 256 чисел. За однобайтового кодування найменше число 0 — усі біти дорівнюють нулю. А якщо перевести всі одиниці в байті (8 бітів), то отримаємо 255. Наявність комбінацій чисел 0 і 1 дасть проміжні значення із заданого діапазону.

У комп'ютері по-різному кодуються числа, значення яких може бути:
- тільки додатним — так звані беззнакові числа й такі, що можуть набувати значень як додатних, так і від'ємних
- тільки цілим і дійсним

Кількість бітів для кодування визначає діапазон значень числа, яке кодується. Значення максимального беззнакового цілого числа, закодованого n бітами, є $2^n - 1$.

Розглянемо таблицю:

Кількість бітів (n)	Мінімальне значення	Максимальне значення ($2^n - 1$)
8	0	255 ($2^8 - 1$)
16	0	65 535 ($2^{16} - 1$)
32	0	4 294 967 295 ($2^{32} - 1$)
64	0	18 446 744 073 709 551 615 ($2^{64} - 1$)

Під час кодування повідомлення відбувається зміна вигляду повідомлення без зміни його змісту.

Результатом кодування цілих чисел зі знаком є прямий код числа. Прямим кодом двійкового подання чисел є форма: знак, значення числа. Найстарший біт (розряд зліва) відводиться під знак: значення 0 для додатного числа і 1 — для від'ємного. Решта бітів визначає значення числа, записаного у двійковій системі. Наприклад, $5_{10} = 101_2$. У 8-бітовому коді в комп'ютері $+5_{10} = 0000\ 0101$, а $-5_{10} = 1000\ 0101_2$.

У таблиці, поданій на с. 22, наведено діапазон цілих чисел для кодування різною кількістю бітів.



Комірки пам'яті комп'ютера, у яких зберігаються дані, мають обмежену кількість елементів — розрядів у двійковому поданні: 8, 16, 32 або 64 (1 байт, 2, 4 або 8) — кодування відбувається бітами, загальна кількість яких є кратною 8.

Спосіб кодування числа і довжина його двійкового коду залежать від типу числового значення, а саме: чи є воно додатним або від'ємним, цілим або дробовим.

Кількість бітів (n)	Мінімальне значення (2^{n-1})	Максимальне значення ($2^{n-1}-1$)
8	–128 (2^7)	127 (2^7-1)
16	–32 768 (2^{15})	32 767 ($2^{15}-1$)
32	–2 147 483 648 (2^{31})	2 147 483 647 ($2^{31}-1$)
64	-2^{63}	9 223 372 036 854 775 807 ($2^{63}-1$)

Знаковий розряд у прямому коді несе інформацію тільки про знак числа і не має кількісного значення. Для подальшого опрацювання від'ємного числа відбуваються перетворення його значущої області коду. У програмуванні оперують цілими числами, для подання яких достатньо 1, 2 або 4 байтів.

Дійсні числа в комп'ютері подано у формі з рухомою крапкою. Ми звикли оперувати числами у формі з фіксованою крапкою: десяткова крапка відокремлює цілу частину числа від дробової. Наприклад: 7.375_{10} або 1101.0112.

Нормалізована форма запису числа (або форма з рухомою крапкою) подана мантисою та порядком.

Мантисою числа є запис цифр числа з крапкою після першої значущої цифри (приклад 1). Щоб отримати правильне значення числа, поряд із мантисою потрібно дописати порядок — степінь 10, завдяки якому значення числа в мантисі буде помножене або поділене на 10 у цьому степені для отримання істинного значення числа (приклад 2). У комп'ютері мантиса та порядок будуть подані у двійковій системі в такій послідовності: знак числа, порядок, закодований як число зі знаком, і мантиса. Оскільки у мантиси перша значуща цифра не 0, то у двійковому коді перед комою завжди буде 1 і біти під десяткову крапку не виділяються.

Розмір точного подання числа залежить від відповідності необхідної кількості бітів запису його мантиси у двійковій системі числення та кількості бітів, які призначено для зберігання числа. Під дійсні числа відводиться 4 або 8 байтів із мантисами 23 і 53 біти відповідно.

У таблиці наведено розподіл бітів для 4-байтового подання числа з рухомою крапкою:

Знак числа	Порядок зі знаком	Мантиса
1 біт	8 бітів	23 біти

Приклад 1.
Для числа 40075.696 (довжина екватора Землі в тисячах кілометрів) мантисою буде 4.0075696, а для 0.75 (такий розмір у міліметрах має найбільша бактерія) — 7.5.

Приклад 2.
Для прикладу 1 запишемо числа з фіксованою та рухомою крапкою:
$40075.696 = 4.0075696 \cdot 10^4$;
$0.75 = \left(7.5 \cdot 10^{-1}\right)_{10}$ або
$0.75_{10} = 0.11_2 = 1.1 \cdot 2^{-1}$
(ділення на 2 у двійковій системі — зсув числа на 1 біт управо).

Запитання для перевірки знань

1. Який біт та яке його значення відповідають за знак числа?
2. Назвіть форми подання дійсних чисел.
3. Чому нормалізовану форму числа ще називають формою з рухомою крапкою?
4. Пригадайте діапазони беззнакових і знакових чисел в однобайтовому поданні. Чи однакова тут кількість закодованих чисел?
5. Чому при однобайтовому поданні цілих чисел зі знаком найменше значення –128 (адже 1 у двійковому коді числа 128 стоїть на місці знака)?
6. Для кодування значень температури повітря (ціле число в інтервалі від –15 до 31) використовується двійковий код.
Яка мінімальна довжина двійкового коду?

Завдання для самостійного виконання

1. Знайдіть у пошуковій системі інтернет відстань від Землі до Місяця і запишіть її значення у формі з рухомою крапкою. Скільки байтів потрібно для зберігання цієї величини у комп'ютері?

2. Маса молекули сірчаної кислоти (H_2SO_4) $163 \cdot 10^{-7}$кг. Запишіть це значення у формах з фіксованою та рухомою крапкою. Скільки байтів необхідно для збереження цієї величини у комп'ютері?

2.3. Кодування символів

Назвіть типи даних. Як кодуються дані в електронних обчислювальних машинах?

Текст кодується двійковими кодами і складається зі слів, які складаються із символів. Отже, кодування текстових даних — це кодування символів. Текстові дані є набором символів — літер, цифр, знаків пунктуації, дужок, знаків арифметичних дій, а також пропусків, які відокремлюють слова. Уведення символу з клавіатури супроводжується появою і передаванням електричних сигналів, які в математичній моделі подано двійковими кодами.

Кодова таблиця ASCII складається з двох частин.

Базова частина таблиці містить коди семи молодших бітів (нумерація кодів від 0 до 127). Перші 32 кодові комбінації (від 0 до 31) відведено для операцій керування, наприклад: переміщення в межах рядка та з рядка на рядок, видалення, звуковий сигнал, підтвердження введення (Enter). Наступні коди (від 32 до 127) відведено для символів латинського алфавіту, цифр, знаків пунктуації та ін.

Базова частина таблиці ASCII є спільною в усіх комп'ютерах. Великі літери англійської абетки починаються з 65-го місця, у двійкових кодах це 01000001. Отже, літері «А» відповідає код 01000001.

Розширена частина таблиці містить коди 128–255 і використовується для кодування символів національної абетки та символів псевдографіки. Тому в різних країнах одному й тому самому коду відповідають різні символи.

Для українських символів використовували 8-розрядні таблиці кодування KOI8-U, KOI8-R, Windows-1251, ISO 8859. Щоб уникнути непорозумінь із національною частиною таблиці кодування, було розроблено та впроваджено 16-розрядний міжнародний стандарт ISO 10646 під назвою Unicode (Юнікод), який має 65 536 кодових комбінацій. Програми MS Windows Office підтримують це кодування з 1997 року.

У MS Word визначають шістнадцятковий код символу в системі Unicode у вікні Символ (рис. 2.6). Щоб відкрити це вікно, потрібно перейти у стрічці Вставлення → групи Символи → списку Символи → до команди Інші символи.

Кожному символу відповідає двійковий код довжиною 8 або 16 бітів: 8 бітів дозволяють закодувати 256 (2^8) символів, а 16 бітів — 65 536 (2^{16}) .

Базову частину таблиці ASCII можна переглянути на сайті

Першою стандартизованою є система кодування **ASCII** (*American Standart Code for Information Interchange*) із використанням двійкового коду довжиною 1 байт.

Довжина у 8 бітів дозволяє утворити $2^8 = 256$ різних кодів.

Юнікод має кілька версій: UTF (*Unicode Transformation Format* — формат перетворення Юнікоду) і UCS (*Universal Character Set* — універсальна таблиця символів). UTF широко застосовується для передавання символів інтернетом.

Залежно від правил кодування програмним засобом текстові файли зберігаються в різних форматах. Наприклад, форматом збереження текстових даних у програмі **Блокнот** є TXT, а у Word — DOC, DOCX.

У нижній частині цього вікна у шістнадцятковій системі числення висвітлюється код вибраного символу. На рис. 2.6 видно, що код великої англійської літери А дорівнює $41_{16} = 1 \cdot 1 + 4 \cdot 16 = 65_{10}$ (місце, як і було визначено).

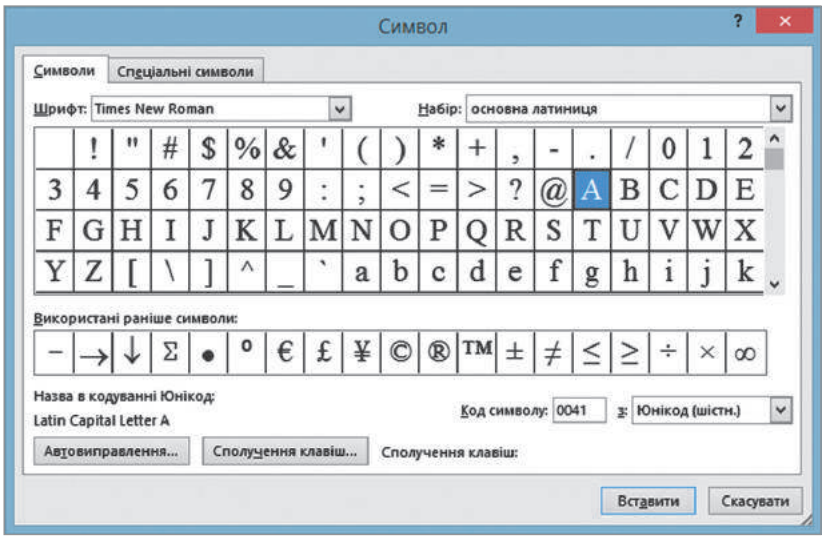


Рис. 2.6. Вигляд вікна **Символ**

Якщо скористатися сполученням клавіші Alt і цифр із додаткової цифрової клавіатури, то в документ буде вставлено символ, код якого набрано на клавіатурі.

Запитання для перевірки знань

1. Що означає термін «текстові дані»?
2. Скільки символів містить кодова таблиця ASCII?
3. Як можна визначити коди символів у текстовому процесорі Word?
4. Яка довжина фрази «Моя країна — Україна» у стандарті Unicode; ASCII?
5. Чи можуть бути в тексті неправильно зазначені коди однотипних літер різних алфавітів?
6. Поясніть, яким чином при введенні з клавіатури однакових літер можна отримати такий вигляд:

Біт

Завдання для самостійного виконання

1. Запустіть програму Word.
2. Виконайте команду **Вставлення** → **Символи** → **Інші символи** — відкриється вікно **Символ**.
3. Зі списку **Шрифт** виберіть шрифт.
4. У вікні **Символи** виділіть літеру **D** і двічі клацніть — вона буде відображена в рядку з назвою **Використані раніше символи**. У цьому рядку виділіть **D** як поточний символ.
5. Передивіться шістнадцятковий код літери із зазначенням таблиці кодування — виберіть таблицю ASCII (шістн.), змініть на Кирилиця (шістн.), Юнікод (шістн.).
   З'ясуйте, чи змінюється код літери.
6. Знайдіть у таблиці серед літер кирилиці літеру **Я** і перегляньте її коди за рекомендаціями в п. 4, 5.
   З'ясуйте, чи відбувались зміни. Відповідь обґрунтуйте.

2.4. Кодування графічних зображень

Наведіть приклади інформації, поданої у графічному вигляді.

Інформація, подана в графічному вигляді, оточує нас в повсякденному житті практично повсюди. Так, «зебру» на дорозі «читають» і пішоходи (місце для переходу), і водії (загальмувати, бути уважним, перевірити, чи вільна дорога) (рис. 2.7). Нам цілком звичні та зрозумілі за змістом піктограми на екранах смартфонів і дисплеях.

Є різні форми подання та збереження графічних даних: неперервна (аналогова) і цифрова. Прикладом аналогового подання є неперервна зміна кольорів на полотні митця або навіть на меблях. Зображення, що виводиться на екран дисплея, складається з окремих точок різного кольору — отримуємо цифрове, дискретне подання.



Рис. 2.7. Графічний спосіб подання даних

Просторова дискретизація — перетворення аналогової форми подання графічних даних на цифрову.

Як відомо, у растровому зображенні найменшим елементом є *піксель*, а прямокутна ділянка з пікселів — це *растр*.

У сучасних комп'ютерах для подання кольору пікселя зазвичай використовується 16 або 32 біти.

Кодування відбувається, оскільки кожному елементу графічного зображення відповідає значення двійкового коду. Приклад кодування растрового зображення наведено на рис. 2.8. Видно, що збільшена літера К побудована з пікселів.

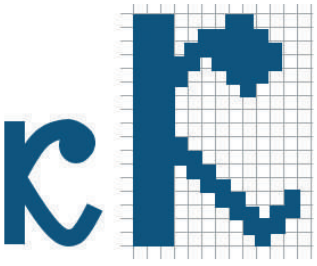


Рис. 2.8. Растрова структура збільшеного зображення літери

Для створення й опрацювання графічних зображень є низка програмних засобів. Залежно від способу збереження та опрацювання зображення їх поділяють на растрові та векторні (є й інші засоби, проте нас цікавить лише спосіб кодування зображення).

Для коду кольору зображення (див. рис. 2.8) достатньо використати біт: світлий тон кодується 0, а темний — 1, і код верхньої частини зображення матиме такий вигляд, як наведено на рис. 2.9.

1	1	1	0	0	0	0	1	1	0	0	0	0
1	1	1	0	0	1	1	1	1	1	0	0	0
1	1	1	0	1	1	1	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1	1	0	0	0
1	1	1	0	0	0	0	0	1	0	0	0	0

Рис. 2.9. Приклад кодування растрового зображення

Для збереження зображення достатньо скористатися координатами точки на екрані та зазначити код її кольору. У випадку кодування растрового зображення зазначаються коди кольорів точок прямокутної ділянки від лівого верхнього кута до правого нижнього.

Зображення розміром 100×100 пікселів, вибране у графічному редакторі, виглядатиме більшим на екрані з роздільною здатністю 640×480 пікселів, ніж за роздільної здатності 1024×768 пікселів.

У сучасних комп'ютерах в основному використовують достатньо великі значення роздільної здатності екрана: 1024×768, 1280×1024 або 1680×1050 пікселів.



Рис. 2.10. Графічне подання колірної моделі RGB



Рис. 2.11. Графічне подання колірної моделі CMYK

Як відомо, векторне зображення складається з елементарних об'єктів — *графічних примітивів* (лінія, дуга, прямокутник та ін.), які описуються математичними формулами. Наприклад, об'єкт квадрат можна описати так: центр — 80, 60 (координати *x*, *y*); сторона — 20; лінія — суцільна; товщина — 0,50; заповнення — відсутнє.

Кожний піксель растрового зображення містить інформацію про його колір. Контури та заповнення такого зображення теж характеризуються кольором. Для кодування кольорів векторного зображення використовують ті самі методи, що й для растрового.

Векторні й растрові зображення відтворюються на моніторах, лазерних і струменевих принтерах, які за принципом дії є растровими. А для відтворення векторних зображень вони перетворюються в набори пікселів. Процес перетворення здійснюється з урахуванням масштабу зображення та роздільної здатності пристрою.

Роздільна здатність екрана — кількість пікселів на одиницю довжини, найчастіше на дюйм (1 дюйм = 2,54 см); позначається ppi (*picel per inch* — пікселів на дюйм).

Чим вища роздільна здатність екрана, тобто більша кількість рядків і точок у рядку, тим вища якість зображення.

Обсяг збереженого зображення залежить також від методу кодування кольору пікселя. Глибина кольору визначає метод кодування кольорів: для кодування кольору чорно-білого зображення достатньо 1 біта, для зображення у відтінках сірого колір кодується 1 байтом — отримують 256 відтінків сірого. Є різні колірні моделі для кодування кольору.

Колірна модель — математична модель опису подання кольорів у вигляді послідовності чисел, кожне з яких називається колірним компонентом.

На екрані ми бачимо зображення завдяки випромінюванню світла точками екрана, а на папері — завдяки світлу, яке відбивається від поверхні аркуша. Для опису кольорів, утворюваних у різний спосіб, створено різні колірні моделі.

Колірні моделі умовно поділено на три класи:

- адитивні (RGB): колір пікселя формується поєднанням базових кольорів червоного (*Red*), зеленого (*Green*) і синього (*Blue*) — рис. 2.10;
- субтрактивні, або додаткові (оскільки модель доповнює кольори адитивної моделі до всього спектра): модель CMY отримала назву від складових *Cyan* (блакитний), *Magenta* (пурпуровий) і *Yellow* (жовтий) — рис. 2.11;
- перцепційні: моделі Lab, HSB, HSL — мають за основу сприйняття кольору людським оком.

Розглянемо спочатку принципи кодування кольору в адитивній моделі RGB. Якщо кожний колір закодувати 1 бітом

(колір є, колір відсутній), отримаємо 3-бітове кодування кольору, бо маємо три базові кольори для кодування. А комбінування базових кольорів дасть 8 (2^3) кольорів. Для кожного базового кольору в сучасних комп'ютерах відводиться 8 бітів, колір має 256 відтінків від 0 до 255.

Усі нульові значення означають відсутність будь-якого кольору — чорний колір, а 255 255 255 — навпаки, білий. За моделлю RGB можна отримати 256^3 різних кольорів (або 2^{24}). Таке кодування називають True Color. Розглянемо таблицю:

Результуючий колір	Базові кольори		
	червоний	зелений	синій
Червоний	11111111	00000000	00000000
Зелений	00000000	11111111	00000000
Синій	00000000	00000000	11111111
Блакитний	00000000	11111111	11111111
Пурпуровий	11111111	00000000	11111111
Жовтий	11111111	11111111	00000000
Білий	11111111	11111111	11111111
Чорний	00000000	00000000	00000000

Таким чином, з таблиці видно, що, наприклад, кольори веселки отримано комбінацією базових кольорів моделі.

До 24-бітного кодування (режим High Color) додається 8-бітний альфа-канал, який задає прозорість в пікселях. Отже, колір кодується 32 бітами, а реально має 256^3 кольорів. У поданні кольору зображення використовується така важлива характеристика графічного зображення, як глибина кольору (або колірна роздільна здатність, або бітова глибина).

Глибина кольору — кількість бітів (обсяг пам'яті) для зберігання й подання кольору під час кодування одного пікселя графічного або відеозображення.

На графічному зображенні, яке створене з використанням моделі RGB і чудово виглядає на екрані дисплея, чимало відтінків не передаються на папір під час друку. Кольорове зображення на папері ми бачимо у відбитому від паперу світлі.

Розглянемо принципи кодування кольору в субтрактивній моделі CMY. Модель CMY описує колір у відбитому світлі. Оскільки поєднання CMY-кольорів не дає глибокого чорного кольору, то під час друку до основних фарб додають чорну — використовують модель CMYK (англ. *blac*K — літеру B не використовують, щоб не плутати з *Blue* моделі RGB).

Однозначно визначає колір лише перцепційна модель Lab.

Графічні редактори надають можливість вибрати режим кольору зображення: кольорове, напівтонове (у відтінках сірого) або чорно-біле, а також вказати коефіцієнт прозорості.



У 1996 році прийнято sRGB (*standard Red Green Blue*) — стандарт подання колірного спектру з використанням моделі RGB для уніфікації її використання в моніторах, принтерах та на інтернет-сайтах.

У процесі опрацювання графічних зображень для отримання високого контрасту користуються технологіями HDR (або HDRI — *High Dynamic Range Imaging*), які перевищують можливості стандартних технологій відображення. Режим екрана HDR можна встановити в комп'ютері з Windows 10 і отримати яскраве деталізоване зображення порівняно зі стандартом SDR/LDR (*Standard Dynamic Range/Low Dynamic Range*) і sRGB.

У моделі Lab є можливість окремо впливати на яскравість, контрастність і колір зображення. Колір визначається трьома параметрами: освітленістю, діапазоном зміни від пурпурового до зеленого та діапазоном зміни від синього до жовтого. Освітленість змінюється в діапазоні від 0 до 100 %. Її максимальне значення відповідає максимальній яскравості кольору. Значення діапазонів зміни кольорів задаються числами від –128 до 127.

Вибираючи колір, можна дізнатися, як він виглядатиме в різних програмах (рис. 2.12).

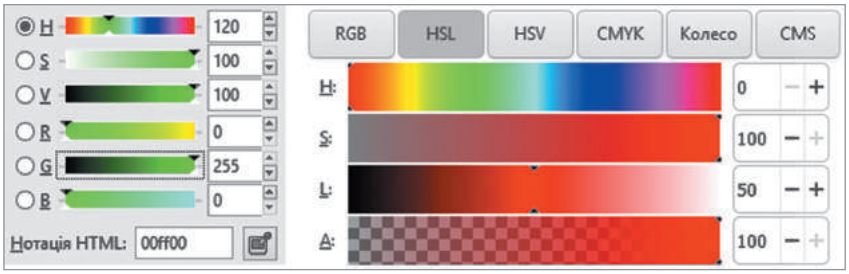


Рис. 2.12. Приклади вибору кольору у вікнах різних редакторів

Щоб вибрати колір об'єкта, створеного у векторному редакторі, слід створити або виділити об'єкт і перейти на стрічку Формат фігури; у групі Стилі фігури вибрати список Заливка фігури, а в ньому — команду Інші кольори заливки; у вікні, що відкрилося, зі списку модель — кодування кольору.

Модель HSB має такі характеристики кольору (рис. 2.13):

- колірний тон (англ. — *Hue*) — колір світла, який ми бачимо. Якщо всі кольори розмістити на колірному колі, то позиція кольору буде змінюватись від 0° до 360°, що й дасть значення параметра тону;
- насиченість (*Saturation*) — характеристика чистоти кольору певного тону. Значення насиченості коливаються від 0 до 100 % у суміші з сірим кольором; тон кольору задається хвилею певної довжини, решта хвиль спектра зливаються в сірий колір. Значення 0 насиченості відповідає сірому кольору;
- яскравість (*Brightness*) або освітленість (*Lightness*) — означає, наскільки світлим (100 %) або темним (0 %) подано колір. Моделі можно називати HSB, коли яскравість замінюється на освітленість.

У поліграфії друк здійснюється блакитною, пурпуровою, жовтою та чорною фарбами, які становлять палітру CMYK. У CMY усі нулі дають біле світло (усі колірні компоненти RGB відбились, і додавати до спектра нічого). Якщо поверхня повністю поглинула світло, то ми бачимо її чорною.

Найяскравіші кольори моделі RGB неможливо передати під час друку за допомогою CMYK, а для найтемніших кольорів моделі CMYK немає аналогів у RGB.

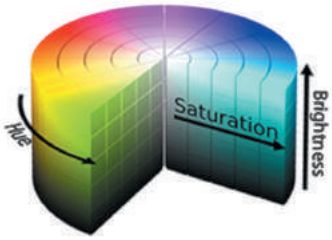


Рис. 2.13. Графічне зображення моделі HSB

Запитання для перевірки знань

1. Що таке просторова дискретизація?
2. Назвіть колірні моделі для опису кольорів.
3. Який обсяг пам'яті необхідно виділити для збереження растрового зображення 64×64 пікселі, якщо в палітрі 16 кольорів?
4. Який принцип кодування кольору в RGB і CMYK.
5. Які моделі точніше описують колір при скануванні та виведенні зображення на друк?
6. Як визначити код кольору в зображенні для сайту?

Завдання для самостійного виконання

1 Дано формати: BMP (*Bitmap*); JPEG (*Joint Photographic Expert Group*); TIFF (*Tagged Image File Format*); PNG (*Portable Network Graphics*); PDF (*Portable Document Format*). Виходячи з формату, наведіть тип і приклад графічного редактора.

Практична робота № 2

ТЕМА. Опрацювання колірних моделей у векторному графічному редакторі

ЗАВДАННЯ: ознайомитися з можливостями редактора щодо налаштування колірної гами зображення з використанням різних колірних моделей.

ОБЛАДНАННЯ: комп'ютер, програмний засіб Microsoft Office.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1 Відкрийте графічний редактор.

2 Намалюйте правильний трикутник. Фігура має бути активною — активною є стрічка Формат фігури.

2. Стрічка **Вставлення** → група **Ілюстрації** → **Список фігур**.

3 Виберіть колір заповнення, у списку Контур фігури → Без контуру.

4 У вікні, що відкриється, перейдіть на вкладку Спектр і доберіть колір або задайте вручну так, щоб у моделі RGB червоний колір мав значення 255, решта — 0.

3. Група **Стилі фігур** зі списку **Заливка фігури** → команда **Інші кольори заливки**.

5 У списку моделей перейдіть до HSL і запишіть у зошиті значення тону кольору: hue (або в іншомовному інтерфейсі редактора — тон).

6 Повторіть попередні дії для створення трикутників зеленого та синього кольорів.

7 У моделі CMYK доберіть колір або задайте так, щоб у моделі RGB два кольори мали значення 255, а третій — 0.

8 Зафіксуйте колір і значення тону, запишіть у зошиті.

9 Повторіть попередні дії, змінюючи попарно значення кольорів (два по 255, третій 0), кількість трикутників має бути 6.

10 Розташуйте зображення трикутників у ряд за кольорами веселки (див. рисунок). Додайте підписи — вкажіть перші літери кольорів моделей RGB і CMYK.

10.



11 Додайте помаранчевий трикутник і запишіть значення кольорів моделей RGB і HSL.

12 Порівняйте кольори останнього трикутника.

Зробіть висновок щодо збігу кодів у колірних моделях.

2.5. Кодування звукових даних

Дайте означення просторової дискретизації. Що означає закодувати графічні дані?

Приклад 1. Звукова доріжка вінілової платівки (у минулому ці платівки були єдиним пристроєм зберігання звуку) неперервно змінює свою форму — так зберігаються звукові дані в неперервному, або аналоговому, вигляді.

Частота дискретизації — кількість вимірювань сигналу за 1 с; вимірюється в Герцах (Гц).
Одне значення за 1 с відповідає частоті f = 1 Гц.
1000 вимірювань за 1 с f = 1000 Гц = 1 кГц.

Одним зі способів передавання інформації є звук. Звукові сигнали надсилають живі істоти, люди спілкуються, не тільки коли перебувають поря, а й на відстані — за допомогою пристроїв. У пристроях відбувається кодування звичного для нас звуку в певні сигнали. Звуком є фізичне явище — коливальний рух частинок середовищ, який поширюється в цьому середовищі.

Під час передавання звуку поширення коливального руху середовищем є неперервним явищем: будь-якому моменту часу відповідає наявність звукового сигналу (приклад 1).

У комп'ютерах відбувається цифрове кодування неперервного (аналогового) сигналу. Сигнал опрацьовується та зберігається у певні проміжки часу, а висота вертикальних ліній у певні моменти часу показує, який сигнал за значенням, тобто визначає амплітуду сигналу.

Оцифрування звуку — процес перетворення звукових даних в аналоговому вигляді у закодований цифровий сигнал.

Дискретизація — процес вимірювання, зберігання та опрацювання значень аналогового сигналу через однакові проміжки часу, які називаються частотою дискретизації.

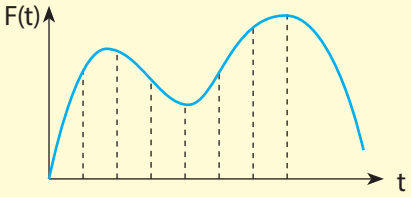


Рис. 2.14. Приклад дискретизації неперервного сигналу

Процес дискретизації показано на рис. 2.14. Чим вища частота дискретизації, тим точніше відбувається перетворення безперервних даних у дискретні, але при цьому зростає і кількість отриманих значень, і, як наслідок, складність їх опрацювання, передавання та зберігання.

Квантування амплітуди сигналу — процес замінення реального значення сигналу, який отримується в результаті вимірювання.

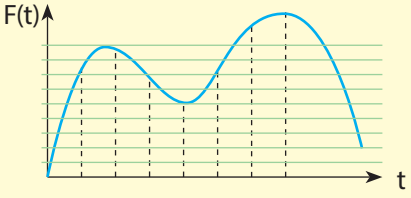


Рис. 2.15. Приклад квантування неперервного сигналу

Найближче значення з набору фіксованих значень називають *рівнями квантування* (рис. 2.15). Набір фіксованих значень є *значеннями ваги розрядів*, наприклад, у двійковій або шістнадцятковій системі числення. У процесі кодування визначають значення сигналу в певні проміжки часу (дискретизація) у вигляді двійкового коду (квантування). Будь-яке число можна подати у двійковій системі числення, але на можливість точного подання значення впливає кількість розрядів або бітів.

Кількість використаних розрядів для подання даних називають розрядністю квантування.

Розглянемо алгоритм оцифрування звуку.

Крок 1	На приймач подається звуковий сигнал (приклад приймача — мікрофон)
Крок 2	Неперервний електромагнітний сигнал мікрофона подається на аналого-цифровий перетворювач (АЦП)
Крок 3	В АЦП здійснюється безпосереднє оцифрування звуку — дискретизація та квантування отриманого сигналу
Крок 4	Сигнал у вигляді двійкового коду зберігається в комп'ютері

Розрядність квантування і частота дискретизації є характеристиками звукових пристроїв записування та відтворення. Наприклад, 24 біти/192 кГц; або 16 бітів/48 кГц. Якість цифрового звуку залежить від розрядності та частоти дискретизації.

Як відомо, звукові дані зберігаються у файлах, які потребують значного обсягу пам'яті.

Залежно від способу кодування є різні способи квантування та стиснення даних. Розрізняють такі аудіоформати файлів збереження закодованого звуку: без стиснення: WAV, AIFF; зі стисненням без втрат: APE, FLAC; зі стисненням із втратами: MP3, WMA (*Windows Media Audio*).

Відеодані для їх опрацювання та зберігання комп'ютером теж кодуються. Кодування звукового супроводу відеоданих нічим не відрізняється від кодування звуку, а зображення у відео складається з окремих кадрів, які змінюються з певною частотою.

Кадри кодуються як звичайне растрове зображення — попіксельно. Якщо закодувати всі кадри, відеодані будуть подані як послідовність змінних графічних об'єктів, описаних у цифровому вигляді. Основою збереження кольорового відеосюжету є використання моделі RGB. Зміна об'єктів відбувається з однаковою швидкістю, але закодований таким чином відеофайл матиме великий розмір.

Існують різні програми кодування відео, які використовують алгоритми стиснення відеоданих. Програма з таким алгоритмом називається кодеком. Кожний кодек зберігає відеофайли у своєму форматі. Кодек виявляє і зберігає кадри, на яких вже відбулась зміна сюжету, такі кадри називають *ключовими*. Дані зі змінами зберігаються в поточному відносно до попереднього кадрі; проміжні між ними кадри взагалі не зберігаються (приклад 2).

Характеристики зберігання відеоданих: *кількість кадрів за секунду* та *роздільна здатність кадру*. Різні стандарти стиснення мають різні характеристики збереження (приклад 3).

Приклад 2. Якщо у відеосюжеті на незмінному тлі відбувається рух об'єкта, то на всіх кадрах, задіяних у цьому відеосюжеті, зміни стосуються лише рухомого об'єкта. Зберігаються тло та деякі зображення під час руху об'єкта.

Приклад 3. Збереження відеоданих у стандарті SIF характеризується 30 кадрами за секунду та роздільною здатністю кадру 352×240 пікселів.

Запитання для перевірки знань

1. Наведіть приклад зберігання звукових даних в аналоговому (неперервному) вигляді.
2. Опишіть процес оцифрування звуку.
3. Дайте означення дискретизації сигналу.
4. Дайте означення квантування сигналу.
5. Поясніть сутність оцифрування відеоданих.
6. Навіщо необхідні алгоритми стиснення для кодування звукових і відеоданих.



Практична робота № 3

ТЕМА. Кодування даних різного формату в комп'ютері.
ЗАВДАННЯ: проаналізувати візуалізацію формату чисел із рухомою крапкою в комп'ютері за допомогою табличного редактора.
ОБЛАДНАННЯ: комп'ютер, табличний редактор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте табличний процесор.
2. У клітинку A1 уведіть 15 одиниць.
3. Скопіюйте це значення в клітинки A2, A3 — активною є клітинка A1.
4. Відкрийте вікно стрічки Основне групи Число.
5. У списку Числові формати виберіть формат Експоненціальний (це подання чисел із рухомою крапкою). Зробіть активною клітинку A2.
6. Відкрийте вікно форматів, виберіть формат Експоненціальний і змініть кількість знаків після крапки.

   Проаналізуйте, скільки нулів після десяткової крапки.
7. Зробіть активною клітинку A3. Виберіть формат Експоненціальний та установіть 0 знаків після крапки.
8. Проаналізуйте: як змінилося зображення числа; розмір чого ви встановлюєте кількістю десяткових знаків; яке значення має порядок числа; що замінює 10 у записі порядку числа.
9. З'ясуйте, скільки існує різних послідовностей із символів «крапка» і «тире» довжиною від 4 до 6 символів включно.
10. Світлове табло складається з лампочок. Кожна лампочка може перебувати в одному з двох станів (увімкнено, вимкнено). Визначте, якою найменшою кількістю лампочок на табло можна передати 100 різних повідомлень.
11. Розгляньте растрову чорно-білу фотографію розміром 13×18 см. У кожному квадратному сантиметрі 450 точок, кожна точка описується 2 бітами. Для розрахунків скористайтесь можливостями табличного процесора.
12. Для збереження растрових зображень розміром 512×256 пікселів відвели 64 Кбайти пам'яті. Для розрахунків скористайтесь табличним процесором.

Зробіть висновок про розміри кодування даних різних типів.

5. Як змінилося зображення числа?

11. Який обсяг фотографії в кілобайтах?

12. Яке максимально можливе число кольорів у палітрі зображення?

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 2

Розділ 3
КОМП'ЮТЕР ЯК УНІВЕРСАЛЬНИЙ ПРИСТРІЙ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ

3.1. Архітектура комп'ютера. Процесор, його будова та призначення

Пригадайте основні пристрої комп'ютера.

Архітектура комп'ютера (англ. *computer architecture*) — це набір дисциплін, які описують функціональність, організацію та реалізацію комп'ютерних систем.

Коли ми говоримо про архітектуру комп'ютера, то насамперед маємо розглядати структуру системної плати (рис. 3.1).

Системна плата

Системна плата — основний компонент персонального комп'ютера, самостійний елемент, який керує внутрішніми зв'язками і взаємодіє із зовнішніми пристроями.

Системну плату також називають материнською (*motherboard*), або головною (*mainboard*), платою. Вона забезпечує зв'язок між апаратними компонентами комп'ютера, такими як процесор, внутрішня та зовнішня пам'ять, відеокарта, а також периферійні пристрої.

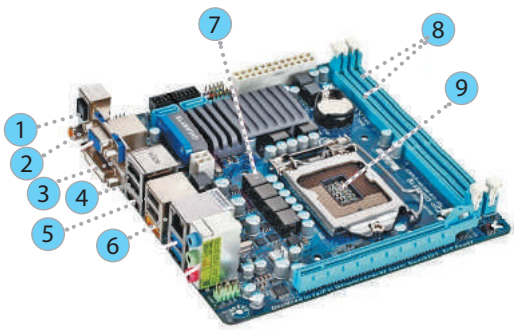
Системні плати зазвичай характеризуються *форм-фактором*, *чипсетом* і *сокетом* (типом процесорного гнізда).

Форм-фактор системної плати (англ. *form factor*) — це стандарт, який задає її габаритні розміри, форму, типи додаткових елементів і їхнє положення та орієнтацію. Він вказує розташування точок монтування, панелі введення-виведення, роз'ємів живлення та різноманітних інтерфейсів підключення.

Найбільш поширеним на сьогодні залишається форм-фактор ATX, який американська компанія Intel вперше представила в 1995 р. Ця лінійка була продовжена форм-фактором Micro-ATX, який підтримує як процесори Intel, так і AMD; він зазвичай використовується у моноблоках.

На початку XXI ст. компанія VIA Technologies розробила значно менше за розміром сімейство форм-факторів ITX, яке наразі представлене Nano-ITX і Pico-ITX. Nano-ITX — це повністю

Перша системна плата була розроблена американською електронною корпорацією IBM і продемонстрована в серпні 1981 року (PC-1). Максимум, що могла підтримувати PC-1 без використання плат розширення — 64 К пам'яті.



1 — VGA
2 — DVI
3 — HDMI
4 — eSATA
5 — USB 3.0
6 — HD-аудіо
7 — ферітові дроселі
8 — слоти DRAM
9 — сокет (роз'єм) процесора

Рис. 3.1. Будова типова системна плата

інтегровані плати, розроблені для споживання дуже низької потужності. Цей тип системної плати був спеціально виготовлений для інтелектуальних розваг, таких як доповнена віртуальна реальність, медіа-центри, смарт-телевізори, автомобільні пристрої тощо. Pico-ITX — це найменший форм-фактор системної плати, винайдений для пристроїв інтернету речей (IoT).

Розміри найпоширеніших форм-факторів

Назва	Розмір, мм
ATX	305×244
Micro-ATX	244×244
Mini-ATX	170×170
Nano-ITX	120×120
Pico-ITX	100×72

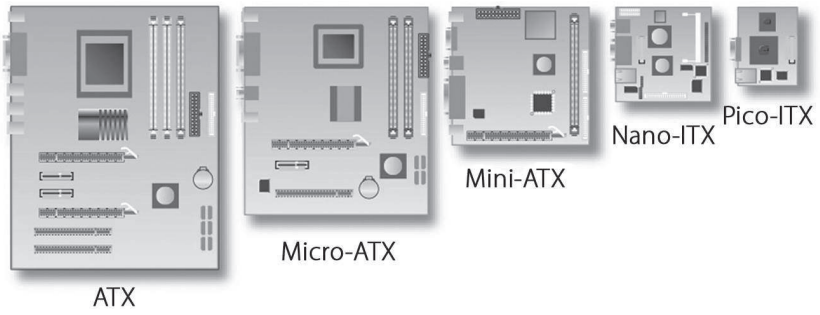


Рис. 3.2. Форм-фактори системної плати

Сучасні плати виконуються на основі чипсетів (*chipset*) — наборів із декількох великих інтегральних схем, що реалізують всі необхідні функції зв'язку основних компонентів — процесора, пам'яті і шин розширення. Чипсет визначає можливості застосування різних типів процесорів, основної і кеш-пам'яті і ряд інших характеристик системи, що характеризують перспективи її модернізації.

Сучасні чипсети забезпечують сумісність встановлюваних на системну плату модулів і дозволяють виконувати автоматичну ідентифікацію типів установлених компонентів.

Усі компоненти системної плати пов'язані між собою системою провідників (ліній), за якими відбувається обмін інформацією. Цю сукупність ліній називають інформаційною шиною, або шиною (*bus*).

- Шина даних забезпечує обмін даними між процесором, картами розширення, встановленими в слоти, і оперативною пам'яттю. Чим більше розрядність шини, тим більше даних може бути передано за один такт і тим вища продуктивність комп'ютера.
- Шина адреси служить для зазначення адреси до якого-небудь пристрою комп'ютера, з котрим процесор проводить обмін даними. Кожен компонент, кожен регістр введення/виведення і кожна комірка оперативної пам'яті мають свою адресу і входять в загальний адресний простір. По шині адреси передається ідентифікаційний код (адреса) відправника і (або) одержувача даних.
- Шина керування передає ряд службових сигналів: записи/зчитування, готовності до прийому/передачі даних, підтвердження прийому даних, апаратного переривання, керування та інших, щоб забезпечити передачу даних.

Для узгодження тактової частоти і розрядності пристроїв на системній платі встановлюють спеціальні мікросхеми, які включають контролер оперативної пам'яті і відеопам'яті (так званий *північний міст*) і контролер периферійних пристроїв (*південний міст*).

Багаторічне суперництво між компаніями AMD і Intel у галузі виробництва процесорів триває. Так, у червні 2016 р. Intel презентувала процесор Intel® Xeon Phi™ Processor 7290 із 72 ядрами та 288 потоковими командами. Навесні 2020 р. компанія AMD представила процесор Ryzen Threadripper 3990X. Цей 64-ядерний процесор на мікроархітектурі Zen 2 здатний виконати одночасно 128 потокових команд.

У сучасних комп'ютерах північний міст інтегровано у кристал процесора разом із контролером пам'яті та графічним процесором. Однією з основних функцій північного моста є забезпечення взаємодії системної плати і процесора, а також визначення швидкості роботи.

Шина, що зв'язує тільки два пристрої, називається портом.

Шина містить слоти для підключення зовнішніх пристроїв, які в результаті стають частиною шини і можуть обмінюватися даними з усіма іншими підключеними до неї пристроями. Контролер шини здійснює керування процесом обміну даними і службовими сигналами.

Північний міст забезпечує обмін даними з процесором, оперативною пам'яттю і відеопам'яттю. Частота процесора в кілька разів більша, ніж базова частота магістралі.

Центральний процесор

Центральний процесор — це пристрій комп'ютера, який виконує операції з опрацювання даних і керує периферійними пристроями.

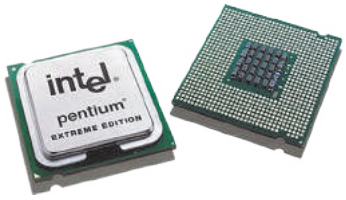


Рис. 3.3. Процесор

Сучасні процесори виготовляються у вигляді мікропроцесорів (рис. 3.3). Фізично мікропроцесор являє собою інтегральну схему — тонку пластинку кристалічного кремнію прямокутної форми площею всього кілька квадратних міліметрів, на якій розміщені схеми, що реалізують всі функції процесора.

Кристал-пластинка зазвичай міститься в пластмасовому або керамічному плоскому корпусі та з'єднується золотими проводками з металевими штирями, щоб його можна було приєднати до системної плати комп'ютера. На рис. 4 наведено схему роботи процесора.

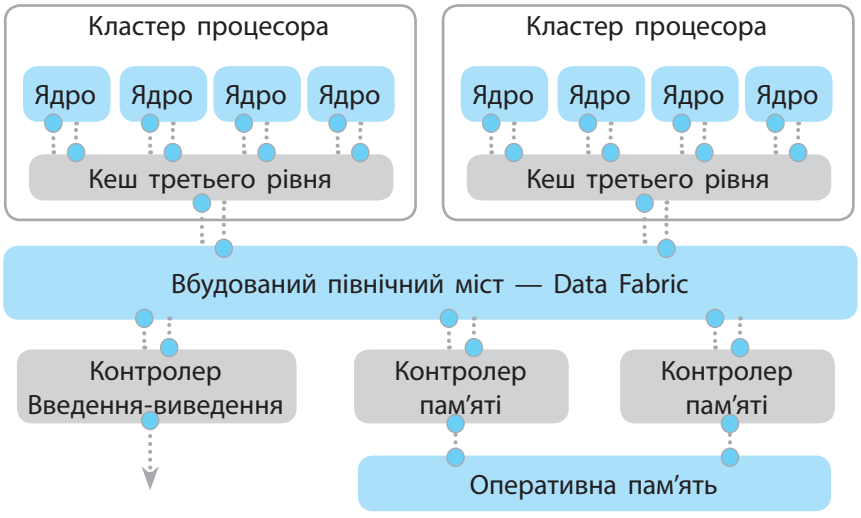


Рис. 3.4. Структура сучасного процесора

Процесор традиційно складається із пристрою управління, арифметико-логічного пристрою та генератора тактових імпульсів. Розглянемо їхню роботу детальніше.

Основні характеристики процесора

- Швидкодія (обчислювальна потужність) — середнє число операцій процесора за 1 с
- Розрядність процесора — максимальна кількість бітів інформації, які можуть опрацьовуватися та передаватися процесором одночасно. Розрядність визначається розрядністю регістрів, у яких розміщуються дані для опрацювання

Більшість процесорів наразі мають розрядність 64 біти. Тактова частота вказує швидкість роботи процесора в герцах (ГГц) — кількість робочих операцій за 1 с. Вона поділяється на внутрішню (визначає темп, з яким процесор опрацьовує внутрішні команди) і зовнішню (визначає, з якою швидкістю процесор звертається до оперативної пам'яті).

Функції процесора

- Опрацювання даних — виконання над ними арифметичних і логічних операцій
- Керування всіма іншими пристроями комп'ютера

Ядро мікропроцесора (*processor core*) — частина мікропроцесора, що містить основні функціональні блоки, або частина процесора, що здійснює виконання одного потоку команд. Таким чином, процесор здатен здійснювати паралельне виконання декількох потоків команд.

- Арифметико-логічний пристрій (АЛП) — призначений для виконання всіх арифметичних і логічних операцій над числовою і символьною інформацією.
- Пристрій управління (ПУ) — формує і подає в усі блоки комп'ютера в потрібні моменти часу певні сигнали управління (керуючі імпульси), обумовлені специфікою виконуваної операції і результатами попередніх операцій; формує адреси осередків пам'яті, використовуваних виконуваною операцією, і передає ці адреси у відповідні блоки комп'ютера; опорну послідовність імпульсів пристрій керування одержує від генератора тактових імпульсів.
- Генератор тактових імпульсів — генерує послідовність електричних імпульсів; частота генеруючих імпульсів визначає тактову частоту машини. Проміжок часу між сусідніми імпульсами визначає час одного такту роботи машини або просто такт роботи комп'ютера.

Типи архітектури процесорів

Ознайомимося з кількома типами архітектури процесорів, які набули найбільшого поширення. Розглянемо таблицю:

Тип	Опис
CISC (англ. *complex instruction set computing*)	Архітектура з повним набором команд. Такі процесори виконують усі команди, прості і складні, за велику кількість тактів. Команд у таких процесорах багато, і компілятори верхнього рівня рідко використовують усі команди
RISC (англ. *reduced instruction set computing*)	Архітектура зі скороченим набором команд. Такі процесори працюють швидше, ніж із CISC-архітектурою, завдяки спрощенню архітектури і скороченню кількості команд, але для виконання складної команди вона складається з набору простих, що збільшує час виконання команди (за більшу кількість тактів)
MISC (англ. *minimal instruction set computing*)	Архітектура з мінімальним набором команд. Такі процесори мають мінімальну кількість команд, усі команди прості і вимагають невеликої кількості тактів на виконання. Але якщо виконуються складні обчислення, наприклад з числами з плаваючою комою, то такі команди виконуються за велику кількість тактів, що перевищує CISC- і RISC-архітектури
VLIW (англ. *very long instruction word* — дуже довга машинна команда)	Архітектура з довгою машинною командою, у якій вказується паралельність виконання обчислень. Такі процесори отримали широке застосування в цифровій обробці сигналів

Запитання для перевірки знань

1. Що таке архітектура комп'ютера?
2. Дайте означення південного та північного мостів, чим відрізняється їхне призначення?
3. Чому материнська плата має таку назву?
4. Що таке такт, чи є він однаковим для різних процесорів? Обґрунтуйте свою думку.
5. Проаналізуйте архітектури процесорів, поміркуйте, в яких саме галузях і яка саме архітектура має переваги.
6. Порівняйте процесори Intel і AMD. Проведіть дослідження, чому саме ці дві фірми є лідерами на ринку виробництва процесорів.

3.2. Пам'ять комп'ютера та її види. Внутрішні та зовнішні запам'ятовуючі пристрої

Пригадайте, що таке пам'ять комп'ютера.

Пам'ять комп'ютера — середовище або функціональна частина комп'ютера, призначена для приймання, зберігання та вибіркового видавання даних.

Розрізняють внутрішню і зовнішню пам'ять (рис. 3.5).

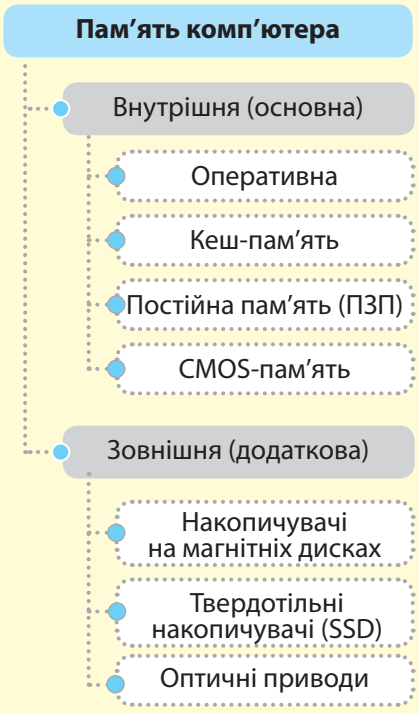


Рис. 3.5. Класифікація пам'яті

Внутрішня пам'ять

Внутрішня пам'ять — це пам'ять, до якої процесор може звернутися безпосередньо в процесі роботи і негайно використовувати її.

Розглянемо функції та основні характеристики цієї пам'яті.

- Оперативна пам'ять (англ. *RAM*, *Random Access Memory* — пам'ять із довільним доступом) — це швидка пам'ять не дуже великого обсягу, безпосередньо пов'язана з процесором і призначена для запису, зчитування і зберігання виконуваних програм і даних, які обробляються цими програмами.

Зазвичай як оперативну пам'ять використовують DRAM — динамічну пам'ять із довільним доступом. Останніми роками стає все поширенішим використання динамічної пам'яті четвертого покоління: DDR4 SDRAM — динамічною пам'ять є тому, що внаслідок малої місткості конденсаторів вона постійно потребує оновлення. Вона є синхронною, оскільки виконує кожну операцію відому кількість часу (або тактів, що дозволяє керувати потоком даних і вибудовувати їх у чергу). Згадаємо, що це пам'ять «із довільним доступом» (RAM), а це означає, що одночасно можна отримати доступ до будь-якої комірки за її адресою на читання або запис, причому завжди за сталий проміжок часу незалежно від її розташування.

- Кеш (англ. *cache*), або надоперативна, пам'ять — дуже швидкий запам'ятовуючий пристрій невеликого обсягу, що використовується при обміні даними між мікропроцесором і оперативною пам'яттю для компенсації різниці у швидкості опрацювання інформації процесором і трохи менш швидкодіючою оперативною пам'яттю.

Кеш-пам'ять реалізується на мікросхемах статичної пам'яті SRAM (Static RAM), на порядок більш швидкодіючих, дорогих і менш ємних, ніж SDRAM, і знаходиться на процесорі. Кеш-пам'ять ділиться на три рівні: I (L1), II (L2) і III (L3). Кожен із рівнів відрізняється за обсягом пам'яті, швидкістю. Причому I рівень — найменший і швидкий, а III рівень — найбільший і повільний. До кожного рівня процесор звертається по

Оперативна пам'ять використовується тільки для тимчасового зберігання даних і програм, оскільки, коли машина вимикається, ми втрачаємо точку входу до даних, які знаходилися в оперативній пам'яті. Доступ до елементів оперативної пам'яті прямий — це означає, що кожен байт пам'яті має індивідуальну адресу.

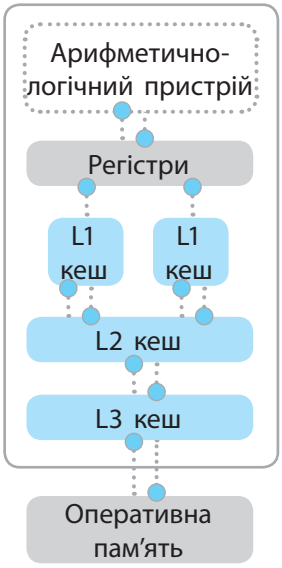


Рис. 3.6. Кеш-пам'ять процесора

У більшості сучасних комп'ютерів BIOS зберігається і на CMOS, і на різновиді флеш-пам'яті EEPROM (англ. *Electrically Erasable Programmable Read-Only Memory* — постійний запам'ятовувач) — електронному незалежному комп'ютерному носієві даних, який можна електрично стирати та перепрограмовувати. Це означає, що один із них доступний лише для читання (резервна копія), а інший — для запису (для запуску нової версії BIOS).

UEFI — це рішення, що підтримує жорсткі диски більшого обсягу, швидше завантажується, значно безпечніше і має графічний інтерфейс і, відповідно, підтримує роботу миші.

черзі (від меншого до більшого): обсяг пам'яті III рівня станом на 2020 рік може сягати 256 Мб, II рівня — від 256 Кб до 8 Мб, I — від 8 Кб до 64 Кб (рис. 3.6).

Кеш-пам'яттю керує спеціальний пристрій — контролер, який, аналізуючи виконувану програму, намагається передбачити, які дані і команди найімовірніше знадобляться найближчим часом процесору, і підкачує їх у кеш-пам'ять. Якщо в кеш підкачати потрібні дані, вилучення їх з пам'яті відбувається без затримки. Якщо ж необхідна інформація в кеші відсутня, то процесор зчитує її безпосередньо з оперативної пам'яті. Співвідношення числа влучень і промахів визначає ефективність кешування.

- Постійна пам'ять (англ. *ROM*, *Read Only Memory* — пам'ять тільки для читання) — незалежна пам'ять, використовується для зберігання даних, які ніколи не вимагатимуть зміни. Зміст пам'яті спеціальним чином зашивається в пристрої при його виготовленні для постійного зберігання. З ROM можна тільки читати.

Тут зберігається програма BIOS (англ. *Base Input Output System* — базова система введення-виведення), яку процесор виконує для перевірки пристроїв під час запуску, завантаження ОС або встановлення нової ОС. Крім того, програма BIOS встановлює потік даних між ОС комп'ютера і приєднаними пристроями (жорстким диском, клавіатурою, мишею, принтером, відеосистемою), управляє споживаною потужністю і живленням. Сучасні комп'ютери використовують замість традиційного BIOS прошивку UEFI (рис. 3.7). Це приклад програмного забезпечення низького рівня, яке запускається під час старту комп'ютера перед тим, як завантажиться ОС.



Рис. 3.7. Вигляд прошивки UEFI BIOS

- CMOS-пам'ять (*Complimentary Metal Oxide Semiconductor Memory*) являє собою пам'ять для зберігання конфігурації комп'ютера. Вона має низьке енергоспоживання і не змінюється при відключенні джерела енергії. Ця пам'ять розташовується на контролері периферії, для електроживлення якого використовуються спеціальні акумулятори.

Зовнішня пам'ять

Зовнішня пам'ять призначена для тривалого зберігання програм і даних, і цілісність її вмісту не залежить від того, ввімкнений або вимкнений комп'ютер. Цей вид пам'яті має великий обсяг і не надто високу швидкодію.

Зовнішня пам'ять не має прямого зв'язку з процесором.

Розрізняють такі види зовнішньої пам'яті: накопичувачі на магнітних дисках, твердотільні накопичувачі, оптичні приводи.

- Жорсткий магнітний диск, або накопичувач на магнітних дисках (*hard* (*magnetic*) *disk drive* (HDD)), складається з декількох намагнічених алюмінієвих дисків зі спеціальним покриттям, на яке за допомогою спеціальної голівки записується і зчитується інформація.

- Твердотільні накопичувачі (*Solid-State Drive*, SSD) — тип сховища, функції якого подібні HDD (рис. 3.9), але засновані на іншій технології. SSD використовують флеш-пам'ять із цифровим доступом, як і USB-накопичувачі. Причина їх появи відображає той факт, що швидкість опрацювання даних у процесорі набагато перевищує швидкість запису даних у HDD.

Жорсткі магнітні диски протягом десятиліть домінували в сегменті зберігання даних, за цей час (з 1950-х років) ємність носіїв зросла у 200 тис. разів, швидкість роботи процесорів теж помітно збільшилась, але швидкість доступу до даних фактично не змінилася, оскільки базується на технології, розробленій ще наприкінці 1970-х років.

Твердотільні накопичувачі забезпечують набагато вищу швидкість роботи з даними порівняно з жорсткими магнітними дисками. Основні характеристики твердотільних накопичувачів такі: малий час доступу до даних: від 100 до 1000 разів швидше, ніж у механічних дисків; висока швидкість, аж до декількох гігабайт за 1 с для довільно розташованих даних; висока надійність, SSD забезпечують рівень збереження даних такий самий, як інші напівпровідникові пристрої.

SSD мають 1 із 4 типів чіпів пам'яті NAND: SLC, MLC, TLC і QLC, кожен з яких може зберігати певну кількість бітів.

SLC (*Single-Level Cells*) здатна зберігати 1 біт інформації.

Пам'ять SLC має високу продуктивність, низьке енергоспоживання, найбільшу швидкість запису і кількість циклів Program/Erase. Пам'ять типу SLC зазвичай використовується в серверах високого рівня, оскільки вартість SSD на основі SLC велика.

Переваги прошивки UEFI:

- підтримує розміри накопичувачів до 9 Пб, працює в 32- або 64-бітному режимі
- забезпечує безпечне завантаження, що запобігає завантаженню комп'ютера несанкціонованими програмами

У липні 2020 р. Асоціація твердотільних технологій JEDEC уклала специфікацію DDR5 SDRAM, а компанія Micron презентувала планку оперативної п'ятого покоління DDR5, яка може використовувати 64-гігабітні чіпи, що збільшує ємність одного модуля до 128 ГБ.

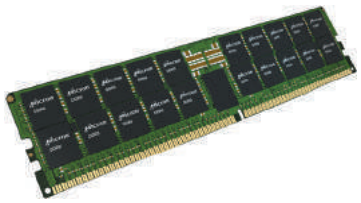


Рис. 3.8. Оперативна пам'ять п'ятого покоління — DDR5 компанії Micron



Рис. 3.9. Твердотільні накопичувачі

MLC (*Multi-Level Cell*) — кожен елемент пам'яті зберігає 2 біти інформації. MLC дешевше від SLC, проте має меншу витривалість і менший ресурс циклів Program/Erase.

TLC (*Triple-Level Cell*) — елемент пам'яті, здатний зберігати 3 біти інформації. Має більшу щільність, але меншу витривалість порівнянно з SLC і MLC, а також відстає від них за швидкістю читання і запису і ресурсу в циклах Program/Erase. Зазвичай TLC NAND використовувалася у флеш-накопичувачах, проте вдосконалення технологій виробництва зробило можливим використання пам'яті TLC і в стандартних SSD.

QLC (*Quad-Level Cell*) — елемент пам'яті, здатний зберігати 4 біти інформації. Наразі є найбільш перспективною технологією, що не поступається за вартістю зберігання 1 Гб даних традиційним жорстким дискам. При цьому за швидкодією і ресурсом на запис QLC-пам'ять лише трохи поступається іншим технологіям NAND.

Взаємодія жорстких дисків із системною платою здійснюється через інтерфейси SATA, PCI, USB, eSATA.

SATA (*Serial ATA*) — розроблений у 2003 р. стандартний інтерфейс підключення жорстких дисків до системної плати. Він використовується для SSD, HDD, а також для CD\DVD\Blue-Ray-пристроїв.

PCI-Express (*Peripheral Component Interconnect Express*) — це засіб взаємодії пристрою з системною платою, що складається з контролера шини і слота. Він є більш швидкісним інтерфейсом підключення пристроїв до системної плати, ніж SATA. SATA SSD майже вчетверо швидше від механічних жорстких дисків при послідовному записі й читанні, а SSD PCI Express перевершують HDD за цим параметром у 7 разів.

- Оптичний привід — пристрій, призначений для зчитування та запису інформації на оптичні носії (диски).

Механічна частина приводу, керована його мікросхемою, забезпечує обертання в ньому диска і зчитування з нього даних за допомогою лазера і системи лінз. Залежно від типів використовуваних носіїв бувають приводи для зчитування/запису CD, DVD і Blu-Ray. Принцип, за яким працюють сучасні оптичні носії інформації, використовується досить давно. По своїй суті CD, DVD і Blu-Ray — вдосконалена вінілова грампластинка. Дані на цих носіях зберігаються у вигляді тонкої спіральної доріжки, нанесеної на спеціальний захищений шар диска, яка складається з мікроскопічних заглиблень і проміжків між ними.

Ученим з університету Саутгемптона вдалося успішно записати інформацію на носій із кварцового скла, використовуючи фемтосекунди і лазерні імпульси для запису даних у 3D-структури кварцу на нанорівні. Імпульси створюють три шари наноструктурованих точок, відстань між якими становить 5 мк. Записана інформація може зберігатися близько 13,8 млрд років за температури до 176 °C і витримувати температуру до 1000 °C.



Рис. 3.10. 360-терабайтний носій

Запитання для перевірки знань

1. Чим відрізняється зовнішня пам'ять від внутрішньої?
2. Що таке постійна пам'ять? Які її функції?
3. Яка роль у роботі процесора кеш-пам'яті?
4. Чому оперативна пам'ять називається енергозалежною?
5. Де має сенс використовувати HDD? A SSD? Наведіть приклади.
6. У минулому десятилітті неможливо було уявити комп'ютер без приводу для оптичних дисків. Поміркуйте, чому сьогодні оптичні носії майже не використовуються. Обґрунтуйте свою думку.

3.3. Будова і алгоритм роботи ЕОМ за фон Нейманом

Що ви знаєте про схему фон Неймана?

Як ви знаєте, принципи, що лежать в основі архітектури ЕОМ, були сформульовані в 1945 р. Джоном фон Нейманом. Йому вдалося узагальнити наукові розробки та відкриття багатьох учених (насамперед Чарльза Беббіджа, котрий уявляв роботу комп'ютера як роботу сукупності пристроїв) та сформулювати на їх основі принципи цього підходу (рис. 3.10).

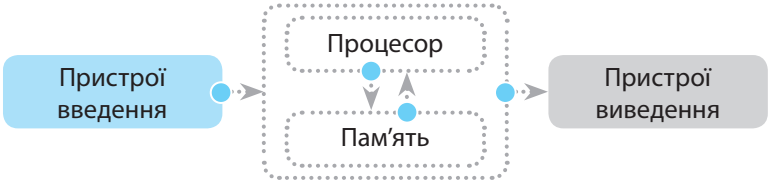


Рис. 3.10. Схема ЕОМ архітектури фон Неймана

Принципи, що лежать в основі архітектури ЕОМ, такі.

- Використання двійкової системи числення. Перевага перед десятковою системою числення полягає в тому, що пристрої можна робити досить простими, арифметичні і логічні операції в двійковій системі числення також виконуються досить просто.
- Принцип однорідності пам'яті. Пам'ять комп'ютера використовується не тільки для зберігання даних, але і програм. При цьому і команди програми, і дані кодуються у двійковій системі числення, тобто спосіб їх запису однаковий. Тому над командами можна виконувати ті самі дії, що і над даними.
- Принцип адресності пам'яті. Основна пам'ять структурно складається з пронумерованих комірок; процесору в довільний момент часу доступна будь-яка комірка. Звідси випливає можливість давати імена областям пам'яті, так, щоб до значень, які в них зберігаються, можна було б згодом звертатися або міняти їх у процесі виконання програми з використанням привласнених імен. Тобто у будь-який момент можна звернутися до будь-якої комірки пам'яті за її адресою.

Цей принцип відкрив можливість використовувати змінні в програмуванні, можливість умовного переходу в процесі виконання програми. Незважаючи на те, що команди виконуються послідовно, у програмах можна реалізувати можливість переходу до будь-якої ділянки коду.

- Принцип послідовного програмного управління. Припускає, що програма складається з набору команд, які виконуються процесором автоматично одна за одною у певній послідовності.



Американський математик Джон фон Нейман сформулював принципи, за якими функціонує більшість сучасних комп'ютерів. Фон Нейман був генієм у тих же областях, що і Леонардо да Вінчі. Він знав багато мов, був фахівцем у фізиці і математиці і мав феноменальну пам'ять.

У нейрокомп'ютерах моделюється робота клітин головного мозку людини, нейронів, а також деяких відділів нервової системи, здатних до обміну сигналами. Поступово народжується новий напрямок, заснований на з'єднанні біологічних нейронів з електронними елементами. За аналогією з Software (програмне забезпечення) і Hardware (електронне апаратне забезпечення) ці розробки отримали найменування Wetware.

Відповідно до принципів фон Неймана комп'ютер складається з арифметико-логічного пристрою — АЛП (англ. *ALU*, *Arithmetic and Logic Unit*), що виконує арифметичні і логічні операції; пристрою управління (ПУ), призначеного для організації виконання програм; запам'ятовуючих пристроїв (ЗП), у т. ч. оперативної пам'яті (ОП) і зовнішнього ЗП; зовнішніх пристроїв для введення-виведення даних.

Програми та дані вводяться в пам'ять з пристрою введення через АЛП. Всі команди програми записуються в сусідні комірки пам'яті, а дані для обробки можуть міститися в довільних комірках.

Найголовнішим наслідком принципів фон Неймана можна назвати те, що тепер програма вже не є постійною частиною машини (як, наприклад, у калькулятора). Програму стало можливо легко змінити. А ось апаратура, звичайно ж, залишається незмінною і дуже простою.

У будь-якій програмі остання команда повинна бути командою завершення роботи.

Команда складається з вказівки, що зазначає, яку операцію з можливих слід виконати, і адрес комірок пам'яті, де зберігаються дані, над якими слід виконати зазначену операцію. А також з адреси осередку, куди слід записати результат (якщо його потрібно зберегти в ЗП).

З АЛП результати виводяться в пам'ять або пристрій виведення. Принципова відмінність між ЗП і пристроєм виведення полягає в тому, що в ЗП дані зберігаються у вигляді, зручному для опрацювання комп'ютером, а на пристрої виведення надходять так, як зручно людині.

ПУ управляє всіма частинами комп'ютера. Від керуючого пристрою на інші пристрої надходять сигнали, що робити, а від інших пристроїв ПУ отримує інформацію про їх стан.

Керуючий пристрій містить спеціальний регістр (*осередок*), який називається лічильником команд. Після завантаження програми і даних у пам'ять у лічильник команд записується адреса першої команди програми. ПУ зчитує з пам'яті вміст комірки пам'яті, адреса якої знаходиться в лічильнику команд, і поміщає його в спеціальний пристрій — регістр команд. ПУ визначає операцію команди, «відзначає» в пам'яті дані, адреси яких вказані в команді, і контролює виконання команди. Операцію виконує АЛП або апаратні засоби комп'ютера.

У результаті виконання будь-якої команди лічильник команд змінюється на одиницю і, отже, вказує на наступну команду програми. Там, де необхідно виконати команду, що не є наступною по порядку за поточною, а відстоїть від даної на якусь кількість адрес, то спеціальна команда переходу містить адресу комірки, куди потрібно передати управління.

Треба підкреслити, що така архітектура, яку ще називають Прінстонською архітектурою, не є єдиною.

Основним осередком квантового комп'ютера є квантовий біт, який скорочено називається кубіт (q-біт). Кубіт — це квантова частинка, що має два базові стани, які позначаються 0 і 1. Двома значеннями квантового біта можуть бути напрямки вниз і вгору спіну ядра атома, два положення електрона в напівпровіднику і т. д. Побудова квантового регістра практично така сама, як і класичного регістра, — це ланцюжок кубітів, з якими можна проводити однобітні та двобітні операції. Проте можливі значення квантового регістра не обмежуються 0 і 1, оскільки, окрім них існують стани суперпозиції, а отже, можна виконувати операції значно швидше. Робоча температура квантового комп'ютера наближається до абсолютного нуля.

Гарвардська архітектура

Гарвардська архітектура була розроблена Говардом Ейкеном наприкінці 1930-х рр. у Гарвардському університеті з метою збільшити швидкість виконання обчислювальних операцій та оптимізувати роботу пам'яті (рис. 3.11).

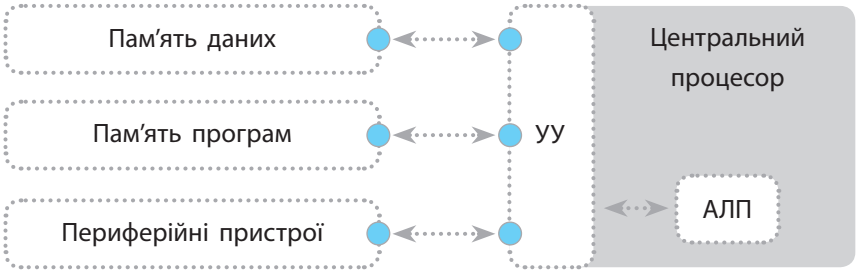


Рис. 3.11. Схема ЕОМ за Гарвардською архітектурою

Гарвардська архітектура широко застосовується у внутрішній структурі сучасних високопродуктивних мікропроцесорів, де використовується окрема кеш-пам'ять для зберігання команд і даних. У той же час у зовнішній структурі більшості мікропроцесорних систем реалізуються принципи архітектури фон Неймана.

Гарвардська архітектура характеризується фізичним поділом пам'яті команд (програм) і пам'яті даних.

В оригінальному варіанті Гарвардської архітектури використовувався також окремий стек для зберігання вмісту програмного лічильника, який забезпечував можливості виконання вкладених підпрограм.

Кожна пам'ять з'єднується з процесором окремою шиною, що дозволяє одночасно із читанням і записом даних при виконанні поточної команди робити вибірку і декодування наступної команди.

Завдяки такому розподілу потоків команд і даних і поєднанню операцій їх вибірки реалізується більш висока продуктивність, ніж при використанні Прінстонської архітектури.

Недоліки Гарвардської архітектури пов'язані з необхідністю проведення більшої кількості шин, а також із фіксованим обсягом пам'яті, виділеної для команд і даних, призначення якої не може оперативно перерозподілятися відповідно до вимог розв'язуваної задачі.

В основі фотонного комп'ютера лежить використання квантів світла — фотонів. Переваги фотонного комп'ютера пов'язані з особливостями поширення світла. Практичною основою фотонного комп'ютера можуть служити фотонні кристали, які здатні пропускати або затримувати світло тільки з певною довжиною хвилі. Зразки фотонних кристалів мають мікронні розміри. Операції будуть проводитися за допомогою згенерованих маленьких лазерів та поширюватись по чіпу за допомогою відбивачів, фотонів.

Запитання для перевірки знань

1. Опишіть роботу комп'ютера, використовуючи схему на рис. 3.11.
2. Назвіть принципи функціонування сучасних комп'ютерів.
3. Поясніть принципи організації роботи комп'ютера за фон Нейманом.
4. Чим відрізняється Прінстонська архітектура від Гарвардської?
5. Які види альтернативних комп'ютерів, на вашу думку, є найбільш перспективними?
6. Знайдіть в інтернеті відомості про ЕОМ, які базувались не на двійковій системі числення.

3.4. Пристрої введення-виведення даних. Пристрої, що входять до складу мультимедійного обладнання

З якими пристроями введення ви вже знайомились на уроках інформатики?

Пригадаємо, що до пристроїв введення належить клавіатури, маніпулятори, сенсорні панелі та екрани, графічні планшети, електронні дошки, ігрові маніпулятори, сканери, камери, мікрофони. До пристроїв виведення належать проєктори, монітори, принтери, навушники, колонки.

Засоби мовного введення дозволяють замість клавіатури, миші та інших пристроїв використовувати мовні команди (або промовляти текст, який повинен бути заздалегідь занесений у пам'ять комп'ютера). Можливості таких пристроїв поки досить обмежені.

Лазерна клавіатура — мініатюрна клавіатура, яка не перевищує розміри брелока, при цьому розміри клавіш не менші за звичайну настільну клавіатуру. Тут використовується лазер, який проєктує зображення клавіатури на будь-яку поверхню.

Монітори

Пригадаємо основні характеристики монітора.

Розмір екрана — визначається довжиною діагоналі (традиційно вимірюється в дюймах).

Роздільна здатність дисплея — кількість пікселів по вертикалі та горизонталі.

Глибина кольору — кількість бітів на кодування одного пікселя.

За принципом роботи існують різні види моніторів (рис. 3.12). Слід зауважити, що електронно-променеві монітори майже не використовуються.

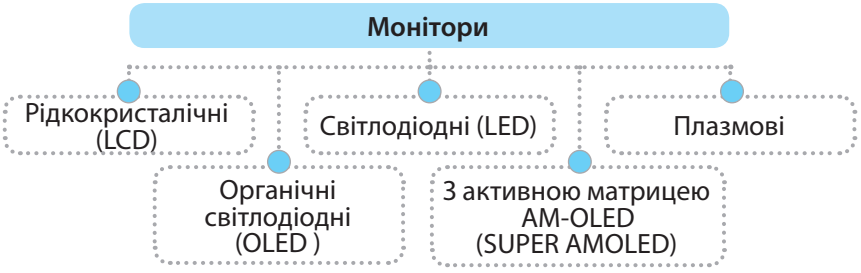


Рис. 3.12. Види моніторів

- Рідкокристалічні (LCD) монітори розроблені на рідких кристалах. Рідкі кристали — особливий стан певних органічних речовин, у якому вони можуть змінювати свою структуру і світлооптичні властивості під дією електричної напруги (рис. 3.13).

Змінюючи за допомогою електричного поля орієнтацію груп кристалів і використовуючи введені в рідкокристалічний розчин речовини, здатні випромінювати світло під дією електричного поля, створюють високоякісні зображення. У рідкокристаличних моніторах використовують тонку плівку з рідких кристалів, поміщену між двома скляними пластинами. Заряди передаються через екран із транзисторів, що забезпечує яскраве і неспотворене зображення. Екран розділений на незалежні осередки, кожен з яких складається із чотирьох частин (три основні кольори і один резервний).

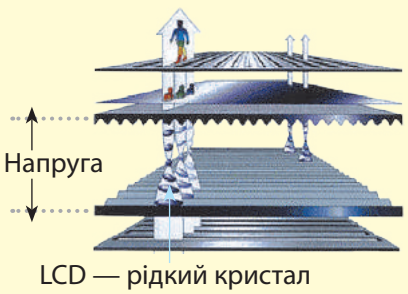


Рис. 3.13. Будова рідкокристалічного монітору

Кількість таких осередків по ширині і висоті екрана називають *роздільною здатністю екрана*.

- Світлодіодні (*Light Emitting Diode,* LED) використовують діоди для підсвічування матриці рідкокристалічного екрана. Можна сказати, що LED-монітор — це відомий нам LCD-монітор, але з використанням підсвічування на світлодіодах (рис. 3.14). Тобто так само, як і в LCD, колір формується завдяки світлофільтрам рідкокристалічної матриці.

Завдяки тому, що світлодіоди є самостійними джерелами світлового випромінювання, вони дозволяють побудувати картинку з максимальною яскравістю і контрастом. Однак у таких моніторів є істотний недолік, це порівняно великий розмір самих світлодіодів. Такі монітори застосовуються в зовнішній рекламі та величезних екранах, на концертах тощо.

- Органічні світлодіоди (англ. *Organic Light Emitting Diode,* OLED) — світлодіоди, у яких електролюмінесценція (випромінювання світла у відповідь на проходження електричного струму) відбувається в пласті органічного напівпровідника, розташованого між двома електродами. Органічні речовини можуть бути молекулами або полімерами.

OLED (AMOLED, Super AMOLED) — технологія створення моніторів для мобільних пристроїв, моніторів і телевізорів (рис. 3.15).

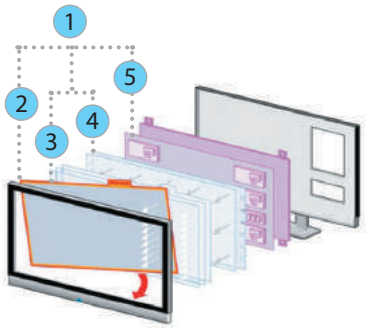
Отже, у OLED-моніторі пікселі випромінюють власне світло, а не підсвічуються, причому в кожному OLED-пікселі є три так званих «мікропікселя», відповідних червоній/зеленій/синій колірній матриці, яка використовується для отримання кольору. Завдяки тому, що дисплеї складаються з декількох органічних плівок, вони можуть бути ультратонкими.

- Монітори з активною матрицею AMOLED і Super AMOLED. За своєю структурою OLED-дисплеї являють своєрідний багатошаровий сендвіч: підкладка та органічний шар затиснутий між двома електропровідними пластинами. Але якщо на підкладку додати шар TFT (*Thin-film transistor* — тонкоплівковий транзистор), то кожним пікселем може керувати окремий транзистор. Тобто можна вмикати і вимикати кожен піксель окремо.

Таке удосконалення назвали активною матрицею (*Active Matrix OLED*), а технологію відповідно AMOLED. Ця технологія використовується в дисплеях мобільних телефонів

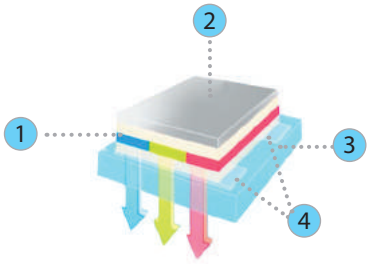
- Плазмові монітори базуються на плазмовій панелі, яка являє собою матрицю газонаповнених комірок, уміщених між двома паралельними скляними пластинами, всередині яких розташовані прозорі електроди, що утворюють шини сканування, підсвічування та адресації. Розряд у газі відбувається між розрядними електродами (сканування та підсвічування) на лицьовому боці екрана й електродом адресації на зворотному боці (рис. 3.16).

Наразі ми не уявляємо собі життя без сенсорних моніторів: згадаємо, наприклад, екран мобільного телефону, плеєра або відеокамери; сенсорний монітор комп'ютера в диспетчерській або в конференц-залі; інформаційно-навігаційний кіоск



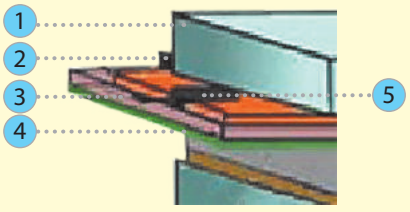
1 — LCD-модуль
2 — РК-матриця
3 — світловод
4 — LED-підсвітка
5 — схема керування

Рис. 3.14. Структура LED-монітора



1 — органічні випромінювачі (OLED)
2 — катод (–)
3 — підкладка
4 — анод (+)

Рис. 3.15. Структура OLED-монітора



1 — зовнішнє скло
2 — живильний електрод
3 — шар діелектрика
4 — захисний шар
5 — скануючий електрод

Рис. 3.16. Будова плазмового монітора

на вулиці чи в торгово-розважальному центрі; банкомат; відеостіна на автошоу або інтерактивний стіл на виставці — всі ці пристрої дозволяють взаємодіяти, приймати рішення і *керувати* складними пристроями за допомогою сенсорів (рис. 3.17).

Слід зауважити, що особливе місце серед моніторів займають монітори-планшети.

Існує низка пристроїв введення. Вебкамера — малорозмірна цифрова відео- або фотокамера, здатна в реальному часі фіксувати відеозображення, призначені для подальшого передавання комп'ютерною мережею. Дігітайзери (графічні планшети) — це ще один пристрій введення графічної інформації. Графічний планшет складається з пера і плоского планшета, чутливого до натиснення або близькості пера.

Введення даних за допомогою світлового пера полягає в дотиках або проведенні ліній пером поверхнею екрана монітора. У наконечнику пера встановлюється фотоелемент, який реєструє зміну яскравості екрана в точці, на якій розташовано перо. За рахунок цього відповідне програмне забезпечення обчислює позицію, «що вказується» пером на екрані, і може залежно від необхідності інтерпретувати її тим чи іншим чином, зазвичай як вказівку на об'єкт, що відображається на екрані, або як команду малювання.



Рис. 3.17. Сенсорний монітор

Принтери

Принтери (*printers*) виконують друк інформації на папері або плівці (результат, який виходить при друку, називають твердою копією).

Ознайомимося з різними видами принтерів та особливостями їхнього друку (рис. 3.18).

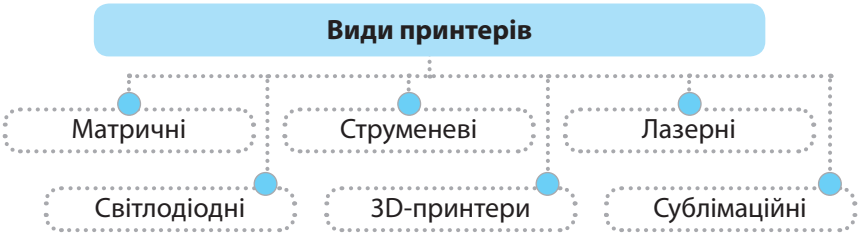


Рис. 3.18. Види принтерів

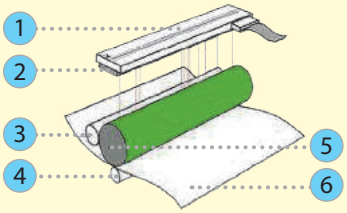


Рис. 3.19. Лазерний принтер

- Лазерний принтер працює за принципом ксерографічного друку, лише зображення формується скануванням фоточутливих елементів принтера променем лазера (рис. 3.19).

Лазерний принтер дозволяє отримати зображення високої якості, яке не боїться вологи і стійке до вицвітання, та складається з картриджа і безпосередньо принтера.

- Світлодіодний принтер (*LED printer*) — паралельна область розвитку лазерних принтерів (рис. 3.20). Швидкість роботи світлодіодних принтерів практично однакова з лазерними. Весь процес друку повністю аналогічний роботі лазерного

1 — світлодіодна лінійка
2 — фокусуючі лінзи
3 — тонерний ролик
4 — ролик перенесення
5 — світлочутливий барабан
6 — надрукований аркуш

Рис. 3.20. Принцип роботи світлодіодного принтера

принтера. Відмінність полягає в способі засвічення фотоциліндра: в світлодіодному принтері для цього застосовують лінійку світлодіодів, яка розташована уздовж всього циліндра.

- Струменевий принтер — такий тип принтера, в якому зображення на носієві формується точковим способом (рис. 3.21). Як робочий елемент використовується рідкий барвник. Відрізняється високою здатністю передачі напівтонів. Чорнило наноситься на папір при проходженні матриці, що друкує. Кольорове зображення формується відразу при одному проході паперу.

- Сублімаційний принтер друкує на щільних поверхнях, вносячи спеціальний барвник під поверхню паперу (рис. 3.22).

Робота сублімаційного принтера заснована на сублімації — переході речовини в газоподібний стан із твердого без рідкої фази. Оптимальне використання принтера цього типу — друк зображень на пластикових картах.

- Матричний принтер — апарат, який формує зображення ударним чином з окремих маленьких точок. Це найстаріший, але досі широко використовуваний тип принтерів.

Високоякісні графічні документи можуть бути створені при використанні графопобудовувачів (*plotters*).

Плотери (рис. 3.23) оснащуються набором пір'я, в який входять рапідографи для малювання ліній різної товщини і різного кольору.

Плотери повільніші за принтери, зате дозволяють отримувати документи великих розмірів — креслення, карти, схеми.

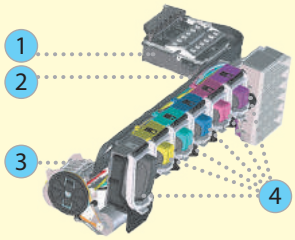
- 3D-принтери — це пристрої для пошарового створення тривимірних об'єктів на основі цифрової тривимірної моделі, параметри якої передаються на процесор пристрою.

Як матеріал зазвичай використовуються кілька видів пластику, хоча останнім часом починають з'являтися й інші матеріали.

Настільний 3D-принтер виглядає як невеликий ящик з металевими направляючими, по яких рухається робочий елемент принтера: екструдер, або лазер (рис. 3.24). Зазвичай 3D-принтери використовуються для створення різних прототипів, ливарних форм і складних деталей, які звичайним чином виготовити неможливо або вкрай важко.

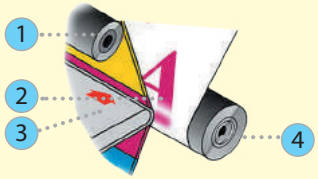
Розглянемо, на яких матеріалах працює обладнання для об'ємного друку.

Порошок: виходячи від параметрів деталі, встановлений на кресленні принтер наносить по периметру підставки сполучну речовину. Поверх неї вкладається порошок, після чого здійснюється спікання. Далі цикл повторюється. За один прохід досягається підйом заготовки на міліметри, тому процес є тривалим, особливо при створенні великих моделей. 3D-принтери можуть працювати з металевою пудрою.



1 — друкуюча голівка
2 — трубки, по яких подаються чорнила
3 — помпа
4 — роздільні картриджі

Рис. 3.21. Струменевий принтер



1 — валик перенесення фарбувальної стрічки
2 — газоподібна фарба
3 — нагрівальний елемент змінної потужності
4 — валик

Рис. 3.22. Сублімаційний принтер



Рис. 3.23. Плотер



Рис. 3.24. 3D-принтер

Гіпс: можна працювати не тільки з гіпсом, а й з різними шпаклівками, цементом. Цим обладнанням користуються для створення статуеток, а також інтер'єрних прикрас. Застосовуючи такий пристрій, можна отримувати твори мистецтва або моделі, необхідні для створення силіконових форм для відливання.

Фотополімер: зроблений у вигляді довгого дроту, накрученого на котушку. Принтер друкує розплавленим пластиком. Надалі він застигає під впливом ультрафіолетового проміння або просто при охолодженні. Такі пристрої використовуються для створення фігурок, шестерень і інших комплектуючих для механізмів.

Віск: 3D-принтери на такому матеріалі застосовуються порівняно рідко. Однак настільки легкі для руйнування моделі є відмінним рішенням при створенні предметів із бронзи шляхом лиття. Воскові вироби поміщаються в шар піску таким чином, щоб залишився тільки отвір для заливки. У нього заливається розплавлений метал. Він спалює віск і займає його місце. Після застигання виходить така сама модель, але вже з бронзи, латуні, золота або алюмінію. Саме таким обладнанням користуються сучасні ливарні майстерні.

Запитання для перевірки знань

1. Які види моніторів наразі популярні? Чим вони відрізняються?
2. Наведіть приклади використання сенсорних моніторів.
3. При викладанні яких саме навчальних предметів під час дистанційного навчання дігітайзер стає незамінним? Наведіть приклади.
4. Чому струменеві принтери зручно використовувати в домашніх умовах?
5. Ударні принтери використовуються і сьогодні. Які властивості роблять їх конкурентоспроможними? Обґрунтуйте свою думку.
6. Які напрямки використання 3D-друку, на вашу думку, будуть найперспективнішими у наступному десятилітті?

3.5. Технічні характеристики складових комп'ютера. Визначення значень властивостей комп'ютера

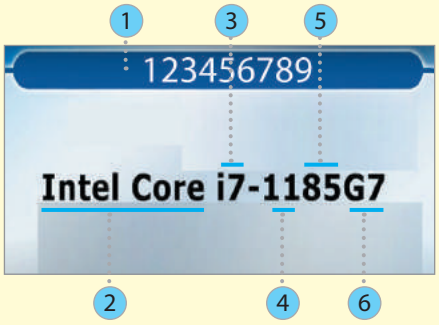
Користувач купив дуже потужний процесор, але не може вставити його у сокет системної плати. Поміркуйте, чому.

Всі ми хоча б один раз стикалися з необхідністю оновлення або купівлі комп'ютера. Щоб правильно вибрати комп'ютер, слід визначити перелік складових, які є суттєвими: системна плата, процесор, оперативна пам'ять, відеокарта та накопичувач пам'яті. Кожна зі складових має власну специфікацію, яку можна проаналізувати, точніше розшифрувати, якщо відомий принцип маркування.

Вибір складових майбутнього комп'ютера починається з вибору процесора. Значну роль відіграє архітектура процесора, кількість ядер, розміри кеш-пам'яті. Наразі на ринку виробництва процесорів є два потужних гравці: компанії Intel і AMD, причому кожна має власні принципи маркування.

Розглянемо спочатку процесори фірми Intel (рис. 3.25). Як бачимо, маркування процесорів включає назву, серію, покоління з SKU-номером (*Stock Keeping Unit* — код товару) і версію.

- Марка. Більшість пристроїв, які використовуються сьогодні для ноутбуків і настільних ПК, мають назву Intel Core. Для робочих станцій і серверів використовуються Intel Xeon. Значно менше поширені такі моделі, як Atom (смартфони і планшети).
- Серія. Основні серії Intel Core — це i3, i5, i7, i9. Фактично номер серії означає продуктивність процесора, тобто чим більший номер, тим вища продуктивність.
- Покоління. На 2021 рік існують процесори 11-ти поколінь.
- Версія. Наприклад, G1-G7 — рівень графіки (лише процесори з новою інтегрованою графічною технологією).

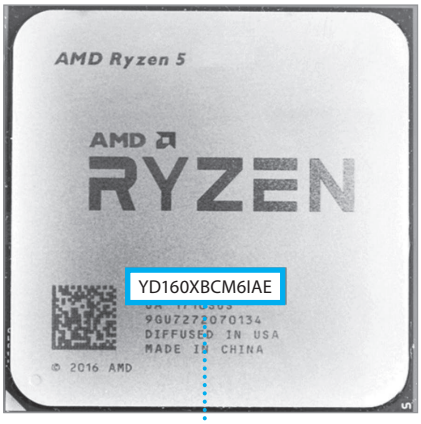


1 — номер процесора,
2 — марка, 3 — серія,
4 — покоління, 5 — код товару,
6 — версія

Рис. 3.25. Приклад маркування процесора фірми Intel

Розглянемо маркування процесорів фірми AMD (рис. 3.26).

Перший символ позначає належність до лінійки сімейства AMD, другий символ — на сферу використання: настільний (D — *Desktop*), серверний (S — *Server*) або мобільний (M — *Mobile*). Три наступні цифри означають базову частоту. Цифра, що йде слідом, вказує на номер моделі конкретного зразка, дві наступні літери — тепловий пакет (вимоги до системи охолодження). Наприклад, A2 або BA вказує на 95 Вт, а AU або BB — на 65 Вт. Далі йде тип корпусу, остання цифра означає кількість фізичних ядер, а наступний символ вказує на конфігурацію кеш-пам'яті.



Y — покоління
D — сфера використання
6 — кількість ядер
AE — конфігурація кеш-пам'яті

Рис. 3.26. Приклад маркування процесора фірми AMD

Швидкість роботи комп'ютера залежить не лише від процесора. Так, у комп'ютерних іграх багато що залежить від відеокарти. Наступним кроком має бути вибір відеокарти.

На сьогодні найбільш впливовими виробниками відеокарт є NVIDIA, AMD, MSI, GIGABYTE, ASUS. Проте всі сучасні відокарти мають графічний сопроцесор. Нині безперечними лідерами є лише дві фірми: NVIDIA і AMD. Ви можете ознайомитися з топ-10 кращих відеокарт 2021 р. за версією сайту 10roar.com/best-graphics-card-brands.

Ознайомимося з характеристиками однієї з кращих відеокарт 2021 року — Gigabyte GeForce GTX 1660 (рис. 3.27).

Для більш детального ознайомлення зі специфікацією сучасних процесорів можна переглянути youtube.com/watch?v=QcxNdXnahOM

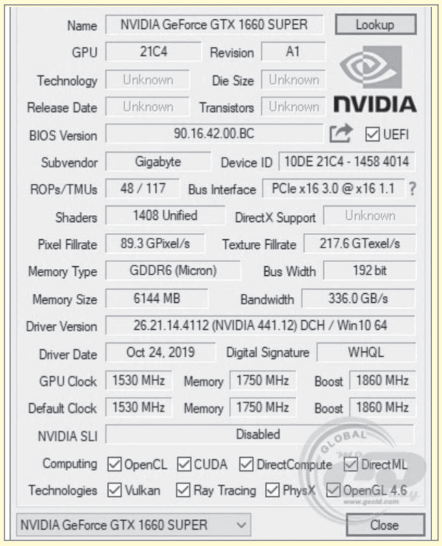


Рис. 3.27. Характеристики відеокарти Gigabyte GeForce GTX 1660

Маркування GTX означає середню і високу продуктивність, а також адаптацію під ігри. Перші дві літери вказують номер покоління (у нашому випадку — 16). В основі відеокарти використовується 12-нм графічний процесор NVIDIA TU116-300 з 1408 CUDA-ядрами, 88 текстурними і 48 растровими блоками. Його базова частота знаходиться на рівні 1530 МГц. Підсистема пам'яті набрана за допомогою GDDR6-мікросхем Micron загальним обсягом 6 Гб. Обмін даними між графічним ядром і пам'яттю здійснюється через 192-бітну шину, яка здатна пропускати 336 Гб інформації за 1 с.

Ви вже детально ознайомилися з властивостями твердотільних накопичувачів і оперативної пам'яті. Для домашнього комп'ютера, який передбачається використовувати для навчання, перегляду відео, серфінгу в інтернеті, легких ігор не потрібно такої продуктивності, як для ігрового. Можна відмовитися від потужної і дороговартісної відеокарти й використовувати відеоадаптер, який має відмінну швидкодію і порівняно низьку ціну.

Основним споживачем оперативної пам'яті є операційна система, на другому місці йде браузер, його синхронізація з іншими пристроями теж вимагає ресурсів.

У відеокарті AMD Radeon Pro SSG буде встановлена SSD-пам'ять обсягом 1 Тб.

Системна плата є об'єднуючим елементом для всіх комплектуючих комп'ютера. Саме від неї залежать швидкість, надійність і стабільність роботи. Крім того, на системній платі розташовуються інтерфейси для підключення зовнішніх пристроїв і мережеві конектори. Швидкість мережі так само залежить від характеристик системної плати. Системну плату доцільно вибирати останньою, оскільки вона підбирається під певний процесор. Її характеристики вже були детально розглянуто раніше.

Запитання для перевірки знань

1. Визначте основні критерії маркування поцесорів. У чому різниця маркування процесорів Intel і AMD?
2. Що впливає на вибір комп'ютера?
3. Яке призначення домашнього комп'ютера?
4. Чи залежить швидкодія комп'ютера виключно від потужності процессора? Відповідь обгрунтуйте.
5. Чому системну плати необхідно вибирати в останню чергу? Відповідь обгрунтуйте.

Завдання для самостійного виконання

1. Знайдіть в інтернеті відомості про сучасні лінійки відеокарт.
2. Створіть інфографіку.

Практична робота № 3

ТЕМА. Конфігурація комп'ютера під потребу

ЗАВДАННЯ: запропонувати варіанти обладнання комп'ютером робочого місця для (виберіть один із запропонованих варіантів): учасника олімпіади з 3D-анімації; учня, який буде навчатись дистанційно; відеоблогера.

ОБЛАДНАННЯ: комп'ютер зі встановленими браузером і текстовим процесором, під'єднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Доберіть за каталогами інтернет-магазинів складові для комплектування комп'ютера з конфігурацією за потребою.
2. Запустіть текстовий процесор і створіть новий документ.
3. Уведіть назви й вартість наведених складових комп'ютера.
4. Визначте складові, від яких залежить відповідність конфігурації комп'ютера зазначеним потребам. Скопіюйте назви, технічні характеристики, посилання на вебсторінки.
5. Запишіть, за якими характеристиками ви вибирали процесор; системну плату; які зі складових не є обов'язковими для конфігурування вашого комп'ютера.
6. Збережіть файл з іменем Практична робота 3 у папці за вказівкою вчителя. Завершіть роботу за комп'ютером.

Зробіть висновок, як визначити конфігурацію комп'ютера під потребу.



3. Складові комп'ютера:
- корпус системного блока з блоком живлення
- системна плата
- процесор; монітор
- оперативна пам'ять
- накопичувач на жорстких магнітних дисках
- SSD (за необхідності) відеоадаптер (якщо не вбудований у системну плату)
- мережева карта
- клавіатура
- миша

3.6. Історія засобів опрацювання інформаційних об'єктів

З якими пристроями для лічби ви вже знайомі?

Розвиток засобів опрацювання інформаційних об'єктів пов'язаний із розвитком людства та пережив декілька етапів.

Перший етап — домеханічний або ручний

Стародавні люди робили зарубки на кістках і каменях, залишали рисочки на стінах житла. Наприклад, знайдена при розкопках стародавнього поселення «Дольні Вестоніці» на південному сході Чехії «вестоніцька кістка» із зарубками (рис. 3.28) являє собою променеву кістку молодого вовка завдовжки близько 17 см з 55 глибокими зарубками.

Перші 25 зарубок були розміщені групами по п'ять. За ними йшла зарубка подвійної довжини, що символізує закінчення цього ряду. Потім із нової зарубки подвійної довжини починався новий ряд зарубок.

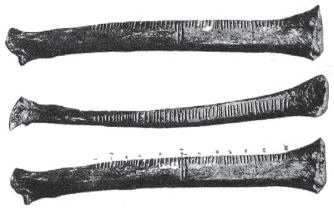


Рис. 3.28. Вестоніцькі кістки



Рис. 3.29. Давньоримський абак



Рис. 3.30. Кіпу, вузликова лічильна система

Креслення Леонардо да Вінчі зберігалися в Національній бібліотеці Іспанії, і лише 13 лютого 1967 року були знайдені американськими дослідниками. Серед креслень вчені виявили ескіз 13-розрядного пристрою з десятизубцовими кільцями для підрахунку сум.

«Годинник, що обчислює» був першою механічною рахунковою машиною, що дозволяла додавати, віднімати, ділити і множити числа.

Їх близький родич — рахункова дошка абак, яка з'явилася в Вавилоні близько п'яти тисяч років тому. Очевидно, що її появі ми зобов'язані бурхливій вавілонській торгівлі. Вавилонський абак використовував шістдесяткову систему числення яка базувалась на пропорціях людського тіла: від числа фаланг пальців на одній руці (не враховуючи великого).

Варіації лічильних дощок були в усіх стародавніх культурах: у Японії — соробан, у Китаї — суньпань. Римляни робили рахівниці з металу (рис. 3.29), пересуваючи кісточки в пазах металевої дошки, а ацтеки — з кукурудзяних зерен.

Кіпу — стародавня лічильна система інків, яка являє собою складні мотузкові сплетення і вузлики з вовни альпаки або з бавовни (на мові *кечуа khipu* означає «вузол», «зав'язувати вузли», «рахунок») (рис. 3.30). В одному кіпу може бути від кількох до 2500 ниток різних кольорів і розмірів. Найдавніший кіпу датується приблизно 3000 роком до н. е. Кіпу був широко поширений в Імперії інків. За словами іспанського хроніста Хосе де Акоста, вся імперія управлялася за допомогою кіпу: рахували число лам, кількість воїнів або зібраного врожаю, вели перепис населення, записували податки; навіть використовувався як календар.

Інки розробили цілу систему зчитування інформації. Кольори шнурів та структура ниток та вузлів містили інформацію про те, що було підраховано та записано. Головний шнур був початком. Що стосується кольорів, то червоний означав армію, військо, білий — срібло, жовтий — золото.

Другий етап — механічний

Історію механічного етапу розвитку обчислювальної техніки можна почати вести з 1492 року, коли Леонардо да Вінчі (1452–1519) розробив креслення лічильної машини і описав його у своїх щоденниках.

У 1850 році французький офіцер Амедіа Маннхейм створив прямокутну логарифмічну лінійку, яка фактично стала прообразом сучасних лінійок і забезпечувала точність до трьох десяткових знаків (рис. 3.31).



Рис. 3.31. Сучасна логарифмічна лінійка

Професор астрономії Грешемського коледжу англієць Едмунд Гюнтер запропонував шкалу з декількох відрізків, розташованих паралельно на дерев'яній або мідній пластині. На кожен відрізок наносилися поділки, відповідні логарифмам чисел або тригонометричних величин. У 1620 році Гюнтер опублікував опис шкали і вона стала пращуром логарифмічної лінійки та багаторазово доопрацьовувалась.

У 1623 році в листі німцю Йогану Кеплеру німецький професор математики і астрономії Вільгельм Шикард (1592–1635) написав про свою машину — шестирозрядний механічний калькулятор, який отримав назву «годинник, що обчислює».

У 1956 році в міській бібліотеці Штутгарта була знайдена фотокопія ескізу годинника Шикарда і в 1960-му група ентузіастів, на основі цієї фотокопії і листів Шикарда, зуміли побудувати діючу модель «годинника, що обчислює» (рис. 3.32).

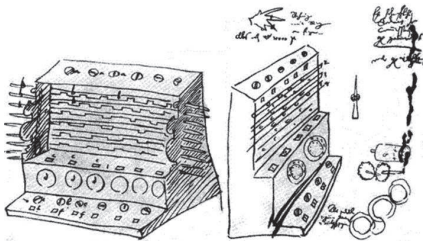


Рис. 3.32. Ескіз годинника Шикарда

Перша механічна обчислювальна машина винайдена геніальним французьким ученим Блезом Паскалем (1623–1662) в 1642 році. Паскаль назвав свій пристрій «Паскаліна». Цей механізм активно використовувався в різних технічних приладах до XX ст. Його перевагою було вміння автоматично складати багатоцифрові числа самим приладом, а принцип пов'язаних коліс майже на 300 років став основою для більшості створюваних обчислювальних машин.

Помітний внесок у розвиток обчислювальної техніки вніс англійський учений Самуель Морленд (рис. 3.33). Лондонська газета в квітні 1666 року писала, що сер Семюель Морленд винайшов два дуже корисних інструменти один служить для додавання і віднімання монет, ваг і мір, інший — для швидкого виконання множення і ділення, а також отримання квадратного і кубічного коренів з наперед заданою точністю.

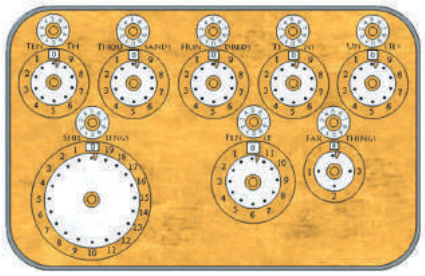


Рис. 3.33. Рахункова машина Морлендаа

Перша обчислювальна машина, яка давала можливість виконувати множення і ділення так само легко, як додавання і віднімання, була винайдена в Німеччині в 1673 році Готфрідом Вільгельмом Лейбніцем (1646–1716) і називалася калькулятором Лейбніца (рис. 3.34). Ідеї, викладені Лейбніцем, мали велику кількість послідовників: Вагнер, Левін, Тоблер, Буркхардт, Кнутцен, Мюллер й ін.



Рис.3.34. Калькулятор Лейбніца

Третій етап — електромеханічний

Коли йдеться про засоби опрацювання інформаційних об'єктів, необхідно навести першу працездатну систему для зберігання і видачі даних по складних запитах, яка була створена в Росії та публічно представлена в 1832 р. Семеном Миколайовичем Корсаковим, який служив у статистичному управлінні Імператорського Міністерства внутрішніх справ у Санкт-Петербурзі. Він запропонував машину, яка б спростила пошук і видачу інформації з картотек.

Інформація зберігалася у вигляді пластин з отворами — перфокарт. Спеціальний механізм дозволяв задати критерій пошуку, за яким машина знаходила потрібний запис. Це була перша обчислювальна машина, яка працювала з множинами. Це також була перша машина для зберігання, пошуку, порівняння і видачі по запитах великих наборів структурованої інформації. Винахід не був підтриманий.

Перший повноцінний аналог машин Корсакова з'явився в кінці XIX ст. в США. Це були електромеханічні табулятори для обробки даних перепису населення. Винахідником табулятора є Герман Голерит (рис. 3.35).

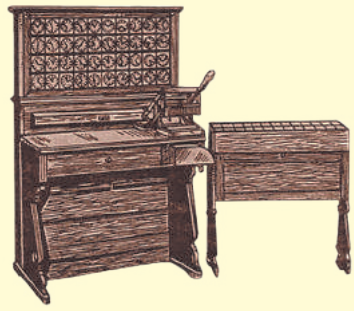


Рис. 3.35. Табулятор Голерита

Четвертий етап — електронний

Неможливо розглядати етапи становлення певних засобів опрацювання інформаційних об'єктів, не спираючись на час, коли вони були винайдені. Потрібно визнати, що зазвичай винаходи мають чітко спрямований воєнний характер. Це особливо яскраво прослідковується у XX ст.

Друга світова війна вплинула на розвиток комп'ютерної техніки. Велика Британія досягла певних успіхів у зломі зашифрованих німецьких переговорів. Код німецької шифрувальної машини «Енігма» (рис. 3.36) був підданий аналізу з допомогою електромеханічних машин, які носили назву «бомби». Така «бомба» була розроблена англійськими математиками Аланом Тюрінгом і Гордоном Велшманом.



Рис. 3.36. Шифрувальна машина «Енігма»

Перші перехоплення передач із таких машин були зафіксовані в 1941 р. Для злому коду в обстановці секретності була створена машина «Колос» (*Colossus*). Специфікацію розробили професор Макс Ньюман і його колеги; «Колос» став першим повністю електронним обчислювальним пристроєм. У ньому використовувалася велика кількість електровакуумних ламп, введення інформації виконувався з перфострічки. Крім Colossus Mk I, було зібрано ще дев'ять моделей Mk II.

Z3 — перша повнофункціональна програмно керована і вільно програмована у двійковому коді з рухомою комою робоча обчислювальна машина, яка мала всі властивості сучасного комп'ютера (рис. 3.38). Вона була створена німецьким інженером Конрадом Цузе (рис. 3.39) у 1941 р. на основі його перших обчислювачів Z1 і Z2. Передбачалося, що новий комп'ютер можна буде використовувати для криптографії і розшифровки закодованих повідомлень. Але єдиний зразок комп'ютера разом з іншими ранніми розробками Цузе був знищений під час нальоту союзницької авіації в 1945 році.



Рис. 3.39. Конрад Цузе

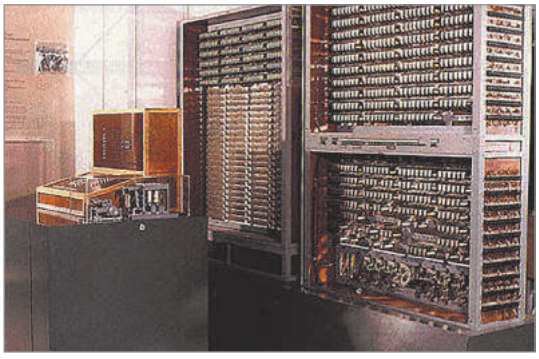


Рис. 3.38. Обчислювальна машина Z3

Друга світова війна вплинула і на розвиток комп'ютерної техніки в США. У 1943 році Джон Моуклі зі своїм студентом Дж. Преспером Екертом почали конструювати електронний комп'ютер, який назвали ENIAC (*Electronic Numerical Integrator and Computer* — електронний цифровий інтегратор і калькулятор) (рис. 3.40).

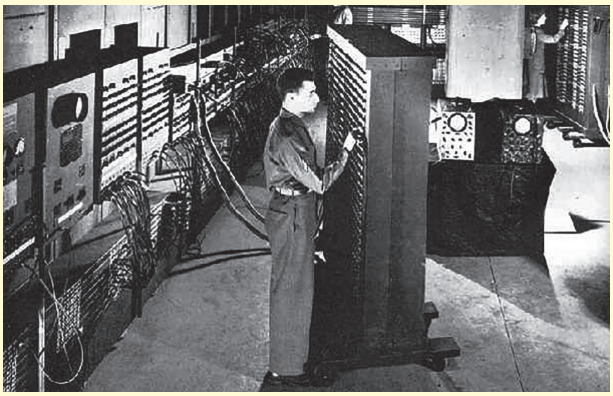




Рис. 3.40. ENIAC3 та його розробники Джон Моушлі та Дж. Преспер Екерт

Покоління електронних обчислювальних машин

Ідея ділити машини з'явилася через те, що за час короткої історії свого розвитку комп'ютерна техніка виконала велику еволюцію як у сенсі елементної бази (лампи, транзистори, мікросхеми та ін.), так і в сенсі зміни її структури, появи нових можливостей, розширення областей застосування і характеру використання.

Більш докладно всі етапи розвитку ЕОМ наведено в таблиці.

Покоління ЕОМ	Роки	Електронні елементи	Швидкодія
I	1950–1960	Електровакуумні лампи (ENIAK, МЕСМ)	10–20 тис. оп./с
II	1960–1965	Транзистори	100–500 тис. оп./с
III	1965–1970	Інтегральні схеми	1 млн оп./с
IV	З 1970	Мікропроцесори	Сотні млн оп./с

16 грудня 1947 року фізик-експериментатор Волтер Браттейн, який працював із фізиком-теоретиком Джоном Бардіном, зібрав перший робочий точковий транзистор, на його зміну прийшли германієві площинні транзистори.

Теоретичні засади у 1948—1950 роках розробив Вільям Шоклі. Згодом кремній витіснив германій, а планарний процес став основною технологією виробництва транзисторів і зробив можливим створення монолітних інтегральних схем.

В історії розвитку обчислювальної техніки якісний стрибок відбувався приблизно кожні 10 років. Такий стрибок пов'язаний із появою нового покоління ЕОМ.

У 1956 році Волтер Браттейн, Джон Бардіно та Шоклі отримали Нобелівські премії з фізики за дослідження напівпровідників і відкриття транзисторного ефекту.

Запитання для перевірки знань

1. Що спільного було в засобах обчислення давніх вавілонян, греків. ацтеків, японців, китайців?
2. Які пристрої розробили Блез Паскаль та Готфрід Лейбніц?
3. Назвіть причини бурхливого розвитку обчислювальної техніки у 1940–1950-х роках.
4. Скільки поколінь ЕОМ ви знаєте? Що обумовило появу кожного нового покоління?
5. Знайдіть в інтернеті відомості про аналітичну машину Беббіджа, Хто співпрацював із Чарльзом Беббіджем?
6. За значеннями яких властивостей можна віднести певну ЕОМ до певного покоління?

3.7. Програмне забезпечення комп'ютера

Назвіть основні складові інформаційної системи.

Пригадаємо, що інформаційна система — це сукупність організаційних і технічних засобів для збереження та опрацювання інформації з метою забезпечення інформаційних потреб користувачів.

Інформаційна система має дві складові: апаратну і програмну, які працюють у неперервній взаємодії. Комплекс технічних засобів і набір програм призначені для її функціонування. Будь-який апаратний пристрій керується програмно.



Рис. 3.41. Класифікація ПЗ

Програмне забезпеченням комп'ютера (ПЗ, Software) — сукупність програм, які призначені для функціонування комп'ютера, узгодження роботи апаратної складової та розв'язування різноманітних завдань.

На рис. 3.41 наведена умовна класифікація, оскільки є програми з властивостями програмного забезпечення інших груп.

Системне ПЗ — це комплекс програм, призначених для забезпечення роботи комп'ютера, узгодження роботи апаратної та програмної складових, розподілу його ресурсів, ефективної роботи інших програм та організації інтерфейсу користувача — методів і засобів керування комп'ютером.

Інструментальне ПЗ — це комплекс програмних засобів для створення інших програм, це середовище програмування, в якому записують коди програм. Результат роботи інструментального ПЗ: компілятори, редактори зв'язків, отладчики, інтегровані системи розробки програмних засобів.

Прикладне ПЗ — це комплекс програм для роботи за комп'ютером у будь-якій сфері діяльності: від виконання завдань для роботи до творчої або розважальної діяльності.

Прикладне та інструментальне ПЗ є великою кількістю програмних засобів для їх використання у будь-якій галузі. Це саме ті програми, які дозволяють користуватись комп'ютером користувачам з різним рівнем знань та умінь.

Оскільки програмне забезпечення виконує функції і керування комп'ютером, і взаємодії з користувачем, то програми класифікують ще в залежності від їх призначення у зв'язку користувач — програма — апаратна складова.



Рис. 3.42. Підпорядкованість ПЗ

Класифікація ПЗ має ієрархічну структуру підпорядкованості — виконання програм кожного наступного рівня базується на виконанні програм попереднього (рис. 3.42).

Базовим ПЗ, тобто програмами базового рівня, є програми базової системи введення-виведення — BIOS (англ. *Basic Input / Output System*), які «прошивають» в мікросхеми постійного запам'ятовуючого пристрою (ПЗП, або ROM — *Read Only Memory*) на етапі їх виготовлення.

BIOS містить набір програм, з яких використовують необхідні для налагодження конфігурації наявних апаратних складових комп'ютера. Після вмикання комп'ютера BIOS тестує апаратну складову та перевіряє його налагодження.

Системне ПЗ за призначенням розподіляють на дві категорії: програми, які забезпечують взаємодію всього програмного забезпечення з програмами базового рівня та апаратною складовою; програми, які відповідають за взаємодію з користувачем. Програмам цього рівня належить операційна система (ОС), яка зберігається на пристроях зовнішньої пам'яті (зазвичай на жорсткому диску).

Службове ПЗ (зазвичай це програми обслуговування) можуть додаватись до складу ОС, але більшість службових програм є для ОС зовнішніми і служать для розширення їх функцій. Програми службового рівня працюють з програмами і базового, і системного рівня.

Прикладне ПЗ призначене для певної проблемної задачі користувача або класу таких завдань (ППЗ часто називають програмним додатком або просто додатком).

Встановлення програмного забезпечення на комп'ютер відбувається за правилами використання певного ПЗ.

Прикладами програмного забезпечення є офісні програмні засоби, системи автоматизованого проєктування, настільні видавничі системи, програмні засоби для створення сайтів, системи аудіо- та відеомонтажу.

Ліцензійна угода (ліцензія) — основний документ, який визначає правила використання програмного забезпечення; права та зобов'язання користувача програмного забезпечення, та яка додається до придбаного продукту у вигляді друкованого або електронного документа.

Невиконання ліцензійної угоди є недотриманням авторських прав розробника ПЗ, що є підставою для розгляду справи в суді.

ПЗ класифікується за правом використання на дві великі підгрупи — *вільне використання* (безкоштовна та відкрита ліцензія), *платне* (комерційна ліцензія) та *умовно-безкоштовне* яке належить до обох груп.

Розглянемо типи ліцензій і їх опис.

Ліцензія	Опис
Безкоштовна	Супроводжує ПЗ, яке в основному поширюються через Інтернет або як додаток до платного комерційного ПЗ. Прикладом є LibreOffice — безкоштовний офісний пакет, повністю сумісний з 32/64-бітними системами, аналог Microsoft Office
Відкрита	Використовується для програмного забезпечення, яке надається на безкоштовній основі разом із вихідним програмним кодом з дотриманням авторських прав і можливістю модифікувати коди співавтором, який придбав програмний засіб. Модифіковане ПЗ розповсюджується на тих же умовах, що і вихідний продукт — не можна перевести в тип платних ліцензій. Відкрита ліцензія має основою міжнародну систему вимог до ліцензій на програмний засіб Open Source Definition (OSD)
Умовно-безкоштовна	Призначена для програмного забезпечення, яке пропонується з обмеженим функціоналом, обмеженими терміном використання або кількістю запусків (так звана пробна версія — trial), поки не відбудеться сплата за повнофункціональний програмний засіб
Платна	Має комерційне програмне забезпечення на основі плати за нього: авансом або договором після отримання копій ПЗ

Комерційне ПЗ поширюється з використанням трьох основних варіантів ліцензування.

OEM (ліцензії при купівлі разом з новим комп'ютером)	Характеризується встановленим ПЗ на комп'ютері і використовувати його можна лише на купленому ПК. Підходить для закупки нового обладнання. Цей вид ліцензії призначений лише для виробників або тих, хто укомплектовує персональні комп'ютери.
Full Package Product (повна упаковка продукту або «коробочний»)	Застосовується в основному для приватних осіб або малого бізнесу. За комп'ютером можуть працювати різні користувачі з різним напрямком ПЗ. Дозволяється змінювати комп'ютер з переустановкою ПЗ визначену кількість разів. Підходить для тих, у кого на наявному комп'ютері відсутнє потрібне ПЗ.
Volume Licensing (об'ємне ліцензування, корпоративна ліцензія)	Компанія отримує одну іменну ліцензію на ПЗ, яка містить інформацію про замовлення (назва, адреса та ін.), ПЗ та ключі для його встановлення. Підходить для компаній з довгостроковим використанням ПЗ.
Subscription (передплата)	Передбачає підписку на ліцензування ПЗ, зазвичай вказується розмір щомісячних або щорічних платежів. Це зручно для компанії, яка купує більше 10 ліцензій або в тому випадку, коли невідомо, чи підійде конкретне ПЗ.

Через стрімкий розвиток комп'ютерної техніки у процесі встановлення програмних засобів може виникнути проблема сумісності різних версій ПЗ. Часто таку проблему вирішують за допомогою правильних налаштувань параметрів сумісності. Для цього в контекстному меню програмного засобу вибирають опцію Виправлення неполадок сумісності.

На екрані з'являється вікно, в якому пропонується вибрати режим діагностики. Також можна звернутися до опцій Використання рекомендованих параметрів, і тоді операційна система зберігає налаштування сумісності автоматизовано.

Запитання для перевірки знань

1. Які основні складові інформаційної системи?
2. Дайте визначення програмному забезпеченню.
3. Перелічте, що входить до складу ПЗ.
4. Що таке ліцензія на ПЗ?
5. Назвіть елементи ієрархічного складу ПЗ.
6. Наведіть приклади використання різних типів ліцензій на ПЗ.

3.8. Класифікація, основні функції та складові операційних систем

Пригадайте структуру ПЗ. Що таке системне ПЗ?

Сучасні ОС характеризуються графічним інтерфейсом користувача, багатозадачністю (можливістю ПК одночасно виконувати кілька програм).

Після вмикання комп'ютера і завантаження BIOS системне програмне забезпечення автоматично завантажується в його основну пам'ять і бере на себе всі функції керування, взаємодія з апаратною складовою відбувається через BIOS. Основою програм системного ПЗ є операційна система (ОС).

Операційна система є комплексом програм, який забезпечує використання ресурсів ПК, взаємодію програм між собою та з апаратною частиною ПК та взаємозв'язок з користувачем.

Програми службового рівня доповнюють функції ОС, тому ми їх розглянемо як такі, що входять до складу ОС. ОС комп'ютера велика за розміром, саме тому в оперативній пам'яті завжди зберігається лише її частина ОС. Решта програм міститься на жорсткому диску і за необхідності завантажується з нього в оперативну пам'ять. Диск, на якому зберігається операційна система, називається системним.

Приклади ОС: MS DOS, IBM PC DOS; MS Windows (98, XP, 7, 8, 10); Mac OS, Mac OS X; OS/2; Linux.

Складові ОС

Ядро ОС	Частина ОС, яка координує розподіл ресурсів ПК для програм, завантажується в оперативну пам'ять при увімкненні ПК і зберігається там увесь час роботи. До складу програм ядра входять планувальник; драйвери пристроїв; мережна підсистема; файлова система. Планувальник ОС розподіляє процес завантаження ОС на частини, які запускаються без втручання користувача
Драйвери	Програми для налагодження та забезпечення керування апаратними пристроями комп'ютера. Програми забезпечують обмін даними між процесором і периферійним обладнанням. Для кожної ОС використовуються її власні драйвери, які реалізовані у вигляді модулів, бібліотек ядра та завантажуються на стадії ініціалізації системи або під час звертання до послуг пристроїв
Системні утиліти	Сервісні програми, які використовуються при роботі або технічному обслуговуванні комп'ютера: діагностичні програми, архіватори, програми для боротьби з вірусами та інші допоміжні програми. Програми полегшують користувачеві взаємодію з комп'ютером. До них приєднуються програми, що забезпечують роботу комп'ютерів в мережі. Вони реалізують мережеві протоколи обміну інформацією між машинами, роботу з розподіленими БД
Операційні оболонки	Інтерфейс для взаємодії користувачів з ОС. Інтерфейс користувача — сукупність засобів ОС і правил, які забезпечують взаємодію користувача з ОС

Існують такі види інтерфейсу користувача:

- графічний — для введення команди ОС за допомогою меню, перемикачів, кнопок, представлених у вигляді графічних елементів (вікон, значків, ярликів, меню, кнопок, списків тощо) — операції можна здійснювати за допомогою маніпулятора миші, сенсорного екрану, килимка, що значно спрощує роботу з комп'ютером;
- текстовий — для введення команди ОС у вигляді набору букв, цифр та символів псевдографіки.

Функції ОС:

- розподіл ресурсів (оперативної пам'яті, процесора, зовнішніх пристроїв);
- планування та організація процесу обробки програм;
- програмна підтримка роботи периферійних пристроїв (дисплея, принтера та ін.) запуск програм на виконання;
- введення/виведення, керування даними;
- допоміжні операції обслуговування;
- діалог з користувачем.

ОС класифікують за різними ознаками (рис. 3.43):

- за кількістю користувачів: *однокористувацька* ОС (обслуговує лише одного користувача); *багатокористувацька* (працює з багатьма користувачами).



Рис. 3.43. Класифікація ОС

У багатокористувацькому режимі кожний користувач починає роботу за комп'ютером під своїм обліковим записом (часто використовують назву на англ. *accoutn*): вводить ім'я (login) та пароль (password);

- за кількістю процесів виконання: *однозадачні* (опрацьовують тільки одну задачу — вже не використовуються: MS DOS, IBM PC DOS); *багатозадачні* (в оперативній пам'яті розміщується одночасно кілька завдань, які почергово опрацьовує процесор);
- за розрядністю: на сьогодні розповсюджені 64-розрядні (необхідно бути уважним при встановленні драйверів периферійних пристроїв);
- за типом інтерфейсу: *текстовий* (інтерфейс командного рядку) або *графічний*.

Для кожного користувача є своє налаштування апаратних і програмних засобів, збереження даних, прав на використання системи. Windows 10 підтримує облікові записи двох типів: Адміністратор і Звичайний користувач

Запитання для перевірки знань

1. Дайте визначення операційній системі.
2. Які програми називаються драйверами?
3. Перелічіть, що входить до складу ОС.
4. Що таке ядро ОС?
5. Перелічіть функції ОС з прикладами.
6. Наведіть приклади ОС за різними ознаками їх класифікації.

3.9. Інсталювання програмного забезпечення. Інсталяція та деінсталяція компонентів операційної системи

Пригадайте, яка частина ПЗ міститься у внутрішній пам'яті комп'ютера, яка зберігається на зовнішніх носіях.

Кожний користувач встановлює на свій комп'ютер те програмне забезпечення, яке йому потрібне. Але є перелік програмних засобів, які присутні практично в усіх ПК. Такими є службові програми, які додатково встановлюють на комп'ютер, наприклад, браузери, антивірусні програми, програмні засоби для створення та опрацювання графічних зображень, звуку, відео та інші.

Дистрибутив містить набір файлів (пакет) програмного засобу, архівованого певним чином, файли з інструкціями для операційної системи (куди і що встановлюється) та безпосередньо інсталятор.

Інсталювання програмного забезпечення — процес його встановлення на комп'ютері.

Більшість програмних засобів розповсюджується на носіях у стиснутій формі — архіві. Встановлення ПЗ виконується спеціальним пакетним менеджером (інсталятором) — стандартною системою, майстром встановлення, який покроково

керує процесом встановлення. Для встановлення програмного засобу необхідно мати його дистрибутив.

Дистрибутив (англ. *distribute* — розповсюджувати) — поширення програми для її інсталяції у вигляді упакованих файлів (або одного файлу).

При запуску інсталятора відбувається копіювання файлів дистрибутива з носія у пам'ять жорсткого диска комп'ютера, перевіряється відповідність комп'ютера вимогам програмного засобу: чи достатньо оперативної пам'яті, місця на жорсткому диску, швидкодії процесора, наявність необхідного додаткового програмного забезпечення, яка версія операційної системи.

Якщо не виникає конфлікту, починається сам процес встановлення програм, при якому майстер установки за допомогою повідомлень та підказок користувачеві здійснює остаточну інсталяцію.

Є програми, які встановлюють без інсталяції (portable-програми), папка з програмним додатком просто копіюється в необхідний комп'ютер. Складні та багатофункціональні додатки вимагають повноцінної інсталяції з дистрибутива.

Грейс Мюррей — американська вчена в галузі комп'ютерних наук і контр-адмірал військово-морських сил США. Вона була однією з перших програмістів комп'ютера Марк I, розробила перший компілятор (комп'ютерна програма, що перетворює (компілює) вихідний код, написаний певною мовою програмування на код, необхідний для виконання програми комп'ютером.



Етапи інсталювання операційної системи

Етап 1	Копіювання файлів операційної системи на жорсткий диск
Етап 2	Визначення конфігурації комп'ютера за допомогою систем автоматичного налаштування
Етап 3	Початкове налаштування всіх програмних модулів ОС. Є модулі, налаштування яких доступні користувачеві для змін
Етап 4	Після завершення початкових налаштувань здійснюється перезавантаження комп'ютера і завершується інсталяція

Програмні засоби можуть мати власну програму інсталяції. Для запуску інсталяції в папці ПЗ потрібно відкрити файл інсталяції. Зазвичай він має назву Setup.exe, або Install.exe (для Windows), Install.sh, або файли з розширенням .deb (для Linux).

Образ диска — це файл, який містить копію ПЗ, наприклад копія дистрибутиву ОС.

Якщо з дистрибутиву відбувається самостійне завантаження ПЗ, то для запуску образу програмного запису необхідно мати за комп'ютером спеціально встановлені програми.

В останніх версіях ОС передбачене створення образів без додаткового програмного забезпечення. Потрібно відкрити у файловому провіднику папку, де зберігається образ ОС і в контекстному меню вибрати Підключити. Для того щоб отримати образ на носії USB є спеціальна утиліта від Microsoft,

У процесі встановлення всі налаштування записуються в спеціальній базі даних операційної системи, так званому «реєстрі». Без записів у реєстрі, програма може не працювати або буде працювати з помилками.

Більшість програмних засобів мають дистрибутиви, поширені як образи ISO.

яка називається Media Creation Tool. Програма проста у виконанні, сама скачає офіційний образ Windows 10 і запише його на флешку.

Пригадаємо, що диск, на якому зберігається операційна система, називається системним.

Для встановлення ОС Windows розглянемо в загальному вигляді алгоритм (вважаємо, що комп'ютер з пустим жорстким диском, а дистрибутив знаходиться на флеш-носії):

Крок 1	Під'єднайте флешку до **USB**-порта, щоб комп'ютер почав завантаження файлів
Крок 2	У вікні **Встановлення Windows** натисніть кнопку **Далі** (якщо параметри «по замовченню» не змінюєте), кнопку **Встановити зараз**
Крок 3	У вікні введіть ключ засобу (ключ може бути на упаковці, диску або отриманий від постачальника ОС). Надалі у вікні працюєте з ліцензією — підтверджуєте Я приймаю умови ліцензії, виберіть тип встановлення **Вибіркова**, виберіть жорсткий диск або розділ для встановлення ОС, натисніть кнопку **Далі**
Крок 4	Відбувається встановлення ОС, за потреби комп'ютер може перезавантажуватись — краще запустіть команду-кнопку **Перезавантажити зараз**
Крок 5	Відбувається базове налагодження конфігурації: налагодіть регіональні параметри, розкладку клавіатури (за потреби встановіть ще одну розкладку або пропустіть крок)
Крок 6	Відбувається налагодження мережі. За потреби, якщо буде запит, виберіть **Налаштувати для особистого використання**, натисніть кнопку **Далі**
Крок 7	Відбувається налагодження облікового запису — процес і вікна залежать від версії ОС. Для використання свого облікового запису, який не є записом Microsoft введіть логін, у відповідь на запит введіть пароль, повторіть його ще раз — Далі
Крок 8	Ознайомтесь з параметрами конфеденціальності, за потреби отримати додаткові відомості натисніть **Детально**. Виберіть функції, які надалі будете використовувати, натисніть **Прийняти**
Крок 9	Завершення встановлення — ОС Windows продовжує процедуру кінцевого налаштування стандартних додатків. По завершенню на екрані **Робочий стіл** — ОС Windows встановлена

При встановленні ОС на комп'ютер вперше жорсткий диск форматують, розбивають на розділи, один із яких і буде системним. При перевстановленні ОС може виникнути потреба в перерозподілі розділів диска.

Інколи виникає потреба вилучити програми з комп'ютера. Оскільки при встановленні є записи в реєстр, системні файли, то при вилученні мають бути знищені всі файли ПЗ в усіх папках та записи вилучення ПЗ за допомогою клавіші видалення може призвести до помилкової роботи решти програм.

Деінсталяція — процес видалення встановлених програм з комп'ютера.

Для деінсталяції користуються програмою деінсталяції конкретної програми або системною утилітою для видалення

програм: для Windows — у вікні Програми та компоненти: Пуск Панель керування Програми та компоненти. У вікні, що відкриється, вибирають необхідну програму та запускають її вилучення. Для Linux — у вікні Центру програмного забезпечення (Панель запуску Центр програмного забезпечення). У вікні, що відкриється, вибирають програму та запускають її вилучення. Інсталювати та деінсталювати програми з комп'ютера може тільки користувач із правами адміністратора.

Запитання для перевірки знань

1. Що таке інсталяція ПЗ?
2. Поясніть значення дистрибутива.
3. Який порядок команд інсталяції програм?
4. Назвіть права користувача для інсталяції та деінсталяції програм за ПК?
5. Поясніть різницю між образом ПЗ та його дистрибутивом.
6. Навіщо для вилучення програм з комп'ютера важливо запускати деінсталяцію?

3.10. Архівування даних. Стиснення даних, види стиснення даних. Архіватори

Пригадайте кодування звукових даних. Навіщо для їх збереження використовується стиснення?

Усі типи даних у комп'ютері мають форму у вигляді двійкових знаків незалежно від їх розміру. Для збереження даних великого розміру, передачі їх електронною поштою, для економії пам'яті на носіях чи у хмарному середовищі.

Стиснення даних — процес перетворення даних файлу в інший вигляд (перекодування) з метою зменшення обсягу їх зберігання.

Існують різні алгоритми стиснення даних, які класифікують за втратами якості даних: без втрат і з втратами.

- Стиснення без втрат дозволяє з отриманих після стиснення даних надалі відтворити початкові дані (розпакувати їх) практично без змін. Якщо початкові дані є важливими на бітовому рівні, використовують саме ці методи стиснення.
- Стиснення з втратами — це перетворення, при якому відтворюються дані, схожі на початкові. Такі алгоритми мають поширення, оскільки значно зменшують обсяг початкових даних. Файли, отримані після стиснення з втратами, мають менший об'єм, ніж при стисненні таких же даних без втрат.

Степінь стиснення залежить від алгоритмів перекодування даних.

Стиснення даних більше використовують для графічних, звукових файлів та відео. Це пояснюється вмістом таких файлів: при перегляді користувач звертає увагу не на бітову комбінацію цифрового зображення (звуку, відео), а сприймає такі дані загалом.

Архівування (упакування) — процес розміщення (завантаження) вихідних файлів в один файл у стиснутому або нестиснутому вигляді. Результат — створення архівного файлу.

Приклади архіваторів: архіви формату ZIP, RAR, (WinRAR — 32-розрядна версія архиватора RAR для Windows, є й інші версії RAR для різних операційних систем: Linux, UNIX, DOS, OS/2). 7-Zip — це програмне забезпечення з відкритим кодом, сумісне з Windows.

Архівний файл є одним файлом, який містить один або кілька файлів у стиснутому або нестиснутому вигляді та службову інформацію про імена файлів, дату та час їх створення або модифікації, розміри файлів на диску та в архіві, степінь стиснення.

Розархівування (розпакування) — процес відновлення файлів з архіву точно в такому вигляді, який вони мали до завантаження в архіві.

Під час розпакування архівів файли, якими вони були в початковому вигляді, розміщуються на диску або в ОП.

Упакування та розпакування файлів здійснюють програмами, що називаються програмами-архіваторами. Вони перекодовують дані зі стисненнями різної степені.

Основними характеристиками таких програмних засобів є швидкодія роботи, степінь стиснення та набір функціональних можливостей. Є архіватори, при запуску яких у користувача є вибір степені стиснення.

Робота з WinRAR може проходити в двох режимах: режимі управління файлами і режимі управління архівами. У режимі управління файлами у вікні WinRAR з виділеними файлами списку виконують різні операції, наприклад, заархівувати їх або видалити. У режимі управління архівами у вікні WinRAR у списку файлів і папок у відкритому архіві виділяють файли і папки і виконують з ними дії, специфічні для архіву, наприклад, розпаковують, тестують, коментують.

Архіватори мають таку функцію, як додавання в архів та відновлення файлів.

Функція попереднього перегляду файлів у режимі «тільки читання» особливо важлива, якщо:

- архів необхідно відновити на комп'ютері без встановленої програми-архіватора;
- часткове або повне відновлення пошкоджених архівів. У деяких випадках архів може виявитися пошкодженим. Як правило, це відбувається при порушенні зв'язку під час сканування даних з інтернету або в результаті дії вредоносних програм. Тому розробники програмного забезпечення попередньо переглянули можливість відновлення файлів — достатньо відзначити в інтерфейсі додатків пункт Залишити на диску пошкоджені файли або вибрати опцію Додати інформацію для відновлення;
- захист даних від модифікацій або несанкціонованого доступу. Створення архіву може супроводжуватись

Більшість сучасних архіваторів також має функцією попереднього перегляду файлів у режимі «тільки читання»: файли, розміщені в архіві, можна переглядати без створення архівів, які саморозпаковуються. Такий архів є запускаємим файлом з розширенням .exe та містить модуль розпакування та заархівовані дані.

встановленням пароля — для успішного розпакування спливає вікно, в яке необхідно ввести пароль;
- створення неперервних архівів. Неперервним називають такий RAR-архів, весь вміст якого розглядається як один потік даних;
- створення резервної копії: з цією метою користуються програмою архівації Microsoft Backup. Запуск програми з меню Пуск: Пуск → Програми → Стандартні → Службові → Архівація даних (маршрут може бути іншим залежно від версії ОС) — відкриється вікно з доступом до використання майстра архівації, відновлення, аварійного відновлення ОС.

Пароль створеного архіву може використовуватись у процесі надсилання архіву з відтермінуванням його розкриття — перед дозволом розкрити надсилають пароль.

Запитання для перевірки знань

1. Що означає термін стиснення; архівування?
2. Поясніть призначення програми архіватора.
3. Для чого призначені архіватори?
4. Наведіть приклади архіваторів.
5. Поясніть різницю між стисненням та архівацією.
6. Поясніть призначення архіву, який сам розпаковується. Чим він відрізняється від звичайного архіву?

Практична робота № 4

ТЕМА. Створення архівів та операції над ними
ЗАВДАННЯ: створити архів і виконати над ним кілька операцій.
ОБЛАДНАННЯ: комп'ютер, папки з файлами для архівування

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.



1. Відкрийте вікно наявного архіватора в комп'ютері.
2. Виділіть файл для додавання в архів.
3. Заархівуйте файл командою Додати. Закрийте вікно.
4. Перейдіть у папку, з якої архівували файл.
5. Перегляньте розмір файлу та цього самого заархівованого файлу. Поясніть різницю.
6. Відкрийте вікно наявного архіватора в комп'ютері.
7. Виберіть папку з неменш як 4 файлами для архівування.
8. Заархівуйте папку. Вікно архіватора не закривайте.
9. Відкрийте заархівовану папку у вікні архіватора.
10. Виділіть один із файлів в архіві. Видаліть його.
11. Виділіть один із файлів в архіві. Скопіюйте його.
12. Виділіть один із файлів в архіві. Перемістіть його.

Зробіть висновок щодо призначення архіватора та його можливостей опрацьовувати архіви.

3.11. Пристрої пам'яті: форматування, копіювання, дефрагментація



Пригадайте, що таке пам'ять комп'ютера.

При встановленні операційної системи на комп'ютер відбувається форматування жорсткого диску з розбиттям його на розділи, один з яких буде містити ОС і називатись системним.

ОС містить команди форматування несистемного диска двох типів: повне та швидке форматування.

Форматування диска — процес програмної розмітки області зберігання даних для будь-яких електронних носіїв інформації.

Навіщо форматувати диск? Для самостійного комплектування апаратної складової ПК; після заміни існуючого жорсткого диска на новий; коли є потреба в переустановці і самостійному налагодженні Windows; можуть бути й інші приклади.

Процес форматування складається з форматування на *низькому* та *високому рівні*.

Якщо сектор виправити неможливо, він залишається з маркером як несправний, і запис даних у такий сектор не відбувається. Після закінчення форматування на диску здійснюється запис таблиці нової файлової системи.

Форматування на низькому рівні відбувається базова розмітка області зберігання даних. Вона виконується на заводі-виробнику і не доступна користувачеві або службам операційної системи. Ця розмітка залишається незмінною весь термін служби диска.

Форматування на високому рівні формує логічні структури диска, які відповідають за правильне зберігання файлів і дозволяють використовувати диск для запису і зберігання даних (створюється файлова система — логічна таблиця, за допомогою якої надалі здійснюється доступ до даних, які будуть зберігатись на диску). Саме цей вид форматування здійснюється службовими програмами ОС.

Розрізняють *повне* і *швидке* форматування диска.

Процес повного форматування повністю очищує жорсткий диск і виконує обнулення файлової таблиці, яка містить інформацію про всі файли — всі дані, які були на диску, стають недоступними і не підлягають відновленню.

Процес діагностики цілісності фізичної поверхні диска полягає у перевірці диска на наявність пошкоджених секторів. Якщо такі є, програма форматування прагне їх виправити, після чого диск стає повністю працездатним весь простір на ньому можна використовувати для зберігання даних.

Якщо скористатись програмами відновлення даних можна відшукати і повторно зберегти практично всі файли, які були на жорсткому диску до початку процесу швидкого форматування.

При швидкому форматуванні видаляються записи про файли та їх місце знаходження на диску і записується нова, порожня файлова система. При цьому дані на диску повністю не знищуються — надалі на місця їх розташування можуть бути записані нові дані, оскільки ці області диска позначаються як такі, що готові до запису.

Може виникнути потреба в редагуванні розмірів дискових розділів (томів). Для створення розділу (тома) на диску має

бути нерозподілений дисковий простір або вільний простір в додатковому розділі. Нерозподілений простір створюють після стиснення або вилучення наявних розділів. При цьому застосовують таку послідовність дій:

Крок 1

Відкрийте в ОС Windows 10 контекстне меню клавіші **Пуск** та запустіть команду **Керування дисками** (є різні дії для відкриття вікна **Керування дисками** в різних версіях ОС) — відкриється вікно. На рис. 3.44 показано дисковий простір без нерозподілених ділянок. Те саме вікно роботи з дисками можна відкрити з вікна **Керування комп'ютером**, яке запускається однойменною командою з **Пуска**.

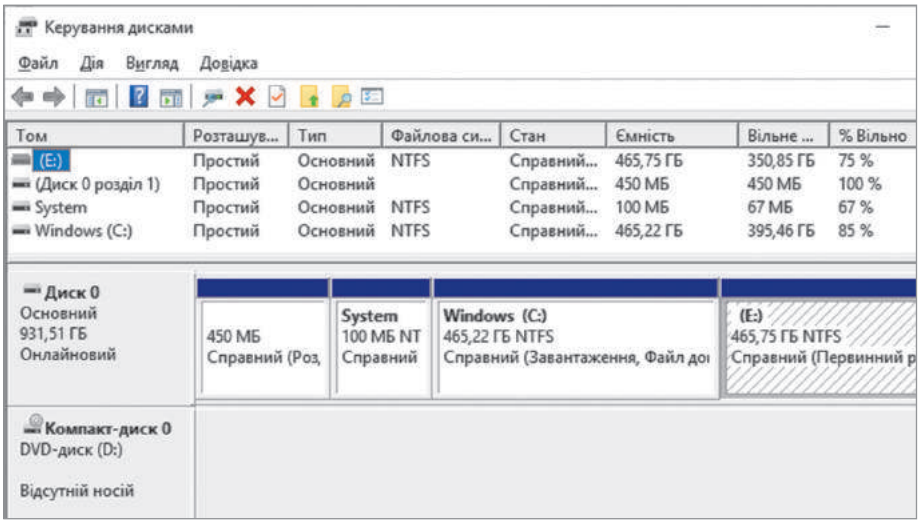


Рис. 3.44. Вигляд вікна **Керування дисками**

Крок 2

У контекстному меню нерозподіленої області диска виберіть команду **Створити простий том** (або відкрийте меню **Дія** і зі списку **Усі завдання** виберіть необхідну команду)

Крок 3

Слідкуйте за вказівками **Майстра створення тома**: введіть розмір тома або підтвердіть запропонований максимальний; для ідентифікації розділу введіть літеру (ми звикли, що розділи жорсткого диска називаються **c:**, **d:**, **e:** — послідовними літерами латинського алфавіту, а можна дати й нові назви, наприклад, **c:**, **m:** або **n:**). Натисніть кнопку **Готово**

Для форматування розділу зі списку Усі завдання вибирають команду Форматувати і у вікні, яке розкриється (рис. 3.45), встановлюють необхідні значення параметрів.

На рис. 3.46 показано вікно форматування розділу e: з наявними параметрами. Видно, що в цьому просторі можна створити новий розділ з введенням його назви (літери латиниці). Форматування розділу може бути недоступним, якщо при його виклику інші програми використовують цей розділ.

Також не форматують системний розділ.

Дефрагментація дисків — процес, при якому здійснюється перезапис фрагментів файлів для їх розміщення в суміжних секторах диска з метою підвищення швидкості зчитування даних із жорсткого диска комп'ютера. Дефрагментацію може здійснювати користувач із правами адміністратора.

Запис файлів на магнітні носії (HDD) здійснюється у вільні сектори. Файли розподіляються на кілька фрагментів, які розміщені в різних частинах диска, оскільки на диску може не вистачати необхідного вільного місця в суміжних секторах. Таке фрагментування записаного файлу уповільнює швидкість зчитування.

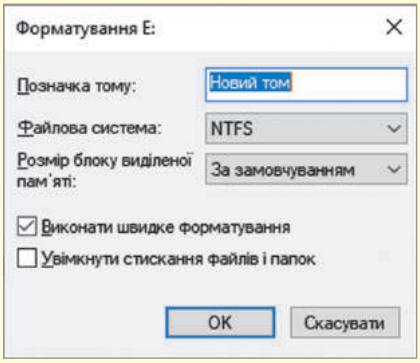


Рис. 3.46. Вигляд вікна форматування розділу жорсткого диска

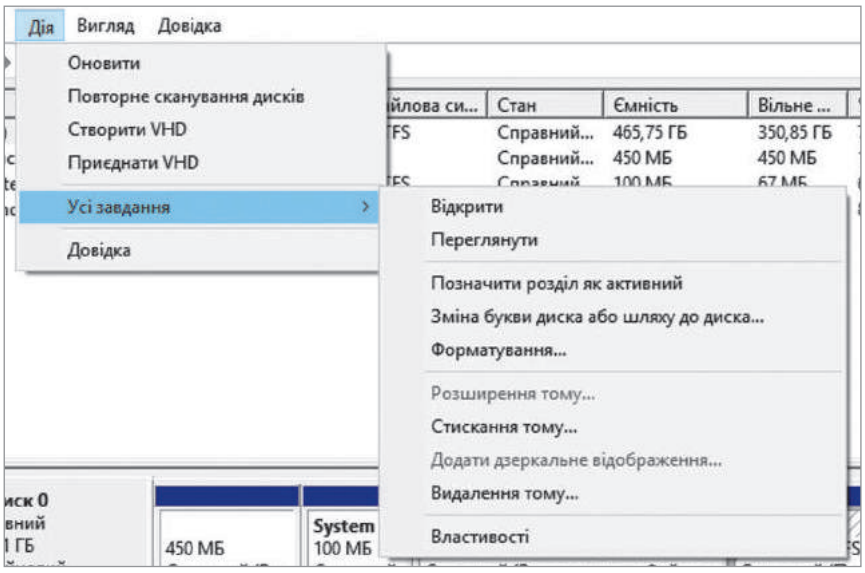


Рис. 3.45. Вигляд вікна з командами списку **Усі завдання**

Під час дефрагментації накопичувач працює більш інтенсивно, що скорочує термін служби SSD.

У твердотільних накопичувачах (SSD) дані зберігаються на чіпах пам'яті, дефрагментація також приведе до впорядкування файлів, але швидкість роботи з файлами не зміниться. У ОС Windows з 8 версії дефрагментація відімкнена. Для перевірки слід звернутися до властивостей диска й у вкладці Сервіс (Tools) вибрати команду Оптимізувати та дефрагментувати диск. Відкриється вікно, в якому буде показано або розклад оптимізації диска, або відімкнення цієї команди.

Запитання для перевірки знань

1. Що означає термін форматування?
2. Поясніть призначення дефрагментації.
3. Охарактеризуйте різні види форматування.
4. Які види дефрагментації ви знаєте?
5. Як створити новий розділ диска?
6. Поясніть призначення оптимізації диска.

Завдання для самостійного виконання

1. Відкрийте контекстне меню клавіші **Пуск** і запустіть команду **Керування дисками**. У вікні переглянте характеристики наявних дисків вашого ПК. Закрийте вікно.
2. Відкрийте контекстне меню клавіші **Пуск** і запустіть команду **Панель керування**.
3. У вікні, що відкриється, виберіть команду **Програми й установки**.
4. У відкритому вікні передивіться програми. Яка програма є «найстарішою» за датою встановлення? А яка «найновішою»? Закрийте вікно перегляду програм.
5. Запустіть команду **Пристої й прінтери** та у вікні передивіться пристрої, під'єднані до вашого ПК.
6. Закрийте всі вікна.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 3

Розділ 4
СТВОРЕННЯ ТА ОПРАЦЮВАННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

4.1. Створення та форматування списків у текстовому документі

Пригадайте означення поняття «текстові дані». З чого складається текст?

Для створення, редагування і форматування текстових документів використовують текстові редактори. Усі вони мають різні можливості і працюють з різними операційними системами. Одним із найпопулярніших серед них є текстовий процесор Microsoft Word (далі Word), який входить до складу стандартного пакета Microsoft Office.

Працюючи з Word, ви вже навчилися виконувати редагування і форматування введеного тексту. Пригадаємо, що редагування передбачає внесення змін до тексту: виправлення помилок, копіювання, вставлення тощо текстових фрагментів, а форматування — надання документу бажаного вигляду шляхом змінення властивостей його об'єктів (рис. 4.1).

Пригадаємо команди форматування об'єктів документа. Щоб вибрати команди форматування символів і абзаців, слід звернутися до вкладок Шрифт і Абзац стрічки Основне, а щоб змінити вигляд сторінки — до команд стрічки Макет. Інші виділені об'єкти форматуються після активування вкладки Формат головного меню — розкривається стрічка Формат.

У межах абзацу можна виконати розрив рядка — примусове переміщення на інший рядок сполученням клавіш Shift + Enter. Розрив використовують в разі вирівнювання абзаца за лівим або правим краєм, за одним або обома краями. У разі вирівнюванні абзацу за обома краями пропуски в рядку можуть розтягнутися на всю ширину абзацу і бути завеликими. Наприклад, не переносяться з рядка на рядок прізвище з ініціалами або рік.

Для форматування кількох абзаців можна скористатися вкладкою Відступи та задати інтервали у вікні Абзац: за рахунок збільшення інтервалу між окремими абзацами (інтервал можна встановити перед і після поточного абзацу) з'являється можливість виокремити деякі абзаци на сторінці.

Коли виникає потреба розташовувати абзаци на одній сторінці (наприклад, в абзаці в один рядок вказано формулу, а далі, в наступному абзаці — пояснення щодо позначень у формулі), слід скористатися командами у вкладці Розташування на сторінці вікна Абзац.

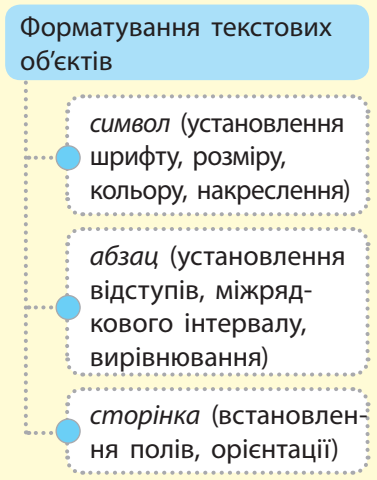


Рис. 4.1. Форматування об'єктів

Команди містяться у відповідній групі або вікнах: у групах подано найчастіше використовувані команди, а у вікнах — додаткові команди на вкладках. Так, командами вкладки **Додатково** вікна **Шрифт** установлюють нестандартні інтервали між символами.

Щоб уникнути розтягування, потрібно поставити нерозтяжний пропуск сполученням трьох клавіш: Ctrl + Shift + пропуск.

Слова через нерозтяжний пропуск сприймаються як одне слово.



Карен Спарк Джон — британська дослідниця, яка розробила технології пошуку, що дозволили користувачам працювати з комп'ютерами, використовуючи слова замість рівнянь та кодів.

Особливим видом форматування кількох абзаців є список.

Список — це сукупність перерахованих абзаців, оформлених у вигляді елементів (пунктів) із позначкою (маркером або номером).

Списки можуть бути впорядкованими і невпорядкованими. Впорядкованим списком є послідовність абзаців у певному порядку (якщо переставити абзаци на інші місця, зміниться логічний порядок списку). Невпорядкованим списком є перелічена послідовність абзаців без необхідності розташовувати абзаци в строго визначеному порядку.

Пригадаємо, що список, у якому кожним елементом є абзац, називають однорівневим. Список, у якому кожний елемент є теж списком, називають багаторівневим.

У Word є можливість створення таких видів списків:

- маркований (однорівневий невпорядкований список); кожен елемент списку — абзац — містить символьну позначку — маркер;
- нумерований (однорівневий упорядкований список) із позначкою у вигляді числа або літери;
- багаторівневий (максимальна кількість рівнів вкладення списків — 9).

Створення та форматування списків відбувається за допомогою команд групи Абзац стрічки Основне (рис. 4.2.)

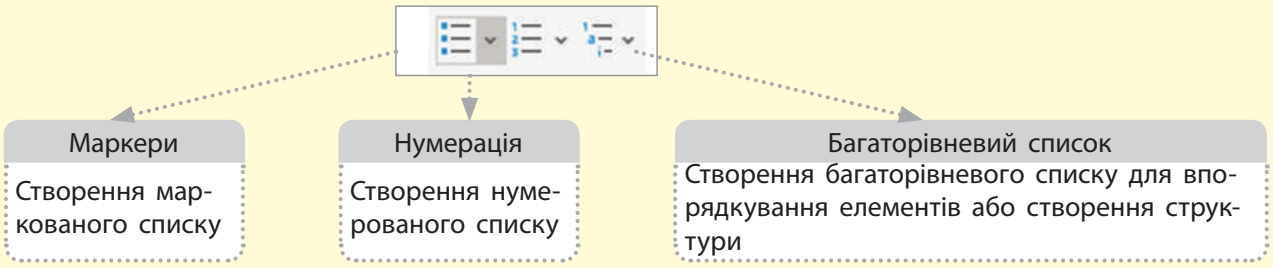


Рис. 4.2. Вигляд команд створення списків на вкладці **Основне**

У Word існує кілька способів для створення однорівневих списків. Розглянемо їх докладніше.

Спосіб 1	Створити абзаци, перетворити їх на елементи списку і форматувати його. Спосіб спрямовано на перетворення кількох вже наявних абзаців на список. Для цього достатньо виділити необхідні абзаци та вибрати список за допомогою команд створення списку в групі **Абзац**
Спосіб 2	Вибрати тип списку, встановити його параметри, почати створювати список. Перед введенням тексту абзацу спочатку вибрати тип однорівневого списку (перед абзацем з'явиться позначка). Після натискання **Enter** як ознаки кінця абзаца наступні абзаци сприйматимуться Word як елементи списку. Для припинення форматування абзаців як списку слід двічі натиснути **Enter**

Можна скористатись і запрограмованими можливостями Word. А саме: перед введенням абзацу проставити певні

символи: * (зірочку) і пропуск або число 1 з «.» (крапкою) і пропуском чи табуляцією (клавіша Tab). Word автоматично відформатує цей абзац у перший пункт маркованого або нумерованого списку, а на початку другого пункту вставить позначку.

Алгоритм вимкнення автоформатування абзацу в елемент списку

Крок 1	Перейдіть у **Файл** → **Параметри** — відкриється вікно **Параметрів**
Крок 2	Виберіть команду **Правопис** і відкрийте однойменне вікно
Крок 3	У вікні **Правопис** активуйте кнопку **Параметри автовиправлення** — відкриється вікно **Автовиправлення**
Крок 4	У вікні **Автовиправлення** перейдіть у вкладку **Автоформат під час введення тексту**
Крок 5	У відкритій вкладці активуйте або зніміть активність необхідних команд (рис. 4.3).

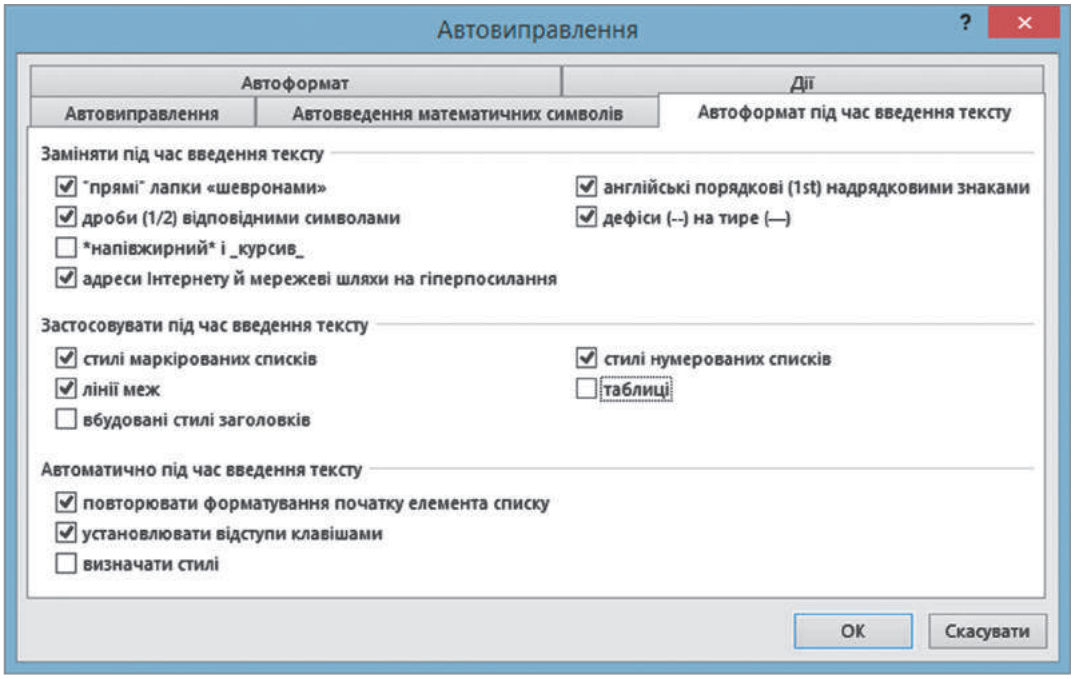


Рис. 4.3. Вигляд вікна автоформатування абзаців у списки

Розглянемо більш детально процес форматування списків.

Якщо активувати стрілку біля однорівневих списків у групі Абзац, то відкриється вікно бібліотеки маркерів нумерованих списків. У цих вікнах слід вибрати позначки.

Для встановлення нового маркера, відсутнього в бібліотеці, потрібно вибрати команду Визначити новий маркер. Після цього відкривається вікно Визначення нового маркера (рис. 4.4), у якому для маркера слід вибрати символ із будь-якого встановленого шрифту — кнопка Символ, графічне зображення з інтернету або наявне на диску вашого комп'ютера — кнопка

Приклад 1.
Створимо маркований список:

Домашні тварини:
- собака
- кіт
- папуга

Для маркерів вибрано символи шрифту Webdings.

Приклад 2.
Створимо нумерований список:

Пори року:
I. Зима
II. Весна
III. Літо
IV. Осінь

Місяці літа:
f) червень
g) липень
h) серпень

Приклад 3.
Створимо багаторівневий список.

Пори року:
I. Зима
  i. грудень
  ii. січень
  iii. лютий
II. Весна
  i. березень ...
III. Літо

Рисунок. У цьому вікні можна відформатувати маркер-символ — кнопка Шрифт та вирівнювання абзаців списку.

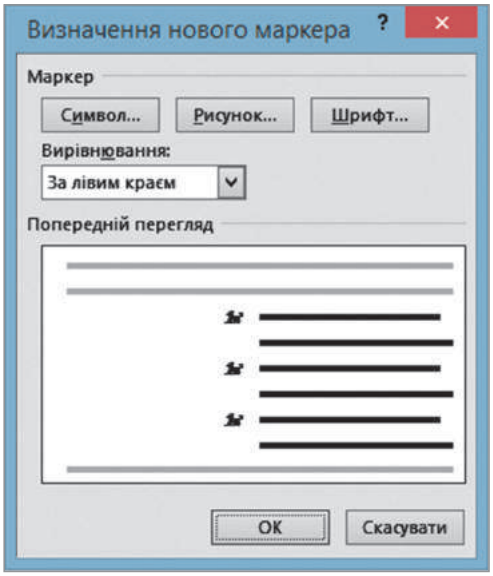


Рис. 4.4. Вигляд вікна налагодження маркованого списку

Аналогічно маркерам вибирають позначки нумерованого списку командою Визначення нового числового формату з бібліотеки нумерованого списку. Встановлення значення номера відбувається у вікні Установити значення нумерації (рис. 4.5).

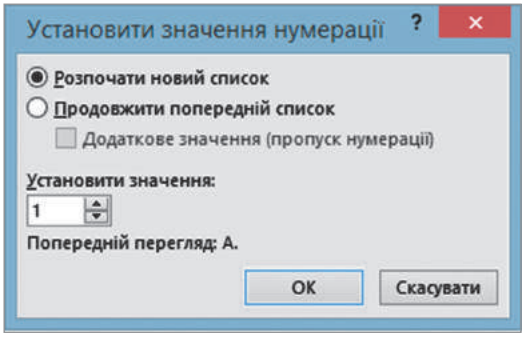


Рис. 4.5. Вигляд вікна встановлення значення списку

З рис. 4.5 видно, що Word надає можливості опрацювання кількох списків, особливо це важливо при копіюванні списків. Якщо після переривання списку звичайними текстовими абзацами є потреба продовжити список, то потрібно вибрати відповідну команду у вікні або розпочати новий список і встановити необхідне значення нумерації.

Для встановлення значення нумерації користуються числами, які зазначають позицію в послідовності вибраного формату номерів. Так, якщо для нумерації вибрано літери абетки, то при значенні 3 нумерація списку почнеться з третьої літери: у вікні передбачено попередній перегляд значення.

У вікнах однорівневих списків міститься команда Змінити рівень списку. Її застосування до виділеного елемента списку зменшує його рівень — так із наявного однорівневого списку можна створювати багаторівневий.

Вікно команди багаторівневого списку містить команди, аналогічні командам однорівневих списків: Змінити рівень списку; Визначити новий багаторівневий список; Визначити новий стиль списку. Ви знайомі лише з першою командою.

Для нового багаторівневого списку відкривається вікно, у якому потрібно налагодити вигляд і позначки списку. Останню команду призначено для створення стилю, який визначить абзаци як елементи списку. (Що таке стилі і як їх опрацьовувати, ви дізнаєтесь далі.)

Для перетворення списку на звичайні абзаци потрібно виділити пункти списку та з бібліотек списків вибрати вигляд позначок немає.

Гарячі клавіші та сполучення клавіш для створення багаторівневого списку:

- Tab — для переходу до списку на нижчий рівень попереднього
- Shift + Tab — для переходу до списку на вищий рівень попереднього

Запитання для перевірки знань

1. Що таке список?
2. Які типи списків пропонує Word?
3. Як абзаци перетворити на список?
4. Опишіть алгоритми створення списків.
5. Які символи автоформатування форматують введений абзац в елемент списку?
6. Назвіть послідовність дій для зняття автоформатування абзаців у списки.

Завдання для самостійного виконання

1. Відкрийте Word і введіть такий текст:

   Етапи вивчення текстового процесора:

2. Уведіть новий елемент нумерованого списку — **1.** пропуск і текст **Структура вікна Word**. — натисніть **Enter**.
3. На кожному новому рядку вводьте:

   Створення нового документа.

   Збереження документа.

   Відкриття документа.

   Форматування тексту.

   Отже, введений текст перетворено на нумерований список.
4. Створіть маркований список за зразком.

   Робота з фрагментами тексту:

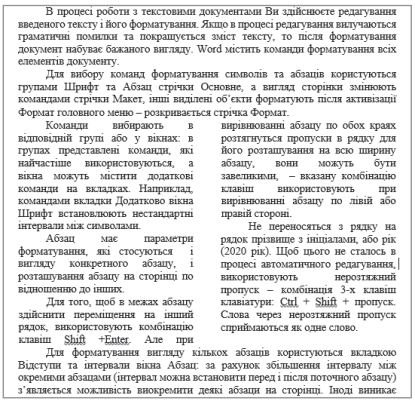
   - виділення;
   - переміщення;
   - копіювання;
   - форматування;
   - видалення.

   Якщо у вас в бібліотеці відсутній запропонований маркер, додайте його відповідною командою вибору нового маркера і виберіть із символів.
5. Створіть багаторівневий список за наведеним зразком.

   1. Форматування сторінки:
      a. Поля
         i. Верхнє.
         ii. Ліве.
         iii. Праве.
         iv. Нижнє.
      b. Орієнтація
         i. Книжкова.
         ii. Альбомна.
   2. Форматування абзаців
      a. …

4.2. Створення та форматування колонок, таблиць у текстовому документі

Пригадайте, як виглядають різні текстові документи. Наведіть приклади.



В процесі роботи з текстовими документами Ви здійсносте редагування введеного тексту і його форматування. Якщо в процесі редагування вилучаються граматичні помилки та покращується зміст тексту, то після форматування документ набуває бажаного вигляду. Word містить команди форматування всіх елементів документу.

Для вибору команд форматування символів та абзаців користуються групами Шрифт та Абзац стрічки Основне, а вигляд сторінки змінюють командами стрічки Макет, інші виділені об'єкти форматують після активізації Формат головного меню – розкривається стрічка Формат.

Команди вибирають в відповідній групі або у вікнах: в групах представлені команди, які найчастіше використовуються, а вікна можуть містити додаткові команди на вкладках. Наприклад, командами вкладки Додатково вікна Шрифт встановлюють нестандартні інтервали між символами.

Абзац має параметри форматування, які стосуються і вигляду конкретного абзацу, і розташування абзацу на сторінці по відношенню до інших.

Для того, щоб в межах абзацу здійснити переміщення на інший рядок, використовують комбінацію клавіш Shift +Enter. Але при вирівнюванні абзацу по обох краях розтягнуться пропуски в рядку для його розташування на всю ширину абзацу, вони можуть бути завеликими, – вказану комбінацію клавіш використовують при вирівнюванні абзацу по лівій або правій сторони.

Не переносяться з рядку на рядок прізвище з ініціалами, або рік (2020 рік). Щоб цього не сталось в процесі автоматичного редагування, використовують нерозтяжний пропуск – комбінація 3-х клавіш клавіатури: Ctrl + Shift + пропуск. Слова через нерозтяжний пропуск сприймаються як одне слово.

Для форматування вигляду кількох абзаців користуються вкладкою Відступи та інтервали вікна Абзац: за рахунок збільшення інтервалу між окремими абзацами (інтервал можна встановити перед і після поточного абзацу) з'являється можливість виокремити деякі абзаци на сторінці. Іноді виникає

Рис. 4.6. Приклад сторінки з колонками

Подивимося, як виглядають сторінки у підручнику. Текст може бути відформатовано в одну або кілька колонок, навіть у межах однієї сторінки. Таке розташування можна побачити у газеті, журналі, на сайті. У випадку задавання різних параметрів форматування для частин одного документа йдеться про розділи (не плутайте з розділом, що є змістовою частиною документа).

Розділ — частина документа з форматуванням, яке відрізняється від форматування решти текстового документа.

У документі є окремі сторінки, на яких текст розміщено в колонках (рис. 4.6), решта параметрів — однакові. Такий документ містить три розділи: розділ до тексту з колонками, розділ тексту з колонками, розділ після колонок до кінця документа. Існує й інший приклад, який стосується орієнтації сторінок (буде два розділи або стільки, скільки разів змінювали орієнтацію).

Щоб створити новий розділ, потрібно застосувати команду Розрив (розрив наявного розділу) зі стрічки Макет.

Для встановлення розриву з переходом на наступну сторінку існує команда Вставлення → Розрив сторінки. Цей вид розриву застосовується в навчальних посібниках, у художніх творах для переходу до іншої змістової теми.

Якщо увімкнути показ недрукованих символів, то на межі розділів побачимо пунктирну лінію з текстом Розрив розділу (вказано вид розриву). Цей недрукований символ розриву розділу за потреби можна вилучити, як вилучаємо будь-які об'єкти тексту. Після цього змінюється форматування або попередньої частини перед символом розділу, або наступної. Створення текстових колонок здійснюють із розривом розділу або без розриву — залежно від вигляду документа.

Колонки тексту — ряд абзаців, розташованих у вертикальні смуги. Зазвичай колонками називають кілька таких смуг.

Види розривів розділу
- Наступна сторінка
- Поточна сторінка
- Парна сторінка
- Непарна сторінка

Алгоритм форматування тексту в колонки

Крок 1	Викличте команду **Макет** → **Стовпці** у групі **Параметри сторінок**
Крок 2	Задайте необхідні параметри: кількість колонок із наявних прикладів або виберіть команду **Більше стовпців** — розкриється вікно **Стовпці** (рис. 4.7)
Крок 3	Вікно **Стовпці** містить наявні в команді **Стовпці** зразки колонок, рахівник **Кількість стовпців** та параметри форматування стовпців. Задайте номер стовпця (колонки), його ширину та проміжок. Автоматично сформатуються стовпці однакової ширини вибором однойменної команди

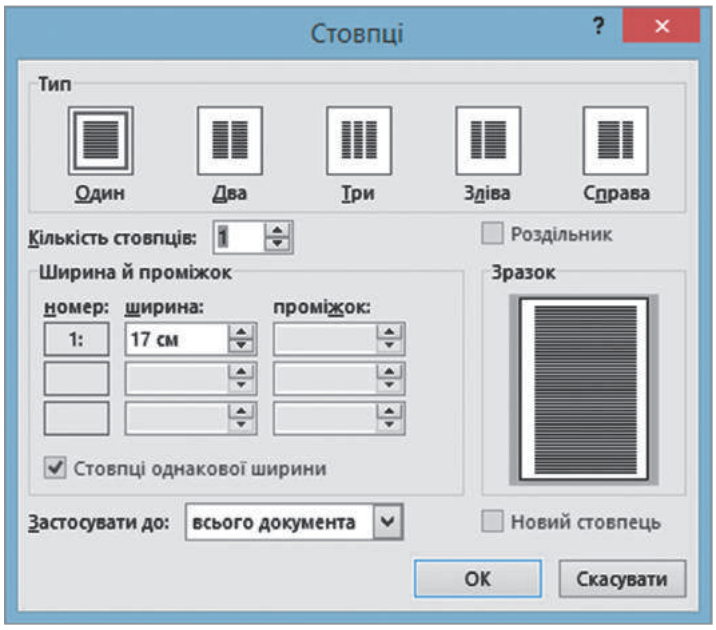


Рис. 4.7. Вигляд вікна **Стовпці**

Форматування тексту в колонки застосовується до всього текстового документа або від поточного розміщення вказівника миші до кінця документа вибором однойменних команд зі списку Застосувати до:, якщо не було виділення тексту.

Якщо для створення або заміни вигляду колонок текст був виділений, в указаному списку з'явиться команда Застосувати до виділеного тексту.

Часто в документах текст необхідно розмістити у вигляді таблиці. Існують кілька способів вставити таблицю в текст. Команди створення та опрацювання таблиць у групі Таблиці стрічки Вставлення наведено на рис. 4.8.



Рис. 4.8. Команди вставлення таблиці в документ

Розглянемо способи створення таблиць: команди, якими можна скористатися, щоб у місці розташування текстового курсора створити таблицю.

- Команда Вставлення таблиць супроводжує двовимірна сітка клітинок. Потрібно вказівником миші вибрати певну кількість рядків і стовпців та створити таким чином таблицю на всю ширину вікна документа. Усі рядки будуть однакові заввишки, а стовпці — завширшки.

У відповідь на виклик команди Вставити таблицю відкривається вікно Вставлення таблиці (рис. 4.9). У ньому треба вибрати необхідну кількість рядків і стовпців та задати, як встановити ширину стовпців: однакового розміру для всіх стовпців, з автошириною, підлагодженою під вміст клітинки та за шириною вікна (подібно до попереднього способу).

- Команда Накреслити таблицю подібна до малювання клітинок таблиці на аркуші олівцем. Якщо в попередньому способі створювалася прямокутна таблиця, то зараз є можливість створити таблицю за виглядом кросворда: з пропущеними в рядках або стовпцях клітинками.

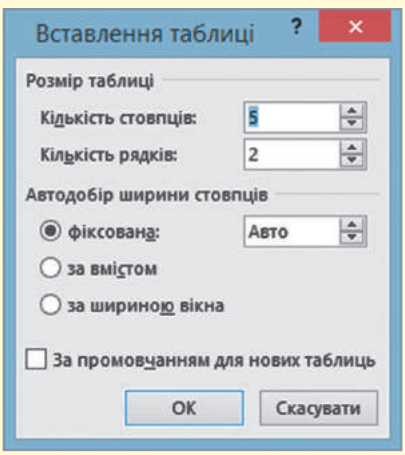


Рис. 4.9. Вигляд вікна **Вставлення таблиці**

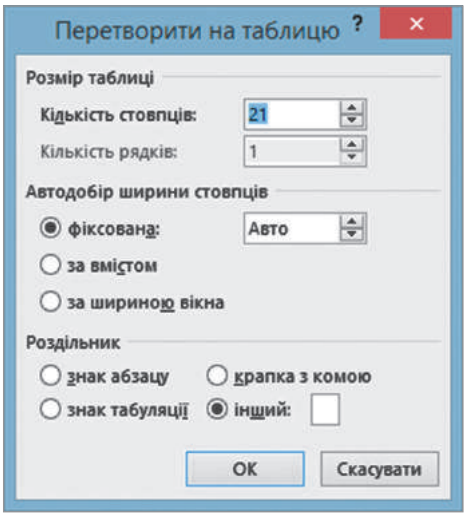


Рис. 4.10. Вигляд вікна **Перетворити на таблицю**

- Команда Перетворити текст дозволяє розмістити виділений текст у створеній таблиці. Вікно Перетворити на таблицю (рис. 4.10) містить ті самі команди, що й вікно Вставлення таблиці, та додаткові команди визначення роздільника.

Роздільник — це символ, який вказує на закінчення тексту клітинки та перехід у наступну клітинку.

У вікні Перетворити на таблицю є найбільш уживані роздільники: знак абзацу, крапка з комою, знак табуляції та пропонується вставити свій роздільник, наприклад знак пропуску. Цей спосіб створення таблиці є доцільним, коли з мережі інтернет копіюється таблиця, а отримуємо текст у колонку по одному слову або фразі в рядку — командою легко такий вигляд тексту перетворити на табличний.

- Команда Електронна таблиця призначена для розміщення в документі Word таблиці Excel. У клітинки цієї таблиці вводяться дані, формули для автоматичного обчислення, як у табличному процесорі.

У таблиці, створеній у Word без допомоги Excel, також можна здійснювати обчислення: формули додаються у вікні, що відкривається викликом команди Формули з групи Дані стрічки Макет (таблиці). Але перерахунок формули при зміненні значення, на яке посилається формула, здійснюється вручну командою оновлення.

Команду **Експрес-таблиці** викликають для створення таблиці в документі за готовим зразком.

Вставлені таблиці мають як межі клітинок неяскраві пунктирні лінії. Під час друку їх не буде видно: текст буде відформатований по клітинках таблиці, але без їхніх меж.

Для налагодження вигляду клітинок і всієї таблиці використовують команди стрічок Конструктор таблиць і Макет. Їх назви з'являються в головному меню, коли текстовий вказівник перебуває в межах таблиці або є виділення частини таблиці, її в цілому.

Стрічка Конструктор таблиць містить команди для налагодження вигляду таблиці за готовими зразками (стилями) або самостійно: вибрати вигляд клітинок (із заповненням чи без нього), вказати, які будуть межі таблиці, як вони виглядатимуть за кольором, товщиною, стилем.

Макет містить команди форматування складових таблиць: висоти рядків та ширини стовпців, напрямку тексту в клітинках, формату клітинок.

Ознайомимося з ними детальніше.

- Клітинку можна розподілити на рядки та стовпці (отримаємо таблицю в таблиці) — це здійснюється командою Розділити клітинки, і в діалоговому вікні зазначають, на яку кількість рядків та стовпців здійснюють розділення.

Різні види розривів застосовуються в підручниках та посібниках: нові тематичні розділи зазвичай починаються з нової сторінки.



- Кілька клітинок інколи об'єднують в одну командою Об'єднати клітинки, наприклад клітинки верхнього рядка об'єднують для назви таблиці.
- Клітинки рядка прибирають командою Розділити таблицю для створення вигляду однієї таблиці як двох незалежних.

Для додавання та вилучення рядків або стовпців вміщено відповідні команди на стрічці Макет. Вона містить команди креслення таблиць і команду Властивості, за викликом якої відкривається вікно — тут налагоджують параметри, що містяться на стрічці конструктора таблиць.

Елементи створеної таблиці можна виділити мишею.

- Якщо навести вказівник миші в область клітинки, над стовпцями або ліворуч від рядків, змінюється вигляд вказівника на стрілочку.
- Якщо клацнути лівою клавішею миші, виділяється елемент: клітинка, стовпець або рядок.
- Якщо затиснути клавішу і протягнути вказівник, одразу виділяться кілька елементів. Застосуванням команд до виділених елементів переформатовують їхній вигляд.

Змінити ширину стовпця вручну можна наведенням вказівника миші на межу між стовпцями (вказівник змінює вигляд) і її перетягуванням — загальна ширина таблиці не змінюється, змінюється ширина стовпців зі спільною межею.

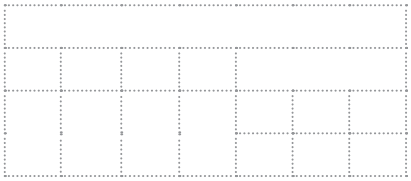
Змінити ширину одного стовпця зі зміною загальної ширини таблиці можна перетягуванням маркерів меж стовпців, що розташовані на горизонтальній лінійці над робочою зоною Word. Аналогічно можна змінити висоту рядків.

У клітинках таблиці розташовується текст у потрібному користувачеві об'ємі. Маркери параметрів абзацу відображаються в межах клітинки на лінійці.

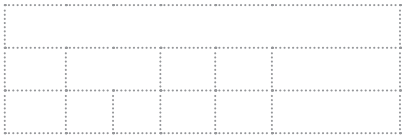
Перехід між рядками і клітинками в межах рядка здійснюється відповідними клавішами на клавіатурі (зі стрілками вгору, вниз) за допомогою клавіші Tab. Якщо вказівник знаходиться в останній клітинці таблиці (нижній правій), рядок у таблицю додається клавішею Tab.

Приклади форматування таблиць

- Таблиця з розділенням та об'єднанням клітинок



- Таблиця з розділенням клітинок та форматованим виглядом меж цих клітинок



- Таблиця з ненульовим значенням інтервалу між клітинками



Запитання для перевірки знань

1. Що таке розділ у текстовому документі?
2. Як ви розумієте поняття **Розрив розділу**?
3. Як відформатувати текст у кілька колонок?
4. Як створити таблицю в текстовому документі?
5. Які символи автоформатування зазвичай використовують при перетворенні тексту в таблицю?
6. Чи можливе перетворення формату таблиці на звичайний текст?

4.3. Стильове оформлення документів



Зверніть увагу на підручники, які ви використовуєте у навчанні. Як відформатовано окремі складові підручника?

Опановуючи можливості текстового процесора, ви створювали невеликі за розміром документи та за потреби виділяли фрагменти й послідовно змінювали один за одним їхні параметри форматування. Такий підхід є незручним для роботи з великими документами та документами, фрагменти яких відрізняються за форматуванням.

Напевне, ви звертали увагу на оформлення підручників: заголовки, підзаголовки, основний та додатковий текст відрізняються форматуванням. Але стиль форматування є однаковим для кожної групи в підручнику загалом.

Стиль — це набір параметрів форматування, який має назву й визначає вигляд текстового або табличного фрагмента документа і встановлюється одночасно для автоматичної зміни всіх параметрів вигляду поточного фрагмента.

Бібліотека стилів міститься в групі Стилі стрічки Основне.

Після відкриття Word пропонуються зразки для створення документа певного вигляду (наприклад, буклета, запрошення або листівки), а при виборі команди створення нового документа — встановлені за замовчуванням параметри форматування на основі шаблона.

Шаблон (від фр. *echantillon*) — зразковий документ, який зберігається у файлі з розширенням .dotx.

Стилі форматування тексту, встановлені в шаблоні за замовчуванням, використовуються у відкритому документі. Шаблонний документ при цьому залишається незмінним, як і його місце розташування на диску.

Шаблоном може бути як зразок документа спеціального призначення (бланки, які надалі заповнюють), так і звичайний текстовий документ із різним форматуванням певних його частин (книжки або статті з виокремленням заголовків частин).

Наявний шаблон можна переглянути та змінити у вікні **Шаблони та Надбудови**. Вікно викликається:

- командою **Шаблони**, яка розташована на стрічці **Розробник** (зазвичай при встановлення офісного пакета стрічка відсутня, її додають у вікні **Файл** → **Параметри** → → **Налаштування стрічки** — в розділі **Основні вкладки** необхідно встановленням галочки додати **Розробник**);
- командою **Файл** → **Параметри** → **Надбудови** — у списку **Керування** вибирають шаблони.

Шаблони більшості стандартних документів містяться в папці Microsoft Office, а також їх можна знайти на офіційному сайті office.com

Шаблон можна редагувати. Для збереження нового шаблона у вікні збереження потрібно вибрати тип файлу .dotx, при цьому краще скористатись командою Зберегти як, щоб залишити незмінним початковий шаблон.

Для створення однотипних документів або частин багатосторінкового документа різними користувачами потрібно зробити шаблон документа. Таким чином, користувач відкриває файл шаблона, працює з цим файлом, а потім зберігає свою роботу командою Зберегти як… із зазначенням типу документа як документ Word (.docx).

Зазвичай для роботи з документами автоматично встановлюється шаблон Звичайний, заснований на однойменному

стилі. Параметри форматування на основі стилів переглядають у панелі Стилі (рис. 4.11), яку відкривають з однойменної групи на стрічці Основне.

Панель містить список назв стилів. Після наведення вказівника миші на певний стиль назва буде показана в рамці і біля неї з'являється стрілка — маркер розкривного списку. Наведенням вказівника миші на маркер розкривається вікно з параметрами стилю, а після клацання по маркеру — список команд для роботи з вибраним стилем.

Параметри стилю вибраного фрагмента можна не лише переглядати. Стилі використовують для форматування вибраного фрагмента (групи символів, абзаців і таблиць), для заміни параметрів форматування фрагмента та ін. Опрацювання стилів передбачає заміну параметрів наявного стилю або створення нового. Можна також створити свій варіант стилю.

Стилі зручно використовувати для багаторазового однакового форматування фрагментів документа. Алгоритм дуже простий і короткий: виділіть потрібний для форматування фрагмент тексту; на стрічці Основне в групі Стилі стиль вибраного фрагмента буде виділено в рамочку; виберіть новий стиль — клацніть вказівником миші

Якщо необхідний стиль не відображено у видимій області групи Стилі, то слід скористатися вікном Додатково (вікно відкривається стрілкою з рисочкою) групи Стиль. У відповідь розкривається вікно з переліком стилів і вказано поточний стиль (рис. 4.12) — у вікні Перелік стилів, як і на панелі стилів.

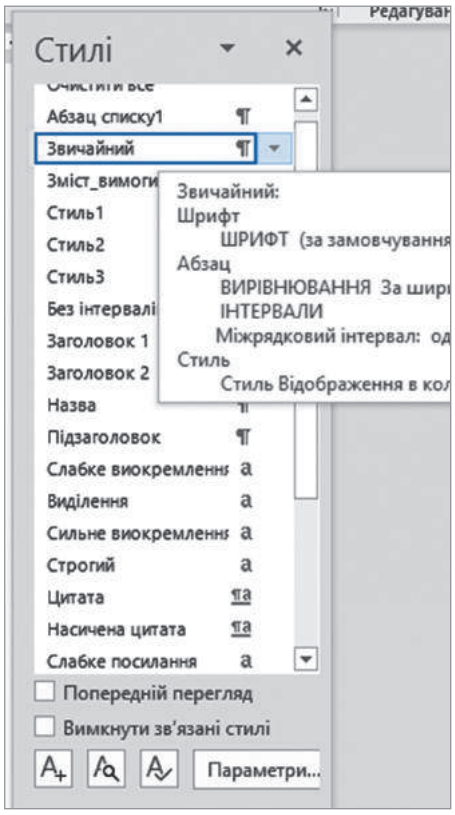


Рис. 4.11. Вигляд панелі **Стилі**

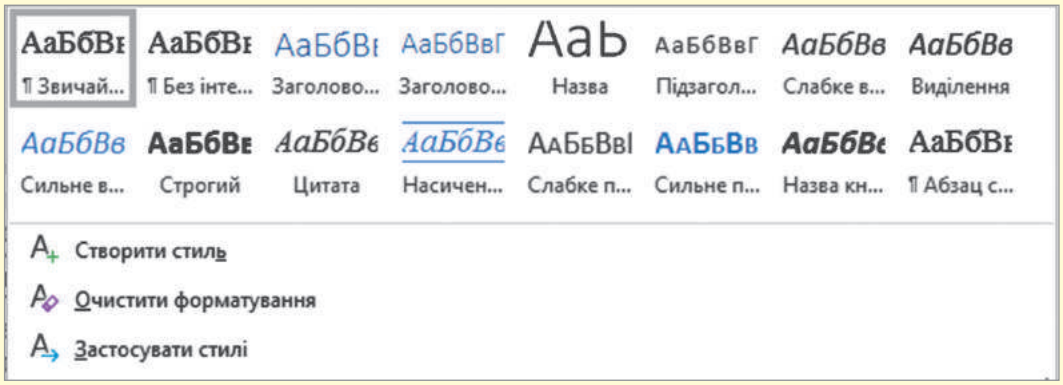


Рис. 4.12. Набір стилів групи **Стиль**

Розглянемо алгоритм змінення параметрів наявного стилю, наприклад Звичайного на інший. Виділіть фрагмент — абзац. Застосуйте до нього форматування шрифту та абзацу за допомогою команд відповідних груп та вікон. Виділення не знімайте. Відкрийте панель Стилі, виберіть стиль Звичайний та вказівником миші перейдіть до списку команд опрацювання стилю. Виберіть команду Оновити Звичайний (або інший — буде вказано назву обраного стилю) відповідно до виділеного фрагмента.

Стиль із новим форматуванням буде запам'ятовано в шаблон, якщо вибрати команду **Нові документи** на основі цього шаблона.

Можна відразу перейти до групи стилів (стрічка Основне), поточний стиль буде взято в рамочку. У його контекстному меню виберіть команду Оновити Звичайний (буде вказано назву стилю) відповідно до виділеного фрагмента.

Цей алгоритм змінює параметри стилю для роботи в поточному документі. Але Word містить команди заміни параметрів стилю для роботи в усіх наступних сеансах у вікні Зміна стилю (рис. 4.13) після виклику команди з панелі стилів.

Для створення власного стилю користуються командою Додати до колекції стилів → Створити стиль. У вікні заміни стилів потрібно вибрати команди вибору документів, у яких можна застосувати новий стиль: тільки в поточному, або в усіх бо стиль буде додано до колекції стилів, або в усіх нових з автоматичним застосуванням параметрів стилю.

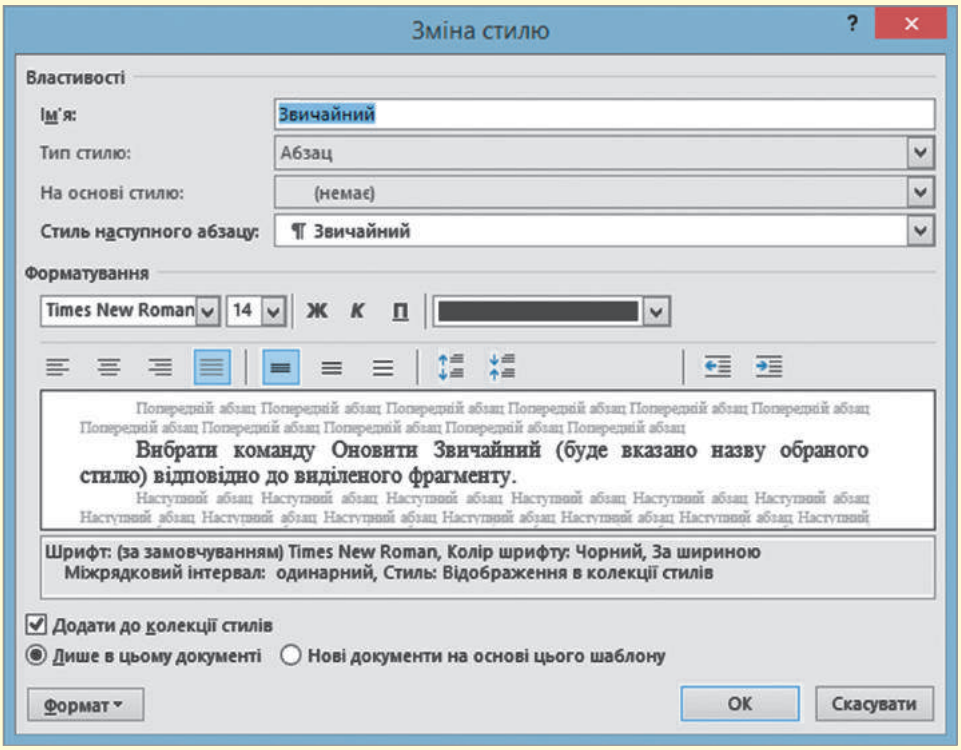


Рис. 4.13. Вигляд вікна **Зміна стилю**

Зверніть увагу, що команду застосування іншого стилю на Звичайний можна відмінити командою Очистити форматування вікна Додатково до виділеного фрагмента.

Запитання для перевірки знань

1. Дайте означення стилю.
2. Що таке шаблон?
3. Чим відрізняється покрокове форматування від форматування з використанням стилів?
4. Які дії можна застосувати до стилю?
5. Як можна використовувати власний створений стиль у сеансах роботи з Word?
6. Що спільне і чим відрізняється результат роботи команди копіювання формату від застосування стилів?

Завдання для самостійного виконання

1. Завантажте у Word будь-який документ.
2. Виділіть абзац.
3. Запустіть команду **Додатково** з групи **Стилі** стрічки **Основне**.
4. У вікні виберіть і запустіть команду **Застосувати стилі** — у відповідь відкриється однойменне команді вікно (рис. 4.14).
5. Клацніть кнопку з командою **Змінити** — відкриється вікно заміни стилю.
6. Змініть параметри форматування та підтвердьте вибір кнопкою **ОК**.

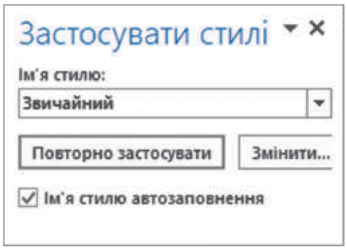


Рис. 4.14. Вигляд вікна **Застосувати стилі**

4.4. Схема документа. Колонтитули

Поясніть доцільність використання стилів і шаблонів у процесі форматування документів.

Для однакового автоматичного форматування фрагментів різного призначення (наприклад, гіперпосилання, виноски, назви) використовують стилі. Але з опису характеристик стилів видно, що вони в основному застосовуються для форматування абзацу.

Заголовок — назва стилю структурного елемента документа.

Найчастіше використовуються такі стилі, як Звичайний, та стилі Заголовок 1 і Заголовок 2.

Стиль Звичайний використовується для оформлення основного тексту документа.

Стиль Заголовок 1 використовується для загальних назв (розділів документа).

Стиль Заголовок 2 підпорядковується стилю Заголовок 1 і використовується для підрозділів документа.

Як уже зазначалося, максимальна кількість рівнів — 9. Але найчастіше використовують 2–4 рівні (приклад 1). Всі абзаци відформатовані на основі стилю Звичайний.

Зазвичай встановлення рівнів абзаців із використанням стилів Заголовки використовується для створення структурованого документа. На основі багаторівневого форматування абзаців із використанням розривів розділів створюють структуру документа.

Структура документа — це схема розміщення складових документа.

Під час перегляду документа в режимі структури його вміст має вигляд маркованого списку.

Приклад 1.

Розглянемо форматування на основі вибору стилів.

Розділ 10.

Тут вказана назва розділу (форматування стилем **Заголовок 1**)

Параграф 10.1

Тут вказана назва параграфа (форматування стилем **Заголовок 2**)

Текст параграфа.

Структура документа залежить від його призначення і виду (лист, наказ, закон, стаття, посібник). Структуру можна створити і після введення тексту в документ.

Алгоритм створення структури документа

Крок 1	Перейдіть до стрічки **Подання**, у групі з таким самим іменем виберіть команду **Структура** — зміниться вигляд документа, а в меню буде показана стрічка **Структура**
Крок 2	Перейдіть у робочу зону і почніть набирати заголовки розділів, параграфів як окремі абзаци
Крок 3	Під час уведення назви параграфа зверніться до команди **Рівень** у стрічці й зменште його. Так опрацьовують кожний заголовок — зменшити, збільшити або залишити на тому самому рівні. Є й інший варіант — уведіть всі заголовки, надалі виберіть кілька й перемістіть їх до іншого рівня
Крок 4	Після створення структури командами стрічки **Структура** перегляньте документ (якщо він є) та закрийте режим структури

Для перегляду структури документа необхідно на стрічці Подання виконати команду Структура (пригадайте, є й інші формати перегляду документа: Режим читання, Розмітка сторінки, Веб-документ, Чернетка). У результаті на панелі інструментів з'являться команди (кнопки) для роботи зі структурою документа, як наведено на рис. 4.15.

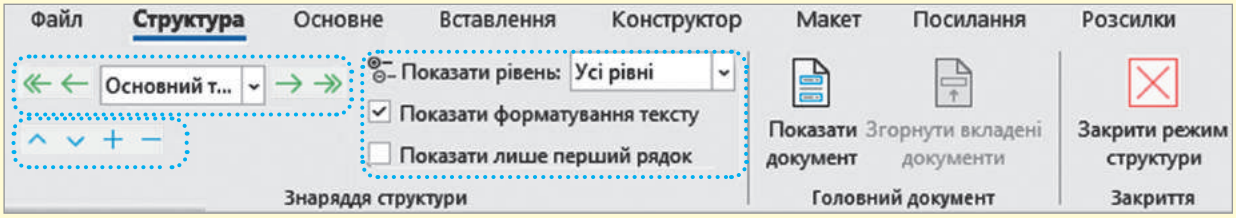


Рис. 4.15. Команди для роботи зі структурою документа

Особливості використання номерів сторінок як колонтитулів:

- номерами сторінок можуть бути не лише числа;
- номери сторінок можна як додавати, так і вилучати:
  - — з кількох або усіх сторінок документа
  - — з першої сторінки
  - — із будь-якої сторінки

Наявність колонтитулів полегшує пошук необхідних розділів під час перегляду документа.

У режимі структури можна не лише переглядати структуру документа, а й змінювати рівні вкладеності заголовків, копіювати й переміщувати великі фрагменти тексту шляхом переміщення заголовків вже створеного текстового документа.

Для змінення рівнів вкладеності розділів використовують кнопки, що розміщені в рамці червоного кольору (див. рис. 4.15), для переміщення назв розділів використовують кнопки, обведені зеленою рамкою, а для перегляду — синьою.

Сторінки зазвичай мають нумерацію. Поряд із номерами можуть бути назви заголовків структурних частин документа або інший текст. Такі елементи на сторінці є колонтитулами.

Колонтитул (від фр. *colonne* — стовпець і лат. *titulus* — напис, заголовок) — напис, який розміщують у верхньому або нижньому полі сторінки.

Розрізняють верхні та нижні колонтитули — відповідно до розміщення на сторінці. Для додавання нумерації сторінок на стрічці Вставлення у групі Колонтитули слід вибрати команду Номер сторінки. У випадному меню можна вибрати:

- режими розміщення номерів: угорі чи внизу сторінки, з вирівнюванням по лівому, правому краю чи посередині;
- команду Видалити номер сторінки для документа з нумерацією сторінок;
- команду Формат номерів сторінки для опрацювання колонтитулів (проста нумерація сторінок є простим прикладом колонтитулів у документі).

Видалити можна номери з усіх сторінок документа; номер на першій сторінці; номер із будь-якої сторінки документа; номери з кількох сторінок документа. Для повернення у звичайний режим розмістіть текстовий вказівник за межами колонтитула та клацніть ЛКМ або скористайтеся командою (червоний хрестик) на стрічці Знаряддя для колонтитулів — Конструктор

Для опрацювання нумерації сторінок слід скористатися командами вікна Формат номера сторінки (рис. 4.16) зі списку Номер сторінки стрічки Вставлення або зі стрічки Знаряддя для колонтитулів — Конструктор, якщо ви вже працюєте з колонтитулами.

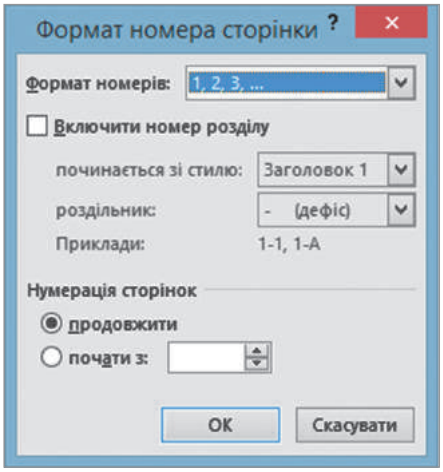


Рис. 4.16. Вікно **Формат номера сторінки**

Вікно Формат номера сторінки містить команди встановлення вигляду колонтитулів (числа, букви, можливість у нумерації сторінок використовувати нумерацію розділів тощо), установлення початкових номерів для сторінок (не обов'язково нумерувати з 1, наприклад, якщо розділи посібника розташовані в різних файлах).

Для вилучення колонтитулів призначена команда Видалити верхній (нижній) колонтитул: необхідно перейти до області колонтитула, вибрати список Верхній (Нижній) колонтитул і запустити команду Видалити верхній (нижній) колонтитул — колонтитул буде вилучено.

Розглянемо особливості встановлення колонтитулів. Слід пам'ятати, що часто на першій сторінці номер не проставляють.

Щоб колонтитул не відображався на першій сторінці, потрібно виконати такий алгоритм. Установіть вказівник миші в ділянку колонтитула і двічі клацніть ЛКМ — з'явиться стрічка конструктора колонтитулів; установіть прапорець для команди Інші для першої сторінки в групі Параметри. Сторінки з нумерацією і без номерів мають належати різним розділам, адже вони вже відрізняються параметром номер сторінки.

Щоб видалити номера не на першій сторінці, перед потрібною сторінкою слід установити розрив розділу (не розрив сторінки!): перейдіть до стрічки Макет, зі списку Розриви виберіть команду Наступна сторінка. Далі перейти до області колонтитула на потрібній сторінці й двічі клацніть ЛКМ; на стрічці вимкнути команду Як у попередньому з групи Навігація — використання цієї команди порушить зв'язок між розділами. Нарешті зробити активною команду Інші для першої сторінки — колонтитул на першій сторінці нового розділу не відображатиметься.

Для видалення нумерації на кількох сторінках відбудуться зміни в кроці 4 наведеного алгоритму: у стрічці активізуйте команду Верхній колонтитул (або Нижній колонтитул — залежить

Щоб додати текстовий колонтитул на сторінку, слід викликати стрічкове меню **Вставлення**, з групи **Колонтитул** вибрати необхідний колонтитул. Вказівник з'явиться в області колонтитула, основний текст буде недоступний для роботи з ним, з'явиться тимчасове стрічкове меню **Знаряддя для колонтитулів** → → **Конструктор.** І можна вводити текст в область колонтитула.

Якщо новий розділ зі сторінками без номерів не має продовжуватися до кінця документа, то слід встановити розрив цього розділу й для останнього нового розділу вставити колонтитули — номери сторінок.

від того, де проставлені номери сторінок) та зі списку, що розкриється, виберіть команду Видалити верхній колонтитул (чи нижній) — буде видалено номери з усіх сторінок розділу.

Опрацювання колонтитулів на парних і непарних сторінках ви опануєте під час роботи над завданням для самостійного виконання.

Запитання для перевірки знань

1. Які режими перегляду документа ви знаєте?
2. Що таке структура документа?
3. Поясніть поняття колонтитула.
4. Як вставити текстовий колонтитул?
5. Як вилучити номер на першій сторінці?
6. Чи може існувати багатосторінковий документ без зазначення нумерації деяких сторінок? Відповідь обґрунтуйте.

Завдання для самостійного виконання

1. Завантажте текстовий процесор і відкрийте багатосторінковий документ.
2. На стрічці **Конструктора** колонтитулів, коли вказівник уже в ділянці колонтитула, в команді **Різні колонтитули** для непарних і парних сторінок і проставте прапорець.
3. Перейдіть у ділянку колонтитула непарної сторінки, запустіть команду **Верхній** (або **Нижній**) колонтитул; виберіть за своїм бажанням зразок вигляду колонтитула; уведіть напис.
4. Повторіть ці самі дії для парної сторінки.
5. Оскільки Word пропонує різні зразки для різних сторінок, відслідковуйте, які зразки краще вибрати. Поверніться до основного тексту документа.

4.5. Автоматизоване створення змісту та покажчиків

Пригадайте, що таке стилі документа. Як встановити форматування назв частин документа одним стилем?

Зміст документа — це структура, створена з форматованих заголовків із зазначенням сторінок їх розміщення в документі.

Багатосторінковий документ на початку або в кінці зазвичай містить зміст. Текстовий процесор має автоматичний режим створення та опрацювання змісту.

До змісту увійдуть правильно оформлені назви структурних елементів документа — до них потрібно застосувати стилі Заголовок 1, Заголовок 2 тощо. Для створення змісту документа потрібно розташувати текстовий вказівник у місце вставлення змісту, перейдіть на вкладку Посилання, пункт Зміст, і вибрати вигляд змісту зі списку. Доцільно вибрати тип змісту з ієрархічним поданням заголовків різних рівнів. Після додавання чи вилучення тексту в документі, розділів чи параграфів слід виділити зміст. Далі зробити оновлення з вибором Оновити все чи тільки Номери сторінок.

Автоматично створений зміст документа показує, з яких частин складається документ і на яких сторінках вони розташовані, а також дозволяє одразу перейти з нього до необхідної частини. Під час створення змісту назви структурних елементів оформлені як посилання на відповідні сторінки.

Під час роботи з документами також може виникнути потреба в додаванні в документ списку термінів, які розглядаються в документі, із зазначенням сторінок, на яких вони згадуються. Це покажчик, його можна створити на стрічці Посилання у групі Покажчик (рис. 4.17).

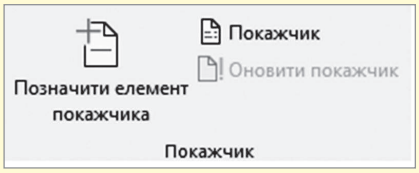


Рис. 4.17. Вигляд групи **Покажчик**

Елементи групи Позначити елемент покажчика і Покажчик призначено для створення покажчика, а Оновлення покажчика, як і змісту, — для оновлення після редагування елементів покажчика або їх видалення.

Розглянемо алгоритм створення покажчика в документі. Щоб створити покажчик, слід виділити текст як елемент покажчика — текстом можуть бути окремі слова, фрази або символ; вибрати і позначити вигляд покажчика та всі його елементи. Word знайде всі елементи з відповідною позначкою, відсортує їх за алфавітом, додасть номери сторінок, видалить повторювані елементи й відобразить покажчик у документі. Після цього створіть сам покажчик.

Алгоритм створення покажчика

1. Визначення слова або фрази як елемента покажчика

Крок 1	Виділіть текст, який ви хочете використовувати як елемент предметного покажчика
Крок 2	На вкладці **Посилання** в групі **Покажчик** виберіть команду **Позначити елемент покажчика** — відкриється вікно (рис. 4.18)
Крок 3	У вікні **Елементи покажчика** налагодьте елемент покажчика: • можна замінити виділений текст на інший у полі **Елемент покажчика**; • вибрати **Параметр вибору місця знаходження елемента**: на поточній сторінці, у діапазоні сторінок або як посилання на введений у полі **Посилання** текст; • налагодьте формат подання сторінки у покажчику
Крок 4	Щоб позначити елемент покажчика, натисніть кнопку **Позначити**. Щоб позначити всі місця в указаному вище діапазоні, які містять даний текст, натисніть **Позначити все**

2. Створення покажчика (*продовження алгоритму*)

Крок 5	Перемістіть вказівник туди, де буде розміщено покажчик
Крок 6	Запустіть команду стрічки **Посилання** → група **Покажчик** → **Покажчик**: відкриється вікно **Покажчик** (рис. 4.19)
Крок 7	У вікні зробіть активною вкладку **Покажчик**
Крок 8	Відформатуйте вигляд списку покажчика на основі команд вибору формату текстових елементів, номерів сторінок, вигляду покажчика

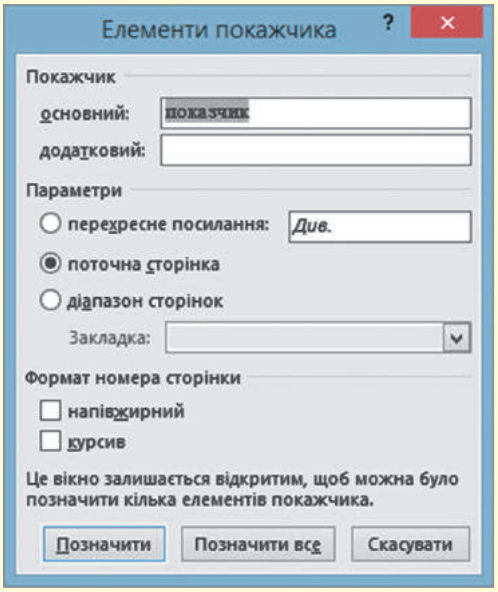


Рис. 4.18. Вигляд вікна **Елементи покажчика**

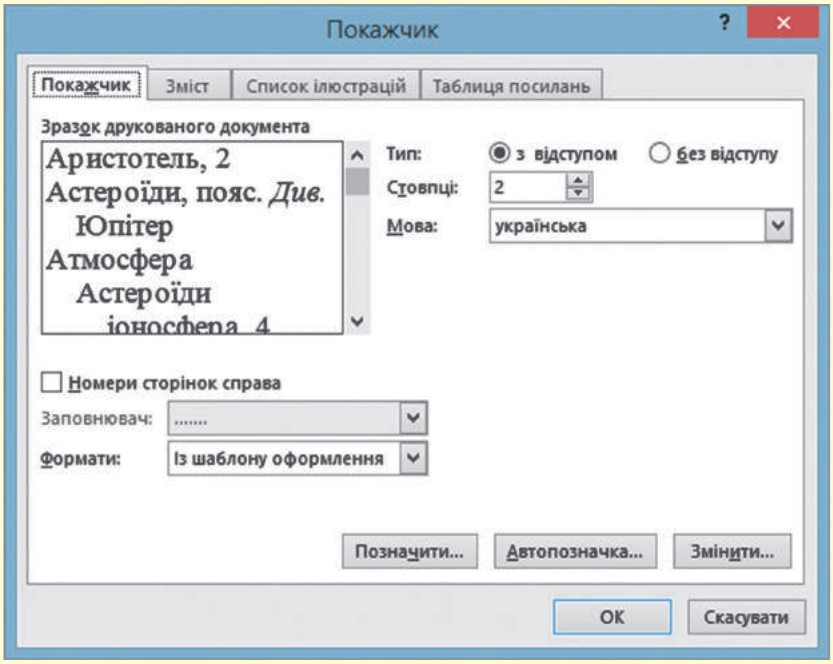


Рис. 4.19. Вигляд вкладки **Покажчик**

Приклад 1. Для елемента покажчика виберемо слово покажчик. Після цього слова в документі з'явиться така позначка:

{XE «покажчик»}

Фігурні дужки означають вставлення в документ кодового поля, а XE є назвою поля для створення покажчика до цього слова.

Приклад 2. У документі є назви планет. Виділимо назву планети у вікні форматування елемента покажчика і виберемо команду перехресне посилання, а в поле введемо слово планети. Вигляд покажчика буде з параметром заміщення:

{XE «Марс» \ t «планети»}

Покажчик варто оновити після редагування елементів, якщо в тексті виявлено помилку, після видалення елемента покажчика відповідною клавішею клавіатури.

Якщо після створення покажчика буде додано нові елементи, для їх відображення слід оновити предметний покажчик. Для цього потрібно знайти поле для елемента, текст якого необхідно змінити. Якщо коди поля покажчика не показані, їх вмикають командою Показати всі знаки з групи Абзац стрічки Основне. Щоб змінити текст елемента покажчика, змініть текст всередині лапок. Щоб оновити предметний покажчик, клацніть його та натисніть клавішу F9 або виберіть команду Оновити покажчик у групі Покажчик на вкладці Посилання.

Запитання для перевірки знань

1. Дайте означення змісту документа.
2. Що таке покажчик?
3. Які особливості підготовки тексту для автоматизованого створення змісту?
4. Опишіть алгоритм створення покажчика.
5. Назвіть особливості створення покажчика.
6. Порівняйте структуру документа зі змістом будь-якого підручника. Що в них спільного?

4.6. Опрацювання складного текстового документа. Макроси

Які дані може містити документ Word? Пригадайте, що таке шаблон документа Word. Яке розширення мають файли документа й шаблона Word?

При створенні нового документа програма пропонує вибрати його шаблон із бібліотеки. Привітання, запрошення на події можуть містити зображення, не обов'язково створені у Word, звуковий супровід. Однією з видів таблиць у документі є таблиця Excel. Таких прикладів розміщення в документі фрагментів з інших програмних засобів може бути багато.

Складний (інтегрований) документ — це документ, який містить фрагменти документів різних типів.

Фрагменти можуть бути створені в різних додатках Windows або MS Office та інтегровані (об'єднані) в одному документі Word. Для створення складного документа має бути документ-джерело, з якого береться фрагмент, і документ-приймач, у який цей фрагмент вставляють. Останній ми інколи будемо називати поточним документом — таким, який у результаті стане складним.

Існує кілька способів створення складного документа.

- За допомогою миші. У межах документа мишею зручно перетягувати виділений фрагмент. Для цього фрагмент необхідно виділити, затиснути ЛКМ і, не відпускаючи, перемістити в потрібне місце, відпустити ЛКМ: реалізується команда переміщення фрагмента.

Аналогічно працюють із мишею при одночасно відкритих вікнах різних документів. Описане перетягування фрагментів аналогічне копіюванню і вставленню, а з утриманою клавішею Ctrl — вирізанню і вставленню.

- З використанням буфера обміну. Для обміну даними в додатках MS Office зручно користуватися буфером обміну. Для цього на стрічці Основне є група Буфер обміну, у ній кнопкою зі стрілкою відкривається область завдань з однойменною назвою.

Усі попередньо скопійовані фрагменти містяться в буфері обміну. В області завдань вони мають позначки, які відповідають програмним засобам, і невеличкі мініатюри скопійованого фрагмента (рис. 4.20). Якщо навести вказівник миші на необхідну позначку, вона стає виділеною і з'являється список із двома командами: Вставити, Видалити.

За допомогою команди Вставити можна вставити потрібний фрагмент у поточний документ у місце перебування текстового вказівника, а за допомогою команди Видалити — видалити фрагмент із буфера обміну.

Додатки **MS Office** сумісні один з одним, тому в текстовому документі можуть бути такі об'єкти: текст документа **WordPad**, таблиця **Excel**, звіт **Access**, малюнки **Paint**, відеокліп.

Робота зі складним документом має певні переваги: окремі складові можуть створювати різні користувачі одночасно за різними комп'ютерами, що прискорює процес створення такого документа.

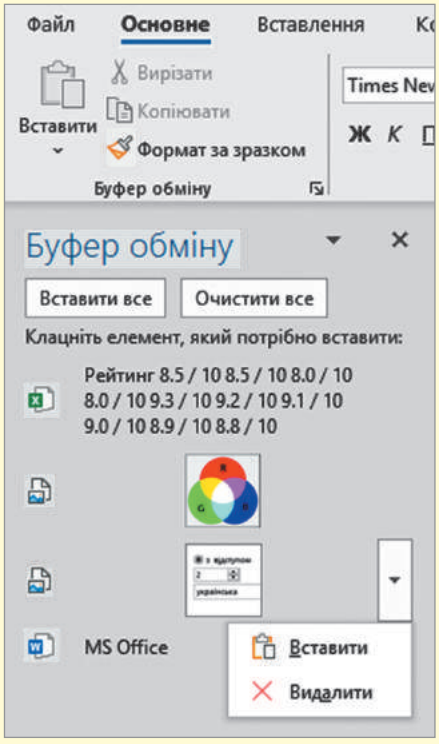


Рис. 4.20. Вигляд області завдань **Буфер обміну**

- З використанням буфера обміну за технологією OLE. За допомогою попередніх способів між фрагментом, доданим у поточний документ, і документом-джерелом розривається зв'язок. Фрагмент вставили, і редагувати його в джерелі не можна. Цю проблему розв'язує використання технології OLE (англ. *Object Linking and Embedding* — зв'язування і впровадження об'єктів).

Таблицю Excel, вставлену в документ Word, редагують засобами Excel. Для цього слід двічі клацнути мишею таблицю, і частина екрана Excel (тільки з командами для роботи з таблицею) активується на екрані монітора.

Для реалізації технології у Word у файлі-джерелі потрібно виділити необхідний фрагмент, скопіювати його в буфер обміну знайомим способом, запустити команду Використати спеціальне вставлення зі стрічки Основне списку Вставити. У відповідь відкривається вікно Спеціальне вставлення (рис. 4.21). У ньому вибрано програму-джерело. Потрібно вибрати формат вставлення в документ-приймач із запропонованих форматів.

Технологія OLE призначена для зв'язування об'єктів різних додатків ОС Windows за певними правилами. Деякі програми повністю підтримують технологію OLE, можуть бути і джерелом, і приймачем об'єктів (наприклад, WordPad, Word), деякі — лише джерелом (графічний редактор Paint).

Об'єкт вставлення називають OLE-об'єктом. Він зберігає зв'язок із додатком створення, тобто з'являється можливість редагування об'єкта засобами джерела з документа-приймача.

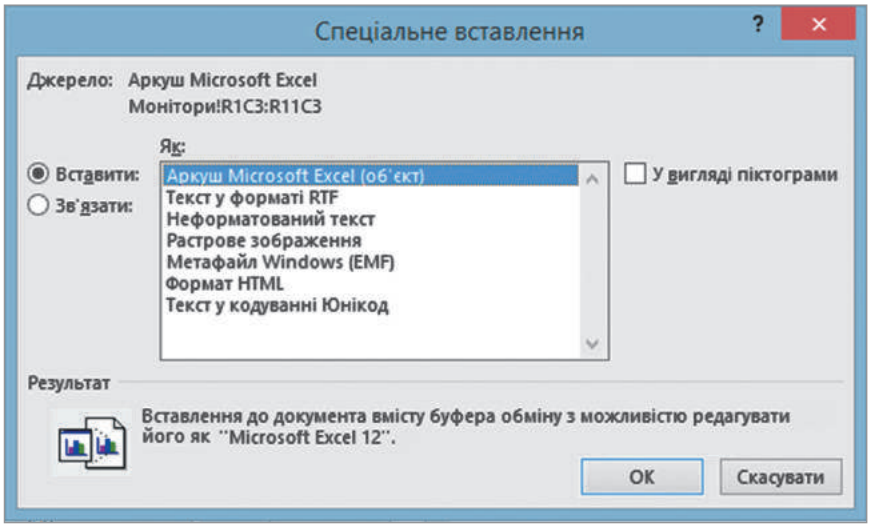


Рис. 4.21. Вигляд вікна **Спеціальне вставлення**

Існують різні способи вставлення формату. Розглянемо їх.

- Командою Вставити вставляють копію об'єкта, залишається зв'язок із програмою — додатком створення об'єкта, зв'язок із файлом-джерелом не підтримується. Цей спосіб є громіздким за витраченою пам'яттю.
- Командою Зв'язати користуються для вставлення об'єкта як гіперпосилання, залишається зв'язок об'єкта з програмою і файлом — джерелами.
- Командою У вигляді піктограми користуються, якщо у файлі-приймачі немає потреби постійно бачити вставлений об'єкт у відкритому вигляді. У вікні з'являється зразок піктограми, вигляд якого матиме вставлений об'єкт у документі-приймачі.

Якщо клацнути зразок мишею один раз, відкриється вікно для заміни вигляду піктограми та підпису. Подвійне клацання мишею по піктограмі розкриває документ-джерело, звичайне його закриття повертає вигляд піктограми. Піктограма є посиланням на файл-джерело з об'єктом. Зазвичай піктограма має вигляд піктограми програмного засобу, в якому створено файл.

У процесі роботи з документом може виникнути потреба у використанні однотипних дій кілька разів. Або у створенні документа, в якому є автоматизоване повторення певних дій. У таких випадках користуються макросами.

Макрос — це набір команд та інструкцій у вигляді єдиної команди з назвою для автоматичного виконання завдання.

Існує кілька способів створення макроса.

- Програмний засіб Microsoft Visual Basic для додатків. У ньому у вигляді кодів програми записати команди для їх виконання. Для цього у стрічці Розробник запускають команду Макрос; у вікні, що відкриється, вводять його назву; запускають команду Створити, далі пишуть програму у вікні програмного засобу.
- Автоматичне створення макроса без використання вікна програмування. Для цього слід налагодити запам'ятовування необхідних команд; надати групі цих команд назву. Далі можна використовувати створений макрос у поточному або інших документах Word

Для створення макроса потрібно перейти на стрічку Розробник (у разі відсутності додати її у вікні Файл → Параметри → Налаштування стрічки → прапорець біля Розробник). Запустити команду Записати макрос — відкриється вікно запису (рис. 4.22), у якому ввести назву макроса (автоматично Word дає назву Макрос 1). Далі слід вибрати, у яких документах буде доступ до створеного макроса зі списку Зберегти макрос як: у поточному чи всіх на основі шаблона Звичайний, та як буде відбуватися запуск макроса: кнопкою чи натисканням клавіш: виберіть команду Призначити макрос клавішам (відкриється вікно Настроювання клавіатури).

Потім у вікні слід ввести сполучення клавіш, натисканням яких і буде запускатися макрос, і підтвердити введення. Команда Записати макрос зміниться на команду Зупинити запис.

Виконайте необхідні команди. Word буде записувати все, що ви клацаєте мишею, і всі клавіші, які ви натискаєте (макроси не підтримують виділення фрагмента тексту мишею — необхідно користуватись клавішами клавіатури). За потреби можна призупинити запис — команда Пауза, виконати якісь дії і продовжити запис командою Відновити записування.

Тепер зупиніть запис макроса командою Зупинити запис.

В обох випадках до стандартних розширень у назвах файлів додається літера m, тип документа буде .docm, а шаблона .dotm (наприклад, шаблон Normal.dotm).

Налагодження і редагування макроса відбувається командою Макрос — відкриється однойменне вікно, яке містить назви всіх макросів, доступних у даному документі, і команди опрацювання вибраного макроса.

Можна відмовитись від вигляду піктограми для об'єкта, перетворити її на повноцінний вигляд. Піктограму виділяють і в контекстному меню вибирають команду **Зв'язаний об'єкт** → **Документ** → **Перетворити...**, у вікні **Перетворення типу документа** потрібно зняти прапорець **У вигляді піктограми**.

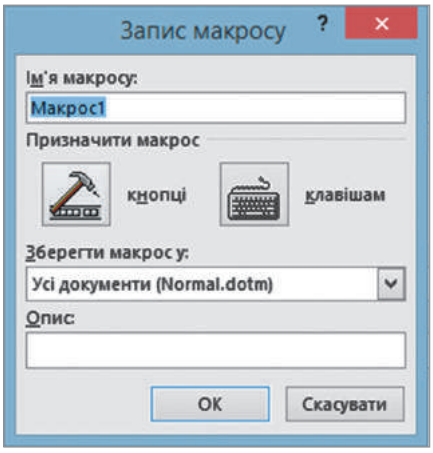


Рис. 4.22. Вікно **Запис макросу**

Документи з макросами зберігають:

- у файлах типу **Документ Word** з підтримкою макросів — користування макросом дозволено тільки в документі створення макроса
- у **Шаблоні Word** з підтримкою макросів — користування макросом дозволено в усіх документах, створених на основі шаблона **Звичайний**.

Запитання для перевірки знань

1. Дайте означення поняттю «складний документ».
2. Що таке макрос?
3. Перелічіть способи додавання в документ Word об'єктів з інших додатків.
4. Опишіть кожний спосіб додавання в документ Word об'єктів з інших додатків.
5. Назвіть особливості створення макроса.
6. Як створити макрос, який запускається кнопкою?

4.7. Налаштування параметрів роботи середовища текстового процесора

Пригадайте, як виглядає вікно текстового процесора при завантаженні.

Налагодження параметрів має значення при одночасній роботі з кількома документами. Особливо важливим у цій роботі є процес копіювання та збереження таких об'єктів, як рисунки, списки, інші фрагменти, форматування яких може відрізнятися від форматування решти документа.

При завантаженні текстового процесора екрани різних комп'ютерів мають однаковий вигляд. Це пояснюється стандартним налаштуванням середовища. Його можна налагодити згідно зі своїми потребами.

До складу меню Файл входить команда Параметри, при виклику якої відкривається вікно Параметри Word (рис. 4.23). У ньому можна переглядати й оновлювати налаштування Word. Загальні параметри дають змогу змінювати інтерфейс користувача, персоналізацію та налаштування запуску для копії Word і документів.

Загальні параметри призначені для налагодження процесу перетягування фрагмента документа, показу спливних підказок про призначення кнопок, команд. Подвійним клацанням документа зі стандартним розширенням відкривається текстовий процесор Word з одночасно завантаженим файлом — ця процедура також налагоджується в загальних параметрах. Решта параметрів Word згруповані за своїм призначенням.

Додаткові параметри задаються командою Додатково:

Параметр	Опис
Редагування	Налагодження параметрів виділення, заміни, форматування тексту, вигляду абзаців, списків
Вирізання, копіювання та вставлення	Вибір режимів вставлення скопійованих текстових об'єктів та їх форматування (особливо це стосується списків) у межах одного чи кількох документів
Розмір і якість зображення	Установлення кількості пікселів на дюйм при виведенні зображення на друк. Можна встановити для всіх зображень або для одного — конкретного
Вміст документа	Відображення тексту та інших елементів: рисунків, закладок, кодів полів
Відображення	Показ смуг прокрутки, лінійки, підказок
Друк	Налаштування друку сторінок у прямому (зворотному) порядку розташування, на лицьовій (зворотній) сторінці для двостороннього друку, розміру паперу

Параметр	Опис
Збереження	Зберігання змін, внесених у шаблони, а також налаштування автоматичного резервного копіювання файлів
Надбудови	Налаштування співпраці Word із сумісними програмними засобами

З панеллю Настроювання стрічки ви вже знайомі — її параметри використовують для редагування стрічок у вікні Word. Однойменними параметрами користуються для налагодження показу команд у панелі швидкого доступу.

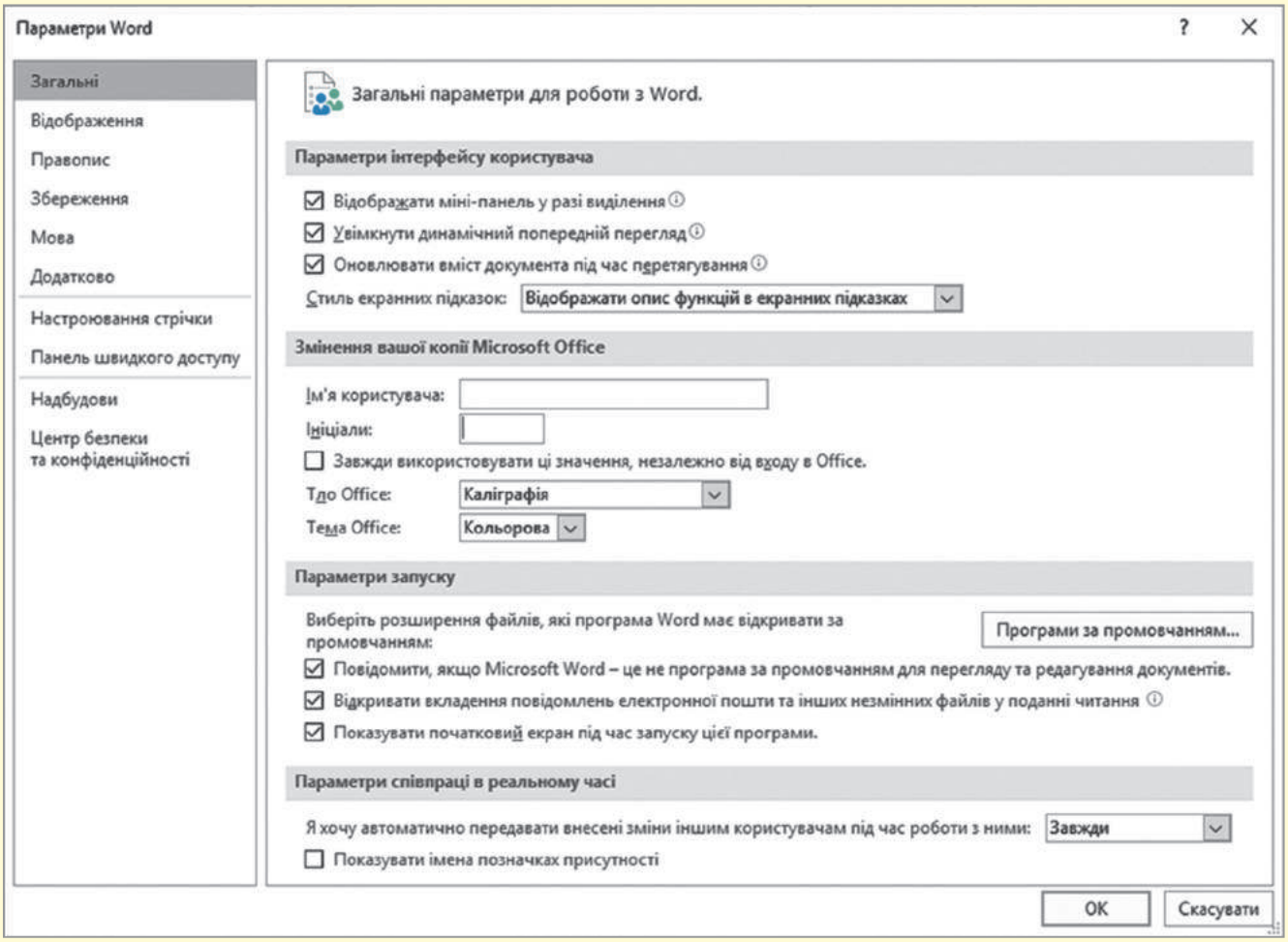


Рис. 4.23. Вигляд вікна налаштування загальних параметрів

Запитання для перевірки знань

1. Ознайомтеся з параметрами налаштування Word, які встановлено на вашому ПК.
2. Які параметри ви хотіли б змінити? Чому?
3. Як налаштувати показ абзаців у вигляді списку при їх копіюванні та вставленні в інший документ?
4. Як налаштувати двосторонній друк?
5. Назвіть способи перегляду кількох документів на екрані.
6. Як визначити, який документ є активним при одночасному відкритті кількох документів?

Практична робота № 6

ТЕМА. Створення текстового документа, що містить об'єкти різних типів

ЗАВДАННЯ: створити буклет і візитівки для Всеукраїнського конкурсу-захисту науково-дослідницьких робіт учнів — членів Малої академії наук (МАН) України.

ОБЛАДНАННЯ: комп'ютер, текстовий процесор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте програму Microsoft Word.
2. Виберіть команду створення нового документа та із запропонованих шаблонів виберіть буклет.
3. Виберіть один із макетів буклета і виконайте команду Створити у вікні, що відкриється. У робочій області розміщений макет сторінок буклета.
4. Заповніть текстові та графічні блоки даними:
   1) на титульній сторінці (третя колонка першої сторінки буклета) у блоці Назва організації уведіть Міністерство освіти і науки України, на наступному рядку — назву вашого закладу;
   2) додайте текстове поле (панель ВСТАВЛЕННЯ, команда Додати текстове поле), розмістіть по центру титульної сторінки; вставте назву заходу;
   3) додайте текстове поле і розмістіть внизу титульної сторінки, введіть дату проведення конкурсу.
5. На зворотній сторінці буклета (друга колонка першої сторінки) в блок із назвою Організація уведіть реквізити закладу: адресу, контактні телефони, у графічний блок додайте емблему закладу.
6. У першій колонці першої сторінки, яка містить текстове поле з назвою Заголовок задньої панелі, розмістіть інформацію про ваш заклад.
7. У першій колонці другої сторінки додайте план проведення конкурсу:
   1) змініть назву в текстовому блоці на План конкурсу;
   2) змініть вміст текстового блоку на перелік доповідачів по порядку їх виступів.
8. В інші колонки сторінки введіть інформацію про роботу секцій МАН, секції виберіть на свій розсуд.
9. Збережіть буклет.
10. Аналогічно п. 2 і 3 виберіть шаблон візитівок та завантажте його для опрацювання.
11. Введіть дані у візитівки на одному аркуші.
12. Збережіть файл із візитівками.

Зробіть висновок щодо можливостей текстового процесора застосовувати готові шаблони документів.

2. Якщо в шаблонах буклети відсутні, введіть у пошукове вікно слова Буклет або Брошура.

3. Буклет має альбомну орієнтацію з трьома колонками для розміщення даних.

- Всеукраїнський конкурс-захист науково-дослідницьких робіт учнів — членів МАН — один із провідних заходів МАН. Він має на меті «виявлення і підтримку обдарованих дітей, залучення учнівської молоді до науково-дослідницької та експериментальної роботи, формування активної громадянської позиції тощо.
- Складається з кількох етапів: відбірний, регіональний, всеукраїнський.
- Щороку понад 100 тис. учнів і учениць беруть участь у відбірних змаганнях.

Більше дізнатися можна на сайті.

Практична робота №7

ТЕМА. Використання стилів для оформлення текстових документів. Структура та зміст документа

ЗАВДАННЯ: ознайомитись із можливостями редактора щодо опрацювання стилів, створення структури та змісту конкретного документа.

ОБЛАДНАННЯ: комп'ютер, під'єднаний до інтернету, текстовий процесор.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. У новому документі Word перейдіть до стрічки ПОДАННЯ і в групі Подання запустіть команду Структура.
2. Створіть структуру нового документа:
   1) у робочій області наберіть заголовок (наприклад, Вступ) і завершіть абзац. Зверніть увагу на поле Рівень у стрічці меню;
   2) наберіть заголовок Розділ 1. Кінець абзацу;
   3) наберіть заголовок Параграф 1. Стрілками команди Рівень перемістіть заголовок до II рівня;
   4) наберіть заголовок Розділ 2 і перемістіть його до I рівня. Кінець абзацу;
   5) виконайте команду Закрити режим структури.
3. У режимі розмітки сторінки після слів Параграф 1 вставте Параграф 2.
4. Змініть стиль написання Параграф 2:
   1) виділіть текст Параграф 2;
   2) відкрийте меню Стиль і виберіть тип стилю Заголовок 2.
5. Відкрийте сайт Вікіпедії, у пошуковий рядок уведіть текст Історія ЕОМ.
6. Копіюйте текст з Вікіпедії та заповнюйте створений документ за логікою структури документа.
7. Виберіть розмір поля сторінок документа — А5.
8. Проставте нумерацію сторінок у документі.
9. Додайте першу сторінку з назвою посередині сторінки Історія обчислювальної техніки та вкажіть своє прізвище в правому нижньому куту сторінки. Завершіть розривом сторінки.
10. Вставте зміст на початку документа. Чи є посилання на першу сторінку?
11. Вилучіть номер із першої сторінки й оновіть зміст (виберіть Оновлення номерів сторінок).
12. Змініть назви розділів, щоб вони відповідали тексту, і знову оновіть зміст (виберіть Оновлення всього).

Зробіть висновок щодо підготовки стилів окремих фрагментів тексту для автоматизації створення змісту.

3. Який стиль має написання введеного тексту?

4. Як змінилось написання тексту?

11, 12. Зверніть увагу на результат.



Практична робота № 8

ТЕМА. Автоматизоване створення покажчиків. Макроси

ЗАВДАННЯ: ознайомитись із можливостями редактора щодо створення та опрацювання тематичних покажчиків і макросів.

ОБЛАДНАННЯ: комп'ютер, під'єднаний до іньернету, текстовий процесор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Для пошуку в інтернеті в адресний рядок уведіть слова МОН булінг — відкриється сайт МОН України зі статтею «Протидія булінгу». Сторінку не закривайте. Скопіюйте текст статті в новий документ.
2. Перейдіть до статті на сайті МОН України і виберіть посилання Корисні посилання щодо теми булінгу.
3. На сторінці, що відкриється, перейдіть за посиланням Протидія булінгу в закладах освіти: системний підхід — відкриється посібник. Скопіюйте сторінки 4 і 5, уставте в документ. Далі працюйте в текстовому процесорі.
4. Виділіть слово булінг. Запустіть команду: стрічка Посилання група Покажчик Позначити елемент покажчика.
5. Повторіть п. 6 і 7 для слова булінгу, для слова цькування виберіть вказівку Перехресне посилання, а в поле посилання введіть див. булінг.
6. Перемістіть вказівник у кінець документа, запустіть команду: стрічка Посилання група ПокажчикПокажчик. Відредагуйте показ сторінок на свій розсуд, клацніть ОК.
7. Перемістіть вказівник на початок слова булінг у будь-якому місці документа.
8. Перейдіть до стрічки Розробник і запустіть команду Записати макрос — відкриється вікно встановлення макроса:
   1) виберіть режим призначення макроса клавішам — відкриється вікно для введення сполучення клавіш;
   2) у вікні перемістіть вказівник у поле встановлення нового сполучення;
   3) натисніть сполучення клавіш — Ctrl + будь-яку літеру;
   4) підтвердьте, клацнувши ОК.
9. Введіть у документі слово цькування і видаліть клавішами клавіатури булінг.
10. Перейдіть до стрічки Розробник і виконайте команду Зупинити макрос.
11. Перемістіть вказівник на початок іншого слова булінг і натисніть сполучення клавіш Ctrl + Q. Відбулась заміна.
12. Збережіть документ у режимі Документ Word з підтримкою макросів.

Зробіть висновок щодо доцільності використання макросів та різного виду посилань у документі.

4. *Примітка* на появу полів у фігурних дужках. У вікні, що відкриється, позначте поточну сторінку і підтвердьте командами кнопок Позначити.

12. Зверніть увагу на тип збереженого файлу

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 4

Розділ 5
СТВОРЕННЯ ТА ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА

5.1. Поняття мультимедіа. Кодування аудіо- та відеоданих

Світ змінюється, світова пандемія, пов'язана з короновірусною інфекцією 2020 року, внесла корективи в наше життя, і зокрема у сферу освіти. Як змінився формат проведення ваших занять?

Під впливом різних чинників, у тому числі розвитку комп'ютерних технологій, акцент у системі освіти змістився в бік дистанційного способу проведення уроків, використання інтернету для навчання. Матеріали цього розділу допоможуть вам опанувати інструменти, що стають як ніколи актуальними.

Ви вже познайомилися з відеоуроками, які вчителі записують, викладають на власний Youtube-канал, а потім надають посилання, щоб ви могли переглянути їх у зручний для вас час.

Мультимедіа (від лат. *multum* — багато, *medium* — середовище) — комп'ютеризована технологія, яка по'єднує різні способи подання даних на одному носієві.

Сьогодні мультимедіа є одним із найперспективніших і найпопулярніших напрямів сучасної інформатики. Його мета — розробка технологій створення й опрацювання продукту, що передає інформацію шляхом використання зображень і текстів, які супроводжуються звуком, відео, анімацією й іншими візуальними ефектами.

Розвиток мультимедійних технологій є логічним продовженням впровадження інформаційних систем в усі сфери життя. Згадаємо, які є об'єкти мультимедіа (рис. 5.1).



Рис. 5.1. Класифікація об'єктів мультимедіа

Кодування аудіоданих

Ви вже вмієте кодувати текстові та графічні об'єкти. Тепер розглянемо, як відбувається кодування звуку в аудіооб'єктах. Згадаємо, що звук як фізичне явище — це коливальний рух частинок середовища, яке поширюється в цьому середовищі.

Оцифрування звуку — послідовність перетворення звичайного звуку в закодований цифровий сигнал.

Ознайомимося з порядком дій з оцифрування звуку. На приймач подається звуковий сигнал (приймач — мікрофон). На АЦП подається неперервний електромагнітний сигнал мікрофона, здійснюється оцифрування звуку — дискретизація та квантування отриманого сигналу. Сигнал у вигляді двійкового коду зберігається в комп'ютері.

Збереження кольорового відеосюжету здійснюється завдяки використанню моделі RGB. Зміна об'єктів відбувається з однаковою швидкістю, проте закодований відеофайл матиме великий розмір.

Експертна група з рухомих зображень (MPEG) — це робоча група ISO/IEC, яка відповідає за розробку міжнародних стандартів стиснення, декомпресії, обробки та кодованого подання рухомих зображень, аудіо та їх поєднання. Зазвичай MPEG проводить на рік 4 засідання, що включають пленарні засідання та засідання підгруп з вимог, систем, відео, кодування відео, 3D Video Audio, 3D Graphics і зв'язку. У зустрічі беруть участь десь понад 400 експертів із майже 20 країн. Вони представляють понад 200 компаній, що охоплюють усі галузі з використанням цифрового аудіо, відео та мультимедіа. Наразі MPEG вивчає нові можливості для стандартів, які відповідають потребам медіа-індустрії (адреса сайту mpeg.chiariglione.org/)

Кодування відеоданих

Кодування звукового супроводу відеоданих не відрізняється від кодування звуку. Зображення у відео складається з окремих кадрів, які змінюються з певною частотою. Кадри кодуються як растрове зображення — попіксельно. Якщо закодувати всі кадри, відеодані буде подано як послідовність змінних графічних об'єктів, описаних у цифровому вигляді.

Існують програми кодування відео, які використовують алгоритми стиснення відеоданих, — кодеки.

Завдяки аналізу кодек виявляє і зберігає кадри, на яких відбулася зміна сюжету. Дані зі змінами зберігаються в поточному кадрі відносно попереднього, проміжні кадри взагалі не зберігаються. Існують також і інші можливості. Наприклад, у відеосюжеті, у якому на незмінному тлі відбувається рух об'єкта, на всіх кадрах, задіяних у ньому, зміни стосуються лише рухомого об'єкта. Зберігається тло і деякі моменти руху об'єкта.

До характеристик збереження відеоданих належить кількість кадрів на секунду, роздільна здатність зображення, глибина кольору. Різні стандарти стиснення мають різні характеристики збереження. Кожний кодек зберігає відеофайли у своєму форматі. Існує велика кількість відео- і аудіоформатів, бо вони створювалися для різних пристроїв і мають різне призначення. Технології опрацювання мультимедійних даних невпинно розвиваються, що зумовлює появу нових форматів файлів і «відмирання» застарілих.

Цифровий відеофайл складається з контейнера і кодека, виражених разом у відеоформаті. Контейнер відеофайлів схожий на коробку, в якій містяться різні дані, зберігається відеопотік, аудіопотік та інші метадані, такі як субтитри.

Кодек можна розглядати як програмне забезпечення, яке дозволяє відеоданим взаємодіяти із зовнішніми платформами, так що, коли контейнер «доставляється» програмами редагування або програвачів, відео буде повністю функціональним.

Формати аудіофайлів

Розглянемо найпоширеніші аудіоформати та їх опис.

Формат	Опис
WAV	Один з перших аудіоформатів. Зазвичай використовується для зберігання нестиснених аудіозаписів (PCM), ідентичних за якістю звуку записів на компакт-дисках (audio-CD). У середньому 1 хв звуку у WAV займає близько 10 Мб
WMA	Windows Media Audio — формат компанії Microsoft, який був представлений як заміна MP3. За заявою Microsoft, має більш високі характеристики стиснення, проте це спростовується деякими незалежними тестами
MP3	MPEG Layer-3 — найбільш поширений в світі звуковий формат. MP3, як і ага-то інших форматів із втратою якості, обмежує діапазон відтворення, який не може сприйняти людина, тим самим зменшуючи розмір файлу. На поточний момент MP3 — не найкращий формат щодо співвідношення розміру файлу до якості звучання. Проте він підтримується значною кількістю пристроїв, і більшість користувачів зберігають свої записи саме в ньому

Формат	Опис
OGG	Відкритий формат, який підтримує кодування аудіо різними кодеками. Найбільш часто в OGG використовується кодек Vorbis. За якістю стиснення формат можна порівняти з MP3, але при цьому менш поширений з точки зору підтримки в аудіопрогравачах і плеєрах. Саме в **OGG** мобільні телефони зберігають та відтворюють голосові повідомлення
AAC	Запатентований аудіоформат, який має великі можливості (кількість каналів, частоти дискретизації) порівняно з MP3 і який дає дещо краще звучання за того самого розміру файлу. Сьогодні є одним з найбільш якісних алгоритмів кодування звуку з втратами. Формат підтримується більшістю пристроїв
FLAC	Популярний формат стиснення без втрат. Не вносить змін в аудіопотік і звук, закодований за його допомогою, ідентичний оригіналу. Часто використовується для прослуховування звуку на звукових системах високого рівня

Порівняємо можливості стиснення аудіоформатів (рис. 5.2).

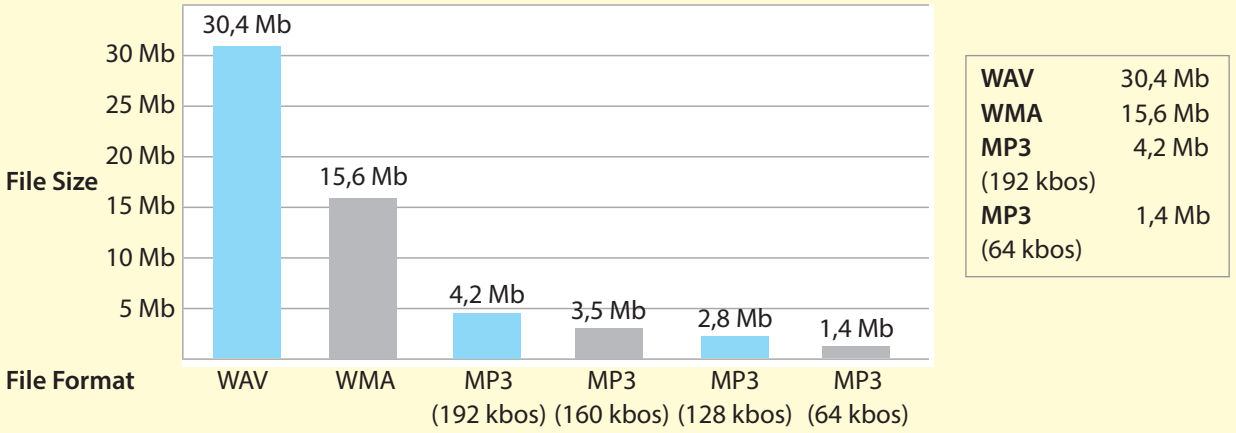


Рис. 5.2. Стиснення аудіоформатів

Формати відеофайлів

Розглянемо найпопулярніші відеоформати та їх опис.

Формат	Опис
MP4 (або MPEG-4)	Формат є одним з найпопулярніших для використання в інтернеті, оскільки стискає відео високої якості у відносно невеликий розмір. Всі основні платформи обміну підтримують MP4, він досить широко застосовується для редагування програм і офлайн-програм. Формат був введений тією самою командою, що встановила стандарти стиснення аудіо та відео: The Moving Pictures Experts Group (*Група експертів з рухомих зображень*)
WMV (Windows Media Video)	Формат був розроблений Microsoft і є основним типом відеофайлів, підтримуваних ОС Windows. Формат популярний для онлайн-використання, оскільки не займає багато місця і володіє ще більш кращими можливостями стиснення, ніж MP4
AVI	Найпопулярніший контейнер, розробником якого є Microsoft. У файлах такого типу можуть зберігатися одночасно аудіо, відео, текст і графіка. Головна перевага AVI полягає в тому, що його підтримують практично всі популярні плеєри та сучасні пристрої

Формат	Опис
AVI (Audio Video Interleave)	Формат сумісний із широким спектром медіаплеєрів, виготовлених до і після його випуску в 1992 році. Він має широкий спектр кодеків і працює в різних системах, таких як Windows, Macintosh і Linux, тому зазвичай використовується за замовчуванням для зберігання відео на комп'ютері. З іншого боку, його високу якість може утруднити стиснення відео, і цей формат відео може бути не ідеальним для потокового передавання
MOV (Apple QuickTime Movie)	Формат, який створений Apple для QuickTime Player і який уважається одним з найбільш привабливих форматів відео. Може зберігати аудіо, текстові та відеоефекти як наслідок, його функції і якість займають багато місця. Також цей формат сумісний з великою кількістю платформ онлайн-обміну
AV1 (AOMedia Video Codec 1.0)	Порівняно новий відеокодек (рік випуску 2018-й), спеціально розроблений для потокового передавання в інтернеті. Тобто він має дійсно гарні можливості стиснення. AV1 забезпечує якість відео Ultra HD без ліцензійних відрахувань. Оскільки його розробка підтримується великими галузевими гравцями (такими як Google, Apple, Netflix, Amazon та ін.), очікується, що найближчими роками формат буде підтримуватися пристроями та платформами всіх цих компаній. Наразі AV1 підтримується останніми версіями Chrome і Firefox
MKV	Формат відео контейнера високої чіткості, включає відео, аудіо, субтитри тощо. У ньому може зберігатися кілька звукових доріжок на різних мовах, під час перегляду навіть у вигляді файлів розміром у кілька гігабайт не буде «провисання» і уповільнення

Запитання для перевірки знань

1. Що таке мультимедіа? Наведіть приклади об'єктів мультимедіа.
2. Що таке кодек? З чого він складається?
3. Що означає стиснення? Навіщо використовувати формати із стисненням?
4. У чому перевага використання форматів **WMA** і **WMV**, коли ви працюєте на комп'ютері з ОС **Windows**?
5. Поміркуйте, навіщо було створено Експертну групу з рухомих зображень.
6. Чи відрізняється кодування тексту і звуку?

5.2. Програмне забезпечення для опрацювання об'єктів мультимедіа

Пригадайте, якими програмами можна опрацьовувати аудіо- та відеодані.

Для опрацювання об'єктів мультімедіа розроблено багато програм: плеєри, грабери, конвертори, редактори, студії.

Програвач мультимедіа (медіаплеєр, англ. *Media player*) — вид комп'ютерних програм, призначених для відтворення мультимедійних файлів.

Існує величезна кількість різноманітних плеєрів (рис. 5.3). Серед них користувачі завжди можуть обрати потрібний їм.

Деякі мультимедіапрогравачі призначені для відтворення тільки аудіо- або відеофайлів і називаються, відповідно, програвачі аудіо (*аудіоплеєри*) і програвачі відео (*відеоплеєри*).

Розглянемо деякі з безкоштовних плеєрів і їх опис.

Програма	Опис
VLC Media Player	Потужний програвач для відтворення більшості аудіо і відеоформатів (MPEG-1, MPEG-2, MPEG-4, DivX, DVD, VCD), що поєднує високу якість і легкість в управлінні. Однією з особливостей є можливість програвання практично будь-якого типу потокового (*streaming*) відео
MKV Player	Безкоштовний відеоплеєр, що дозволяє відтворювати на комп'ютері файли у форматі MKV і забезпечує повноцінне підтримання цього формату
Media Player Classic	Потужний медіаплеєр, що має багато корисних особливостей і налаштувань, наприклад відтворення другої звукової доріжки
KMPlayer	Безкоштовний програвач, який відтворює більшість популярних аудіо і відеоформатів, має широкий вибір скінів і кольору оформлення, можливість програвання певного відрізка матеріалу (початкова та кінцева точки)



Рис. 5.3. Логотипи найпопулярніших плеєрів

Грабери — програми, які дають змогу захопити відео (від англ. *Video capture* — захоплення відео), процес перетворення відеосигналу із зовнішнього джерела в цифровий відеопотік за допомогою персонального комп'ютера та його запису у відеофайл з метою подальшого оброблення, зберігання або відтворення.

Захопити відео можна із зовнішнього носія або з екрана монітора (вести запис ігор, фільмів, робити скріншоти, причому як усього вікна, так і обраної ділянки). Більшість з них має повний набір необхідних інструментів для захоплення відео з екрана зі звуком.

До найбільш поширених граберів належать: Movavi Screen Capture, Bandicam, Camtasia Studio, Icecream Screen Recorder. Більш детально про роботу з програмами захоплення відео ми поговоремо далі (§ 5.3).



Рис. 5.4. Піктограми найпоширеніших конверторів

Конвертори — комп'ютерні програми, які перетворюють дані, подані у файлі, з одного формату в інший.

Існує багато як платних, так і безкоштовних конверторів (рис. 5.4). Усі їх легко використовувати, вони мають інтуїтивно зрозумілий інтерфейс, великий вибір підтримуваних форматів, високу швидкість конвертації, багатофункціональність. Поміж найбільш популярних необхідно насамперед назвати Format Factory, Movavi Video Converter, Wise Video Converter Pro, Any Video Converter Free, Total Video Converter.

Ще одним суттєвим представником програмного забезпечення для опрацювання мультимедіа є відеоредактори.

Існує багато відеоредакторів: PinnacleStudio, Free Video Editor, MS Producer, Adobe After Effects, Adobe Premiere, Ulead MediaStudio, SONY Vegas Pro.

Відеоредактор — це програма, що містить набір інструментів, за допомогою яких створюють і редагують відеофайли на комп'ютері.

Відеоредатори поділяються на дві групи: ліцензійні (тобто платні та з відкритим кодом) і безкоштовні. Ознайомимося з трійкою лідерів серед ліцензійних відеоредакторів.

Програма	Опис
Adobe Premiere Pro	Вважається провідним програмним пакетом для редагування відео як у галузі кіноіндустрії та телебачення, так і інтернету. Творчі інструменти, інтеграція з іншими програмами й службами, а також потужні технології дають змогу перетворювати відеозаписи на дивовижні фільми та відео.
Final Cut Pro	Відеоредактор від компанії Apple трохи поступається в популярності. Проте визнається одним із кращих інструментів для професійного монтажу відео. Перевагою цього відеоредактора є спрямованість на роботу з дронами
Vegas Pro	Як і будь-який інший відеоредактор, містить велику колекцію звукові ефектів та професійних аудіо плагінів; при монтажі дозволяє робити динамічну розкадровку, має вбудовані таймлайни; ланцюжки спецефектів на 4х різних рівнях, професійні опції грейдингу, стабілізація відео та відстеження руху

Серед безкоштовних програм необхідно назвати насамперед два відеоредактори Canva і Filmora Video Editor. Зупинимось на них трохи детальніше.

Canva (canva.com/uk_ua/) — це безкоштовна онлайн-платформа для дизайну, за допомогою якої ви можете створювати відео. Містить велику бібліотеку шаблонів, стандартних відео, анімованих елементів і композицій для створення короткого відео та його публікації в інтернеті.

Можна використовувати фотографії, зображення значки, ілюстрації, анімації, наклейки, та інших графічних елементів, створених професійними дизайнерами, або створювати власний дизайн разом з іншими користувачами за допомогою інструменту співпраці.

Не менш популярним відеоредактором є Filmora Video Editor. Ця програма не займає багато місця на жорсткому диску, швидко завантажує матеріал і так само швидко працює, не є ресурсозатратною. Відеоредактор дозволяє виконувати монтаж будь-якої складності, працювати з різними форматами, має величезну кількість різноманітних вбудованих ефектів. Крім того у вільному доступі в магазині від виробника можна доповнити різними переходами, ефектами, музичними композиціями тощо. Робота з відео на часовій стрічці (*timeline*) проста і інтуїтивно зрозуміла.

Зареєструватись у Canva дуже легко: за допомогою облікового запису Facebook або Google. Після цього можна обрати тип дизайну: Відео, Відео для Facebook, Відео зі слайдами, Відеоколаж, Відео для YouTube, Історія для Instagram або Заставка для YouTube.

Часте оновлення та нарощування функціоналу робить Filmora Video Editor відмінним стартовим інструментом для опанування відеомонтажу новачком.

Запитання для перевірки знань

1. Опишіть класифікацію систем опрацювання мультимедійних даних.
2. Наведіть приклади використання граберів та конверторів.
3. Які системи опрацювання аудіо- та відеоданих ви використовуєте?
4. Наведіть приклади застосування відеоректорів.

5.3. Захоплення аудіо й відео, створення аудіо- і відеофрагментів

Чи доводилося вам створювати власний відеокліп?

Перш ніж створити кліп, потрібно принаймні отримати відео. А щоб записати відео з екрана монітора, необов'язково стояти перед ним з відеокамерою. Існує безліч програм, що захоплюють все, що відображається на екрані, і записують звук, який виводиться на колонки чи навушники (рис. 5.4).

Ці програми безкоштовні та платні, з потужним функціоналом і мінімалістичні тощо. Ознайомимося з їх описом у таблиці детальніше. Ми ж з вами працюватимемо з ShareX (getsharex.com/) — однією з найкращих безкоштовних програм для захоплення відео та створення скриншотів.

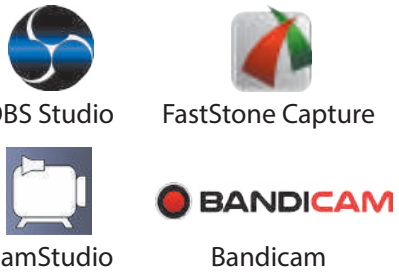


Рис. 5.4. Логотипи програм захоплення

Програма	Опис
OBS Studio	Безкоштовна багатофункціональна програма, яка дозволяє вести запис або трансляцію в мережі не лише того, що відбувається безпосередньо на екрані, а й відео з інших пристроїв (наприклад, з вебкамери).
FastStone Capture	Дуже потужна безкоштовна програма для користування в особистих цілях та освітній діяльності. Вона може якісно записувати відео з екрана, робити скриншоти, редагувати та переглядати їх; може записувати відео та створювати скриншоти, як весь екран, так і окрему його частину. FastStone Capture виконана в стилі «мінімалізм», її меню достатньо компактне та зручне. До недоліків можна віднести орієнтацію на ОС Windows. Програма дозволяє зберігати результат роботи в найбільш поширених відеоформатах, має багато плагінів та стабільно працює практично в усіх операційних системах.
CamStudio	Безкоштовна програма, призначена для записування всього, що знаходиться на екрані комп'ютера. Дозволяє вести запис у форматах AVI, MP4 і SWF. FastStone Capture — непоганий вибір для тих, хто створює навчальні відеокурси, презентації, ігрове відео. Серед переваг — і можливість додавати ефекти при переміщенні та натисканні курсора. FastStone Capture підтримує формати BMP, GIF, JPEG, PCX, PNG, TGA, TIFF і PDF.
Bandicam	Платна програма для тих, хто любить записувати свої дії в іграх. Це дуже якісна програма для захоплення відео та створення скриншотів як всього екрана, так і його частини. Bandicam дозволяє записувати відео практично в будь-яких іграх: та у найрізноманітніших додатках: Skype, вебкамери, відеочати, просто з робочого столу.

Програма ShareX встановлюється буквально у два натискання. Після першого запуску ShareX у головному вікні ви побачите поточні налаштування гарячих клавіш для запису екрана або його області, а також створення скриншотів (рис. 5.5).

Доступ до інших функцій захоплення, наприклад, записування відео в анімований GIF або розпізнавання тексту зі скриншоту, можна знайти в розділі Захоплення (рис. 5.6).



ShareX

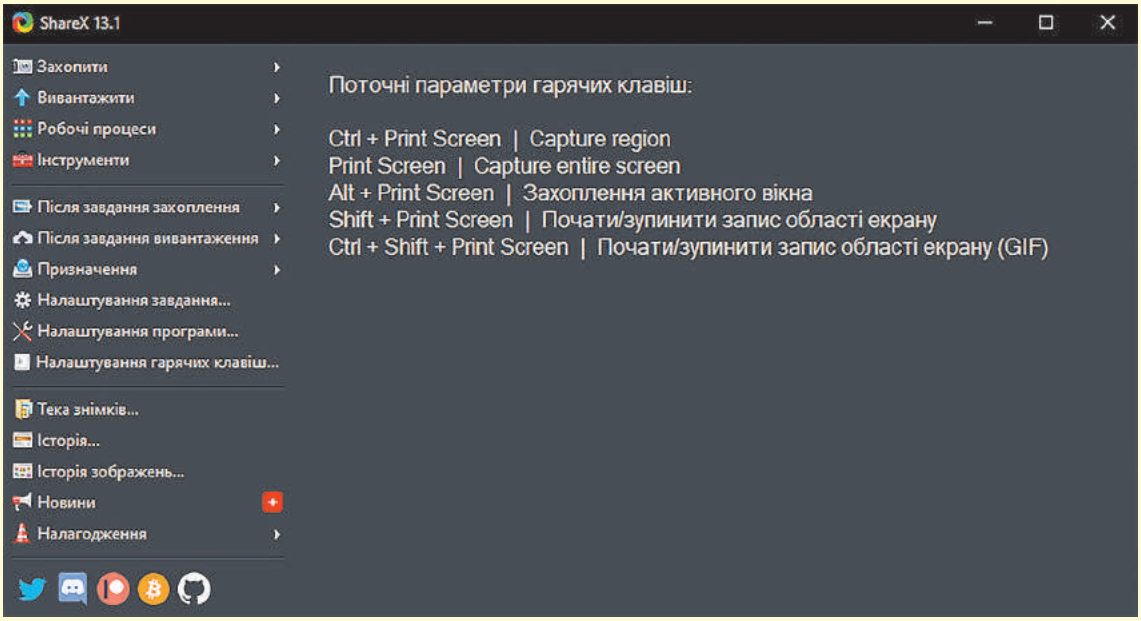


Рис. 5.5. Головне вікно програми

Відкривши пункт Після завдання захоплення ви побачите, що саме програма буде виконувати відразу після того, як був зроблений знімок екрана або записано відео робочого столу (виділені пункти — включені). Ви можете відімкнути непотрібні пункти і ввімкнути необхідні.

Якщо відімкнути Зберегти зображення в файл і ввімкнути Зберегти зображення в файл як, то під час збереження скриншота або відео відображатиметься діалог вибору місця збереження (інакше — збереження в стандартну папку) (рис. 5.7).

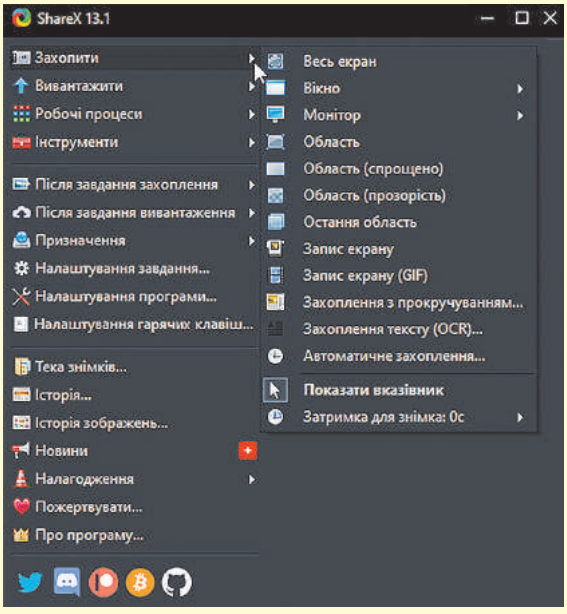


Рис. 5.6 . Меню команди «Захопити»

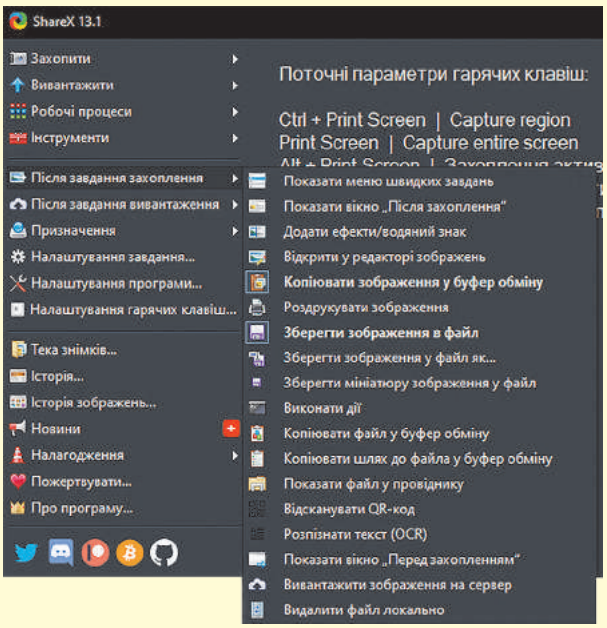


Рис. 5.7. Меню команди «Після завдання захоплення»

Для роботи програми важливо здійснити налаштування: Налаштування завдань → Запис екрану → Налаштування запису екрану. Якщо пункт Шлях до ffmpeg виділено червоним, то натисніть Завантажити і завантажте його.

У вікні Налаштування завдань можна налаштувати параметри експорту відео (для більшості завдань, пов'язаних із записом робочого столу, вже виставлені оптимальні параметри).

Компонент ffmpeg потрібний для кодування відео з екрана, а також використовується у вбудованому конвертері відеофайлів.

Алгоритм записування звуку

Крок 1 Активуйте команду **Джерело звуку** у вікні Налаштування завдань (рис. 5.8)

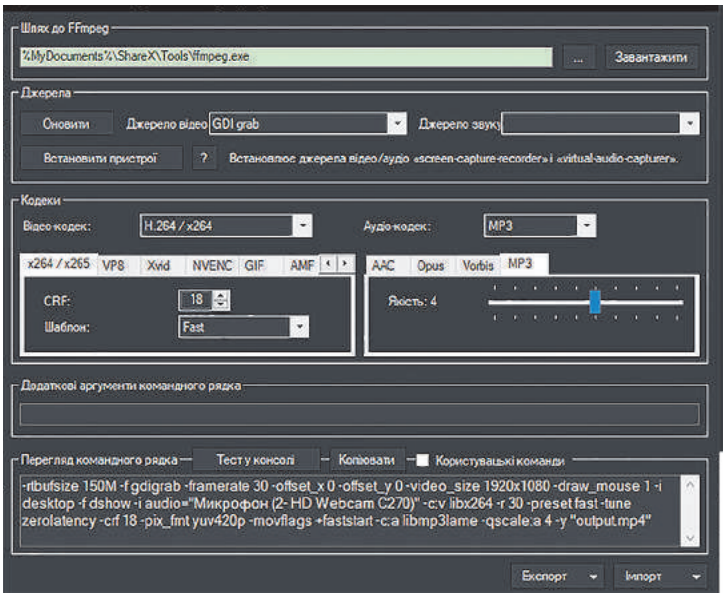


Рис. 5.8. Меню «Налаштування команд»

Крок 2 Натисніть сполучення клавіш SHIFT+PrintScreen і виділіть курсором ту ділянку екрана, яку потрібно записувати

Крок 3 Натисніть клавіші SHIFT + PrintScreen — це приведе до зупинки захоплення відео та появи в головному вікні файлу з результатом роботи програми

Виконавши команду Налаштування завдань → Завантаження → Імена файлів, можна задати шаблони для імен скриншотів і відео. Доступні для використання в шаблонах параметри з'являться в меню при натисканні в поле введення шаблону.

За допомогою команди Налаштування завдань → Захоплення → Розпізнавання тексту можна змінити мову за замовчуванням для розпізнавання тексту зі скриншотів на одну з тридцяти запропонованих. За замовчуванням вибрано англійську.

У розділі Гарячі клавіші можна задати власні поєднання клавіш для різних дій, додати нові дії з новими поєднаннями або видалити ті, які не використовуються.

У списку зроблених скриншотів і записаних відео можна виконати дії над ними, клацнувши ПКМ потрібний елемент. До речі, скриншоти до матеріалів параграфа зроблено за допомогою програми ShareX.

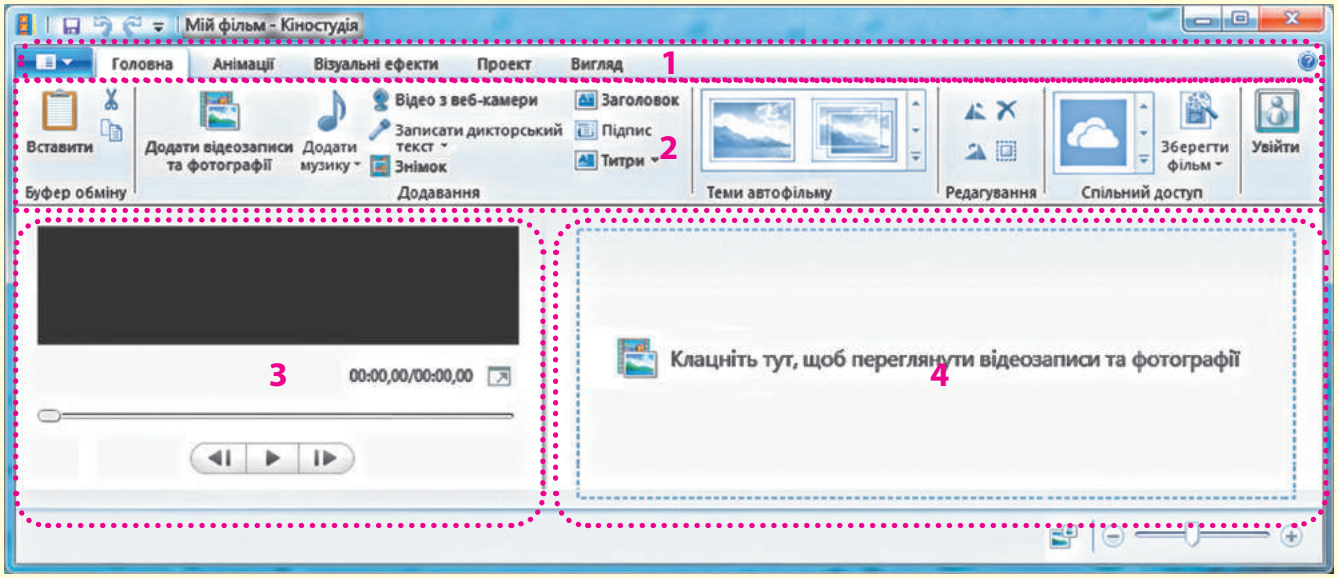
Запитання для перевірки знань

1. Коли доречно скористатися програмою захоплення відео?
2. Які основні функції програм захоплення відео? Наведіть приклади таких програм.
3. Перелічіть способи створення скриншоту.
4. Опишіть алгоритм запису анімованого GIF.
5. Як можна розпізнати текст зі скриншотів?

5.4. Побудова аудіо- й відеоряду. Додавання до відеокліпу ефектів

Ви готові приступити до створення відеокліпу?

Для створення відеокліпів ми користуватимемося кіностудією Windows Live — це компонент ОС Windows. Робоче вікно Кіностудії Windows поділено на чотири ділянки (рис. 5.9):



1 — область вкладок, де можна побачити основні напрямки роботи та способи налаштування програми.
Це вкладки **Основне**, **Анімація**, **Візуальні ефекти**, **Проект**, **Вид**, а також основне меню самої програми, де вона пропонує створити, відкрити або зберегти проєкт, опублікувати або зберегти фільм і отримати довідкову інформацію щодо програми

2 — панель управління, яка залежно від вибраної вкладки має різний набір інструментів і відповідно вигляд

3 — вікно попереднього перегляду, де можна прокрутити завантажений відеоролик і побачити результати своїх дій

4 — вікно проєкту (редагування або розкадровки), де виконуються всі робочі операції з фото-, відео- і аудіоматеріалів

Рис. 5.9. Робоче вікно Кіностудії Windows

Проєкт — файл, який містить відомості про порядок розташування та час відтворення аудіо- й відеокліпів, відеопереходи, відеоефекти, назви, титри тощо. Після збереження проєкту його файл можна відкрити пізніше в середовищі відеоредактора, щоб внести до нього зміни.

Проєкти, створені у програмі Кіностудія, мають розширення .wlmp.

Для додавання змісту (контенту) відкрийте панель керування вкладки Основне, де на цей випадок є цілий набір опцій у відповідному розділі. Завантажте фотографію (рис. 5.10) і відеозапис (рис. 5.11) з комп'ютера. Знайдіть в інтернеті або додайте з комп'ютера файл з музичним записом і захопіть відео з вебкамери (захоплення із записом відеозвернення).

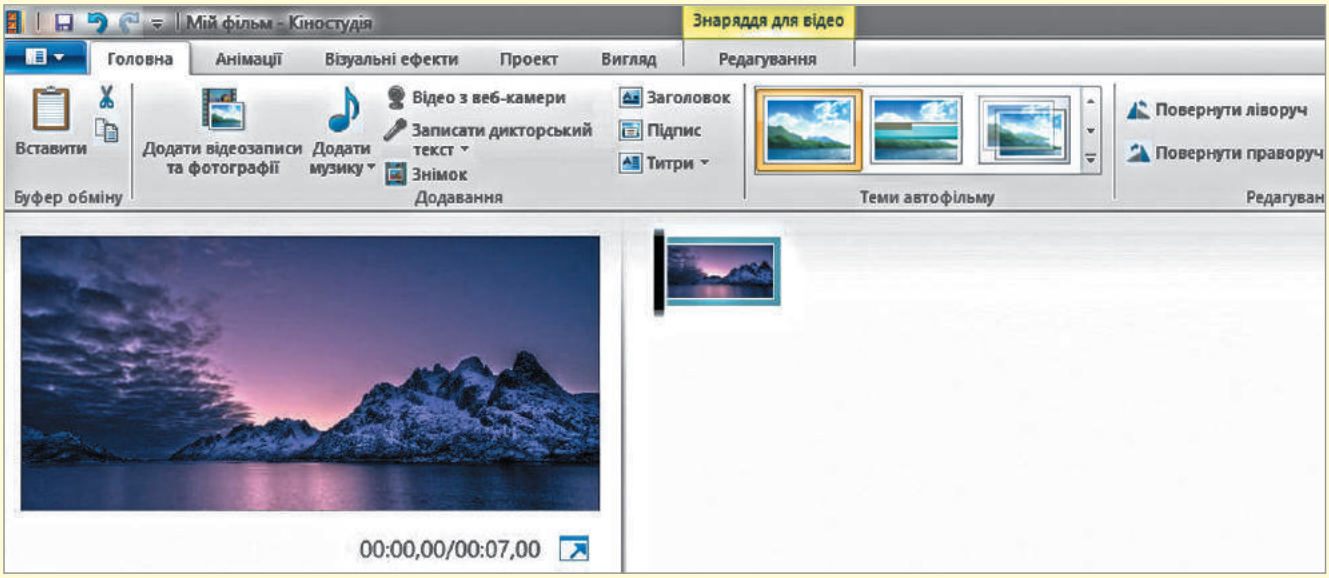


Рис. 5.10. Вигляд робочого вікна після додавання фотографії.

Далі запишіть закадровий текст (зручна функція для озвучування відеороликів), зробіть моментальний знімок (своєрідний Принтскрин вашої роботи) і додайте текстовий супровід на початку і в кінці відеоролика (Назва та Титри).

Існують різні способи додавання відеозаписів і фотографій на початку роботи над новим проєктом.

Курсор виглядає як чорна вертикальна лінія, яку можна встановити в будь-якій частині завантаженого відеофрагмента. Щоб отримати більш точне розташування, можна просто захопити курсор мишею і відстежити потрібний кадр у вікні попереднього перегляду.

Оперуючи великими відеофрагментами, можна розтягнути «стрічку часу», клацнувши її правою клавішею миші (ПКМ) вибравши Збільшити (+).

Способи додавання відеозаписів і фотографій

- Вибрати пункт **Перетягніть сюди відеозаписи та фотографії** або клацнути, щоб переглянути їх
- Натиснути піктограму **Додати відеозаписи та фотографії** на вкладці **Основне**
- Додати потрібне одночасно в декілька файлів, утримуючи Ctrl або Shift, натиснути кнопку **Відкрити**

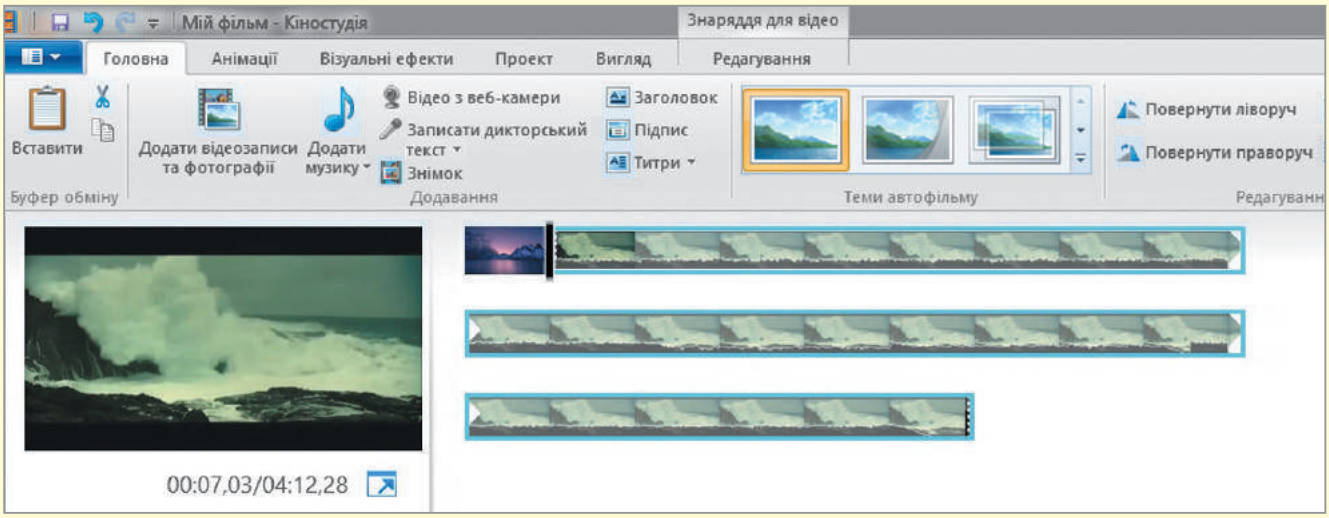


Рис. 5.11. Вигляд робочого вікна після додавання відео.

Для нарізання відеоряду існує низка інструментів (рис. 5.12).

Інструмент	Опис
Розділити	Міститься на панелі управління вкладки **Правка** у **Засобах роботи з відео**; викликається ПКМ по курсору **Розділити** — розрізає відеоряд на відрізки
Встановити початкову точку	Розташовується аналогічно; видаляє з відеоряду всі приготовані кадри
Встановити кінцеву точку	Видаляє з відеоряду всі наступні кадри
Засіб усічення	Дозволяє задати часовий проміжок фрагмента і автоматично вирізає зайве

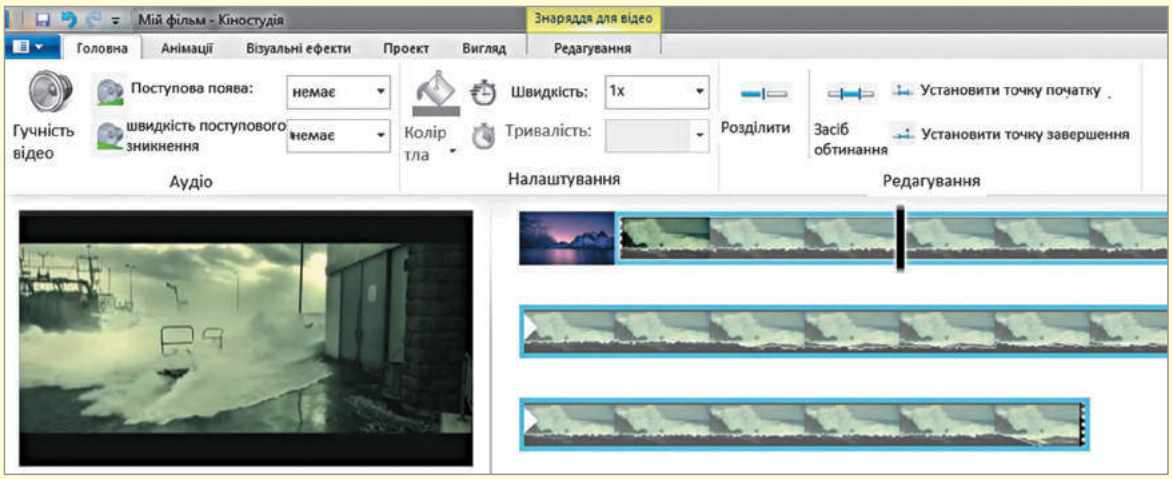


Рис. 5.12. Відображення команд меню «Знаряддя для відео»

Автофільм є найпростішим способом привернути увагу глядачів до вмісту відеоролика.

Додавши фотографії та відеозаписи до проєкту, натисніть кнопку Автофільм на вкладці Основна, після чого буде автоматично додано переходи з перехресним затуханням, автоматичні ефекти панорамування та масштабування, заголовок і титри, а також відобразиться запит на додавання саундтреку.

Якщо натиснути кнопку Так, за допомогою функції Автофільм вміст буде автоматично пристосовано до музики. Якщо вибрати команду Автофільм, усе ще можна повернути.

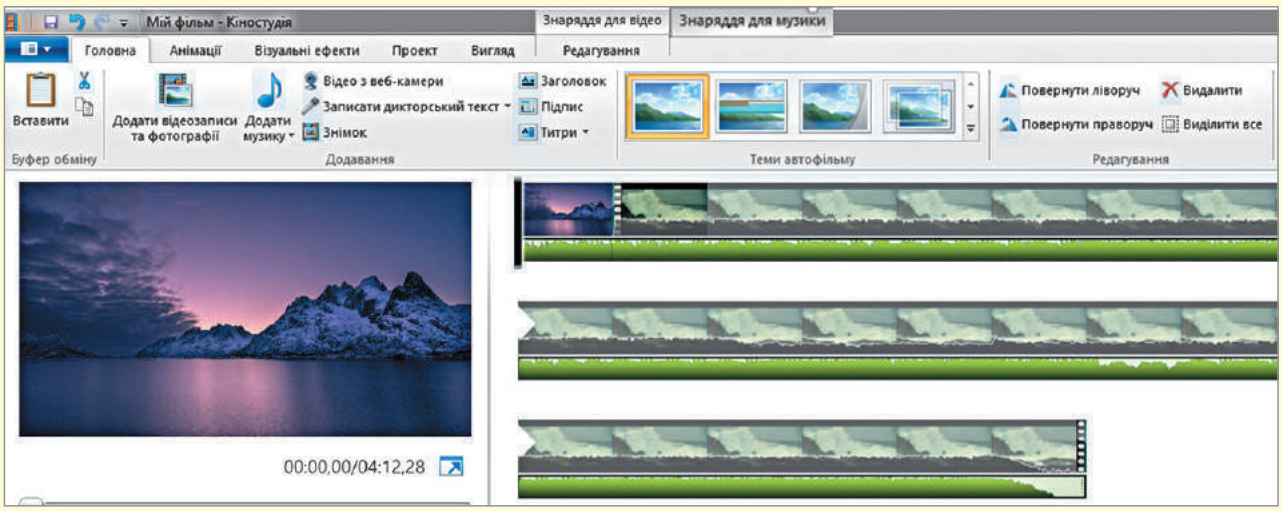


Рис. 5.13. Вибір теми Автофільму

Щоб позначити назву відеоролика на початку, забезпечити його титрами в кінці і забезпечити можливість текстового супроводу в процесі демонстрації, на вкладці Основне є такі можливості (рис. 5.13):

- Назва — розміщує титри на початку фільму;
- Тема — дозволяє супроводжувати підписами окремі кадри;
- Титри (з можливістю вибору зі списку) — розміщується в основному в кінці фільму.

Алгоритм додавання саундтреку до фільму

Крок 1	Натисніть кнопку **Додати музичний файл** на вкладці **Основне** на стрічці
Крок 2	Виберіть потрібну композицію та натисніть кнопку **Відкрити**. Після додавання музики буде доступна вкладка **Знаряддя для музики — Параметри**. У функції **Автофільм** відобразиться запит на додавання саундтреку до фільму. Якщо вибрати пункт **Ні**, то саундтрек можна додати пізніше. Щоб розділити композицію в окремій точці аркуша розкадрування, слід вибрати фотографію або відеозапис, перед яким слід її розділити
Крок 3	Клацніть **Розділити** на вкладці **Знаряддя для музики — Параметри на стрічці**
Крок 4	Перетягніть і перемістіть композицію в будь-яке потрібне місце на аркуші розкадрування
Крок 5	Щоб додати кілька композицій до фільму, виберіть фотографію або відеозапис, до яких потрібно додати нову композицію
Крок 6	Клацніть стрілку вниз у правому нижньому куті кнопки **Додати музичний файл** на вкладці **Основне**, щоб отримати доступ до розкривного меню
Крок 7	Клацніть **Додати музичний файл** у цій точці. Виберіть іншу композицію

Алгоритм додавання заголовків, підписів і титрів

Виберіть фотографію або відеозапис, перед яким потрібно відобразити заголовок. Натисніть кнопку Заголовок на вкладці Основне на стрічці і введіть заголовок фільму в області з написом [Введіть текст тут]. Можна також: змінити шрифт, розмір, колір, ефекти та інші параметри заголовка за допомогою вкладки Знаряддя для тексту — Форматування на стрічці; визначити за тривалістю тексту, скільки секунд відображатиметься заголовок; додати підписи до будь-якої частини фільму.

Потім відтворіть фільм і в потрібній точці натисніть кнопку Підпис на вкладці Основне стрічки. Можна відредагувати появу підпису, його відображення на екрані тощо за допомогою вкладки Знаряддя для тексту — Форматування

Додайте своє ім'я в титрах у кінці фільму. Клацніть останню фотографію або відеозапис на аркуші розкадрування. Натисніть кнопку Титри на вкладці Основне на стрічці. Відредагуйте появу титрів, їхнє відображення на екрані тощо за допомогою вкладки Знаряддя для тексту — Форматування.

Для додавання переходів та ефектів потрібно додати переходи до вмісту, то виберіть вкладку Анімації на стрічці, щоб

- Можна навести вказівник миші на параметри, щоб переглянути в режимі реального часу, яка одна фотографія або відеозапис переходитиме до іншої у вікні попереднього перегляду.
- Щоб переглянути інші параметри, необхідно клацнути стрілку вниз у правому нижньому кутку.

відобразити параметри переходу від однієї фотографії або відеозапису до іншої (рис. 5.14).

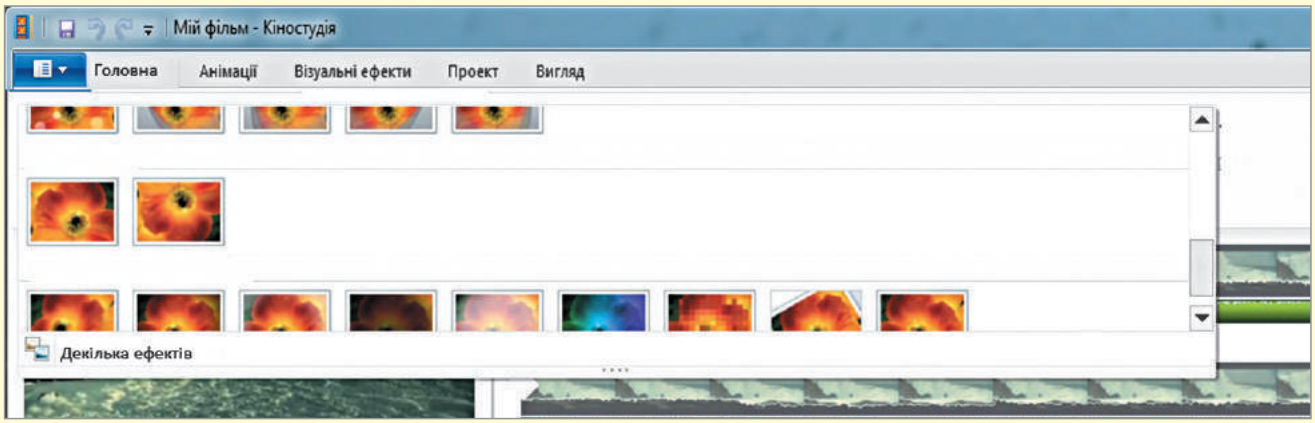


Рис. 5.14. Вибір ефектів

Функція Автофільм дає змогу автоматично додати перехід із перехресним затуханням до кожного фрагмента вмісту, але змінити перехід так само просто, як і вибрати фотографію або відеозапис. Далі слід перейти на вкладку Анімації і вибрати інший перехід, щоб застосувати його до вмісту.

Алгоритми налаштування додаткових можливостей такий: знайдіть і клацніть потрібний перехід, щоб автоматично додати до відеозапису. Щоб додати перехід до кількох елементів, виберіть діапазон, клацнувши фотографію або відеозапис, з яких потрібно розпочати, а потім виберіть файл, яким завершуватиметься фільм, утримуючи натиснутою клавішу Shift

Виберіть потрібний перехід і застосуйте його до вибраного діапазону (рис. 5.15).

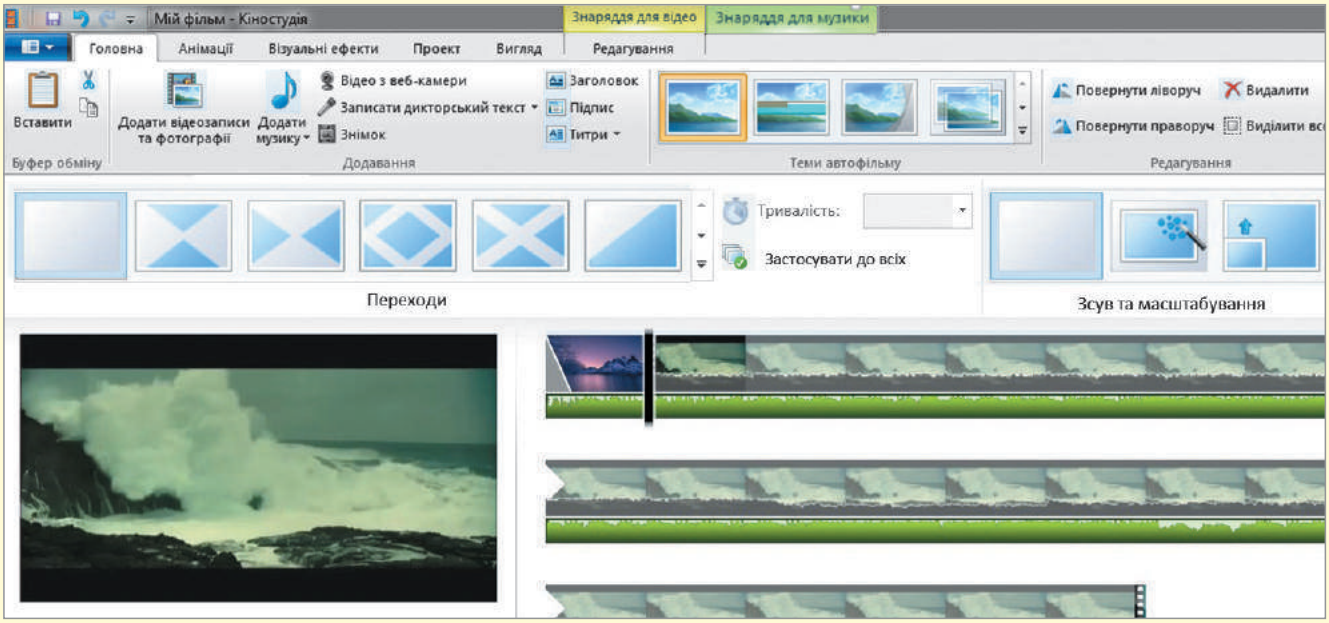


Рис. 5.15. Вибір переходів між фрагментами (відео чи фотографій)

Для додавання ефектів до фотографій виберіть вкладку Анімації на стрічці, щоб відобразити параметри панорамування та масштабування окремих фотографій під час їхнього відображення. Клацніть стрілку вниз у правому нижньому куті для перегляду інших параметрів. За допомогою функції Автофільм до фотографій автоматично додаються різноманітні ефекти панорамування та масштабування.

Якщо потрібно змінити ефект панорамування та масштабування, перейдіть на вкладку Анімації та (вибравши фотографії) виберіть будь-який ефект. Перейдіть на вкладку Візуальні ефекти на стрічці, щоб відобразити ефекти, які можна застосувати до фотографій і відеозаписів.

Наведіть вказівник миші на кожен ефект, щоб переглянути, як він виглядатиме, перш ніж застосувати його до фотографії або відеозапису. Знайшовши потрібний ефект, клацніть його, щоб автоматично додати.

Щоб видалити ефект, не застосовуючи його до вмісту, потрібно клацнути **Без ефектів** у меню візуальних ефектів (перше ліворуч). Після цього ефект зникне.

Запитання для перевірки знань

1. Що таке відеоредактор?
2. Назвіть основні об'єкти вікна програми **Кіностудія Windows**.
3. Як додати титри до відеокліпу?
4. Як вставити до проєкту відеофільму відео або фото з файлу?
5. Як додати ефект до відеофільму?
6. Опишіть призначення функції **Автофільм**.

5.5. Налаштування часових параметрів аудіо- та відеоряду. Засоби перетворення аудіо- й відеоформатів

Як, на вашу думку, можна урізноманітнити відео?

Відеозаписи, записані на відеокамеру чи імпортовані з файлів, можуть містити й звук. Дуже рідко трапляється ситуація, коли тривалість відео та аудіо збігається, тому виникає необхідність налаштовувати часові параметри.

Тривалість відображення зображень, заголовка та титрів можна змінювати за допомогою відповідної властивості в групі Настроювання (рис. 5.16).

Якщо аудіозапис відтворюватиметься не повністю, до нього можуть бути застосовані ефекти Поступова поява або Швидкість поступового зникнення в групі Аудіо на вкладці Редагування. Крім того, можна регулювати гучність відтворення звуку чи вимкнути звук для відеокліпу.

Якщо відеозапис не містить звуку або не потрібно відтворювати наявний звук, до відеокліпу можна додати музичний супровід у вигляді аудіозапису чи звуковий коментар.

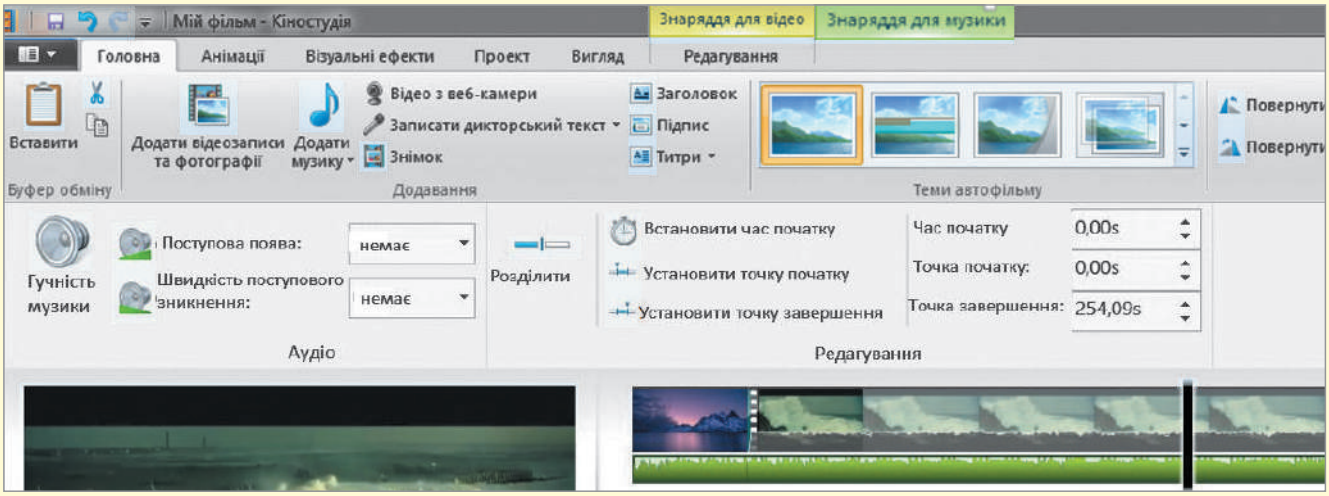


Рис. 5.16. Відображення коман меню «Знаряддя для музики»

Якщо тривалість аудіозапису довша, ніж відеоряду, аудіозапис буде автоматично «обрізано» наприкінці відеокліпу.

У Кіностудії Windows одночасно можна відтворювати лише один саундтрек, тобто водночас звук із фільму, саундтрек і дикторський текст відтворити не можна. Для цього потрібно виконати певні дії. Для додавання дикторського тексту (за умови, що вміст у проєкті Кіностудії розташовано у правильному порядку) відкрийте звукозаписувач у папці Стандартні. Відкривши проєкт Кіностудії Windows Live у фоновому режимі, натисніть кнопку Почати записування у звукозаписувачі та елемент керування Відтворити у проєкті Кіностудії. Додайте дикторський текст до фільму. Промовивши дикторський текст, натисніть кнопку Зупинити записування у звукозаписувачі та збережіть аудіофайл у папці на комп'ютері.

Поверніться до Кіностудії Windows Live, знайдіть точку на часовій шкалі аркуша розкадрування, на якій слід додати текст (з початку або в поточній точці). Виберіть команду Додати музичний файл під зображенням ноти на вкладці Основне. Перейдіть до аудіофайлу і додайте до проєкту.

Дикторський текст можна відредагувати, клацнувши вкладку Знаряддя для музики на стрічці. Спочатку слід перемістити аудіодоріжку в місце у фільмі, де потрібно розпочати дикторський текст, а потім почати відтворення фільму. Можливо, звук і відео не синхронізовані; не звертаючи уваги на відео, слід прослухати фрагмент тексту, у якому має починатися звук, а потім клацнути Установити точку початку. Звук буде налаштовано так, щоб початкова точка перебувала в місці, у якому зразу було розташовано аудіодоріжку, що сприяє ефективній синхронізації звуку з фільмом.

Для додавання саундтреку та дикторського тексту створіть фільм, як планувалось (за допомогою функції Автофільм або вручну), і додайте саундтрек. Експортуйте фільм (залежно від роздільної здатності, завжди рекомендується значення 1080 р). Експортувавши та зберігши фільм, відкрийте новий проєкт у Кіностудії Windows Live. Виберіть команду Додати

Способи отримання відеозапису чи аудіокліпу потрібної тривалості

- Приховати початок або кінець певних відео- чи аудіокліпів
- Змінити швидкість відтворення відео в групі Настроювання на вкладці Редагування відповідно для зменшення чи збільшення тривалості кліпу
- Додати до відеокліпу заголовок, титри тощо

відеозаписи та фотографії; додайте нещодавно експортований фільм. Клацніть Додати музичний файл. Виконайте наведені вище кроки, щоб додати дикторський текст. Щоб відрегулювати рівень звуку, клацніть Змішати на вкладці Основне, щоб підвищити гучність фільму порівняно з аудіодоріжкою (у цьому разі дикторським текстом) або, навпаки, пересунувши смугу ліворуч або праворуч.

Існують платні і безкоштовні конвертери. Відзначимо Format Factory — багатофункціональний безкоштовний конвертер для перетворення відео, аудіо форматів та файлів зображень. Насправді це графічний інтерфейс для трьох найефективніших інструментів конвертації: FFmpeg для відео, Mencoder для аудіо і CxImage для зображень. Format Factory конвертує більшість наявних відео- та аудіоформатів. Його інтерфейс простий у використанні та інтуїтивно зрозумілий.

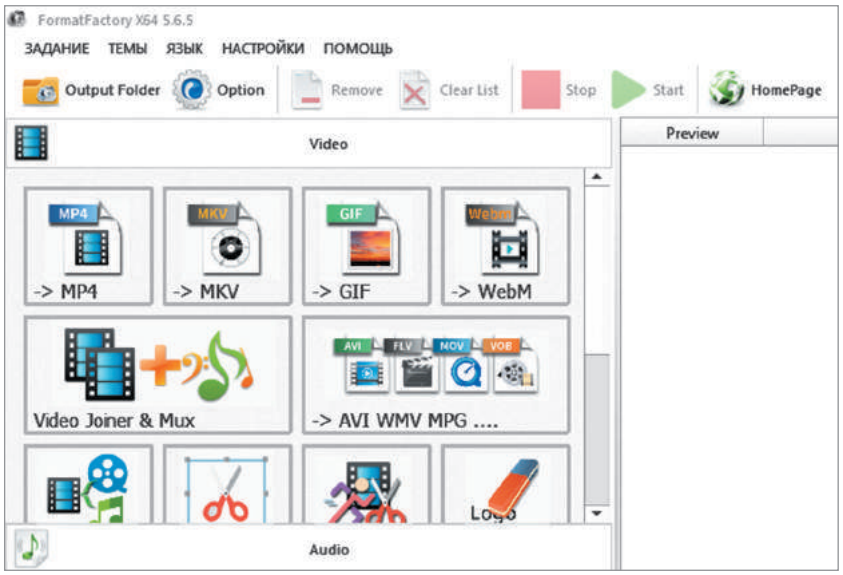


Рис. 5.17. Головне меню програми "Formft Factory"

Кіностудія Windows Live працює не з усіма форматами як в якості завантаження, так і зберігає результат роботи лише у двох форматах: MP4 і WMV, якщо йдеться про відео, та M4A і WMa для аудіо.

Вибір конвертера залежить від вихідного формату файлу та формату, в який його потрібно перетворити. Конвертація файлів полягає в переписуванні файлів в іншому форматі.

Що стосується перетворення зображень, то Format Factory не лише надає можливість конвертації в популярні формати, а й має додаткові функції — зокрема, збільшення і обертання зображень, та присвоєння їм тегів.

Запитання для перевірки знань

1. Назвіть причини необхідності налаштування часових параметрів.
2. Чим відрізняється налаштування часових параметрів для аудіо від відео?
3. Що таке конвертер, коли його доводиться використовувати?
4. Як записати дикторський текст?
5. Чи можна використовувати одночасно декілька саундтреків?
6. Опишіть послідовність дій при додаванні дикторського коментаря у кліп, до якого ви вже додали аудіодоріжку.

Завдання для самостійного виконання

1. Запишіть на смартфон невелике відео вашої прогулянки. Надішліть собі на пошту для подальшого завантаження на комп'ютер. Відконвертуйте ваше відео за допомогою онлайн-сервісу (наприклад https://www.online-convert.com/) або FormatFactory у формат MP4. Завантажте на Google-диск і надайте посилання вчителю.

5.6. Сервіси для роботи з аудіо- й відеоданими та публікування їх в інтернеті. Подкаст

Якими інтернет-сервісами ви користуєтесь?

Коли людина створить власний відеокліп, то, безумовно, захоче поділитись своїми здобутками з максимально великою кількістю людей. А це означає, що відео потрібно опубліку-вати в інтернеті. Звісно, можна викласти кліп на власну сторінку в соціальних мережах, але там його побачать тільки друзі та подруги, або підписники. Тому мова зараз піде про відеохостинг і онлайн-платформи публікації відео в інтернеті.

Відеохостинг — сайт, який дозволяє завантажувати та переглядати відео у браузері, наприклад через спеціальний програвач.

Компанія може використовувати контент на сайті або в соціальних мережах для просування бренду, і часто UGC виявляється набагато ефективніше звичайного контенту.

Подібні сервіси слідують принципу — контент генерує користувач (User-generated content, UGC, або користувацький контент). UGC — це оригінальний контент, який створюється аудиторією бренду. Таким контентом може бути все що завгодно — починаючи від відгуків і коментарів у блозі і закінчуючи фото- і відеороликами.

Наразі безумовним лідером серед відеохостингів є YouTube. Він дуже добре зарекомендував себе в соціальних мережах, блогосервісах і медіакаталогах.

Більшість сайтів відеохостингу тематично не обмежують своє наповнення. Однак деякі відео-хостинги займають спеціалізовані сектори, пропонуючи тематичні портали. Особливе місце займають сервіси публікації наукового, науково-популярного і навчального відеоконтенту.

Використання UGC вигідно і компанії, і користувачам. Компанії воно вигідно отриманням фантастичної кількості свіжих ідей і унікального контенту, на генерування яких у неї могло б піти багато безсонних ночей або серйозних грошових вкладень. Створення UGC дозволяє користувачам співпрацювати з улюбленим брендом, реалізувати себе творчо.

Підтримка форматів:
- Windows Media Video (AVI)
- 3GP (стільникові телефони)
- AVI (Windows)
- MOV (Mac)
- MP4 (ipod/портативні системи)
- MPEG, FLV (Adobe Flash)
- MKV (H.264). Робота з HD-відео.

Основні вимоги до таких відеохостингів — це скільки місця надається для зберігання відеофайлів, які формати вони підтримують, якість відтворення відео, простота використання.

Можливість додавати відео з мобільного телефону і веб-камери, а також проста навігація та зручне редагування проілюстровано на рис. 5.18. Для завантаження відео необхідно скористатися власним обліковим записом Google.

Для завантаження відео увійдіть у свій обліковий запис і виберіть вказівку Завантажити. Для опублікування завантаженого відео зазначте його назву, опис, ключові слова та інші налаштування для відео, за якими інші користувачі зможуть його знайти. Перегляньте список завантажених відео на каналі та внесіть до них зміни за допомогою Менеджера відео у Творчій студії. За потреби відредагуйте (або видаліть) завантажене відео.

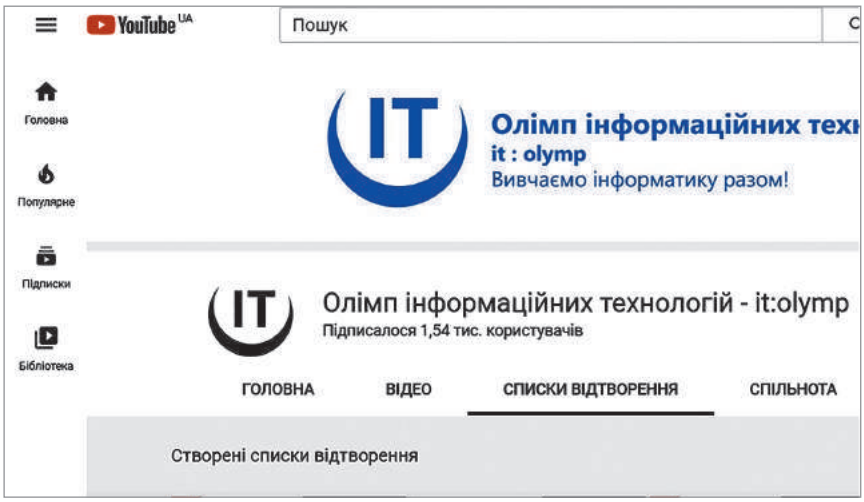


Рис. 5.18. Головне меню програми "Formft Factory"

Кіностудія Wsidows Live надає можливість опублікування відео на YouTube. Для цього потрібно виконати такі кроки. На вкладці Основне стрічки Office виберіть YouTube у розділі Спільний доступ. Відобразиться запит на вхід у службу YouTube за допомогою імені користувача та пароля.

Дотримуйтесь інструкцій зі спливного вікна Публікація в YouTube і введіть заголовок, опис, теги, категорію та дозволи для фільму. Натисніть кнопку Опублікувати — кіностудія опублікує фільм у службі YouTube. Перегляньте фільм в інтернеті (на компютері), або натиснути кнопку Закрити.

Досить популярною платформою є Vimeo (vimeo.com/) — відеохостинг, зручний для використання та роботи з HD-відео. Відео високої якості під час завантаження конвертується в формат дозволу 720 р із частотою 24 кадрів на секунду. На сервісі можна завести платний і безкоштовний облікові записи; платний має набагато більше переваг і можливість заливати HD-відео не лише раз на тиждень.

Подкаст (англ. *podcast*), помилковий підкаст, радіопрограма, — цифровий медіа-файл або низка таких файлів, які розповсюджуються інтернетом для відтворення на портативних медіа-програвачах або комп'ютерах.

За змістом подкасти можуть нагадувати радіошоу, звукову виставу, містити інтерв'ю, лекції тощо, що належить до усного жанру. Подкастинг дозволяє самостійно обирати те, що ви хочете слухати чи хочете дивитися саме у той час, коли вам зручно.

Подкаст регулярно оновлюється відповідно до тематики. Якщо, наприклад, підписатися на подкаст про садівництво, то він буде оновлюватися відповідно до пори року — наприклад, у якому місяці краще саджати нарциси чи як боротися влітку зі шкідниками.

Перш ніж завантажувати відеозаписи, слід створити канал користувача. Це можна зробити у **Творчій студії** або під час першого використання вказівки **Завантажити**: буде запропоновано створити канал користувача, на якому зберігатимуться завантажені відеозаписи. Відео, збережене у файлі, можна вибрати в структурі папок або просто перетягнути мишею у вказану ділянку

Термін «podcast» є поєднанням назви портативного програвача музики iPod і слова «broadcast». Ведучого або автора подкасту часто називають подкастером, а подкаст як сферу діяльності — подкастингом.

Цей спосіб поширення аудіо та відео через інтернет існує лише кілька років і дозволяє створювати матеріали кожній людині, яка того бажає. Цим подкастинг і відрізняється від, наприклад, радіостанції, де працюють професійні журналісти.

Подкастинг можна надати для перегляду чи прослуховування всім, хто зацікавиться в інтернеті.

Запитання для перевірки знань

1. Що таке відеохостинг?
2. Чи можна завантажити свій відеокліп на YouTube засобами Кіностудії Windows? Якщо так, які дії слід виконати?
3. Як завантажити відео на YouTube?
4. Що таке подкаст?
5. Як можна поділитися створеним відео з друзями та подругами, не використовуючи відеохостинг або соціальні мережі?
6. Знайдіть в інтернеті відомості про відеохостинги для мобільних телефонів. Поміркуйте, чому вони набувають популярності.

Завдання для самостійного виконання

1. Виберіть подію, яка, на вашу думку, є важливою, та запишіть, користуючись смартфоном, її обговорення вами та вашими друзями. За допомогою кіностудії, або іншого відеоредактора створіть слайдшоу з декількох світлин, які ілюструють подію. Додайте як дикторский ваш коментар. Збережіть створений фільм і опублікуйте на вашому каналі. Надайте посилання вчителю.



5. За замовчуванням ShareX зберігає захоплення відео у форматі MP4.

ПРАКТИЧНА РОБОТА № 9

ТЕМА. Захоплення та конвертування аудіо- чи відеоданих
ЗАВДАННЯ: розробити відеоурок щодо створення автоматизованого змісту засобами текстового процесора Microsoft Office Word.
ОБЛАДНАННЯ: комп'ютер, текстовий процесор, програма для захоплення відео ShareX.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Створіть план відеоуроку, запишіть текст мовного супроводу.
2. Налаштуйте роботу мікрофона
3. Виберіть файл, на якому плануєте показувати роботу зі стилями та створення змісту.
4. Запустіть програму для захоплення відео ShareX:
   1) відкрийте файл з вихідними даними;
   2) сполученням клавіш Shift+PrtScr виділіть ділянку екрану для захоплення відео;
   3) задайте в колонтитулах номери сторінок; надайте заголовкам відповідні стилі; створіть нову сторінку та розмістіть на ній зміст;
   4) завершіть роботу програми, повторно натиснувши Shift + PrtScr, або виконавши команду Завершити у нижньому лівому кутку екрана;
   5) перейменуйте створений відеофайл як lessson.
5. Підготуйте створений файл для роботи у відеоредакторі Кіностудия Windows Live. Перетворіть MP4 у формат Microsoft для відео — WMV:

1) відкрийте програму конвертації Format Factory;
2) виберіть як вихідний файл уже створений файл lesson;
3) виберіть у лівому екрані необхідний формат (WMV);
4) виберіть папку для зберігання відконвертованого файлу;
5) натисніть команду Старт і перегляньте результат.

Зробіть висновки: як можна використовувати програму захоплення відео; чим корисна програма конвертації форматів. Спробуйте вивести окремо аудіоряд, виберіть формат для аудіофайлу, обгрунтуйте свій вибір.

ПРАКТИЧНА РОБОТА № 10

ТЕМА. Створення відеокліпу. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду

ЗАВДАННЯ: створити відеокліп, вибравши одну із запропонованих тем.

ОБЛАДНАННЯ: комп'ютер із доступом до інтернету, відеоредактор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Знайдіть в інтернеті відомості з обраної теми, збережіть матеріали до власної папки.
2. Запустіть відеоредактор. Створіть, використовуючи програму відеоредактора, відеокліп на вибрану тему тривалістю до 7 хв.
3. Розробіть сценарій відеокліпа. Для цього:
   1) передбачте титул із назвою відеокліпа,
   2) виберіть фотографії та відео зі збережених у папці;
   3) додайте до кожної фотографії підписи з назвою;
   4) установіть тривалість показу кожної фотографії — 5 с;
   5) передбачте кінцеві титри з власним прізвищем як сценариста та режисера відеокліпа;
   6) виберіть музичний супровід.
4. Установіть переходи між усіма об'єктами відеоряду — на власний розсуд.
5. Додайте відеоефекти до деяких або до всіх об'єктів відеоряду.
6. Синхронізуйте час відтворення аудіо- та відеоряду. Для цього обріжте музичний супровід.
7. Перегляньте відеокліп, за потреби відредагуйте.
8. Збережіть відеокліп як фільм (не як проєкт) у вашій папці з назвою Практична_10.

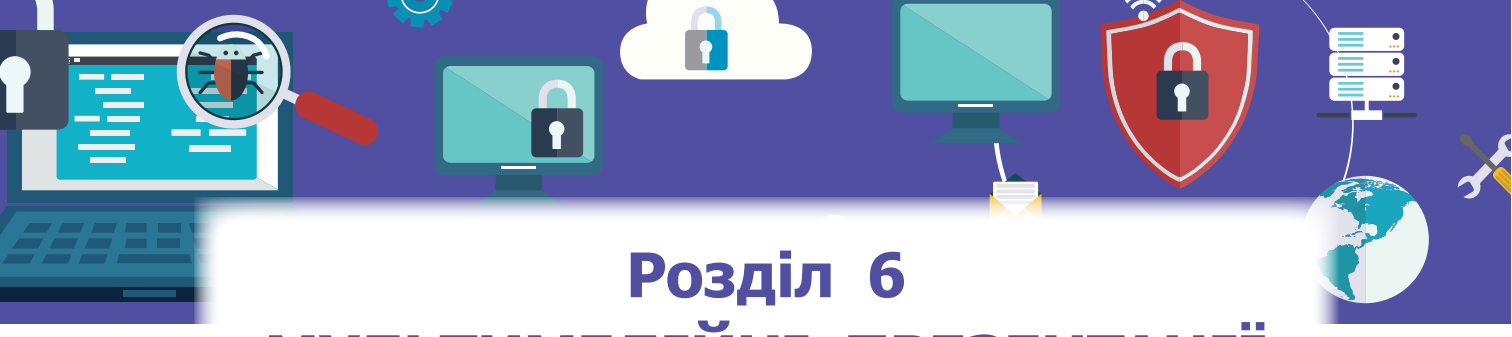
Зробіть висновки: яким чином можна синхронізувати аудіо- та відеоряд фільму; що краще з естетичних міркувань: налаштувати відеоряд під тривалість аудіоряду чи аудіоряд під тривалість відеоряду.



1. **Теми для створення відеокліпу**
   - Найбільші зоопарки світу
   - Дивовижний світ комах
     Сонячна система
   - Суперкомп'ютери

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 5



Розділ 6
МУЛЬТИМЕДІЙНІ ПРЕЗЕНТАЦІЇ

6.1. Етапи розробки презентації. Критерії оцінювання презентації

До яких дій краще вдатися, щоб продемонструвати свої досягнення, результати спостережень, розповісти про якийсь товар або послугу тощо?

Щоб якнайкраще донести до аудиторії певну ідею та надати вичерпну інформацію про якийсь об'єкт у максимально стислій і наочній формі, зазвичай використовують презентацію.

Комп'ютерна презентація — це набір електронних документів, які підготовлені для перегляду на екрані комп'ютера або на великому проекційному чи іншому екрані.

Презентації використовуються для представлення найрізноманітніших видів людської діяльності. Презентації за способом подання можуть бути потоковими або слайдовими.

Ми досить детально розглянули процес створення потокових презентацій під час створення та опрацювання об'єктів мультимедіа. Тепер зосередимося на другому типі презентацій — слайдовому.

Як і будь-який інший продукт, презентація вимагає чіткого алгоритму створення.

Потокова презентація відтворює послідовність (потік) об'єктів із попередньо визначеною тривалістю показу кожного з них, тобто у вигляді кліпу чи відеофільму.

Слайдова презентація відтворює послідовність слайдів — окремих екранних сторінок, кожна з яких може містити текстові, графічні, відео та аудіооб'єкти, анімацію, гіперпосилання. Тривалість показу регулюється доповідачем.

Контент (англ. *content* — вміст) — інформаційне наповнення презентації. Контент може містити текст, статичну графіку, анімаційні елементи, мультимедійні елементи: відео, аудіо тощо.

Етапи розробки крмп'ютерної презентації

Етап 1	Попереднє планування презентації
Етап 2	Створення презентаційних матеріалів
Етап 3	Художнє оформлення слайдів
Етап 4	Підготовка до демонстрації
Етап 5	Попередній перегляд презентації
Етап 6	Збереження презентації

З усіма етапами створення презентації ви вже ознайомилися раніше. Коротко пригадаємо основні з них. Спочатку визначається тема, мета, завдання, цільова аудиторія та кількість слухачів. Чітко сформульовані тема й мета дозволять визначити цільову аудиторію, зміст і форму подання матеріалу.

Цільовою аудиторією є група користувачів, на яку сфокусовано зміст презентації; коло відвідувачів, зацікавлених в інформації, товарах або послугах, що презентуються.

Потім складається сценарій презентації, визначається кількість слайдів, відбувається пошук і створення наповнення. Окрема увага приділяється створенню плану та власне тексту. Ця підготовка тісно пов'язана з роботою MS Word.

Далі робота триває в середовищі опрацювання презентацій, адже починається розробка шаблону, вибір кольорів і шрифтів. Суттєву роль в успішності презентації відіграє її художнє оформлення та дотримання ергономічних вимог.

Ергономіка (від грец. *ergon* — робота, *nomos* — закон) — наука, яка вивчає робочі процеси з метою створення оптимальних умов праці, що сприяє підвищенню продуктивності праці.

Основні ознаки ергономічної презентації

- Тематична послідовність
- Структура за принципом «проблема – розв'язання»
- Короткі речення, яскраві заголовки
- Горизонтальне розташування інформації
- Важлива інформація — у центрі
- Зображення із підписами
- Читабельний шрифт
- Колірна палітра за правилами колористики
- Не більше 5 месенджів (основних думок)
- Цікаві ілюстрації (графіки, діаграми тощо)

Художнє оформлення слайдів відіграє важливу роль у сприйнятті презентації. Пригадаємо, що *композиція* — об'єднання елементів твору в єдине художнє ціле, в якому візуально найбільш яскраво розкривається його зміст, характер, призначення. *Колористика* — наука про колір, яка вивчає основні характеристики кольорів, кольорові контрасти, поєднання кольорів, колірну гармонію та мову кольорів. Тож наведемо деякі корисні, на нашу думку, рекомендації.

- Привабливою виглядає презентація, в якій слайди об'єднані єдиним колірним рішенням та мають однаковий на всіх слайдах (крім першого і останнього) формат і розташування заголовків.
- Краще не використовувати стандартний шаблон, а створити власний (ця тема висвітлюється далі).
- Не варто використовувати більш ніж 5 різних кольорів. Причому три з них — це базові кольори, а два — їхні відтінки.
- Колір шрифту має бути максимально контрастним відносно кольору тла.
- Для надання презентації динамічності можна також застосувати звукові та анімаційні ефекти.

Підготовка до демонстрації передбачає вибір технічних засобів. Зазвичай це комп'ютер, проектор, екран і колонки. Переглядаючи готову презентацію, варто ще раз звернути увагу на оформлення слайдів і їх послідовність, на кількість слів і речень на слайдах і доопрацювати в разі потреби. Потрібно пам'ятати, що ніщо так не знижує рівень довіри як наявність помилок у тексті. Тож слідкуймо за грамотністю.

Існує безліч онлайн-сервісів, які допомагають дібрати палітру кольорів. Найпопулярнішими є ColorExplorer, COLOROTATE, Color Blender, ColorWizard.

Насамкінець презентацію необхідно зберегти: вибрати формат файла Презентація PowerPoint (*.pptx). Завершену роботу можна також зберегти у форматі Демонстрація PowerPoint (*.ppsx). Такі файли відкриваються відразу в режимі перегляду.

Критерії оцінювання комп'ютерної презентації

- Відповідність вмісту презентації вибраній темі
- Доцільність вибору макету й тем оформлення слайдів
- Доречність дизайну та ефектів
- Логічність і структурованість матеріалу
- Зручність застосування елементів керування переглядом



Сет Годін и одна з його порад.



Наразі одним із найпопулярніших трендів комп'ютерних презентацій є принцип мінімалізму. Його сформулював німецький і американський архітектор і дизайнер Людвігом Міс ван дер Рое: «Менше — означає більше» (*Less is more*).

Корисно ознайотися з деякими порадами від Сета Годіна, американського підприємеця, економіста, фахівця з інформаційних технологій, колишнього віце-президента з маркетингу компанії Yahoo. Варто також подивитися його презентацію «Виправте свій справді поганий PowerPoint».

5 правил дивовижної презентації

- На слайді завжди не більше шести слів.
- Скажіть «Ні» неякісним фотографіям.
- Відмовтеся від безцільних складних ефектів зміни слайдів.
- Ніколи не використовуйте вбудовані звукові ефекти.
- Не використовуйте роздруківки слайдів як роздатковий матеріал.

Запитання для перевірки знань

1. В чому різниця між потоковою та слайдовою презентацією?
2. Що таке цільова аудиторія?
3. Які основні етапи розробки презентацій?
4. Навіщо використовують колірні палітри? Якою має бути кількість кольорів?
5. Що таке ергономіка? Як вона впливає на успіх презентації?
6. Як ви розумієте вислів Сета Годіпа: «Припиніть використовувати слайд як телесуфлера»? Чи згодні ви з цим? Обґрунтуйте свою думку.

Завдання для самостійного виконання

1. Створіть презентацію на тему «5 правил ергономічної презентації».
2. Завантажте на Google-диск.
3. Надішліть посилання вчителю.

6.2. Стильове оформлення слайдів презентації. Елементи дизайну презентацій

З яким пакетом програм ви вже працювали?

Існує низка програм для створення слайдових комп'ютерних презентацій. Ми традиційно працюємо з пакетом Microsoft Office, тому використовуватимемо застосунок для створення презентацій PowerPoint, який є складовою Microsoft Office.

Як відомо, при виклику застосунку ми можемо вибрати або порожню презентацію, або презентацію з певної теми. Але в будь-якому випадку програма передбачає, що наша презентація почнеться з титульного слайду (рис. 6.1).

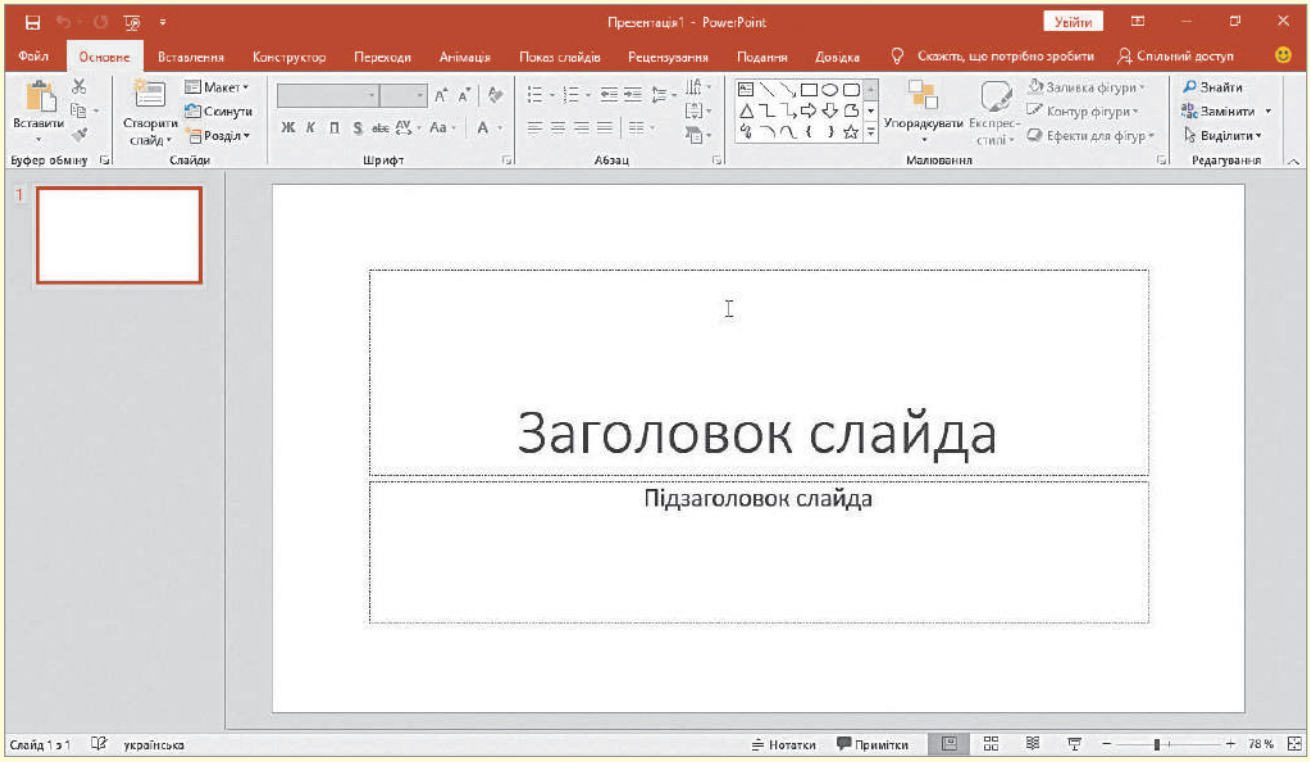


Рис. 6.1. Титульний слайд

Шаблони комп'ютерних презентацій можуть мати різні стилі оформлення слайдів, які називаються темами. Кожна тема має свій набір макетів.

Макет слайда — набір заповнювачів для подання контенту на слайді, що визначає, які об'єкти будуть подані та яким чином розташовані.

Усього PowerPoint пропонує дев'ять типів макетів (рис. 6.2).

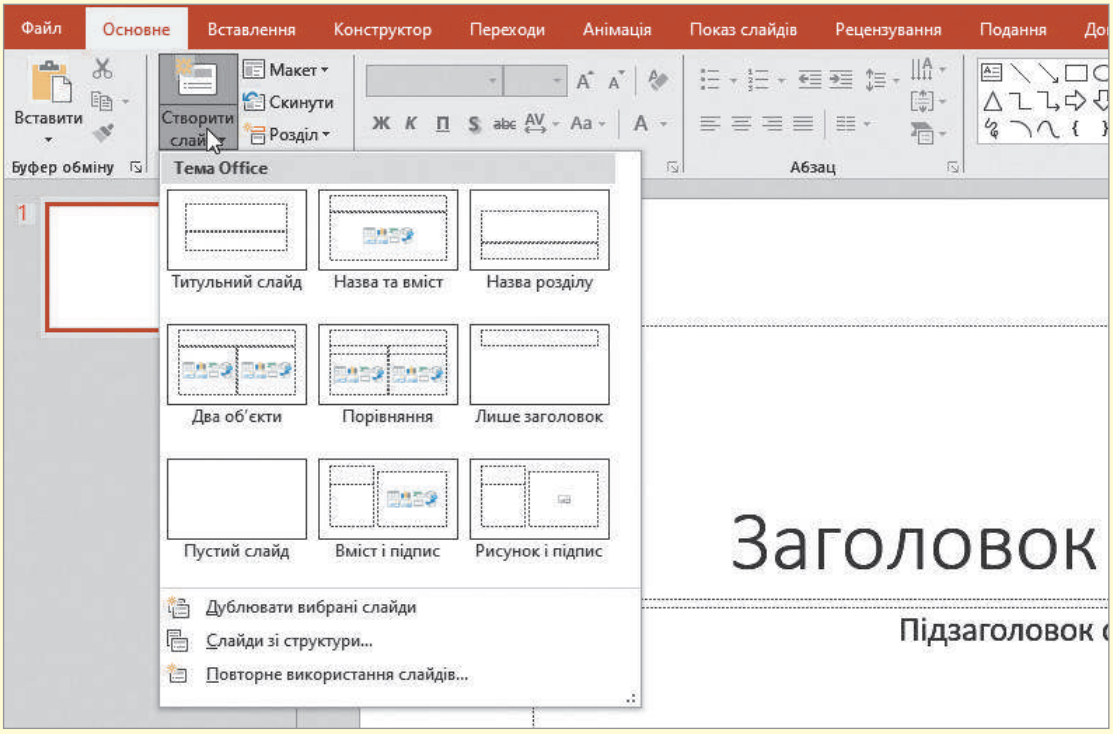


Рис. 6.2. Типи макетів

Способи створення нового слайда

Спосіб 1

Натиснути команду **Створити слайд** на основній стрічці або використати «гарячу клавішу» CTRL + M. При цьому за замовчуванням додається макет слайда **Заголовок** і **Контент**. Він містить заповнювачі для заголовка, маркованого тексту, зображень

Спосіб 2

Відкрити команду **Створити слайд**, в контекстному меню вибрати макет (рис. 6.3)

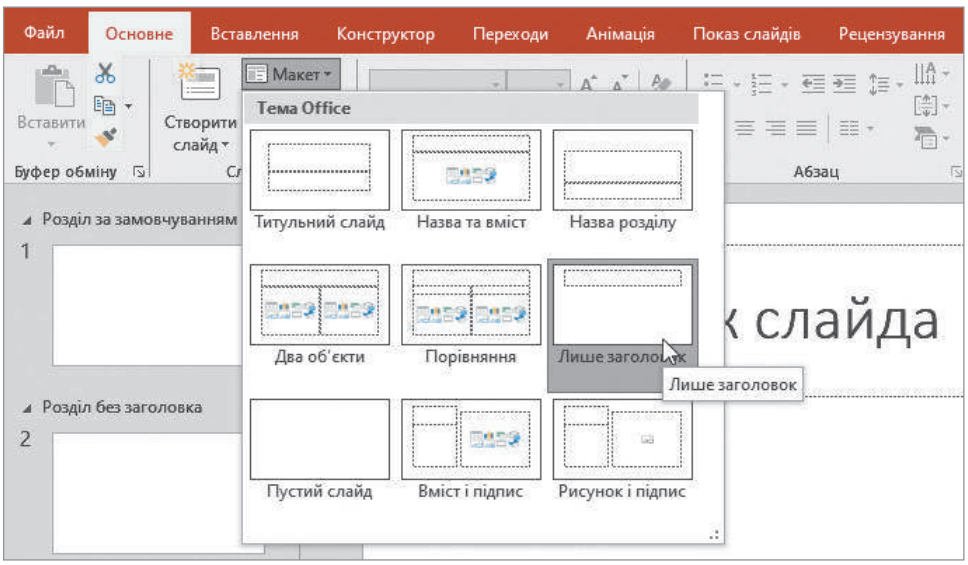


Рис. 6.3. Вигляд розкривного списку з типами макетів слайду

Спосіб 3

Якщо вказівник миші знаходиться на полі навігації достатньо натиснути клавішу Enter або ПКМ і в контекстному меню обрати команду **Створити слайд**.
Новий слайд буде успадковувати тип макету попереднього слайду, у випадку створення слайду після титульного буде створено слайд із макетом за замовчуванням (рис. 6.4)

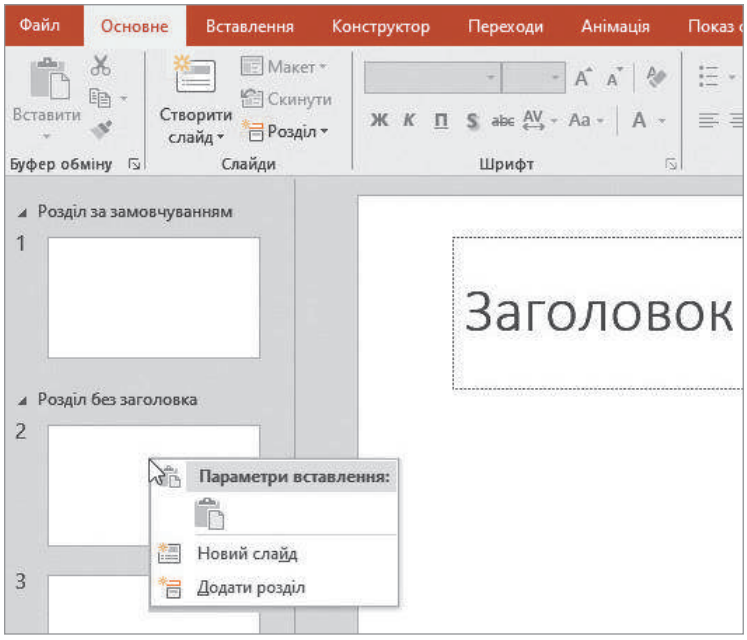


Рис. 6.4. Створення слайду на області структури

Усі вбудовані макети слайдів, окрім макета Порожній слайд, містять заповнювачі вмісту — контейнери, що відображаються у вигляді пунктирних прямокутників.

Контейнери різних типів призначені для розміщення в них тих чи інших об'єктів слайда: це може бути заголовок, текст, таблиця, діаграма, графічний елемент SmartArt, малюнок, картинка з галереї, відео та звук.

Текст на слайді може бути розміщений в одну або дві колонки. Слайд може цілком займати растрове або векторне зображення. На слайді можуть бути розміщені відразу кілька об'єктів різних типів: текст і малюнок, текст і діаграма тощо. Більшість слайдів містять заголовки.

З об'єктами, розміщеними на слайді, можна працювати так само, як у векторному графічному редакторі. Виділений об'єкт можна переміщати по слайду, змінювати його розміри, здійснювати угруповання об'єктів, а також переміщувати об'єкти на різні шари слайда. Таким чином ми створюємо стильове оформлення презентації.

Ви вже познайомилися з означенням стилю, згадаємо, що *стиль* — це сукупність значень властивостей різних об'єктів.

У будь-який момент роботи над комп'ютерною презентацією стиль її оформлення можна змінити. Для цього варто скористатися наявними темами презентації, які можна вибрати на стрічці у вкладці Дизайн.

Макети слайдів визначають розмір і колір шрифтів, ефекти і фоновий малюнок (тема). Найпростішими є макети, що містять порожні заготовки слайдів із заголовком і без нього.

Стиль оформлення слайда задає:

- формат символів (шрифт, розмір символів, колір, накреслення, ефекти тощо)
- формат тла (колір, наявність, розміщення та вид графічних об'єктів)
- додаткові кольори
- формат графічних та інших об'єктів

Шаблон презентації — це схема презентації з одного чи кількох слайдів, які оформлені з дотриманням певного задуму.

Обравши певну тему, можна використати її модифікації, натиснувши варіанти, та пройтися усіма пунктами: кольори, шрифти, ефекти, стилі — кожен з яких надає величезну кількість додаткових варіантів налаштування. Так само можна самостійно вибрати формат тла, натиснувши у вкладці Дизайн кнопку Формат тла, або вибравши цю команду з контекстного меню, натиснувши ПКМ, коли курсор знаходиться безпосередньо на слайді.

Можливості роботи з форматом тла аналогічні роботі з фігурами в текстовому процесорі, оскільки тут використовується вбудований векторний графічний редактор, функції якого доступні застосункам пакету Microsoft Office.

Власний шаблон користувач може створити, вибравши в вкладці Подання команду Зразок слайду (рис. 6.5).

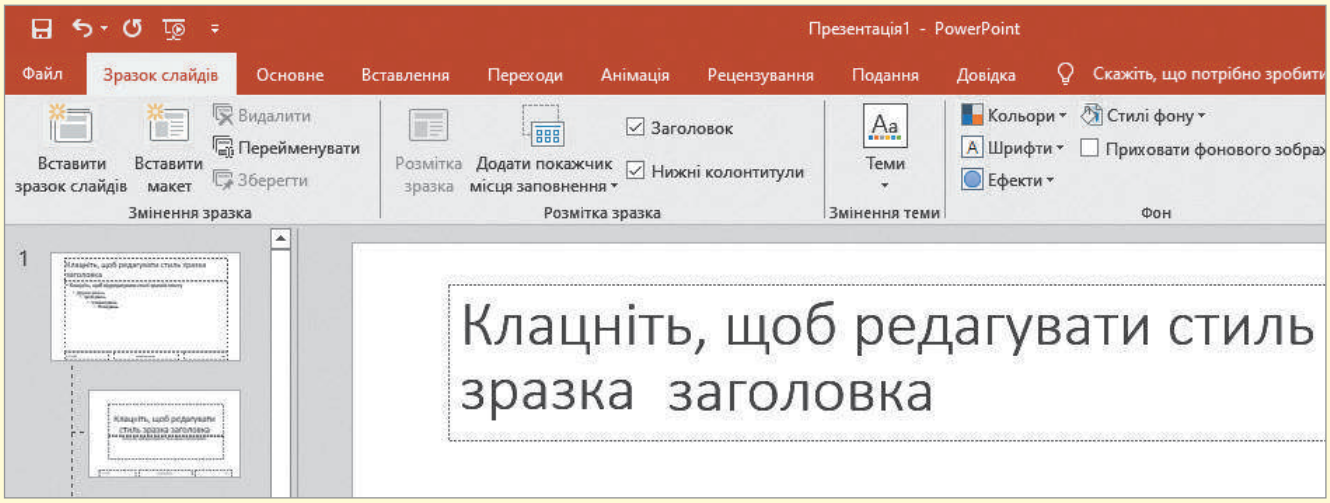


Рис. 6.5. Створення власного шаблону

Залежно від теми та мети презентації можна вибрати теми оформлення з колекції PowerPoint або онлайн-шаблони, або розробити власні.

PowerPoint пропонує готові проєкти презентацій — шаблони. Потрібний шаблон можна відшукати у вікні програма відразу після її запуску

Ми бачимо на полі навігації всі можливі макети. Причому створюючи стиль першого слайду (його легко відрізнити за розмірами), ми надаємо такий самий стиль всім іншим макетам, якщо ми хочемо, щоб стиль решти відрізнявся. Вибираємо макет и налаштовуємо стиль спеціально для нього. Слід пам'ятати декілька правил стильового оформлення слайдів.

- Дотримуйтеся правил композиції, колористики, ергономіки — це три кити дизайну.
- Не можна писати будь-який текст замість наповнювачів Заголовок, оскільки він буде відображатись у вашій презентації і його не можна вилучити.
- Якщо розмістити зображення на першому слайді, то воно буде успадковане іншими макетами.
- Слід зберегти розроблений вами шаблон з типом файлу Шаблон презентації (рис. 6.6).

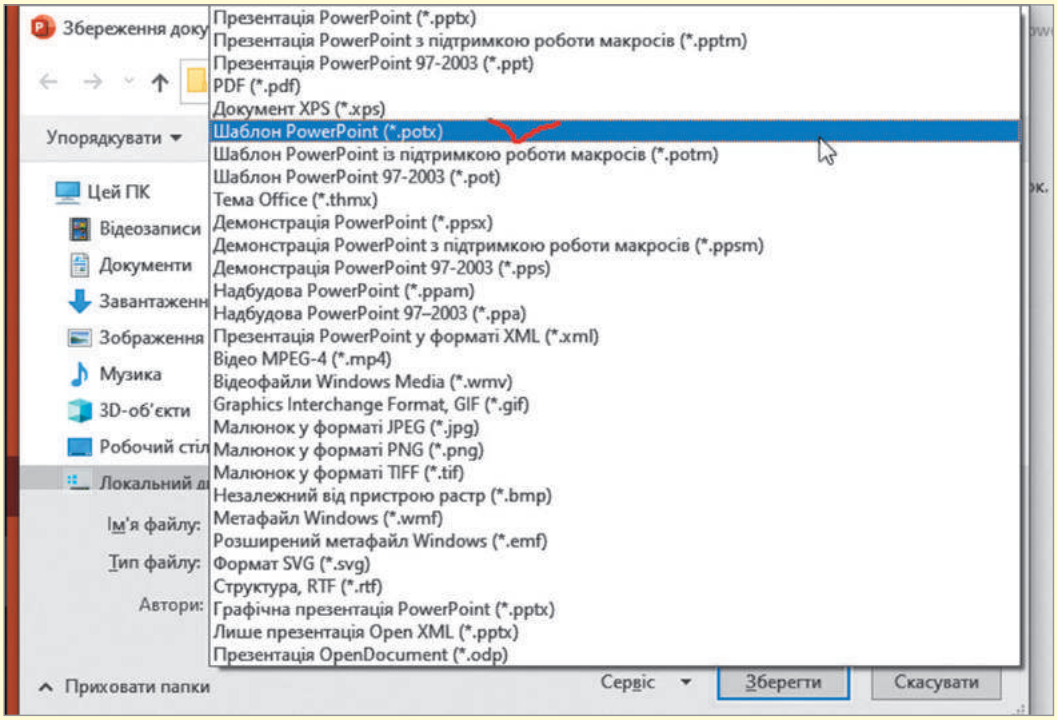


Рис. 6.6. Збереження розробленого шаблону

Якщо ви зберегли шаблон, то завжди зможете його знайти в вкладці Дизайн як користувацький. Для цього достатньо відкрити теми, як наведено на рис. 6.7.

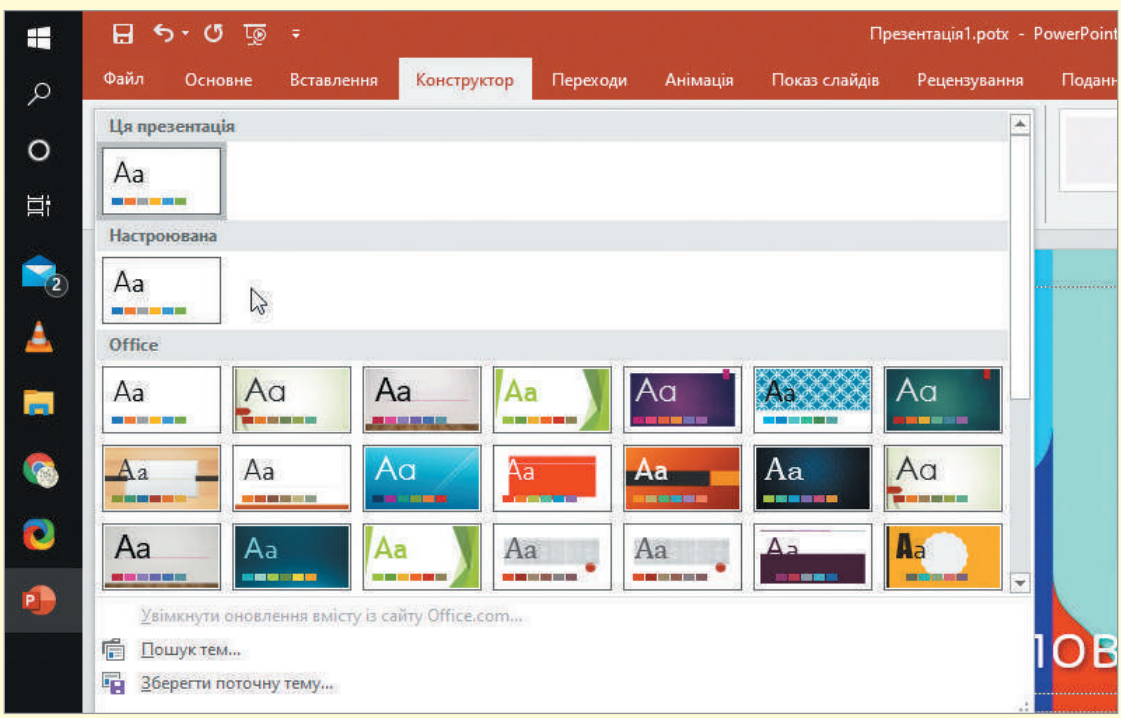


Рис. 6.7. Створення презентації на базі розробленого шаблону

Досить часто презентація потребує візуалізації поданих даних. Ви вже створювали організаційні діаграми за допомогою графічного елемента SmartArt. Текстовий редактор пропонує досить великий набір різноманітних макетів (рис. 6.8).

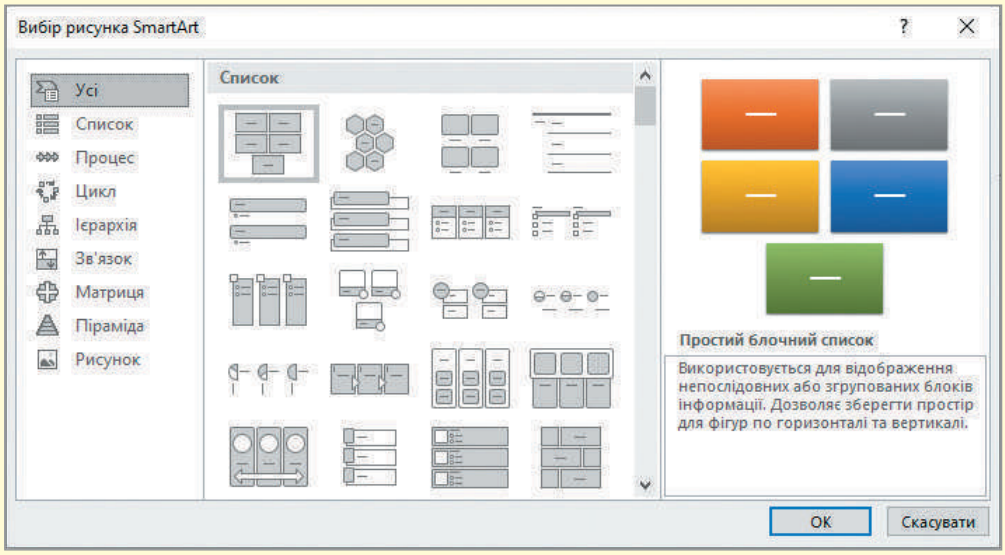


Рис. 6.8. Вибір макета графічного елемента SmartArt.

Способи заповнення контентом SmartArt

Спосіб 1	• У полі **Текст** елемента SmartArt Можна додавати певні фігури, змінювати їх рівень та порядок розташування за допомогою відповідної вкладки **Конструктор**
Спосіб 2	• За допомогою **Область тексту** вкладки **Конструктор** Можна вводити та редагувати текст, який має відображатися на певному елементі **SmartArt**. Область тексту відображається ліворуч від обраного макету. Коли ви додаєте або редагуєте текст у **Області тексту**, діаграма оновлюється автоматично — фігури додаються або видаляються
Спосіб 3	• Вставленням скопійованого заздалегідь тексту в **Область тексту** Виділивши увесь текст в **Область тексту**, можна відформатувати його стандартними способами (наприклад, за допомогою основної стрічки)

Можна також скористатися командами вкладки Формат функції Робота з рисунками SmartArt (рис. 6.9).

Вміст вкладок **Конструктор** і **Формат** дозволяє створювати справжні дизайнерські шедеври.

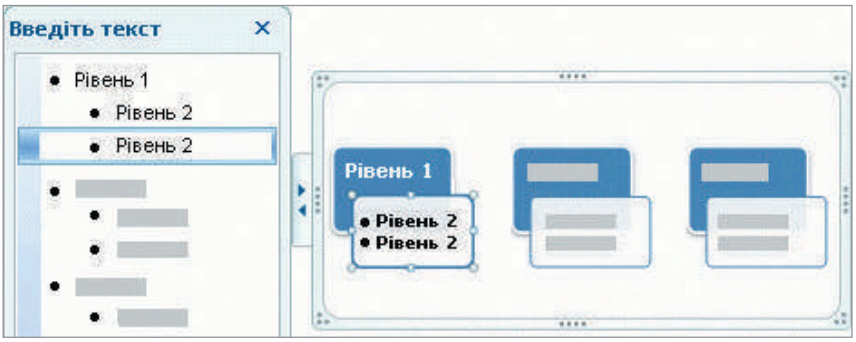


Рис. 6.9. Вигляд розкривного списку **Область текста** графічного елемента SmartArt

Додаткові можливості відображення графічного елементу SmartArt надають команди Обрати колір і Стилі вкладки Конструктор (рис. 6.10, 6.11).

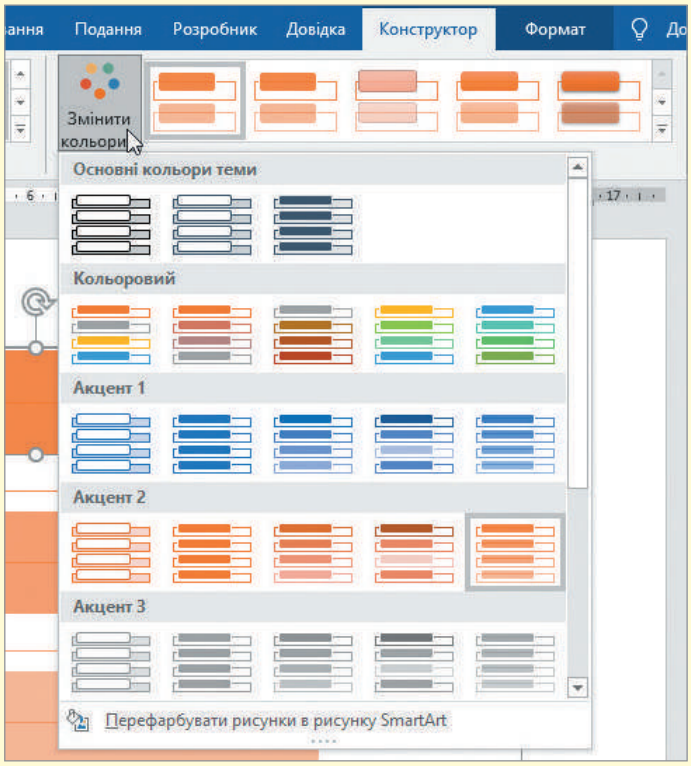


Рис. 6.10. Вигляд розкривного списку команди **Обрати колір** графічного елемента SmartArt

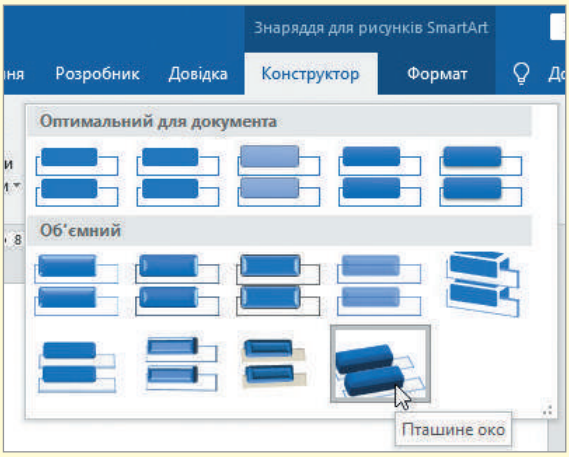


Рис. 6.11.Вигляд розкривного списку команди **Стилі** графічного елемента SmartArt

Запитання для перевірки знань

1. Скільки макетів пропонує PowerPoint?
2. Як вибрати тему і чи можна використовувати власні варіанти?
3. Що таке стильове оформлення презентацій?
4. Як розробити власний шаблон; яких правил необхідно дотримуватися?
5. Коли доцільно використовувати організаційні діаграми?
6. Одну й ту саму дію ми можемо виконувати різними способами. Проаналізуйте ці способи і поясніть, чому в нас з'явилась така можливість.

Завдання для самостійного виконання

1. Розробіть власний шаблон для презентації з вибраного вами навчального предмету (біології, хімії, математики, інформатики тощо). Збережіть як шаблон і надішліть учителю.

6.3. Проектування та розробка розгалужених презентацій. Гіперпосилання і елементи керування

Розкажіть про свій досвід створення презентацій

Гіперпосиланням можна зробити будь-який об'єкт слайда, наприклад текстовий фрагмент, зображення, фігуру. Зовні текстове гіперпосилання відрізняється від звичайного тексту підкресленням, а також тим, що в режимі демонстрації слайда в разі наведення на посилання курсору миші той набуває форми руки з піднятим вказівним пальцем.

Говорячи про програму створення комп'ютерних презентацій, зазвичай згадують три основні елементи: анімацію, керовану анімацію (тригери), навігацію (гіперпосилання), які надають родзинки презентаціям. Наразі ми будемо говорити саме про навігацію.

За структурою презентації можуть бути лінійними або розгалуженими. У *лінійних презентаціях* переходи можливі від попереднього слайду до наступного і навпаки. Більшість презентацій саме лінійні. *Розгалужені* (*інтерактивні*) презентації відрізняються тим, що в них можливі переходи від однієї групи слайдів до іншої. На слайдах можна розташовувати спеціальні елементи керування — гіперпосилання.

Гіперпосилання використовують під час показу презентації для переходу від одного слайда до іншого (так звана навігація), запуску якоїсь програми, відтворення кліпу, відкривання вебсторінки сайту або файлу тощо.

Алгоритм створення гіперпосилання

Крок 1 Виділіть фрагмент тексту на слайді. Виконайте команду **Вставка → Гіперпосилання**

Крок 2 У вікні **Вставка** гіперпосилання, що відкриється, пов'яжіть гіперпосилання з певним об'єктом — файлом, вебсторінкою, місцем в документі (слайдом), новим документом або адресою електронної пошти.
Для цього виберіть на панелі у лівій частині діалогового вікна тип об'єкта і виберіть його у ділянці праворуч (рис. 6.12).

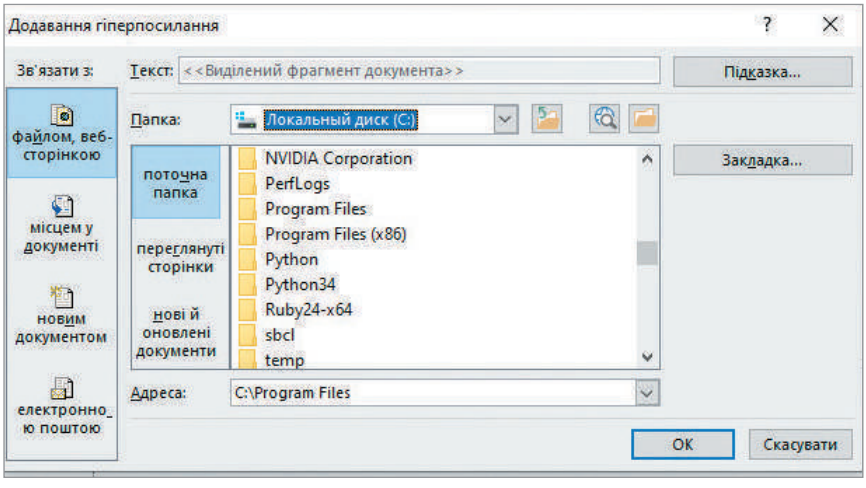


Рис. 6.12. Вигляд меню команди **Додавання гіперпосилання**

Якщо посилання вказує на вебсторінку або файл, то для їх відкривання запускатиметься відповідна програма. Причому після її закриття, знову повертаємось в презентацію, на той слайд, з якого ми переходили за гіперпосиланням (рис. 6.13).

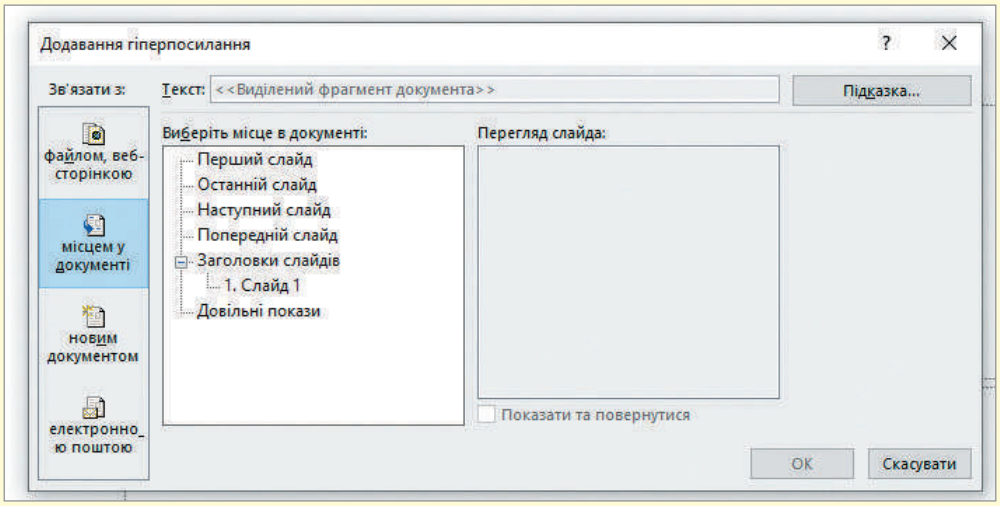


Рис. 6.13. Вибір слайду для додавання гіперпосилання

Крім гіперпосилань, на слайдах можна розташовувати й інші елементи керування — так звані кнопки дій (рис. 6.14). PowerPoint пропонує 12 кнопок. Розглянемо, які саме.

Кнопки	Назва	Призначення
	Назад	Перехід на попередній слайд
	Далі	Перехід на наступний слайд
	На початок	Перехід на перший слайд
	В кінець	Перехід на останній слайд
	Додому	Перехід на початковий слайд
	Відомості	Перехід на слайд з інформацією про автора презентації
	Повернення	Перехід на останній переглянутий слайд
	Фільм	Запускає перегляд доданого відеокліпа на слайді
	Документ	Запускає виклик програми
	Звук	Використовується для відтворення доданого аудіофайлу
	Довідка	Перехід на слайд з довідковою інформацією
	Настроювана	Для налаштування за вибором розробника презентації

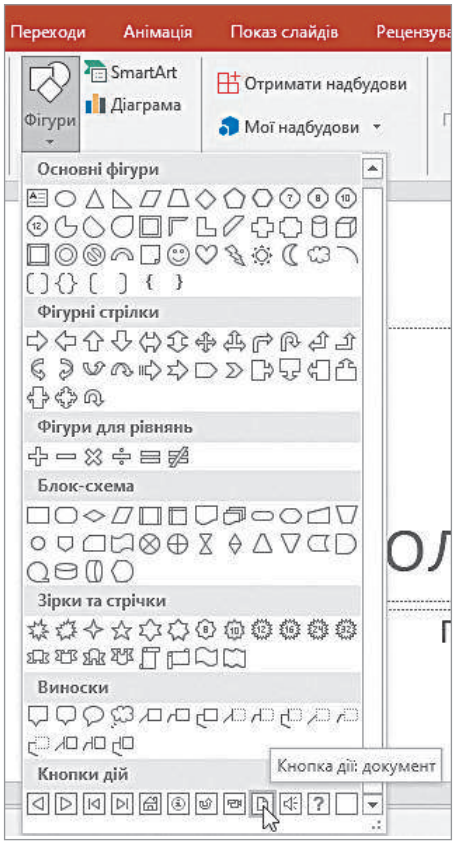


Рис. 6.14. Елементи керування — кнопки дій

Кнопки дій — це низка фігур, які зображують кнопки, для яких уже призначено одну з найчастіше вживаних дій.

Кнопки дій додають до слайда за допомогою Вставка → Фігури. Останні фігури це і є кнопки. Після вибору кнопки бажаного вигляду та клацання на певному місці слайда з'явиться зображення кнопки та відкриється діалогове вікно Настроювання дії (рис. 6.15).

Кнопки дій також використовують під час показу слайдів для виконання певних операцій, наприклад, навігаційних переходів, запуску програм, відкриття файлів тощо.

Існує дві можливості виконати дію при натисненні на ліву клавішу миші, або при наведенні на кнопку вказівника миші. В обох випадках робота ідентична: вибір дії зі списку (рис. 6.16): перехід за гіперпосиланням, виклик програми, запуск макроса, звук.

Перехід за гіперпосиланням, у свою чергу, надає можливість вибору однієї з пропонованих на рис. 6.16 дій.

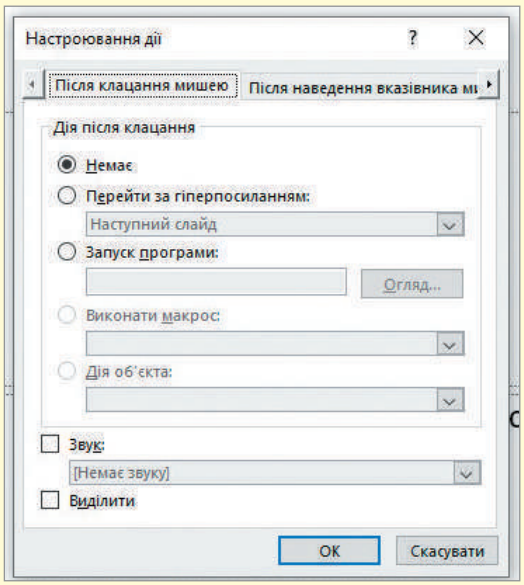


Рис. 6.15. Меню **Настроювання дії** обраного елемента керування

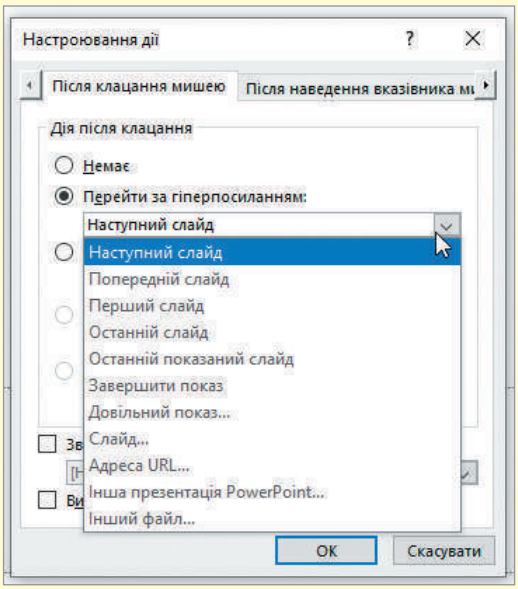


Рис. 6.16. Розкривний список команди **Перейти за гіперпосиланням**

Налаштування, притаманні кнопкам дій, можна застосувати до будь-яких об'єктів слайда, наприклад, до фрагментів тексту, графічних об'єктів тощо.

Після налаштування роботи кнопки можна відформатувати саму кнопку, використовуючи відомі вам команди: змінити фігуру, колір заливки та контуру, змінити тип контуру, додати ефекти і текст.

Додатково можна налаштувати презентацію таким чином, щоб навігація здійснювалась певним маршрутом, не дозволяючи користувачу його змінювати. Це досягається лише однією командою: Показ слайдів → Налаштування показу слайдів і в меню, що відкрилось, слід відмітити Автоматичний (повний екран) (рис. 6.17).

Роботу в презентації можна закінчити також натисненням клавіши Esc.

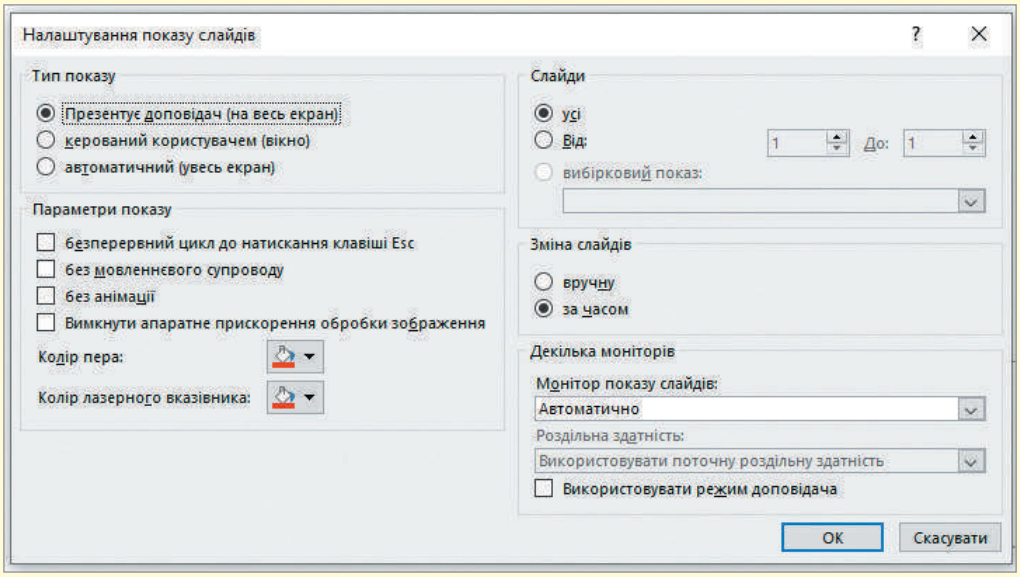


Рис. 6.17. Налаштування показу презентації

Запитання для перевірки знань

1. З яким об'єктом на слайді може бути пов'язано гіперпосилання?
2. На який об'єкт може здійснюватися перехід після вибору гіперпосилання на слайді презентації під час її демонстрації?
3. Як створити гіперпосилання для переходу до перегляду певного слайда під час демонстрації?
4. Як створити гіперпосилання на слайді презентації для переходу на веб-сторінку під час демонстрації?
5. Як повернутися до демонстрації презентації, якщо було вибрано гіперпосилання, яке відкрило вікно текстового процесора?
6. Які існують кнопки дій? Опешіть їх призначення.

Завдання для самостійного виконання

1. Створіть комп'ютерну презентацію «Моя прогулянка».
2. Зв'яжіть презентацію та кліп, створений під час виконання завдання з розділу 5 (§ 5.5.)

6.4. Елементи анімації в презентації

Пригадайте, що таке анімація, де ви вже використовували комп'ютерну анімацію.

Анімація — це спеціальний відео- або аудіоефект, який можна додати до тексту або до певного об'єкту (зображення, фігури, рамки). Прийшов час анімації і керованої анімації.

Згадаємо, що в PowerPoint анімація поділяється на анімацію об'єктів та анімацію зміни слайдів (або переходів між слайдами). Розглянемо анімацію об'єктів. Для налаштування анімації перейдемо на вкладку Анімація (рис. 6.18).

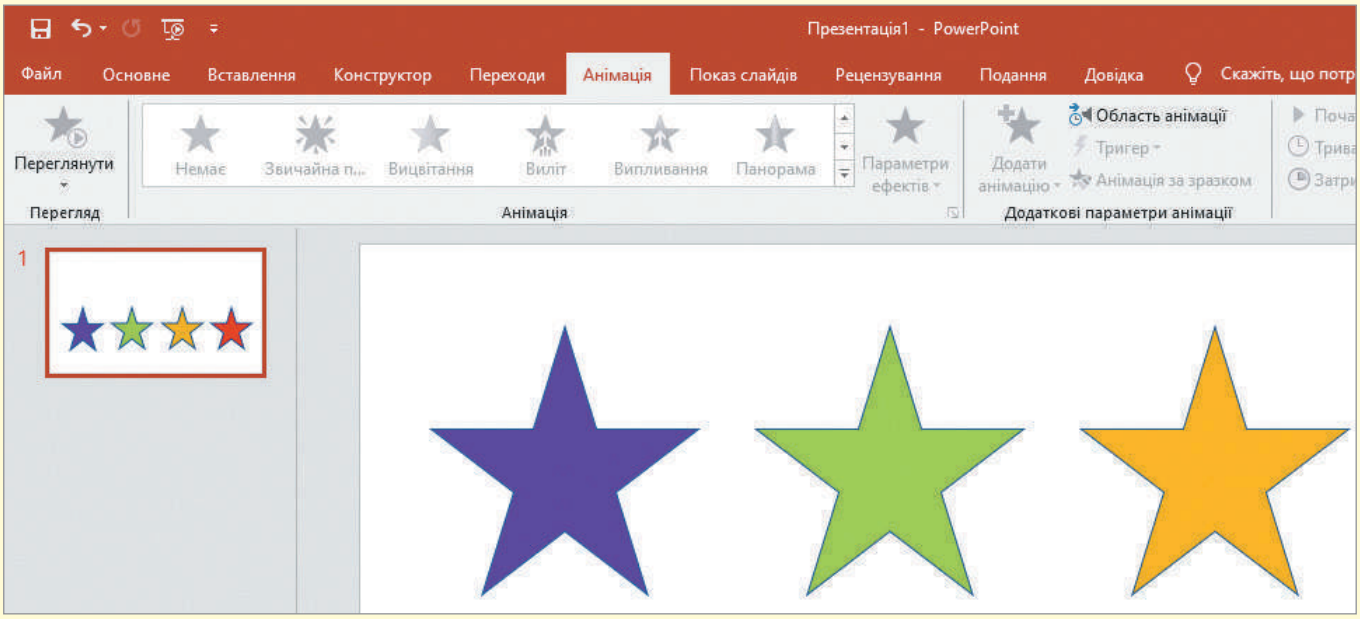


Рис. 6.18. Вкладка **Анімація**

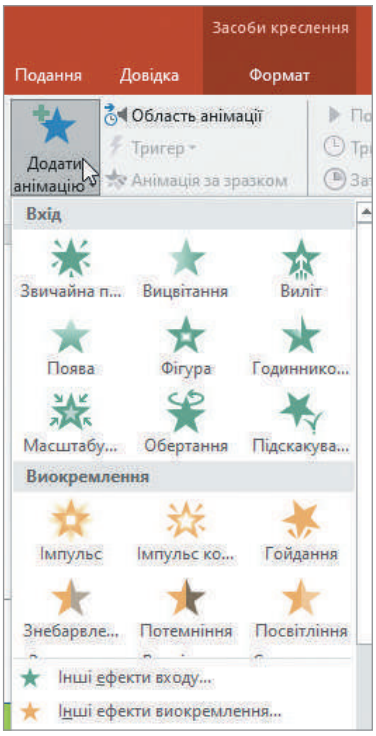


Рис. 6.19. Вибір ефекту анімації

Для того щоб застосувати анімацію, нам потрібно вибрати об'єкт, інакше всі команди, щодо анімації будуть неактивними (на скриншоті їх пофарбовано в сірий колір).

Після виділення об'єкту можна натиснути кнопку Додати анімацію (рис. 6.19). PowerPoint визначає чотири типи анімації:

Вхід	Поява на слайді об'єкта (ефекти, що відносяться до цього типу, пофарбовані у зелений колір)
Виокремлення	Можливість видозміни об'єкта на слайді — зміна розміру, кольору, прозорості, гойдання, обертання, тощо. Ці ефекти відображаються жовтим кольором
Вихід	Зникнення об'єкта зі слайду (вони можуть носити назву таких самих як і ефекти входу, але мають протилежну дію) і відображаються червоним кольором
Переміщення	З самої назви випливає, що це переміщення на слайді за певною траєкторією, яку вибрав або створив розробник презентації (цей тип позначається сірим кольором)

Обраний ефект має певні параметри: наприклад, для ефектів Вхід і Вихід це може бути напрямок, для виокремлення — збільшення чи зменшення розміру.

Під час налаштування ефектів варто натиснути кнопку Область анімації. У правій частині екрану з'явиться панель, на якій відображатимуться ефекти анімації, застосовані на активному слайді (рис. 6.20), це дозволить розширити можливості.

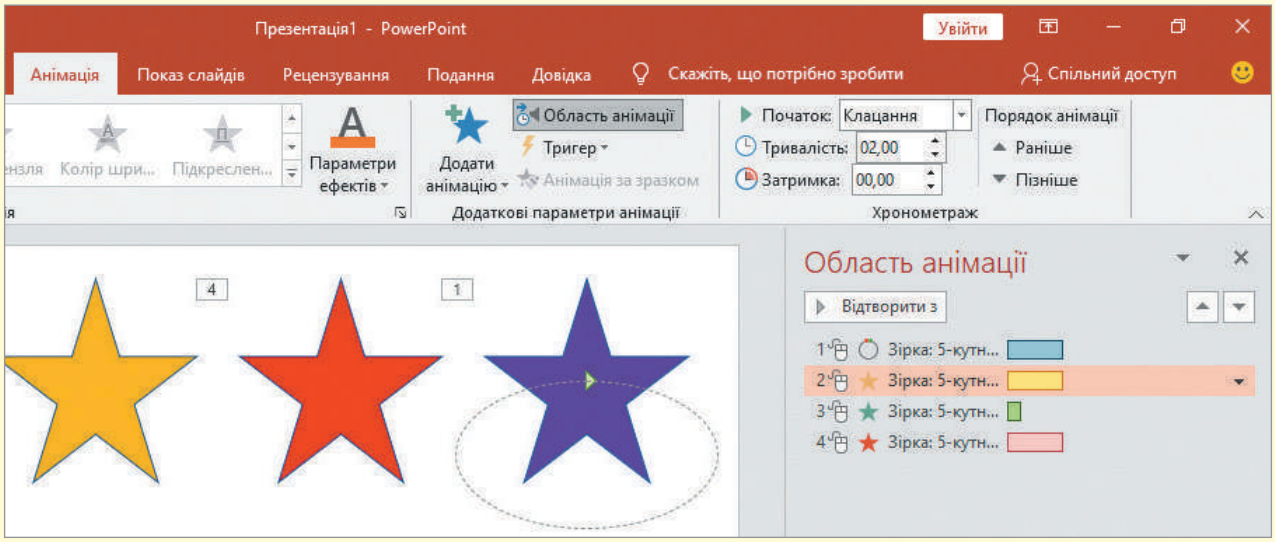


Рис. 6.20. Панель **Область анімації**

Способи налаштовувати параметри ефекту

Спосіб 1

Натиснувши кнопку **Параметри ефектів**, обрати зі спаданого списку прийнятний параметр (рис. 6.21)

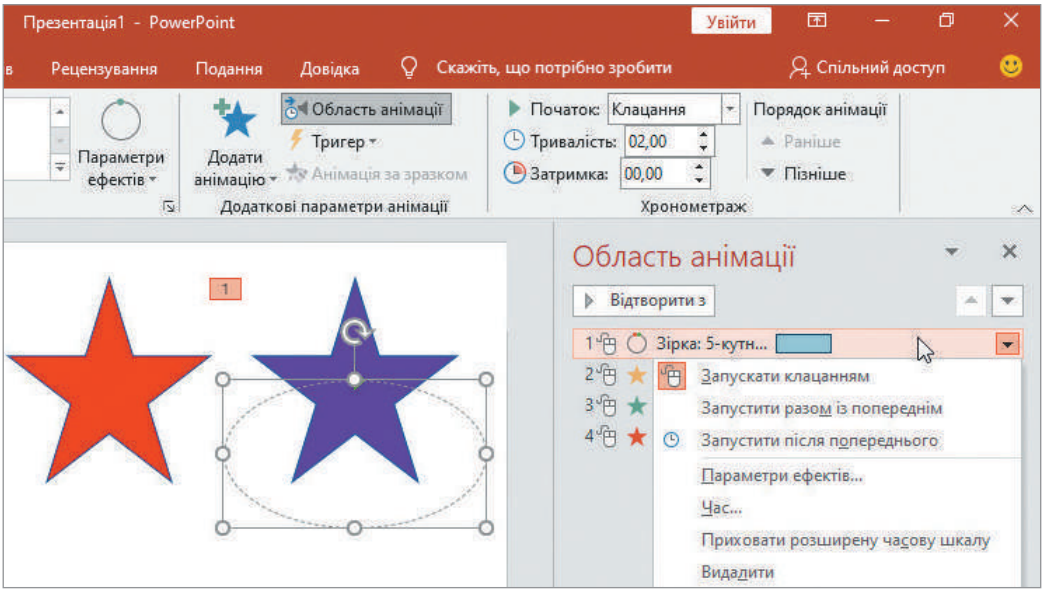


Рис. 6.21. Контекстне меню обраного ефекту в **Області анімації**

Спосіб 2

У полі **Область анімації** встановити вказівник миші на ефекті і натиснути ПКМ, у контекстному меню обрати команду **Параметри ефектів**, вибрати потрібний

Існує два напрямки використання анімації у презентаціях: анімація об'єктів на слайдах і переходи — анімація зміни самих слайдів.

Скориставшись однією з можливостей, наведених на рис. 6.22, можна налаштувати виконання ефекту анімації.

- Після клацання — ефект анімації розпочнеться після натиснення лівої кнопки миші.
- З попереднім — встановлює початок застосування ефекту анімації одразу після появи попереднього об'єкта, а у випадку першого об'єкта слайда — одразу після початку демонстрації даного слайда.
- Після попереднього — розробник презентації встановлює інтервал часу, через який, після закінчення попереднього ефекту, розпочнеться обраний ефект.

Налаштувати ці значення також можна на вкладці Анімація, відкривши для цього меню Початок, а також у полі Область анімації → Час (рис. 6.23).

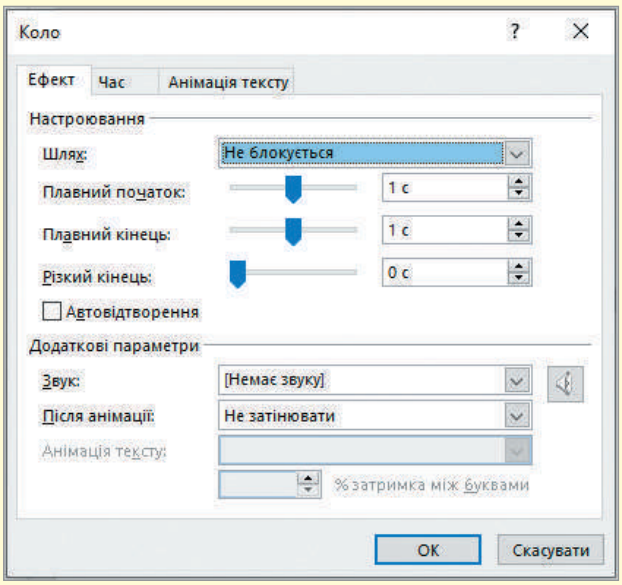


Рис. 6.22. Параметри ефектів в контекстному меню **Області анімації**

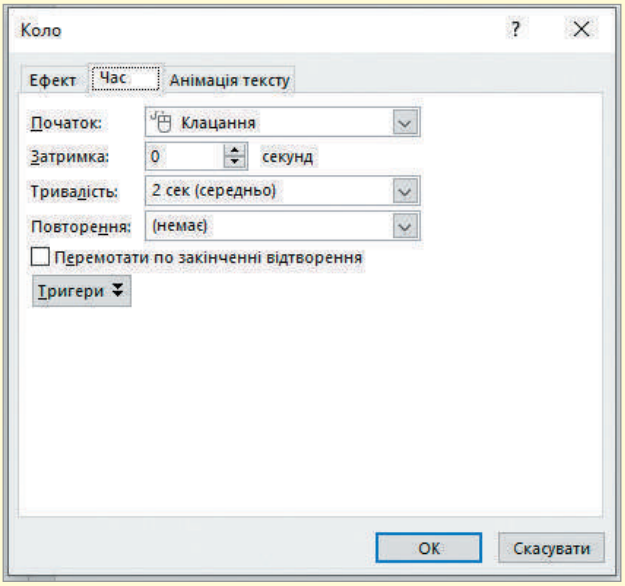


Рис. 6.23. Час в контекстному меню **Області анімації**

Тригер в PowerPoint — це інструмент, який дозволяє створити «гарячу клавішу» для запуску анімації, аудіо- та відеоефектів, причому зробити це досить легко.

Крім зазначеного, можна налаштувати потрібну кількість разів виконання вибраного ефекту об'єкта від Немає до До закінчення слайду, або поставити число (наприклад, 5, 20 і т. д.).

Під час демонстрування презентації глядач може лише спостерігати за анімацією, яку налаштував розробник презентації. Але досить часто ми стикаємось із ситуацією коли потрібно певним чином керувати послідовністю анімаційних ефектів. Звісно ми можемо скористатися кнопками дій і гіперпосиланнями, які розглядали у попередньому параграфі.

Існує ще один інструментарій, який дозволяє виконувати певні дії стільки разів, скільки захоче користувач (глядач), а саме — тригер (у перекладі з англійської означає «спусковий гачок», «затвор»).

Тригер — засіб анімації, що дозволяє задати умову дії або часу виділеному елементу. При цьому анімація запускається після клацання.

До кожного об'єкта можна додати кілька ефектів кожного типу, програмуючи досить складну його поведінку.

За допомогою тригера можна задати дію будь-якого об'єкта. Причому послідовність цих дій ми можемо вибирати в залежності від власного вибору та ситуації.

Наявність тригера значно підвищує інтерактивність презентації.

Приклад.

За допомогою тригера можна створити інтерактивний тест на базі презентації. Як варіанти відповіді можуть виступати слова, цифри або зображення. При натисненні на неправильний варіант відповіді, відповідь зникає, в іншому випадку спрацьовує будь-який ефект анімації, що змінює розмір, колір або розташування об'єкта. Таким чином за допомогою тригерів можна створити вікторини, ігри, інтерактивні плакати тощо

Тепер необхідно лише вибрати ефект анімації, до якого потрібно застосувати тригер, і виконати один із таких способів.

Способи вибору ефекту анімації

Спосіб 1

Натиснути на команду **Тригер** у вкладці **Анімація** і у випадному списку обрати об'єкт на який необхідно натиснути, щоб спрацював певний ефект анімації (рис. 6.24).

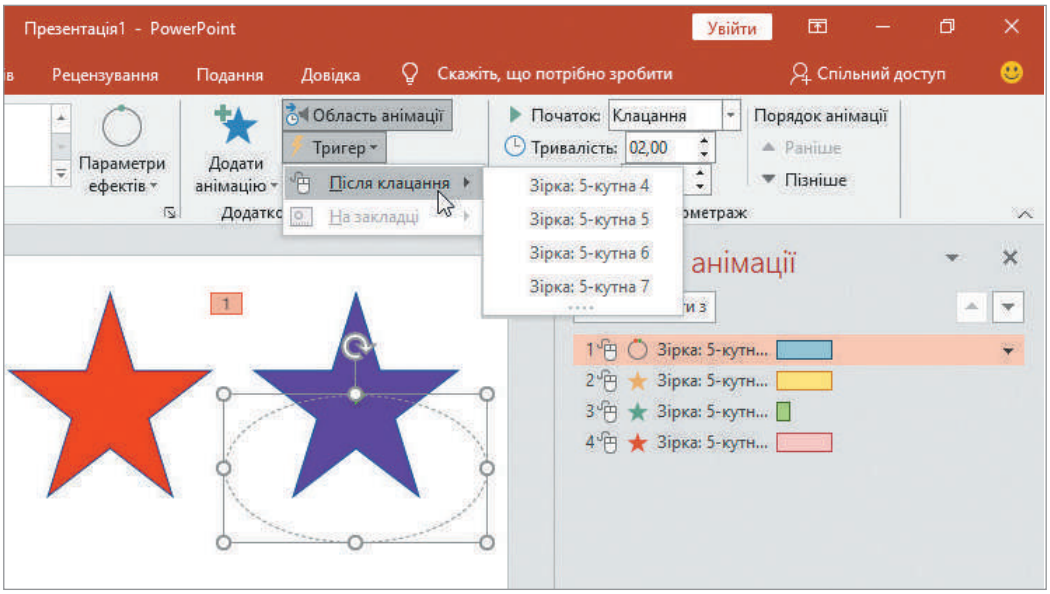


Рис. 6.24. Вибір тригера

Спосіб 2

Вибрати потрібний ефект у полі **Область анімації**, натиснути ПКМ і у контекстному меню обрати команду **Час**.
У ній вибрати команду **Тригери** та натиснувши на команду **Почати виконання ефекту** після клацання у випадному списку всіх об'єктів, що є на слайді обрати потрібний (рис. 6.25).

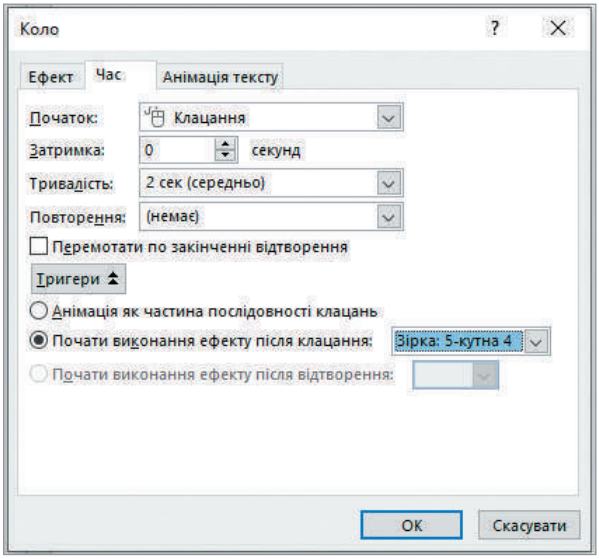


Рис. 6.25. Налаштування тригера

Пригадаємо про ще одну можливість анімації, яку надає програма для створення комп'ютерних презентацій — це зміна слайдів з величезною кількістю різноманітних ефектів, з якою ви познайомилися ще в 6 класі.

Запитання для перевірки знань

1. Що таке анімація в PowerPoint?
2. Які типи анімації розрізняють у PowerPoint?
3. Що таке тригер?
4. Назвіть способи додавання анімації.
5. Які параметри можуть бути у ефектів типу Виокремлення?
6. Чи можливо власноруч створити шлях переміщення об'єкту? Який інструмент дозволить це зробити?

Завдання для самостійного виконання

1. Створіть слайд «Салют». Для цього розташуйте на слайді фігуру, при кожному натисканні на яку, на слайді з'являються зірочки, що імітують салют. Забезпечте вихід зі слайду лише при натисканні клавіши Esc.
2. Створіть слайд «Магазин». Для цього розташуйте на слайді поличку із зображенням продуктів і візок (усі зображення знайдіть в інтернеті, пригадайте як вибрати зображення лише з прозорим тлом). При натисканні на певний продукт його зображення зникає з полички та з'являється у візку. Забезпечте зворотню дію.

6.5. Додавання відеокліпів, звукових ефектів і мовного супроводу до слайдової презентації

Пригадайте, які є об'єкти презентації.

До презентації PowerPoint можна додати кліп, звуковий супровід, наприклад музику, дикторський текст або аудіоефекти тощо.

Алгоритм вставлення відеокліпа зі свого комп'ютера

Крок 1	Виберіть слайд, на якому ви хочете відтворити відеокліп
Крок 2	На вкладці **Вставлення** натисніть стрілку під кнопкою **Відео** та виберіть **Відео на моєму ПК**
Крок 3	У діалоговому вікні оберіть потрібний відеозапис
Крок 4	Виберіть команду **Вставити**

Щоб вставити аудіофайл, необхідно у вкладці Вставлення вибрати команду Аудіо на моєму ПК, а у діалоговому вікні Вставити аудіофайл — необхідний файл.

Пригадаємо, що формати WMV і WMA є форматами, розробленими корпорацією Microsoft. Для запису дикторського тексту (мовного супроводу) потрібно виконати такий алгоритм.

Крок 1	На вкладці **Вставлення** → **Аудіо** виберіть команду **Записати аудіо**
Крок 2	Спочатку введіть ім'я аудіофайлу, потім натисніть кнопку **Записати** та проговоріть необхідний текст.
Крок 3	По закінченні натисніть команду **Стоп** і **ОК**. На слайді з'явиться піктограма звуку, яку можна переміщувати на слайді

Програма PowerPoint дозволяє налаштовувати параметри відтворення (рис. 6.26).

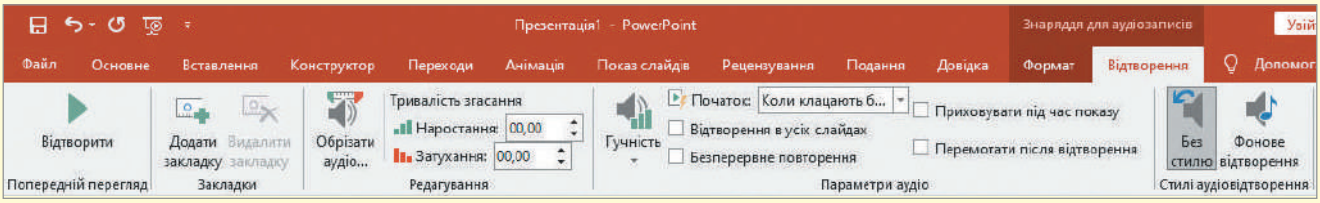


Рис. 6.26. Налаштування параметрів відтворення

Натиснувши на піктограму звуку, ми отримуємо вкладку Знаряддя для аудіозаписів із вкладкою Відтворення.

Розглянемо таблицю:

Операція	Опис дії
Обрізка аудіофайлу	Натиснути кнопку **Обітнути**, а потім за допомогою червоного та зеленого повзунків налаштувати потрібну тривалість
Налаштування наростання або затухання звуку	Змінити значення в полях **Тривалість згасання**
Налаштування гучності відтворення	Вибрати команду **Гучність**, а потім потрібне значення
Налаштування способу запуску аудіофайлу	Клацнути стрілку розкривного списку та вибрати один із параметрів: • клацнути кнопку миші Аудіофайл автоматично відтвориться, якщо клацнути кнопку миші • автоматично Аудіофайл автоматично відтвориться після переходу на слайд із ним • клацнути піктограму Аудіофайл відтвориться, якщо клацнути піктограму
Налаштування способу відтворення аудіофайлу в презентації	Вибрати один із параметрів: • відтворення в усіх слайдах Відтворення одного аудіофайлу на всіх слайдах • безперервне повторення Аудіофайл повторюватиметься, доки ви вручну не зупините його, натиснувши кнопку Відтворити / Призупинити • Фонове відтворення Аудіофайл відтворюватиметься на всіх слайдах у фоновому режимі

Для запису мовного супроводу на комп'ютері має бути або мікрофон, або вебкамера.

Коли створення об'єкта завершено, для повернення до презентації достатньо клацнути мишею поза його межами або натиснути клавішу Esc.

Вбудований об'єкт суттєво збільшує розмір файлу презентації, оскільки зберігається разом із документом.

З'ясуємо, що означає технологія OLE.

OLE (англ. *Object Linking and Embedding* — зв'язування та вбудовування об'єктів) — технологія зв'язування та вбудовування об'єктів в інші документи й об'єкти, яка розроблена корпорацією Microsoft. Проте є обмеження. Вміст можна вставляти лише з програм, які підтримують OLE, і інстальовані на комп'ютері.

Щоб дізнатися, які типи вмісту можна вставити, на вкладці Вставлення в групі Текст слід натиснути кнопку Об'єкт. У полі Вставлення об'єкта перелічено типи об'єктів, які можна використовувати.

Для вставлення об'єкту слід виконати команду Вставка → → Об'єкт, у списку Тип об'єкта вибрати потрібний рядок і клацнути кнопку ОК. Запуститься програма для створення об'єкта, в результаті чого створений об'єкт буде вставлено у слайд.

Об'єкти, вставлені у презентацію в такий спосіб, називають вбудованими, тобто такими, що підтримують прямий зв'язок із материнською програмою. Після вставлення вбудованого об'єкта його можна легко редагувати подвійним клацанням, яке відкриє відповідну програму або надасть доступ до її засобів, або через контекстне меню.

Щоб зменшити розмір файлу презентації, об'єкт можна зберегти в окремому файлі, а потім дати на нього посилання так само, як це роблять для великих зображень. Для цього у діалоговому вікні Вставка об'єкта слід вибрати перемикач Створити з файлу, за допомогою кнопки Огляд знайти потрібний

файл, встановити прапорець Зв'язати і клацнути ОК. Зв'язаний об'єкт оновлюється, якщо його вихідний файл змінено.

Вставлення зв'язаного об'єкта є доцільним, якщо вихідний набір даних великий або складний. Але якщо зв'язок буде втрачено, то замість діаграми ми отримаємо сірий квадрат.

На відміну від зв'язаного, вбудований об'єкт втрачає зв'язок з батьківським файлом, зміни в якому не приведуть до змін у вбудованому об'єкті, який так і називається, оскільки вбудовується у презентацію. При цьому обсяг нашої презентації суттєво збільшується за рахунок вбудованого об'єкту. Тепер, коли ми розібралися в чому різниця між вбудованим та зв'язаним об'єктами, розглянемо алгоритм, як вставити об'єкт.

Приклад. Зв'язана діаграма, вбудована з даних у програмі Microsoft Excel, зміниться, якщо вихідні дані Excel змінено. Вихідний файл має бути доступний на комп'ютері або в мережі, щоб зберегти зв'язок між вставленим об'єктом і вихідними даними.

Крок 1	У програмі, в якій містяться потрібна інформація для презентації, виділіть і скопіюйте те, що потрібно вставити як об'єкт
Крок 2	У PowerPoint помістіть курсор миші на місце, де має відображатися об'єкт
Крок 3	На вкладці **Основне** в групі **Буфер обміну** відкрийте меню **Вставити** та виберіть пункт **Спеціальне вставлення** (рис. 6.27)
Крок 4	Щоб вставити інформацію як *зв'язаний об'єкт*, виберіть пункт **Вставити посилання**; щоб вставити інформацію як *вбудований об'єкт*, виберіть пункт **Вставити**, а у полі **Як** натисніть запис зі словом «об'єкт» у назві її імені. Наприклад, якщо ви скопіювали інформацію з документа Word, натисніть кнопку **Об'єкт документа Microsoft Word**
Крок 5	Виберіть слайд, на якому потрібно розташувати об'єкт
Крок 6	На вкладці **Вставлення** у групі **Текст** натисніть кнопку **Об'єкт** (рис. 6.28).

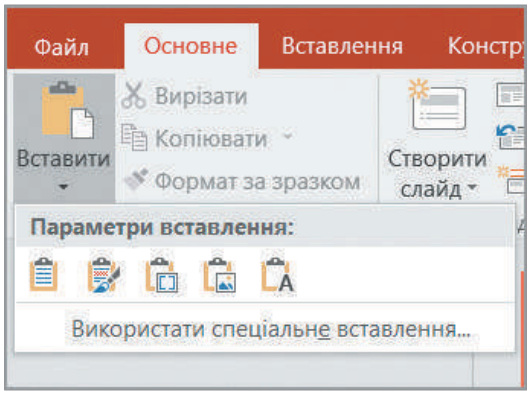


Рис. 6.27. Параметри вставлення з буферу обміну

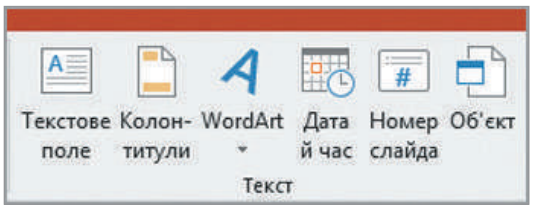


Рис. 6.28. Вигляд групи **Текст**

Крок 7	Натисніть кнопку **Створити з файлу** (рис. 6.29).
Крок 8	У полі **Файл** введіть ім'я файлу або натисніть кнопку **Огляд**, щоб вибрати зі списку
Крок 9	Установіть прапорець **Зв'язати**

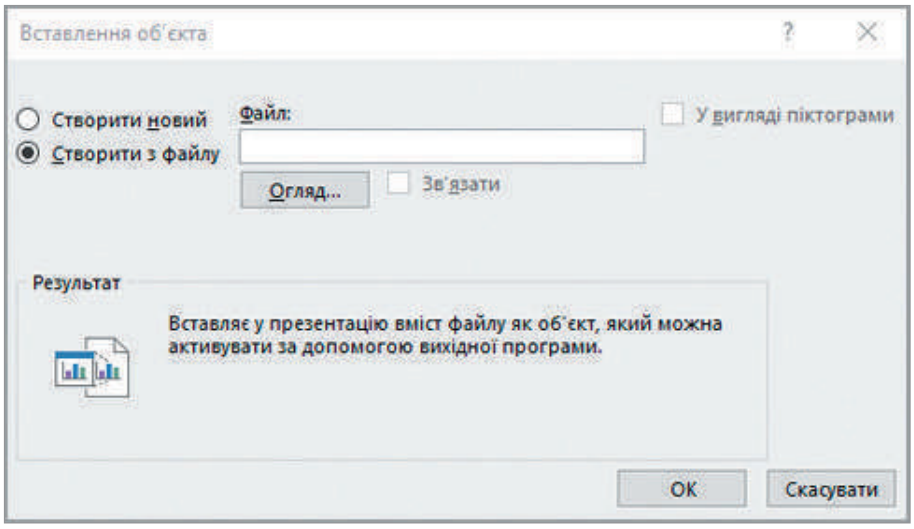


Рис. 6.29. Вигляд меню **Вставлення об'єкта**

Далі потрібно виконати одну з таких дій:

- щоб відобразити вміст презентації, зніміть прапорець Відображати як піктограму;
- щоб відобразити піктограму, яку клацнули, щоб переглянути об'єкт, установіть прапорець Відобразити як піктограму.

Щоб змінити зображення або підпис за замовчуванням, необхідно натиснути кнопку Змінити піктограму, а потім вибрати потрібну піктограму зі списку піктограм. За потреби можна ввести підпис у полі Підпис.

Запитання для перевірки знань

1. Як додати відеокліп до слайду?
2. Чи можна редагувати аудіофайл засобами PowerPoint? Якщо так, опишіть алгоритм.
3. Що таке технологія OLE?
4. Яка різниця між вбудованим та зв'язаним об'єктом, переваги та недоліки?
5. Поміркуйте, чи можливо зв'язати відеокліп з презентацією. Які обмеження при цьому виникають?
6. Чи достатньо мати веб-камеру для запису дикторського тексту? Свою відповідь обґрунтуйте.

Завдання для самостійного виконання

1. Відредагуйте презентацію, виконану як завдання до § 6.3

6.6. Керування показом презентації

Ми створили презентацію і настав момент показу. Що для цього потрібно?

Програма PowerPoint надає досить розвинений інструментарій для налаштування показу презентації (рис. 6.30).

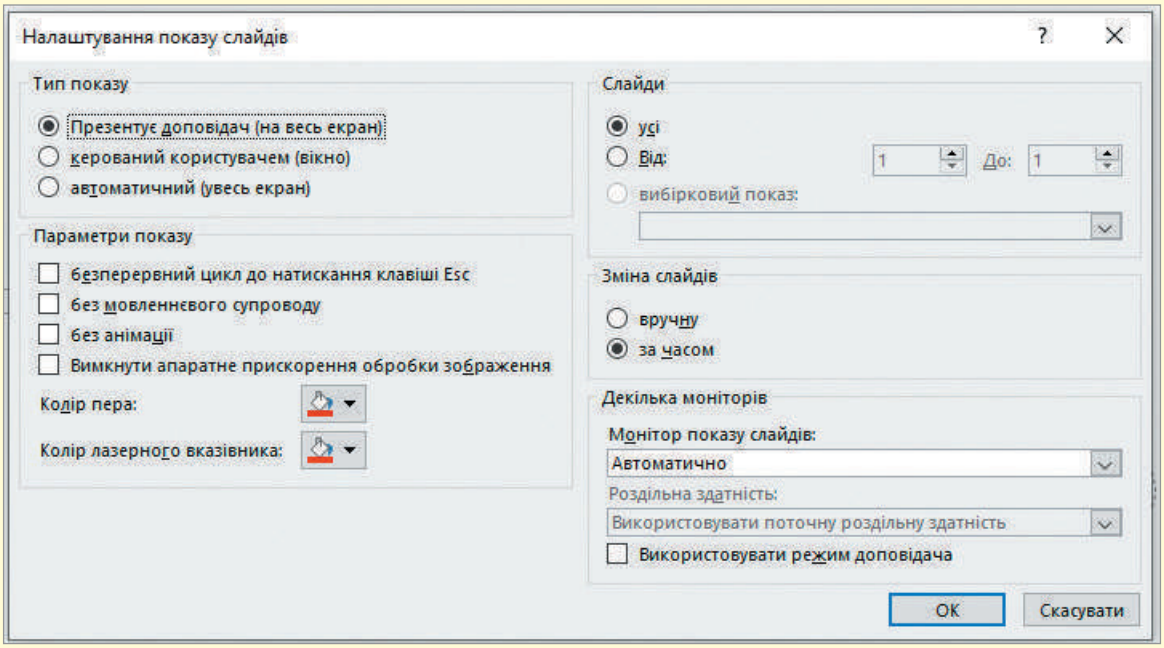


Рис. 6.30. Налаштування показу слайдів

У процесі показу слайдів можна скористатись елементами керування, які з'являються в режимі демонстрації (рис. 6.31).



Рис. 6.31. Меню керування презентацією в режимі демонстрації

Щоб перейти до будь-якого слайда презентації, потрібно клацнути ПКМ елемент керування екран, вибрати команду Перейти до слайда, увести потрібний номер слайда в полі Слайд і клацнути кнопку OK.

У меню налаштування презентації можна вибрати, які саме слайди відображатимуться в режимі демонстрації. За замовчуванням вибираються всі. Проте можна вибрати, які слайди відображатимуться під час показу або створити довільний показ (рис. 6.32).

- Щоб почати презентацію з першого слайду, слід на вкладці **Показ слайдів** натиснути кнопку **Відтворити з початку** або клавішу F5
- Щоб почати презентацію з певного слайду, слід вибрати його і натиснути команду **З поточного слайду** на вкладці **Показ слайдів** або клавіші Shift + F5

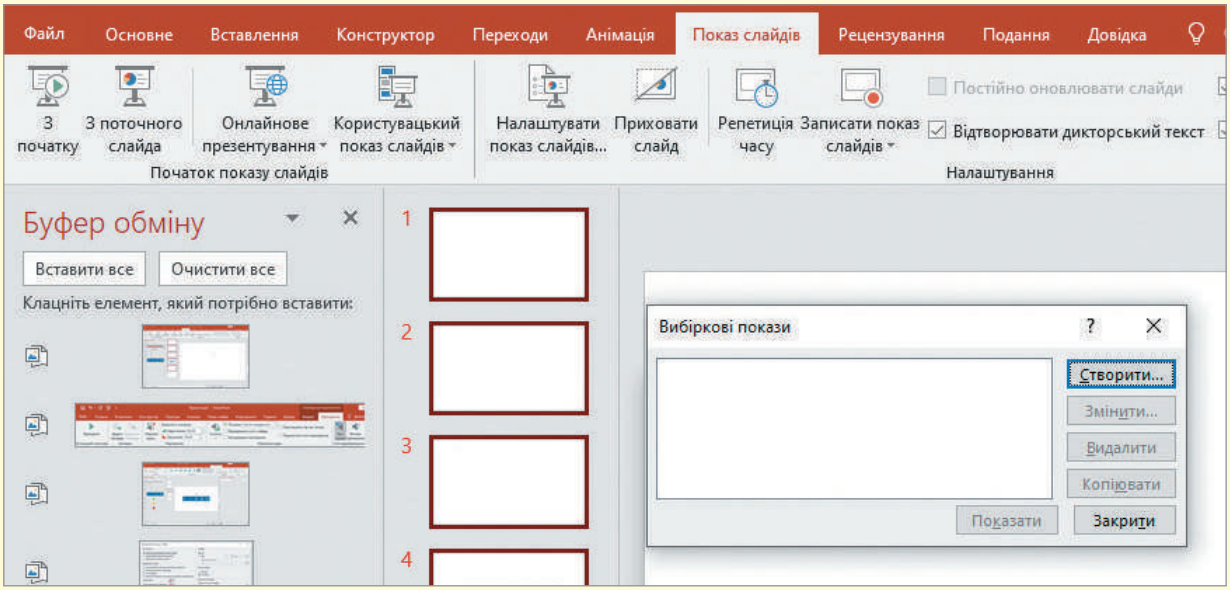


Рис. 6.32. Вибірковий показ

Для вибору вибіркового показу потрібно на вкладці Показ презентації вибрати команду Довільний показ, команду Створити і у діалоговому вікні задати ім'я довільного показу. Далі слід вибрати слайди і налаштувати послідовність показу слайдів кнопкою перемістити вгору і вниз (рис. 6.33), зберегти, в налаштуванні показу вибрати команду Довільний показ і створену послідовність.

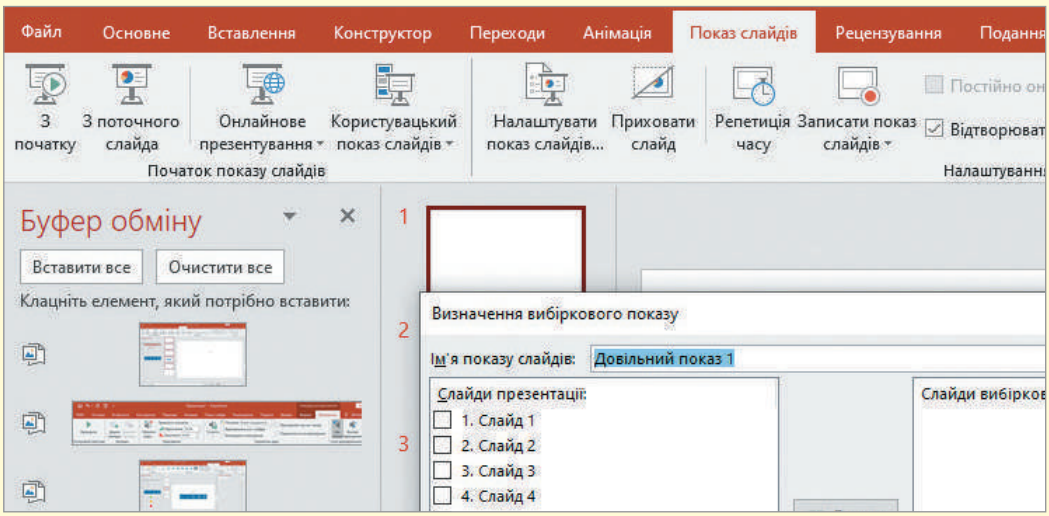


Рис. 6.33. Визначення слайдів у вибірковий показ

Ми розглядали можливість створення розробником навігації та обирали команду автоматичне відтворення в меню Налаштування показу. Згадаємо, що під час демонстрації зникають елементи керування і користувач може лише натискати на кнопки дій та клавішу Esc для завершення показу.

Запитання для перевірки знань

1. На якій вкладці налаштовується **Показ презентації**?
2. Якими гарячими клавішами можна скористатись?
3. Чи можна під час показу презентації перейти на довільний слайд? Як це здійснити?
4. Коли елементи керування недоступні під час демонстрації презентації?
5. Поміркуйте, в яких випадках зручно скористатись можливостями довільного показу.
6. Як закрити від користувача певні слайди на презентації? Опишіть всі способи.

Завдання для самостійного виконання

1. Об'єднайте в одну презентацію завдання, виконані в § 6.4.
2. Налаштуйте вибірковий показ слайдів презентації.

Практична робота № 11

ТЕМА. Проектування та розробка розгалужених презентацій за визначеними критеріями. Використання вбудованих та зв'язаних об'єктів у презентації

ЗАВДАННЯ: виберіть одну з наведених тем для створення презентації (бажано тему з практичної роботи № 10).

ОБЛАДНАННЯ: комп'ютер із доступом до інтернету, відеоредактор.



Теми для створення презентації:
- Найбільші зоопарки світу
- Дивовижний світ комах
- Сонячна система
- Суперкомп'ютери

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Знайдіть в інтернеті відомості з вибраної теми, збережіть матеріали у папці.
2. Створіть титульний слайд із назвою презентації та відомостями про автора.
3. На другому слайді розмістіть принаймні п'ять назв об'єктів із вибраної теми.
4. Для кожного об'єкта створіть окремий слайд і розмістіть на ньому фотографію і стислу анотацію, що з'являється, якщо на фотографію навести вказівник миші.
5. Створіть перехід із другого слайду з кожної назви на відповідний слайд (перехід за клацанням миші).
6. Забезпечте повернення з кожного слайду на другий (зі змістом) за допомогою кнопки дії (за клацанням миші).
7. На останньому слайді розмістіть відповідну карту (світу, Сонячної системи), на якій позначте місцерозташування об'єкта трикутником. При його клацанні презентація переходить на слайд з інформацією про відповідний об'єкт.
8. Збережіть презентацію на Google-диску та надайте посилання вчителю.

Зробіть висновки: коли має сенс використовувати гіперпосилання, а коли достатньо використовувати кнопки дій.

Практична робота № 12

ТЕМА. Розробка презентацій з елементами анімації, відеокліпом, звуковими ефектами та мовним супроводом

ЗАВДАННЯ: додати до презентації, створеної у практичній роботі № 11, анімацію, відеокліп, мовний супровід за алгоритмом.

ОБЛАДНАННЯ: комп'ютер із доступом до інтернету, відеоредактор.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. На першому (титульному) слайді додайте ефект Вхід → → Виліт згори до блоку з темою презентації, а до блоку з інформацією про автора — ефект Вхід → Виліт знизу.
2. Додайте до всіх фотографій на слайдах ефект Вхід → Випливання з параметром → Після попереднього та затримкою на 0,5 с.
3. Додайте слайд із відеокліпом, який було створено під час роботи з практичною роботою № 11.
4. Засобами PowerPoint (Робота з відео → Монтаж) скоротіть відеокліп до 2 хв. Вимкніть у відеокліпі звук.
5. Засобами PowerPoint (Вставка → Звук → Записати звук) додайте дикторський текст на слайд з відеокліпом. Якщо він відтворюється довше за відео, то обріжте звук.
6. Виберіть на власний розсуд та налаштуйте переходи між слайдами, забезпечте автоматичний перехід від титульного до слайду зі змістом. Інші слайди — за клацанням миші.
7. Розмістіть презентацію на власному Google-диску.
8. Надайте доступ вчителю.

Зробіть висновки: щодо використання анімаційних ефектів у презентації; як додано відеокліп та мовний супровід у презентацію; чи потрібно використовувати механізм зв'язування презентації та відеокліпу.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 6

Розділ 7
ТЕХНОЛОГІЇ ОПРАЦЮВАННЯ ЧИСЛОВИХ ДАНИХ У СЕРЕДОВИЩІ ТАБЛИЧНОГО ПРОЦЕСОРА

7.1. Типи величин у текстовому процесорі. Уведення даних

Що таке електронна таблиця? Для чого її застосовують?

Як вам відомо, табличний процесор — це прикладна програма для працювання даних різних типів, поданих в електронних таблицях (ЕТ). Існують різні табличні процесори, і один із них — табличний процесор Microsoft Excel (далі — Excel), який входить до складу комерційного пакета Microsoft Office.

Інтерфейс Excel подібний до інтерфейсів Word і PowerPoint.

Щоб ознайомитися з усіма типами даних, які опрацьовує Excel, потрібно виконати наведений алгоритм.

Excel

Крок 1	Відкрийте книгу табличного процесора у меню стрічки
Крок 2	Перейдіть на вкладку **Число** і відкрийте її
Крок 3	Перейдіть до вікна **Формат клітинок** (рис. 7.1)

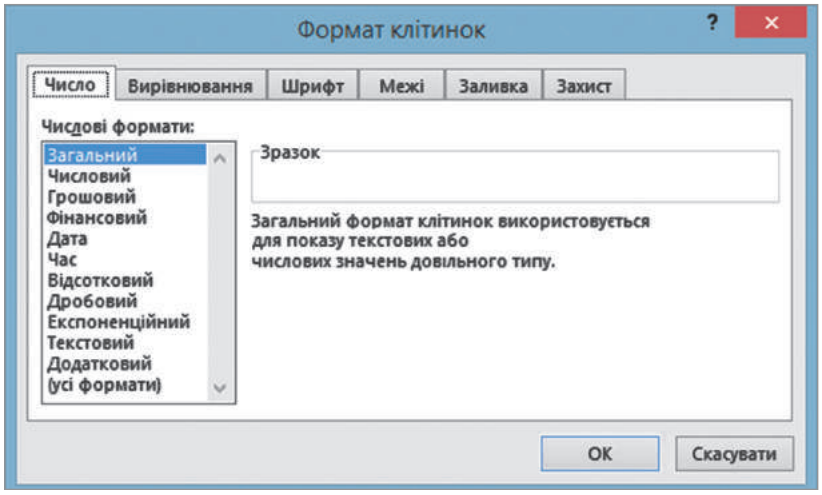


Рис. 7.1. Вигляд вкладки **Число** вікна **Формат клітинок**

За особливостями опрацювання дані в табличному процесорі умовно розподіляють на три типи: числові, текстові, формули. Дату і час, а також грошовий формат відносять до чисел. При введені даних Excel автоматично розпізнає їх тип. Набір цифр Excel сприймає як число, а символ «‘» і введені цифри — як текст, введення формули починається зі знаку «=».

Розглянемо кожний тип даних окремо.

До числового типу належать:

- цілі та дійсні числа;
- грошової одиниці (після числового значення автоматично проставляється вибрана грошова одиниця);
- дата (можна встановити формат виведення дати; підтримуються всі найчастіше використовувані формати. У пам’яті комп’ютера такі дані зберігаються як число, що відображає кількість днів від початку двадцятого століття до введеної дати);
- час;
- відсоток (після числа автоматично проставляється %).

Для виведення на екран дійсних чисел у Excel є можливість установити кількість цифр після десяткової точки. Також можна встановити вигляд чисел в експоненціальному вигляді (з рухомою крапкою), наприклад, число 0,512342 буде записано як 5,12E-01.

У більшості випадків табличний процесор автоматично визначає формат даних. Щоб це відбувалося коректно, необхідно дотримуватися певних правил. Ознайомимося з ними.

Сучасні табличні процесори: LibreOffice Calc, Microsoft Offise Excel, SuperCalc, GNumersc з пакета GNOME Office та ін.
Лідером за популярністю є Excel, йому віддають перевагу понад 80 % користувачів.

Типізація даних в ЕТ необхідна для раціонального використання пам’яті, відображення даних у прийнятному форматі та можливості коректного опрацювання даних різного типу.

- Починайте введення від’ємного числа зі знаку «–» або беріть число у круглі дужки.
- Починайте введення числа у грошовому форматі зі знаку долара ($).
- Для введення відсотків після числа набирайте символ відсотка (%).
- Для введення простого дробу вводьте число у такому форматі: Ціла частина → → Пропуск → Чисельник / Знаменник. Цілу частину вводьте завжди (навіть якщо вона дорівнює нулю), оскільки в разі її відсутності Excel може сприйняти дані як дату.
- Під час введення десяткового дробу для відокремлення цілої та дробової частин використовуйте кому.
- Для введення даних в експоненціальному вигляді використовуйте символ E або e. Наприклад, щоб ввести число $2 \cdot 10^5$, набирайте 2e5.

У табличному процесорі є група функцій для опрацювання текстових даних. Під час введення цифр програма розпізнає їх як дані числового типу. Якщо ці дані й надалі будуть опрацьовуватись як текстові, введення слід починати з символу «‘», або зробити активною клітинку для введення, у вікні Формат клітинок (див. рис. 1) призначити клітинці текстовий формат, а потім увести дані.

Введення даних здійснюється безпосередньо в активну клітинку таблиці або в рядок формул, який розташований у верхній частині вікна програми безпосередньо під панелями інструментів.

Текст є обов’язковим атрибутом будь-якої таблиці, оскільки з його допомогою оформляють її шапку (заголовок), а також надають необхідні коментарі.

Дані однакового вигляду (цифри як число і цифри як текст) у клітинках із різним форматом автоматично мають різне вирівнювання: текстові за правим краєм клітинки, числові — за лівим. Вирівнювання можна встановлювати командами групи вирівнювання вкладки Основне.

Ширина стовпця не завжди відповідає ширині введених даних. Для числових даних у клітинці з'являється решітка, наприклад «###». А текстові дані «перетікають» в сусідні клітинки, якщо ті вільні.

Якщо в сусідніх клітинках присутні дані, створюється враження, що текст «обрізаний», але в рядку формул видно, що вміст клітинки в повному обсязі, просто його не видно. Необхідну ширину стовпця зручно встановити з автоматичним вибором найдовшої ширини даних (будь-якого типу): поьрібно навести вказівник миші на межу між стовпцями (він змінить вигляд) і клацнути.

Оскільки таблиці часто складаються з однотипних або повторюваних даних, Excel містить засоби автоматизації введення. Для роботі з числами зручно використовувати автозаповнення протягуванням миші. У правому нижньому кутку рамки поточної клітинки є чорний квадратик — маркер заповнення. У разі наведення покажчика миші (зазвичай він має вигляд товстого білого хреста), він набуває форми тонкого чорного хрестика.

Перетягування маркера заповнення розглядається як операція «розмноження» вмісту клітинки в горизонтальному або вертикальному напрямі, залежно від того, куди тягнути мишу.

- Якщо клітинка містить значення, то під час перетягування маркера відбувається копіювання клітинок.
- Якщо виділити дві клітинки, відбувається заповнення арифметичною прогресією: Excel автоматично визначає крок прогресії і додає числа з урахуванням кроку.

Щоб точно визначити умови заповнення клітинок, слід скористатися заповненням з викликом команди Прогресія. У відповідь відкриється вікно для встановлення параметрів прогресії. На рис. 2 показано вікно програми з вибраними параметрами для завдання для самостійного виконання.

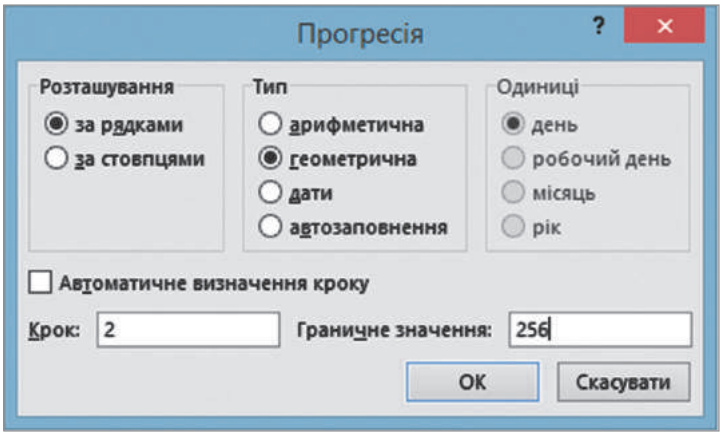


Рис. 7.2. Вигляд вікна **Прогресія**

Для переходу з рядку на рядок в межах клітинки можна скористатися сполученням клавіш Alt + Enter.

- Першою була програма VisiCalc (Visible Calculator). Її розробив Дон Брікклін у 1979 р. для комп'ютерів Apple II.
- Значний вплив на розвиток табличних процесорів зробив пакет LOTUS 1-2-3 фірми Lotus Development. Тривалий час цей табличних процесорів був найпоширенішим, зараз він є другим після Excel.

У 1982 р. фірма Microsoft випустила Multiplan і він став прототипом табличних процесорів нового покоління. У травні 1985 р. був офіційно представлений табличний процесор Excel, розроблений для комп'ютерів Macintosh. У наступних версіях Excel нарощував можливості і ставав зручнішим у використанні.

Запитання для перевірки знань

1. Якого типу дані можуть бути в клітинці?
2. Що належить до числових даних у Excel?
3. Як показати Excel, що введений набір цифр є текстом?
4. З чого починають введення формул?
5. Як створити та змінити формат введення дійсних чисел у клітинку?
6. Як введену дату перетворити на число?

Завдання для самостійного виконання

1. У клітинку A1 уведіть 1, у клітинку A2 — 2.
2. Виділіть діапазон A1:A2, встановіть на маркер автозаповнення вказівник миші — він змінить вигляд.
3. Затисніть ЛКМ і, не відпускаючи, протягніть вказівник миші вниз — у стовпці отримано значення 1 2 3 4 … .
4. Зробіть активною клітинку A1. Для цього на вкладці **Основне** перейдіть до групи **Редагування** і відкрийте список **Заповнення**. Виберіть команду **Прогресія** — відкриється вікно встановлення параметрів прогресії.
5. Виберіть **Розташування за рядками**, **Тип прогресії** — **геометричний**, установіть **Крок 2**, а **Граничне значення** — 256.
6. Натисніть кнопку **ОК** — у рядку отримали числа (степені числа 2).

7.2. Формули. Опрацювання формул

Які особливості введення формульних даних в клітинку?

Ви вже знаєте, що у клітинку табличного процесора можна ввести числові дані (згадаємо, що до них належить дата, час, дані грошового формату), текстові дані та формули.

Сучасні табличні процесори дозволяють автоматизувати різноманітні сфери бухгалтерської і економічної діяльності, здійснювати фінансово-економічний і статистичний аналіз, знаходити кращі варіанти для розв'язування складних економічних задач за певними критеріями.

Формула — вираз, який складається з констант, функцій і посилань на клітинки, з'єднаних знаками операцій.

Запис формули починається знаком рівності (=).

Дужки дозволяють змінювати стандартний порядок виконання дій у формулі. Якщо клітинка містить формулу, то в рядку формул буде відображено формулу клітинки, а безпосередньо в клітинці — результат її обчислення.

Зазвичай після введення формули у клітинку показує результат її обчислення, а в рядку формул відображає формулу. Для перегляду саме формул необхідно змінити параметри відображення вмісту клітинки — в додаткових параметрах (Файл → Параметри → Додатково) зробити активною команду Відображати формулу в клітинках замість обчислених результатів.

У формулі *константи* — це конкретні дані, введені в клітинку, а *посилання* — вказівки для формули, звідки брати дані.

Посиланнями можуть бути як окремі клітинки, так і блоки клітинок. *Блоки* — прямокутні ділянки: горизонтальний — частина рядка, вертикальний — частина стовпця або клітинки, які утворюють прямокутну ділянку.

Блоки вказують назвами клітинок через «:» — перша та остання для рядка чи стовпця, чи діаметра прямокутної ділянки. Наприклад: горизонтальний блок — B10:K10; вертикальний блок — B10:B15; прямокутний блок — B10:K15.

Формула може містити посилання, тобто адреси клітинок, уміст яких використовується в обчисленнях. Це означає, що результат обчислення формули залежить від числа, що міститься в іншій клітинці. Значення, що відображається в клітинці з формулою, перераховується у разі змінення значення клітинки, на яку вказує посилання. Оскільки клітинки з формулами копіюють як і клітинки з простими даними, має значення, який тип посилання використовується в формулі.

Посилання діляться на відносні, абсолютні, та змішані.

Відносними називають посилання, при яких табличний процесор пам'ятає розташування клітинки з даними відносно клітинки з формулою і під час копіювання у формулу підставляються дані з клітинки з таким самим відносним розташуванням (приклад 1).

Абсолютними називають посилання, при яких адреса клітинки з даними не змінюється під час копіювання формули.

Елементи номера клітинки, які використовують абсолютну адресацію, передуються символом $ (приклад 2).

Змішаними називають комбіновані посилання, в яких використовуються відносне та абсолютне посилання, але для однієї складової назви клітинки (приклад 3).

У разі копіювання формул із відносними посиланнями змінюються клітинки з початковими значеннями, а з використанням абсолютних посилань клітинки не змінюються. Щоб автоматично змінити тип посилання, слід встановити курсор на це посилання у формулі та натиснути клавішу F4.

У посиланнях на клітинки інших аркушів перед адресою клітинки додається назва аркушу з знаком «!» (приклад 4).

Копіювання і переміщення клітинок у Excel можна здійснювати методом перетягування або через буфер обміну. Під час роботи з невеликою кількістю клітинок зручно використати метод перетягування, а з великими діапазонами — скористатися буфером обміну.

Автозаповнення формулами виконується так само, як автозаповнення числами.

У ході автозаповнення до уваги береться тип посилань в формулі: відносні змінюються відповідно відносних розташувань копії і оригіналу, абсолютні залишаються без змін.

Щоб присвоїти клітинці ім'я, потрібно звернутися до команди Формули → Визначити ім'я — у результаті розкриється вікно, в якому слід вписати ім'я. Редагування імені, створення нових та вилучення наявних здійснюється у вікні Диспетчера імен, яке викликається із вкладки Формули.

Імена та адреси клітинок і діапазонів, використані у формулах, називають посиланнями.

Приклад 1.
У клітинку B4 вставили формулу = A4+$B$1. A4 — відносне посилання, ліва клітинка у рядку від клітинки з формулою. При копіюванні формули з B4 у клітинку C10 дані будуть підставлені з B10.

Приклад 2.
У прикладі 1 $B$1 є абсолютним посиланням і в клітинці C10 будуть дані з $B$1. Формула матиме такий вигляд: = B10+$B$1.

Приклад 3.
K10 — відносне посилання; $K$10 — абсолютне посилання; $K10 і K$10 — змішані посилання: в першому при копіюванні не змінюється стовпець, а в другому — рядок.

Приклад 4.
У клітинку Аркуша1 вставляється значення клітинки з Аркуша3: отримаємо такий вигляд посилання: Аркуш3!K17. Якщо Аркуш3 було перейменовано, то буде його назва.

Запитання для перевірки знань

1. Що містить формула?
2. Які типи посилань ви знаєте?
3. Як показати абсолютне посилання на клітинку?
4. Як показати використання блоку?
5. У клітинці C10 міститься формула =$A$10*B10. Який її вигляд буде при копіюванні в клітинку K105?

Завдання для самостійного виконання

1. У клітинки діапазону C1:K1 уведені числа від 1 до 9, в клітинки діапазону B2:B10 теж уведені числа від 1 до 9.
   Яку формулу потрібно ввести в клітинку C2, щоб під час її копіювання в клітинки діапазону C2:K10 отримати таблицю множення, наведену на рис. 7.3?

	A	B	C	D	E	F	G	H	I	J	K
1		*	1	2	3	4	5	6	7	8	9
2		1	1	2	3	4	5	6	7	8	9
3		2	2	4	6	8	10	12	14	16	18
4		3	3	6	9	12	15	18	21	24	27
5		4	4	8	12	16	20	24	28	32	36
6		5	5	10	15	20	25	30	35	40	45
7		6	6	12	18	24	30	36	42	48	54
8		7	7	14	21	28	35	42	49	56	63
9		8	8	16	24	32	40	48	56	64	72
10		9	9	18	27	36	45	54	63	72	81

Рис. 7.3. Приклад таблиці множення

7.3. Обчислювальні алгоритми в середовищі табличного процесора

Які типи даних може містити клітинка Excel? Що входить до складу формули?

Алгоритм — це послідовність команд для виконавця, що визначає, які дії та в якому порядку потрібно виконати для розв'язання певного завдання.

Табличний процесор широко використовується для розв'язування задач із різних галузей. Як і в програмуванні, спочатку слід скласти математичну модель задачі та реалізувати в алгоритмі розв'язування в таблиці.

За умовою задачі в таблицю потрібно ввести початкові дані. Модель у таблиці реалізується введенням необхідних формул у клітинки: здійснюється аналіз, у яких клітинках будуть формули, що формули міститимуть: константи, функції, посилання, якого типу посилання. Також з'ясовується, у яких клітинках міститимуться однотипні формули: зручно ввести формулу в одну клітинку, а надалі скопіювати її в інші.

Необхідно визначити, в якій клітинці буде показано результат розв'язування задачі.

Після завершення створення таблиці можна відформатувати її клітинки: виділити кольором, у деяких клітинках відформатувати вміст, наприклад, позначити його жирним накресленням. За потреби можна графічно візуалізувати отримані результати.

Для розв'язування задач використовують три базові алгоритмічні структури: слідування (лінійна), розгалуження, повторення (цикл).

Лінійні алгоритми — алгоритми, в яких вказівки виконуються послідовно в порядку їх подання.

Прикладами використання лінійних алгоритмів (рис. 7.4) можуть бути задачі для знаходження суми ряду чисел, їх середнього значення, задачі з даними для побудови математичних графіків, для підрахунку статистичних результатів опрацювання ряду чисел. У задачах знаходження максимального (мінімального) значення серед інших, необхідно порівнювати величини, вибирати більше (менше), але завдяки наявності в бібліотеці стандартних функцій їх розв'язання теж можна здійснити за допомогою лінійного алгоритму.

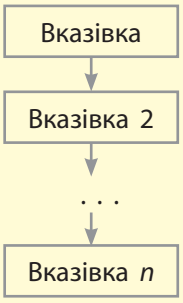


Рис. 7.4. Блок-схема лінійного алгоритму

Алгоритми з розгалуженням — алгоритмами, в яких відбувається вибір виконання певних вказівок на основі результату обчислення логічного виразу.

Алгоритми з розгалуженнями містять три складові (рис. 7.5):

- логічний вираз (умова);
- складова з константою або виразом для значення логічного виразу істина;
- складова з константою або виразом значення логічного виразу хибність.

Залежно від результату обчислення логічного виразу — істина (True) або хибність (False) — виконується одна із двох останніх складових. Алгоритм використовується і без останньої складової (скорочений розгалужений алгоритм).

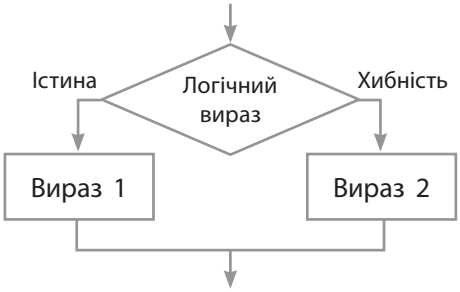


Рис. 7.5. Фрагмент блок-схема алгоритму з розгалуженням

Логічний вираз може бути **простим** — у ньому використовуються величини та знаки порівняння: =, >, >=, <, <=, <> (не дорівнює), і **складеним** — у ньому використовуються прості логічні вирази, поєднані логічними функціями та (AND), або (OR), виключне або (XOR), заперечення (NOT).

Результат обчислення таких виразів показано в таблиці:

Значення виразів				
A	**B**	**A AND B**	**A OR B**	**A** XOR B
Хибність	Хибність	Хибність	Хибність	Хибність
Істина	Хибність	Хибність	Істина	Істина
Хибність	Істина	Хибність	Істина	Істина
Істина	Істина	Істина	Істина	Хибність

Функція NOT(вираз) завжди має протилежне значення виразу:

NOT(TRUE) = FALSE
NOT(FALSE) = TRUE

Посилання на окремі клітинки та клітинки в блоках можуть мати різний вигляд:

- стовпець і рядок (наприклад, K102);
- порядковий номер рядка та порядковий номер стовпця (наприклад R10C3, який вказує на клітинку, що міститьсмя на перетині десятого рядка та третього стовпця), у звичному форматі це клітинка C10;
- ім'я клітинки; щоб присвоїти клітинці ім'я звертаються до команди **Формули** → → **Визначити ім'я** — розкривається вікно, в якому слід уписати ім'я. Редагування імені, створення нових імен і вилучення наявних здійснюється у вікні **Диспетчера імен**, яке викликається з стрічки **Формули**.

Алгоритми з розгалуженням в Excel реалізують у межах однієї клітинки за допомогою стандартних логічних функцій. Логічний вираз є порівнянням значення клітинки з константою або значенням іншої клітинки, або з результатом виразу.

Якщо в програмуванні складовими є блоки програмних вказівок, то в таблицях ними можуть бути різні значення, різні посилання, різні формули.

Прикладом використання алгоритму з розгалуженням може бути фрагмент таблиці, в якому залежно від віку дитини у клітинці вказується дитячий садок, школа. Логічним виразом буде вік < 6, де вік — це посилання на клітинку з віком особи. А вказівками 1 і 2 буде просто текст «дитячий садок», «школа».

Алгоритми з повторенням — алгоритми, в яких передбачено повторення деяких команд.

Для уникнення нескінченного повторення такі алгоритми завжди містять умовні вирази, від результату обчислення яких залежить припинення чи виконання повтору.

В умовному виразі використовують змінний параметр, значення якого впливає на результат виразу. Такі алгоритми часто реалізують за допомогою формул, у яких клітинки посилаються самі на себе. Щоб не було помилки, спочатку потрібно здійснити налагодження використання циклічних посилань.

Запитання для перевірки знань

1. Назвіть основні типи алгоритмів.
2. Наведіть приклад лінійного алгоритму.
3. Наведіть приклад алгоритму з розгалуженням.
4. Що таке логічний вираз в алгоритмі з розгалуженням?
5. Які логічні функції ви знаєте?
6. Наведіть приклад складного логічного виразу.

Завдання для самостійного виконання

1. Складіть блок-схему фрагмента обчислення значення y за формулою (алгоритм універсальний і x може набувати будь-якого значення) $y=x\sqrt{x-5}$.
2. Які зміни будуть у блок-схемі попереднього прикладу, якщо y обчислюється за наведеною формулою?
   Складіть блок-схему фрагмента обчислення значення y (алгоритм універсальний і x може набувати будь-якого значення) за формулою $y=\dfrac{1}{x\sqrt{x-5}}$.
3. Складіть блок-схему фрагмента обчислення значення розв'язків квадратного рівняння (алгоритм універсальний і a, b, c можуть набувати будь-яких значень)

$$x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}.$$

7.4. Призначення й використання основних функцій табличного процесора

Які типи алгоритмів ви знаєте? З чого складаються формули в табличному процесорі?

Для реалізації алгоритмів розв'язування різноманітних задач табличний процесор містить велику бібліотеку функцій.

Функція — це заздалегідь визначена формула, згідно з якою виконуються обчислення за заданими величинами, які називають аргументами, в зазначеному порядку.

Аргументом функції може бути число, посилання на клітинк (діапазон), вираз, для обчислення якого також можуть використовуватися функції.

Для використання функції у формулах спочатку вказують ім'я функції, потім у дужках — список аргументів подається через крапку з комою.

Є функції без аргументів — із порожніми дужками.

Вставлення функції завжди здійснюється безпосередньо в активну клітинку або у рядку формул. Для цього спочатку слід виділити клітинку, потім вказати функцію, аргументи та підтвердити командою ОК. Зазвичай ім'я функції не вписують, а викликають з бібліотеки зі стрічки на вкладці Формули.

Для вставлення функції у клітинку слід виділити клітинку, в яку потрібно вставити функцію. Якщо функція є складовою формули, почніть вводити формулу, а у місці вставлення функції перейдіть до наступного пункту. Далі потрібно перейти на вкладку Формули. Якщо ви знаєте групу, до якої входить функція, розкрийте список групи. Якщо не знаєте, в якій групі може бути функція, або група не показана на вкладці Формули, використайте команду Вставити функцію — відкриється вікно Вставлення функції (рис. 7.6).

Це вікно містить список для вибору групи функцій: Категорія. Після вибору категорії у вікні будуть показані її функції у алфавітному порядку. У списку категорія є команда Всі і тоді у відповідному вікні шукають необхідну функцію.

Нарешті можна вибрати необхідну функцію — відкриється вікно Аргументи для вибраної функції з полями для введення її аргументів. Далі можна увести аргументи функції, підтвердити дію клацанням кнопки ОК.

Якщо у вікні Аргументи назву аргументу позначено жирним накресленням, то він є обов'язковим і відповідне поле обов'язково потрібно заповнити. Аргументи, назви яких наводяться звичайним шрифтом, можна опускати.

У нижній частині вікон Вставлення функцій і Аргументи наведено опис призначення функції, а за наявності вказівника в полі аргументу — призначення змінного аргументу.

Усі функції згруповано за призначенням і розподілено за категоріями. З назви груп, наведених на вкладці **Формули**, зрозуміла галузь їх використання (наприклад, математична, текстова, логічна, фінансова, дата й час).
Для функцій, які використовуються для спеціальних обчислень, є група **Інші**.

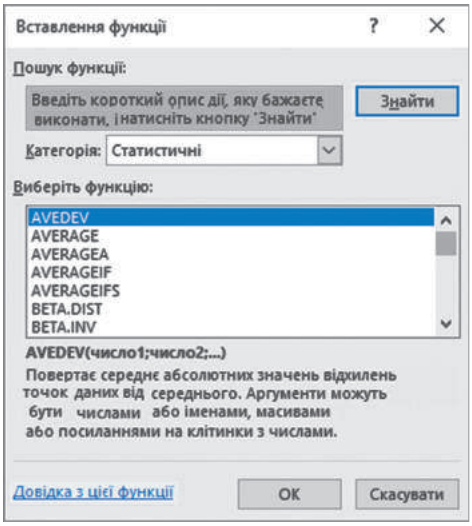


Рис. 7.6. Вікно **Вставлення функції**

Особливості введення: константи вводять з клавіатури, посилання — вручну або переходять з аркушу книги з потрібними даними і вибирають клітинки вказівником миші. Тип посилання — клавішею F4.

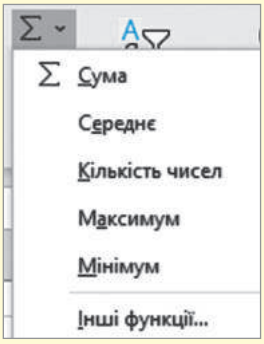


Рис. 7.7. Перелік найуживаніших функцій

Щоб вибрати одну клітинку, її потрібно виділити і клацнути лівою клавішею миші (ЛКМ), щоб вибрати кілька клітинок діапазону — затиснути ЛКМ на першій клітинці, «протяжкою» виділити необхідний діапазон і відпустити ЛКМ.

Щоб вибрати клітинки, розташовані не поруч, їх можна виділити з використанням клавіші Ctrl на клавіатурі. Але не для всіх функцій допустимі значення не з сусідніх клітинок.

Найпоширеніші для реалізації лінійних алгоритмів функції винесені окремо на вкладках Формули і Основне.

Із рис. 7.7 зрозумілі назву цих функцій і їхнє призначення.

Із наведеного переліку функція підрахунку суми — SUM належить категорії Математичні, а функції Середнє — AVERAGE, Кількість — COUNT, Максимум — MAX, Мінімум — MIN знаходяться в категорії Статистичні (на вкладці Формули ця категорія у списку Інші).

Для реалізації алгоритму з розгалуженням використовують логічну функцію ЯКЩО (IF; ЕСЛИ). Функція містить три аргументи (рис. 7.8).

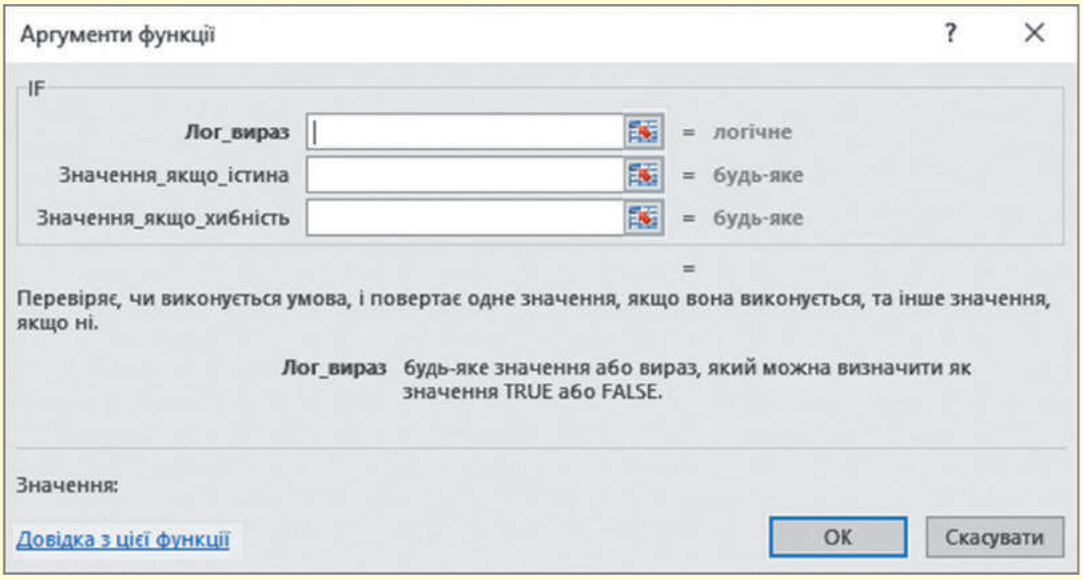


Рис. 7.8. Вікно **Аргументи функції**

Приклад.

Підрахуємо середнє значення зросту десяти учнів класу, а також максимальний зріст, мінімальний зріст та скільки є учнів заввишки 172 см. Початкові дані подано на рис. 7.9.

Для цього використаємо відповідні функції Сума(C2:C11), Максимум(C2:C11), Мінімум(C2:C11) і ПідрахунокЯкщо з умовою =172 і Діапазоном(C2:C11;»=172»).

В останній функції в полі аргументу умови лапки не ставлять, табличний процесор їх проставляє автоматично.

Алгоритм із розгалуженням реалізується і функцією підрахунку значень, які відповідають певним умовам ПідрахунокЯкщо (COUNTIF) — функція міститься в групі Статистичні.

До таблиці можна додати стовпець із реакцією на значення зросту: для зросту більш ніж 175 вивести Вже виріс, а інакше, тобто менше або дорівнює, — Ще підростеш. Для цього в клітинку D2 потрібно вставити функцію ЯКЩО(C2>175; «Вже виріс»; «Ще підростеш»). Як і в попередніх підрахунках, текст у лапки можна не брати — вони проставляються автоматично. Протяжкою миші отримаємо значення функції в усьому стовпці.

На рис. 4 таблицю показано двічі: з формулами та результатами обчислення формул.

	A	B	C	D
1	№	Учень	Зріст	
2	1	Учень 1	178	=IF(C2>=175;"Вже виріс";"Ще підростеш")
3	2	Учень 2	169	=IF(C3>=175;"Вже виріс";"Ще підростеш")
4	3	Учень 3	172	=IF(C4>=175;"Вже виріс";"Ще підростеш")
5	4	Учень 4	168	=IF(C5>=175;"Вже виріс";"Ще підростеш")
6	5	Учень 5	172	=IF(C6>=175;"Вже виріс";"Ще підростеш")
7	6	Учень 6	174	=IF(C7>=175;"Вже виріс";"Ще підростеш")
8	7	Учень 7	172	=IF(C8>=175;"Вже виріс";"Ще підростеш")
9	8	Учень 8	178	=IF(C9>=175;"Вже виріс";"Ще підростеш")
10	9	Учень 9	176	=IF(C10>=175;"Вже виріс";"Ще підростеш")
11	10	Учень 10	182	=IF(C11>=175;"Вже виріс";"Ще підростеш")
12	Середнє значення зросту		=AVERAGE(C2:C11)	
13	Максимальний зріст		=MAX(C2:C11)	
14	Мінімальний зріст		=MIN(C2:C11)	
15	Кількість із 172 см		=COUNTIF(C2:C11;172)	

	A	B	C	D
1	№ з/п	Учень	Зріст	
2	1	Учень 1	178	Вже виріс
3	2	Учень 2	169	Ще підростеш
4	3	Учень 3	172	Ще підростеш
5	4	Учень 4	168	Ще підростеш
6	5	Учень 5	172	Ще підростеш
7	6	Учень 6	174	Ще підростеш
8	7	Учень 7	172	Ще підростеш
9	8	Учень 8	178	Вже виріс
10	9	Учень 9	176	Вже виріс
11	10	Учень 10	182	Вже виріс
12	Середнє значення зросту		174,1	
13	Максимальний зріст		182	
14	Мінімальний зріст		168	
15	Кількість із 172 см		3	

Рис. 7.9. Візуалізація прикладу аналізу зросту учнів

Запитання для перевірки знань

1. Як називається вкладка з командами для вставлення формул?
2. Яким знаком відокремлюються аргументи в описі функції?
3. Наведіть приклад функцій для реалізації лінійного алгоритму.
4. Наведіть приклад функцій для реалізації алгоритму з розгалуженням.
5. Який алгоритм введення стандартних функцій у формулу?
6. Як використати у функції іншу функцію як аргумент?

Завдання для самостійного виконання

1. Відкрийте табличний процесор.
У клітинку A1 уведіть назву x, у клітинку B1 уведіть назву y, а в C1 — назву z. У клітинку A2 значення. У клітинку B2 уведіть формулу = A2*A2, а в C2 — формулу = A2 + B2. Зверніть увагу на значення у B2 і C2.

Перейдіть у налаштування показу формули замість її значення: **Файл** → **Параметри** → → **Додатково**. У параметрах **Відображення** цього аркушу поставте прапорець біля команди **Відображати формули в клітинках** замість обчислених результатів. Як змінився вигляд таблиці?

Практична робота № 13

ТЕМА. Розв'язування задач на обчислення
ЗАВДАННЯ: проаналізувати дані про населення України за областями.
ОБЛАДНАННЯ: комп'ютер із встановленим табличним процесором, підключений до інтернету.



5. Порівняйте з даними таблиці.

9. Отримали 100 %?

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте Вікіпедію. Здійсніть пошук даних за фрагментом Список областей України за чисельністю населення.
2. Відкрийте табличний процесор.
3. Скопіюйте таблицю з Вікіпедії до таблиці Excel; початок таблиці з клітинки A1.
4. У стовпці G порахуйте населення області на основі формули =E3*F3.
5. Протяжкою миші вставте формулу підрахунку в усі рядки.
6. У клітинці G30 порахуйте сумарну кількість населення України, у G31 — максимальне значення населення області, в G32 — мінімальне, G33 — середнє.
7. У стовпці H порахуйте відсоткове співвідношення населення області до населення України. Здійсніть операцію ділення в клітинці H3 (формула =G30/$G$30) і скопіюйте формулу в увесь стовпець.
8. Для стовпця H установіть формат клітинок — відсоток.
9. У клітинці H30 підрахуйте суму значень стовпця H.
10. У стовпці I на основі використання функції IF виведіть текст Більше 2 млн чи Менше 2 млн залежно від кількості населення в області.
11. У клітинці I31 на основі функції COUNTIF підрахуйте кількість областей, населення яких менше за 1 млн, в I32 — більше за 2 млн.
12. Збережіть файл із назвою Населення України.

Зробіть висновок щодо типу використання посилань у формулах.

7.5. Умовне форматування

Як у формулу вставити функцію?

Після розв'язування задачі в середовищі табличного процесора буває потрібно здійснити форматування таблиці. Тоді треба виокремити початкові дані, за необхідності позначити кольором клітинки з проміжними результатами.

Для візуальної інтерпретації результатів обчислення табличний процесор містить команду умовного форматування.

Умовне форматування — автоматичне форматування клітинок на основі встановлених правил для візуальної інтерпретації та аналізу значень клітинок.

Форматування називається умовним, оскільки правила форматування створюють на основі умовного виразу. Існують стандартні правила, також є можливість створювати свої власні, необхідні для конкретної таблиці.

Для здійснення умовного форматування потрібно вибрати однойменну команду на стрічці Основне. Відкривається список, у якому Excel пропонує вибрати правила форматування. На рис. 7.10 показано правила виділення клітинок на основі порівняння їх значень з конкретним значенням.

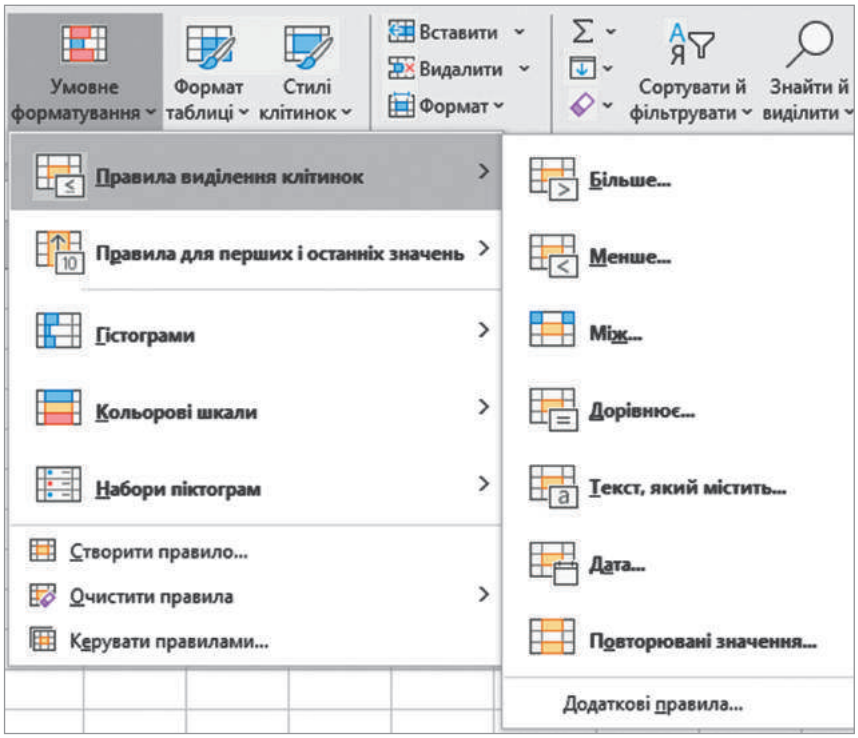


Рис. 7.10. Список правил умовного форматування

Вибір знаку порівняння відкриває вікно формування умовного виразу (рис. 7.11), в якому можна вибрати колір клітинок.

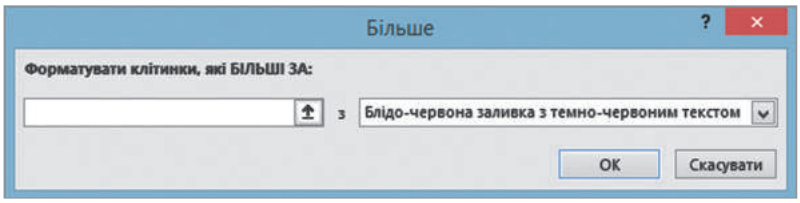


Рис. 7.11. Вікно правила форматування з умовою Більше

- Якщо умовний вираз, до складу якого входить значення клітинки, має значення True, то клітинка буде відформатована, якщо False — форматування не відбудеться.
- Якщо значення форматованих клітинок зміниться, автоматично відбудеться їхнє переформатування.

Умовне форматування дає змогу автоматично виділити клітинки та діапазони клітинок кольором та інтерпретувати їх значення гістограмами або вставленням піктограм.

Умовне форматування автоматично змінює формат клітинки на заданий, якщо виконується задана умова.

Для розуміння, як спрацьовує умовне форматування, розглянемо вже відомий нам приклад про зріст учнів.

Приклад.

Виділимо клітинки зі значенням зросту, більшим за середнє значення, червоним кольором, а з меншим — зеленим. Для цього:

1) виділимо діапазон C2:C11 і звернемося до умовного форматування з правилом виділення клітинок Більше;

2) у поле значення, з яким порівнюють значення клітинок діапазону, введемо C12 (клітинка із середнім значенням); залишимо форматування червоним кольором;

3) повторимо дії для правила Менше і виберемо зелений колір.

Клітинки буде відформатовано (рис. 7.12).

Якщо, наприклад, для Учня 4 його зріст ввели з помилкою та виправили на 175, то відбудеться автоматичне переформатування — клітинка зафарбується в червоний колір і зміниться текст в сусідній клітинці.

	A	B	C	D
1	№ з/п	Учень	Зріст	
2	1	Учень 1	178	Вже виріс
3	2	Учень 2	169	Ще підростеш
4	3	Учень 3	172	Ще підростеш
5	4	Учень 4	168	Ще підростеш
6	5	Учень 5	172	Ще підростеш
7	6	Учень 6	174	Ще підростеш
8	7	Учень 7	172	Ще підростеш
9	8	Учень 8	178	Вже виріс
10	9	Учень 9	176	Вже виріс
11	10	Учень 10	182	Вже виріс
12	Середнє значення зросту		174	
13	Максимальний зріст		182	
14	Мінімальний зріст		168	
15	Кількість із 172 см		3	

Рис. 7.12. Приклад таблиці з умовним форматуванням

Правила умовного форматування:

- правила виділених клітинок, які основою має порівняння значення клітинок із введеним значенням;
- правила для перших і останніх значень, при використанні якого форматуються клітинки з кількома найбільшими або найменшими значеннями у ряді значень, а також при порівнянні з середнім значенням ряду;
- правила форматування на основі вставлення в клітинку із значеннями гістограм, яка визначає відсоток значення конкретної клітинки по відношенню до клітинки з максимальним значенням у ряді: довжина стрічки гістограми відповідає значенню, чим воно більше — тим довша стрічка;
- правила вставлення піктограм мають мету показати різними піктограмами більші та менші значення у ряді;
- правила кольорових шкал мають основою використання дво- та триколірні відтінки для візуалізації значень. Рівень яскравості кольору вказує на великі, малі та середні значення ряду.

Щоб застосувати власне правило, користувачу необхідно скористатися командою Створити правило — відкривається вікно, з якого можна вибирати правила форматування.

Перші чотири правила є уточненням наявних правил бібліотеки. Також існує правило форматування значень, які зустрічаються в ряді один раз, і є можливість створити нове правило на основі введеної формули.

У вікні типів правил можна вибирати тип і формулу, ввести уточнення значень або відредагувати в рядку редагування. Командою Форматувати відкривається вікно, в якому форматують вигляд клітинки: її тло, тип меж, вміст.

Якщо скористатись командою Керувати правилами, то відкриється вікно (рис. 7.13) у якому дублюється команда створити правило. Якщо скористатись командою Редагувати правило, відкриється вікно з переліком типів правил (див. рис. 7.12).

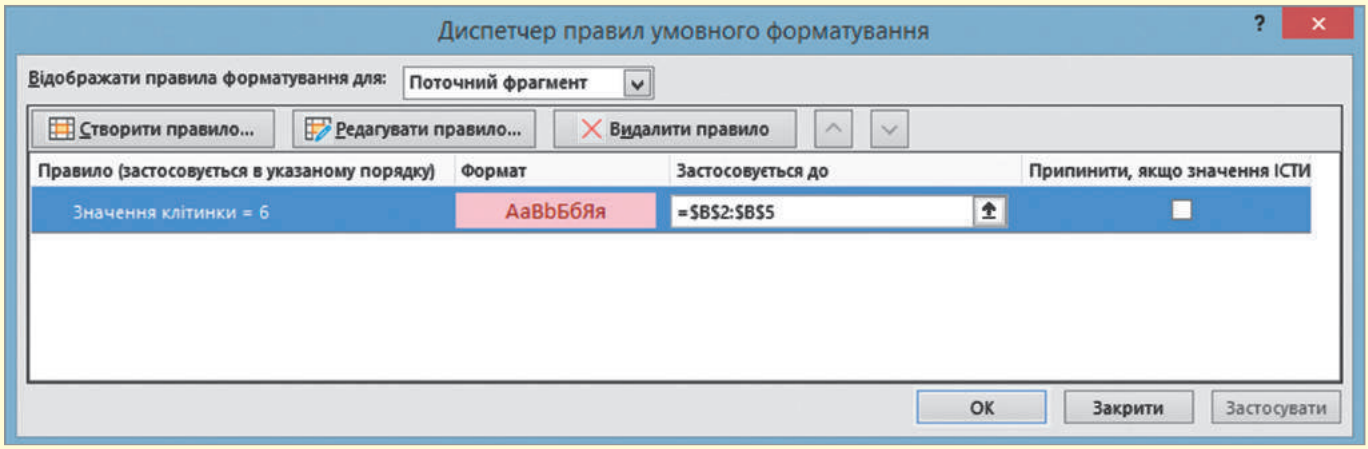


Рис. 7.13. Вигляд вікна команди керувати правилами

Щоб кардинально замінити правила умовного форматування, краще спочатку виконати команду Очистити правила, вибрати тільки для виділених клітинок або для всього аркуша та підтвердити вибір кнопкою ОК.

Запитання для перевірки знань

1. Що означає форматування клітинки?
2. На чому засновані правила умовного форматування?
3. Назвіть правила умовного форматування.
4. Чому автоматичне форматування клітинок містить назву «умовне»?
5. Як змінити наявні правила умовного форматування?

Завдання для самостійного виконання

Поміркуйте, як отримати візерунок на основі умовного форматування.

За зразком, наведеним на рис. 7.14, заповніть у зошиті порожню ділянку.

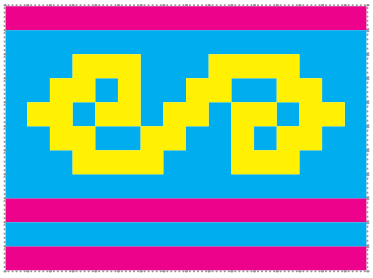


Рис. 7.14. Зразок візерунка

Практична робота № 14

ТЕМА. Умовне форматування
ЗАВДАННЯ: здійснити візуалізацію значень у таблиці про населення України.
ОБЛАДНАННЯ: комп'ютер із встановленим табличним процесором.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте у табличному процесорі файл Населення України (результат практичної роботи № 13).
2. Виділіть стовпець із площею областей.
3. Застосуйте до виділеного діапазону умовне форматування на основі правил вставлення гістограм.
4. До цього самого діапазону застосуйте правила вставлення піктограм.
5. Виділіть стовпець із кількістю населення в областях.
6. Кілька разів до значень виділеного діапазону застосуйте умовне форматування: для значення населення більше 2 млн; менше 1 млн.
7. Виділіть стовпець зі значеннями густоти населення та використайте умовне форматування для 10 найбільших значень.
8. Виділіть клітинку A3.
9. Виконайте команду Основне → Умовне форматування → Створити правило — відкриється вікно створення правил.
10. У вікні виберіть тип правила Використовувати формулу... .
11. У поле введення формул уведіть = G3<$G$33 (населення області менше за середнє по областях України), підтвердіть кнопкою OK.
    Протяжкою скопіюйте форматування в увесь стовпець.
12. Збережіть свою роботу командою Файл → Зберегти.

Зробіть висновки щодо можливостей табличний процесор щодо автоматичного форматування клітинок.

7.6. Створення та налагодження діаграм



Як краще проаналізувати результати контрольної роботи класу: порівнювати оцінки учнів в списку чи побудувати стовпчикову діаграму?

Для змістового навантаження графіків, діаграм важливо правильно вибрати тип і оформлення.

Візуалізація даних під час розв'язування задач в середовищі табличного процесора дає змогу одразу побачити загальну картину результатів дослідження, показати велику кількість даних на обмеженому проміжку, закцентувати увагу на деяких елементах ряду даних, порівняти фрагменти даних тощо.

Візуалізація даних досягається побудовою графіків, діаграм тощо. Так, діаграма містить назву, підписи на осях, за потреби — значення в певних точках діаграми. Якщо вона будується для кількох рядів значень, то буде різнокольоровою: кольори відповідають різним рядам.

Існують типи діаграм, в яких теж використовуються різні кольори, але до них додається легенда з відповідністю кольору певним значенням.

Алгоритм вставлення діаграми

Крок 1

Виділіть клітинки із значеннями. На стрічці **Вставлення** у групі **Діаграми** виберіть вигляд клітинки або відкрийте вікно **Вставлення діаграм** (рис. 7.15).

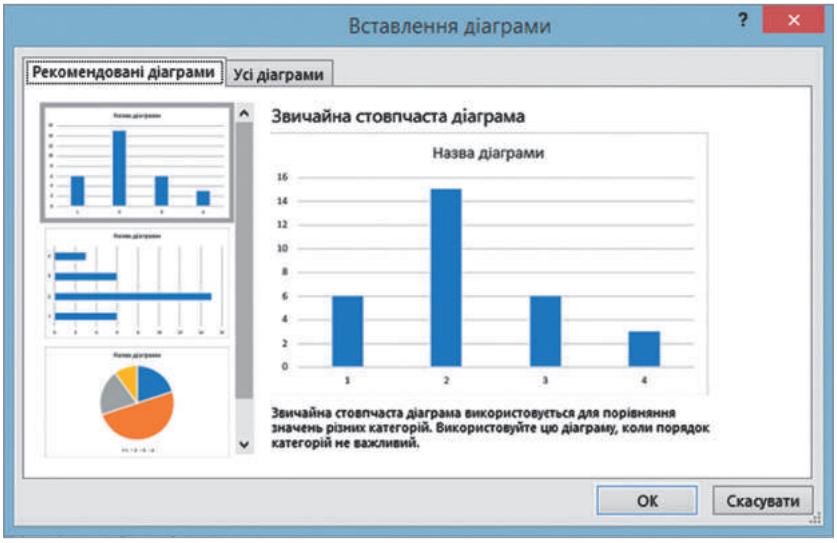


Рис. 7.15. Вигляд вікна **Вставлення діаграм**

Крок 2

На вкладці **Рекомендовані діаграми** на основі аналізу значень Excel одразу пропонує певний тип діаграм та їх вигляд. Виберіть з вкладки **Усі діаграми** тип діаграми на основі запропонованих зразків

На рис. 7.16 показано вигляд вкладки для двох рядів значень.

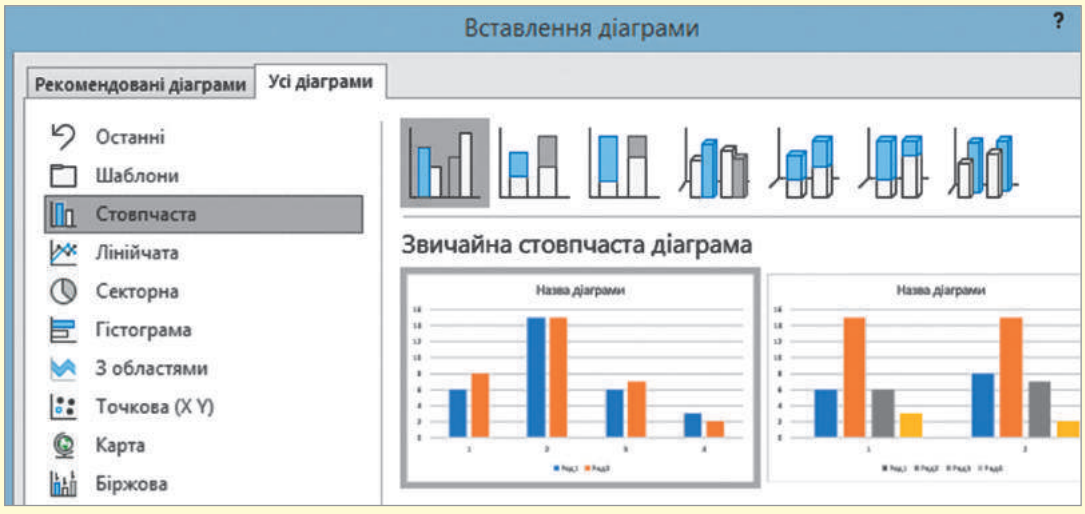


Рис. 7.16. Вигляд вкладки вікна **Вставлення діаграм**

Розглянемо типи діаграм з прикладами їх використання.

- Гістограми. Гістограми можуть бути як вертикальними (стовпчиковими), так і горизонтальними. Гістограми використовують для наочного подання динаміки зміни даних у часі або розподілу даних у вигляді прямокутних ділянок, розмір яких відповідає значенням, отриманих в результаті дослідження (приклади 1, 2)

Приклад 1. На рис. 7.17 подана стовпчикова діаграма даних зросту учнів класу.

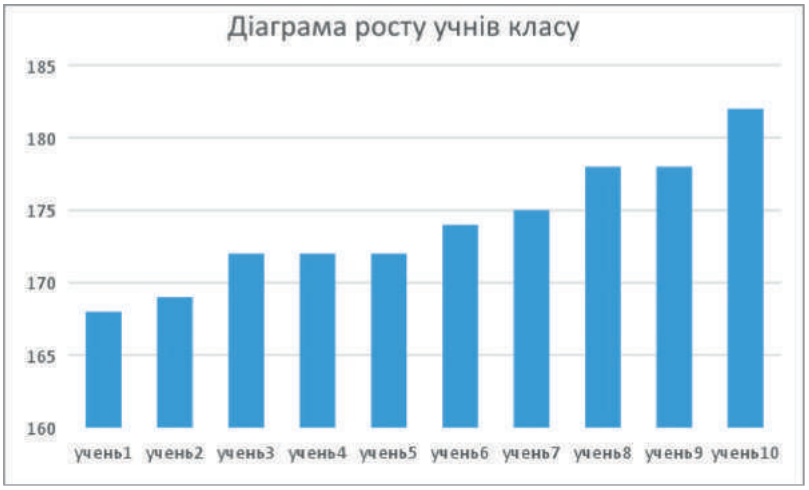


Рис. 7.17. Гістограма аналізу зросту учнів

Приклад 2.
На гістограмі зручно аналізувати, наприклад, успішність учнів класу з певного навчального предмету.
Якщо ввести їхні оцінки в таблицю Excel і за допомогою функції FREQUENCY (ЧАСТОТА) порахувати кількість учнів, які отримали високий бал (інтервал 10–12), достатній (7–9), середній (4–6) і низький (1–3), то на побудованій гістограмі будуть чотири стовпці. Їх висота відповідатиме кількості оцінок, які потрапляють у той чи інший інтервал.
До таблиці з оцінками з одного предмету можна додати кілька інших предметів і побудувати двоколірну діаграму.

- Кругові діаграми. Коли є набір значень і підрахована сума цих значень, зручно використовувати кругову діаграму для відображення відсоткового внеску кожного значення в загальну суму (приклад 3).

Приклад 3.
Якщо побудувати кругову діаграму, наприклад, з успішністю учнів, одразу буде видно, який відсоток учнів потрапляє в той чи інший знаннєвий інтервал (рис. 7.18).

Іншим яскравим прикладом може бути діаграма про кількість мешканців регіонів в загальній кількості населення в Україні.

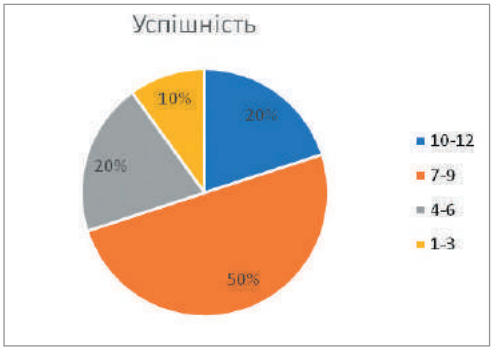


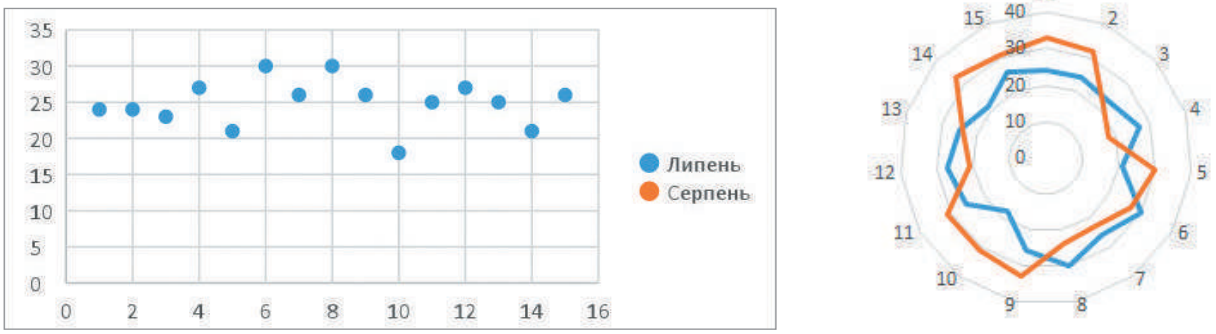
Рис. 7.18. Приклад кругової діаграми

- **Точкові діаграми.** Особливу популярність в дослідженнях мають точкові діаграми та їхній різновид — бульбашкова діаграма. Розмір бульбашки вказує на значення а її розташування на певний ряд досліджень.

Для наочного аналізу рядів даних зручно користуватись точковою діаграмою (для температурних режимів серпня) або пелюстковою (приклад 4).

Приклад 4.

На рис. 7.19 проілюстровано порівняння температурних режимів липня і серпня.



а б

Рис. 7.19. Приклади діаграм: точкової (а) і пелюсткової (б)

- **Комбіновані діаграми.** Ми розглянули деякі діаграми, в яких для порівняння значень кількох рядів даних використовували на основі діаграм одного типу. У Excel також є можливість створювати комбіновані діаграми, в зображенні яких для різних рядів даних використовуються різні типи діаграм (приклад 5).

Приклад 5. На рис. 7.20 наведено комбіновану діаграму двох рядів температурних даних за липень і серпень.

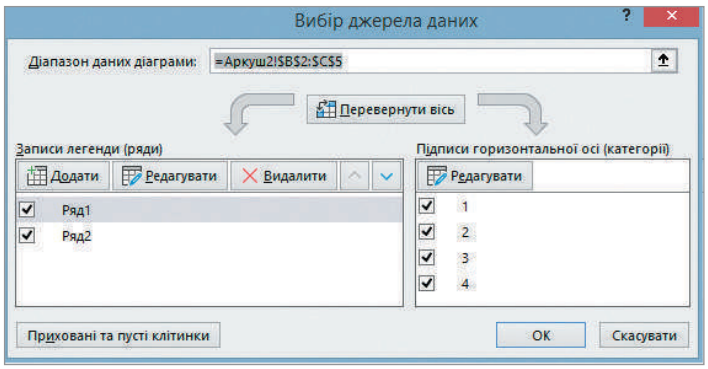


Рис. 7.20. Приклад комбінованої діаграми

Для налагодження вигляду діаграми та її редагування діаграму необхідно виділити. У головному меню з'являються додаткові вкладки: **Конструктор** і **Формат**. Якщо у вікні **Конструктор** діаграм або з контекстного меню діаграми викликати команду **Вибрати дані**, відкривається вікно редагування даних, на основі яких побудована діаграма.

Поле, за яким здійснюється сортування, називається **ключовим**, або **ключем сортування**.

Ряди можна видаляти, редагувати та додавати — для обох останніх команд відкриється вікно Редагування ряду. Також є можливість редагувати підписи на осі командою Редагувати (рис. 7.21), і у вікні, яке відкриється, вставити посилання на значення для підпису. Остання команда дуже важлива, оскільки в Excel у більшості типів діаграм вертикальна вісь містить значення, на основі яких побудована діаграма, а горизонтальна базується просто на переліку порядкового номера значення.

Форматування окремих частин діаграм здійснюється їх виділенням та вибором відповідних команд в контекстному меню. Відкривається область форматування, в якій змінюють необхідні параметри.

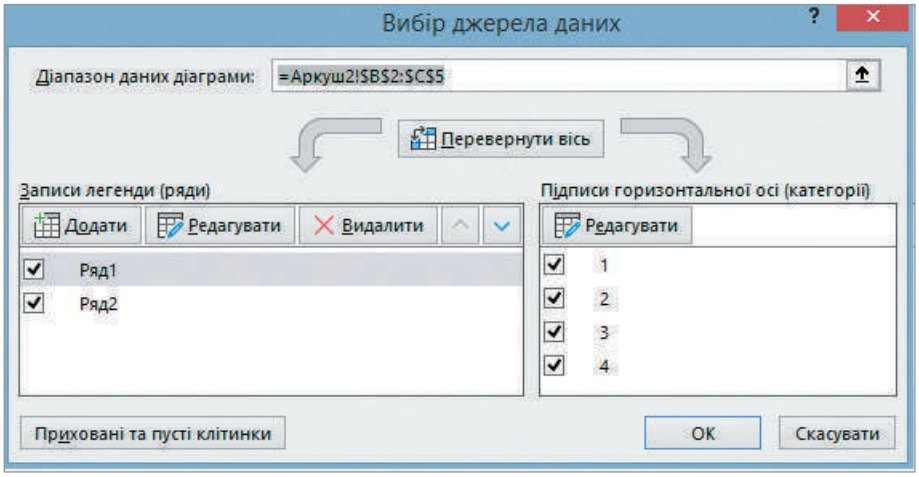


Рис. 7.21. Вигляд вікна редагування даних для діаграми

Як бачимо з рис. 7.21, діаграма побудована для двох рядів даних і в цих рядах по чотири значення, тому й підпис значень містить числа 1–4. У процесі редагування у поле значень вставляють посилання на клітинки початкових значень і їм будуть відповідати підписи осі.

Запитання для перевірки знань

1. Для чого використовують побудову діаграм?
2. Коли для ряду даних зручно створити гістограму, а в яких кругову діаграму?
3. Наведіть алгоритм вставлення діаграми.
4. На що вказує легенда в діаграмі?
5. Як змінити підписи на осях діаграми?
6. Яке призначення команд вкладки Формат для діаграм?

Завдання для самостійного виконання

1. Відкрийте таблицю **Населення України**.
2. Виділіть стовпці з густотою населення та побудуйте гістограму: **Вставлення** → **Гістограма** (звичайна або об'ємна стовпчаста).
3. Перейдіть до **Редагування гістограми**.
4. У вікні викличте **Редагування підписів** і уведіть діапазон клітинок із назвами областей.
5. Підтвердіть кнопкою **OK**.
6. Виділіть стовпці гістограми з густотою населення. У контекстному меню викличте команду **Змінити тип діаграми** — відкриється вікно **Редагування діаграми**.
7. У списку типу діаграми для ряду 2 замініть вигляд діаграми на свій смак.
8. Підтвердіть кнопкою **OK**.

7.7. Упорядковування даних у таблицях. Автоматичні та розширені фільтри

Які типи функцій в табличному процесорі ви знаєте?

Ви вже створювали в табличному процесорі таблиці з даними й розподіляли ряди даних за рядками і стовпцями таблиці на свій розсуд. Зазвичай для опрацювання даних, які мають багато властивостей, вносять в таблицю як список.

Список Excel — таблиця, оформлення якої відповідає певним вимогам. До оформлення таблиці існують певні вимоги.

- Список складається з рядків, які називають записами.
- Стовпці списку, які називають полями, містять дані одного типу.
- Верхній рядок списку, заголовок, містить мітки (імена) відповідних полів, і його формат (шрифт, колір фону і т. п.) повинен відрізнятися від формату записів.
- Всередині списку не повинно бути порожніх рядків і стовпців, якими список відокремлюється від іншої частини робочого листа.
- На робочому аркуші не варто розміщувати ще що-небудь, крім списку, а робочий лист рекомендується іменувати назвою списку.

Сортування списків

Під сортуванням списку, як і будь-якого іншого набору об'єктів, прийнято розуміти розташування його записів у певному порядку.

Способи сортування списків

- За допомогою кнопок Сортування за зростанням і Сортування за спаданням панелі інструментів ОСНОВНЕ
- За допомогою команди меню ДАНІ → група Сортування і фільтр → Сортування

Наведена команда дозволяє сортувати список за один прийом максимум за трьома полями (первинний ключ, вторинний тощо) — рис. 7.22.

Записи можна розташовувати в порядку зростання (зменшення) значень числових полів, у алфавітному (або зворотному алфавітному) порядку значень текстових полів, у хронологічному порядку полів типу дата і час.

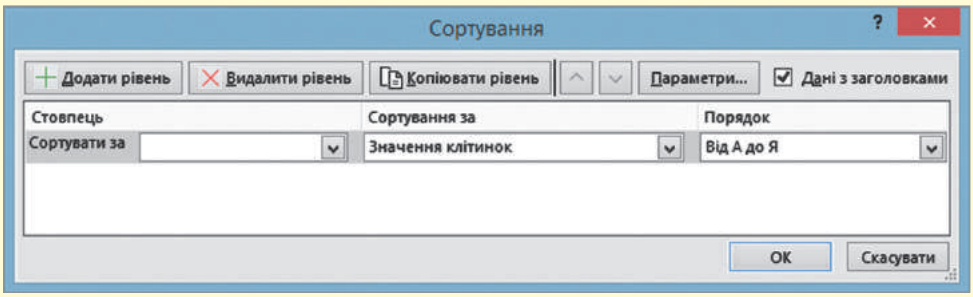


Рис. 7.22. Вигляд вікна Сортування

За більш ніж за трьома полями список сортується послідовно: за первинним ключем, відсортовані списки в межах первинних ключів відсортовуються за вторинним ключем і т. д.

Для сортування списків таблицю необхідно виділити. Якщо виділити лише стовпець, значення в якому сортують, необхідно обов'язково запускати команду Автоматично розширити виділений діапазон. Інакше буде відсортований лише виділений стовпець і дані в таблиці не відповідатимуть початковим.

Сортувати можна і частину списку, попередньо виділивши.

Приклад 1.

Є список учасників, наприклад, ІІІ етапу олімпіади з інформатики. Особливість цієї олімпіади така, що завдання для учнів 9–11 класів, для яких існує ця олімпіада, однакові. Список учасників містить такі поля: ПІБ, Клас, Область, Населений пункт, Освітній заклад, Бали за кожну задачу і Підсумковий бал.

У списку прізвища учасників розташовані так, як дітей розсаджували у приміщеннях. Члени журі впорядкують список за класами, а потім за балами від вищих до нижчих. Одразу стає видно, скільки учасників від кожної паралелі і скільки буде в них переможців (50 %). Після цього до списку додадуть поле Диплом. А представники адміністрації області впорядкують список за областями, за населеним пунктом, і дізнаються, як виступили їхні учасники.

Приклад 2. Повернемося до прикладу 1: представники адміністрації району переглядатимуть дані учасників олімпіади тільки свого району. Решту даних приховають, застосувавши фільтр.

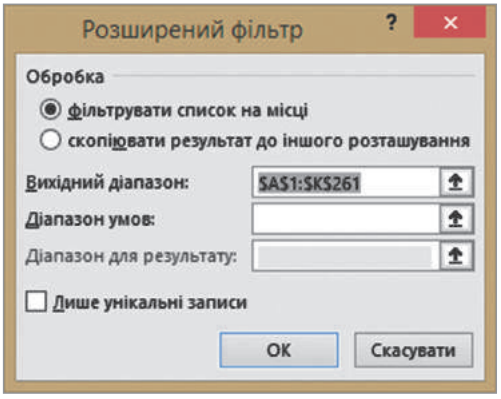


Рис. 7.23. Вікно налагодження розширеного фільтру

Фільтрування списків

Під фільтруванням списку розуміють процес, у результаті якого для перегляду доступні записи, які задовольняють критеріям фільтрування. Решта записів приховані.

Існує простий фільтр і розширений. У Excel команди фільтрування містяться на стрічці Дані у групі Сортування і фільтр.

Простий фільтр викликається командою Фільтр. Після цього біля кожної назви поля встановлюється розкривний список, який містить значення цього поля. Скасуванням або проставлянням прапорця біля значення можна задати умови фільтрування — вибрати значення для перегляду.

За допомогою простого фільтру (команда Фільтр) фільтрують тільки за умовою для одного поля. Відфільтрований список теж можна фільтрувати. Так створюється ієрархічне фільтрування. Для знаття режиму фільтрування і перегляду всього списку слід повторно скористатися командою Фільтр або командою Очистити.

Розширений фільтр дозволяє виконувати умови фільтрування будь-якої складності: створювати критерії з умовами за декількома полями і декількома умовами, обчислювані критерії; переміщувати копію отриманої в результаті фільтрування вибірки в інше місце робочої книги.

Щоб викликати розширений фільтр, потрібно виконати команду ДАНІ → група Сортування і фільтр → Додатково. Відкриється вікно введення діапазонів списку та умов фільтрування (рис. 7.23). Для відфільтрованих значень можна вибрати команду сортування.

Видно, що умови (критерії фільтрування) розміщують в діапазоні, тобто спочатку підготовлюють клітинки з умовами фільтрування, а надалі викликають розширений фільтр.

Під час створення діапазону критеріїв слід пам'ятати, що:

- діапазон умов складається не менше ніж з двох рядків (перший рядок — заголовки, які доцільно копіювати із заголовків стовпців списку, наступні — відповідні критерії);
- якщо умови розташовуються в одному рядку, це означає одночасність їх виконання (тобто вважається, що між ними поставлена логічна операція І);
- для істинності критерію, що складається з умов, які розташовано в різних рядках, має виконуватися хоча б один із них (тобто вважається, що вони з'єднані логічною операцією АБО);
- інтервал критеріїв має розташовуватися вище або нижче списку, або в іншому місці, або на іншому аркуші;
- від списку критерії мають відокремлюватися хоч б одним порожнім рядком, якщо їх розміщено вище або нижче списку;
- в інтервалі критеріїв не повинно бути порожніх рядків.

У вікні Розширений список в поле Вихідний діапазон вводимо діапазон розміщення даних таблиці, а в Діапазон умов — C1:D3, налаштовуємо перегляд відфільтрованого списку.

Приклад 3. На рис. 7.24 у списку виберемо команду для умов фільтрування.

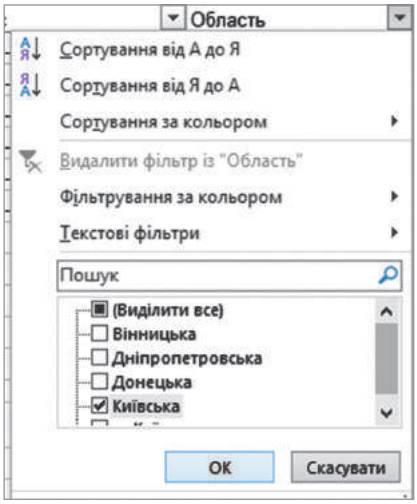


Рис. 7.24. Приклад списку фільтрування

Приклад 4.

Звернемося до прикладу 1. Наприклад, потрібно переглянути, як виступають учні 9 класу Полтавської області і учасники з Львівської області.

На рис. 7.25 наведено фрагмент таблиці. Над списком додано кілька рядків, в які продублювали заголовки полів і введено критерії.

	B	C	D	E	K
	Прізвище, ім'я, по батькові	Клас	Область	Місто	Місце
		9	Полтавська		
			Львівська		
	Прізвище, ім'я, по батькові	Клас	Область	Місто	Місце

Рис. 7.25. Приклад введення умов для аналізу результатів

Запитання для перевірки знань

1. Які таблиці в MS Excel називають списком?
2. Як впорядкувати данні за кількома параметрами?
3. Який результат фільтрування даних?
4. Яка різниця між простим і розширеним фільтром? Наведіть приклади.
5. На що варто звертати увагу під час сортування списку за значеннями в одному стовпці?
6. Чи можна розширений фільтр замінити багаторазовим використанням простого фільтру? Відповідь обґрунтуйте прикладами.

Завдання для самостійного виконання

1. Знайдіть у пошуковій системі результати IV етапу Всеукраїнської учнівської олімпіади, предмет і рік виберіть на свій розсуд.
2. Впорядкуйте таблицю за паралеллю класів (за збільшенням), балами (за зменшенням).
3. Додайте фільтр, залиште область або клас.

Практична робота № 15

ТЕМА. Упорядкування даних у таблицях. Автоматичні та розширені фільтри

ЗАВДАННЯ: відпрацювати команди сортування та фільтрування на прикладі таблиці Населення України.

ОБЛАДНАННЯ: комп'ютер, табличний процесор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте файл Населення України (результат практичних робіт № 12, 13).
2. Виділіть у таблиці стовпець із кількістю населення областей (G3:G29). Застосуйте до таблиці сортування за збільшенням населення областей: Дані → кнопка Сортування → активізуйте команду Автоматично розширити… → ОК.
3. У таблиці заголовки стовпців розташовано у два рядки. Відредагуйте так, щоб вони містилися в одному рядку.
4. Виділіть всю таблицю (верхня межа виділення — назви стовпців). Застосуйте до таблиці сортування за алфавітним порядком назв областей: Основне → кнопка Сортувати й фільтрувати → Настроюване сортування → у вікні виберіть в полі Сортувати за Область, в наступному — За значеннями, в наступному — Від А до Я. Підтвердіть ОК.
5. Відсортуйте таблицю за кольором стовпця з кількістю населення (G3:G29). Алгоритм, як у п. 4, лише змінюється стовпець сортування і умова буде За кольором.
6. Виділіть таблицю. Виконайте команду Основне → кнопка Сортувати й фільтрувати → Фільтр.
7. Перейдіть до стовпця А і в списку фільтрування виберіть Текстові фільтри → Починається з → уведіть К.
8. До відфільтрованої таблиці додайте фільтр за стовпцем Площа і в списку виберіть значення менше 25 000.
9. Відмініть фільтрування: Дані → Фільтр.
10. Додайте угорі таблиці чотири порожніх рядки. В перший рядок скопіюйте назви стовпців.
11. Відфільтруйте таблицю за такою умовою: показати області з кількістю населення більше ніж 1 млн і густотою більше від 100 або менше за 50. Для цього введіть такі умови фільтрування, як наведено в таблиці.
12. Виділіть таблицю і запустіть команду Дані — в групі Сортування й фільтр виберіть Додатково. У вікні, що розкриється, в полі Вихідний діапазон буде показано виділений діапазон — вся таблиця. Перемістіть вказівник в поле діапазона умов і виділіть умови разом із назвами (див. таблицю). Підтвердіть дію кнопкою ОК.

Зробіть висновки щодо можливостей Excel для створення умов перегляду вибіркових записів таблиці.

2. Зверніть увагу на результат.

3. Перенесення на другий рядок в межах клітинки здійснюється за допомогою сполучення клавіш ALT + ENTER.

4. Зверніть увагу на результат.

6. Біля кожної назви стовпців з'явилися кнопки списку умов фільтрування.

7. З'ясуйте, скільки областей показано.

8. Чи змінилася таблиця?

11.

Площа	Густота	Населення
	>100	>1000000
	<50	>1000001

7.8. Проміжні підсумки. Створення форм уведення-виведення

Пригадайте режим введення даних у таблицю з використанням прогресії. Які стандартні функції ви знаєте?

Розглянемо таблицю зі значеннями температури кожного дня всіх літніх місяців (приклад 1). Як бачимо, завдання не складне — достатньо скористатись стандартною функцією AVERAGE. Але коли здійснюється опрацювання великої кількості значень, вручну виконувати такі підрахунки не зручно.

У Excel передбачено розв'язання проблеми — на вкладці Дані у групі Структура є команда Проміжні підсумки.

Проміжні підсумки — команда автоматичного обчислення проміжних та загальних підсумків у списку для вказаних стовпців.

Підсумками є результат обчислення суми, середнього, максимального, мінімального з ряду значень, кількості певних значень або повтору одного.

Приклад 1.

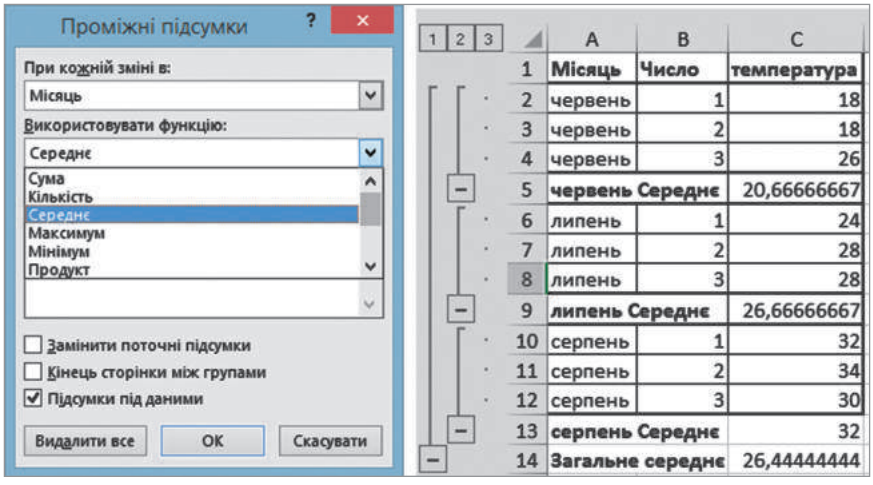
На рис. 7.26 подано дані тільки за три дні кожного місяця. Потрібно порахувати середнє значення температури за все літо та за кожний місяць окремо.

Місяць	Число	Температура
червень	1	18
червень	2	18
червень	3	26
липень	1	24
липень	2	28
липень	3	28
серпень	1	32
серпень	2	34
серпень	3	30

Рис. 7.26. Приклад таблиці для підрахунку середньої температури

Приклад 2.

Результат використання команди Проміжні підсумки для розглянутого прикладу 1 наведено на рис. 7.27.



	A	B	C
1	Місяць	Число	температура
2	червень	1	18
3	червень	2	18
4	червень	3	26
5	червень Середнє		20,66666667
6	липень	1	24
7	липень	2	28
8	липень	3	28
9	липень Середнє		26,66666667
10	серпень	1	32
11	серпень	2	34
12	серпень	3	30
13	серпень Середнє		32
14	Загальне середнє		26,44444444

Рис. 7.27. Приклад використання проміжних підсумків

Далі ознайомимося з правилами використання команди Проміжні підсумки.

- Таблицю потрібно оформляти як список — кожний стовпець має заголовок.
- Усі дані подавати у стовпцях одного типу.
- Таблиця має містити стовпець з однаковими значеннями — основи для обчислення підсумків.
- У таблиці відсутні порожні рядки та стовпці.
- Використання команди видаляє функції, залишає форматування таблиці.
- Загальний підсумок обчислюється на основі даних таблиці, а не через проміжні обчислення.

Для розв'язування задач у середовищі Excel у таблицю вводять початкові дані. Це можна здійснити вручну або за допомогою використання ряду низки значень прогресії, якщо це дозволяють необхідні дані.

Для того щоб виконати алгоритм застосування команди Проміжні підсумки, потрібно опрацювати таблицю для виконання правил використання команди. Слід виділити таблицю та виконати Дані → Структура → Проміжні підсумки. Відкриється вікно налагодження параметрів функції (див. рис. 7.27). Потім треба вибрати стовпець зі списку При кожній зміні в. (мається на увазі зміна значення у стовпці) і необхідну функцію з списку Використовувати функцію, потім вибрати Режим обчислення і підтвердити дію кнопкою ОК

Excel також надає можливість зручного введення даних за допомогою форм. Щоб скористатись цією можливістю потрібно додати на панелі швидкого доступу піктограму форми (рис. 7.28).

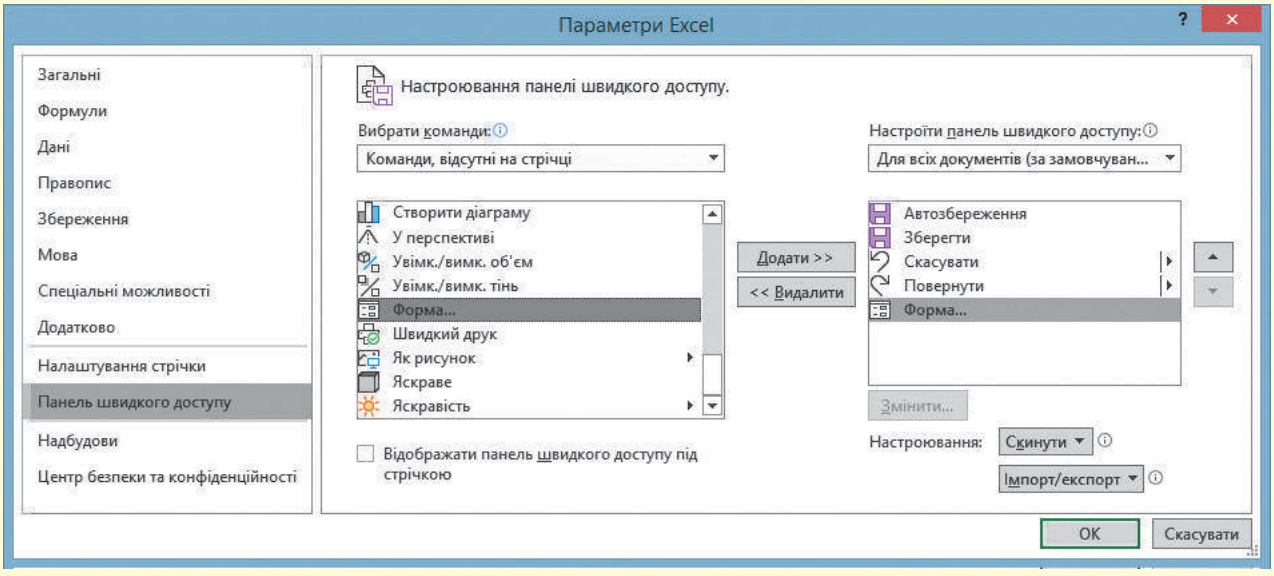


Рис. 7.28. Вікно налагодження панелі швидкого доступу

Для введення даних за допомогою форм потрібно перейти до Файл → Параметри, у вікні відкрити вкладку Панель швидкого доступу та у списку вибору команд вибрати Команди, відсутні на стрічці. Далі у переліку команд вибрати Форма, натиснути кнопку Додати (команда з'явиться у правому полі), і ОК — кнопка форми з'явилась на панелі швидкого доступу.

Як саме використовувати форму для додавання значень також розглянемо на прикладі погоди літніх місяців.

Приклад 3.

Додамо до таблиці стовпці Вітер для введення його потужності та Опади для введення значень, наприклад: відсутні, короткочасний дощ, дощ, злива.

Порядок виконання

1. Створимо таблицю з заголовками стовпців Місяць, Число, Температура, Вітер, Опади.
2. Уведемо у стовпець Місяць перше значення — червень.
3. Виділимо два рядки (заголовки і значення) — рис. 2.29.
4. На панелі швидкого доступу мишею запустимо Форму — відкриється вікно форми для аркушу з виділенням (див. рис. 7.29).
5. У вікні всі заголовки стовпців і перший уже мають значення.
5. Введемо значення в решту стовпців першого рядку значень.
6. Запустимо команду Створити — значення з'являться в таблиці, а поля у формі будуть порожні для введення значень в наступному рядку.
7. Виконаємо п. 5 і 6, а коли таблиця буде заповнено, запустимо команду Закрити — вікно форми закриється.

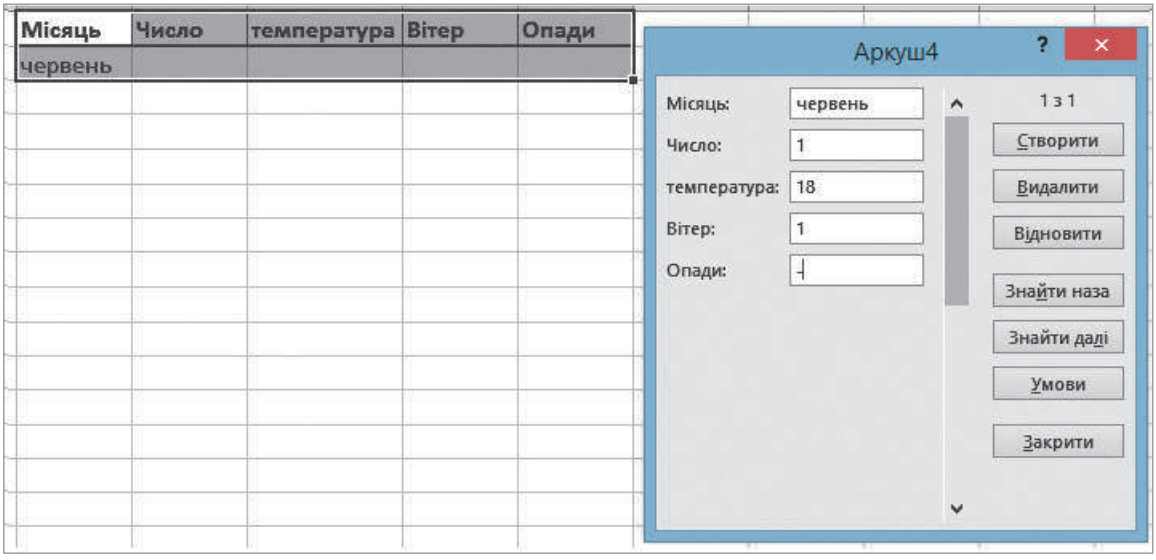


Рис. 7.29. Приклад уведення даних в таблицю за допомогою форми

Форма уведення є вікном діалогу, яке використовується для додавання, редагування, пошуку та видалення даних в рядках таблиці.

Форму зручно використовувати, якщо рядок в таблиці широкий і уведення та опрацювання даних в ньому увесь час потребує горизонтального прокручування. Для використання форми всі стовпці таблиці повинні мати заголовки.

Форми також використовують для виведення в ній значень того рядку, для якого було попередньо введено умови пошуку.

Запитання для перевірки знань

1. Поясніть, що таке проміжні підсумки?
2. Поясніть, навіщо використовувати форму для уведення даних в таблицю?
3. Наведіть приклади доцільного використання проміжних підсумків.
4. Що відбувається з результатами функцій в таблиці після використання команди **Проміжні підсумки**? Поясніть, чому.
6. Чи можна піктограму форми додати на стрічку?, Якщо так, який алгоритм?

Завдання для самостійного виконання

1. У табличному процесорі заповніть таблицю за зразком, наведеним на рис. 7.26.
2. Побудуйте гістограми різного типу для перегляду температури. Порівняйте їх вигляд і з'ясуйте, яка найкраще візуалізує дані.



Виконайте тестове завдання з автоматичною перевіркою результату

Тест 7

Практична робота № 16

ТЕМА. Проміжні підсумки. Форма уведення даних

ЗАВДАННЯ: заповніть таблицю погодних умов літніх місяців за допомогою форми уведення даних та використайте команду **Проміжні підсумки** для аналізу даних.

ОБЛАДНАННЯ: комп'ютер, табличний процесор.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відкрийте табличний процесор. Заповніть перші два рядки за зразком (див. рис. 7.29).
2. Уведіть назву місяця — червень.
3. Виділіть у таблиці рядок із заголовками та рядок з одним значенням.
4. Установіть піктограму форми на панель швидкого доступу.
5. Відкрийте форму уведення даних, клацнувши піктограму форми.
6. Уведіть по 5 значень даних для кожного літнього місяця.
7. Закрийте форму.
8. Виділіть таблицю зі значеннями і заголовками стовпців.
9. Запустіть команду Проміжні підсумки зі вкладці Дані групи Структура.
10. У вікні, що відкрилося, виберіть стовпець Місяць, функцію Середнє для температури, підтвердіть кнопкою ОК.
11. Виділіть оновлену таблицю. Повторно виконайте команду Проміжні підсумки.
12. У вікні, що відкрилося, залиште стовпець Місяць, функцію Максимум для сили вітру, підтвердіть кнопкою ОК.

Зробіть висновки: які можливості для створення умов уведення даних в таблицю та автоматизації їх аналізу.

Розділ 8
ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

8. Основні поняття алгоритмізації

8.1. Поняття алгоритму. Властивості алгоритмів

Чим здавна користувалися люди для виконання обчислень? Наведіть приклади.

Користування різними інструкціями, таблицями, списками правил тощо часто дозволяє не вдаватися в сутність задачі, що розв'язується, тобто формалізувати обчислювальний процес. Списки формальних правил виконання обчислювальних робіт пізніше отримали назву алгоритми. Нині алгоритми широко застосовуються в науці, техніці, побуті (рис. 8.1).

З алгоритмами та основними поняттями алгоритмізації ви вже знайомі з курсів інформатики попередніх класів. Як відомо, існує багато означень алгоритму, які за сутністю мало відрізняються одне від одного. Далі ми користуватимемося наведеним означенням алгоритму.

Алгоритм — це скінченна послідовність команд (інструкцій, вказівок), виконання яких у визначеному порядку приводить до розв'язування певного завдання.

Команда — це конкретна інструкція (вказівка), що визначає, яку і як виконувати дію (операцію). Наприклад, додати два числа, зупинитися на світлофорі на червоне світло, натиснути відповідну кнопку в ліфті, вивести на екран монітора зображення піраміди.

Алгоритми завжди розробляють для певного виконавця.

Виконавцем алгоритму називають живу істоту чи пристрій, що його виконує.

Кожен виконавець володіє набором команд, які він може виконувати. Це означає, що не кожний алгоритм може бути виконаний будь-яким виконавцем. Так, алгоритм розв'язування квадратного рівняння, доступний учням 8 класу, учень 2 класу виконати не може.

Сукупність усіх команд, які може виконувати виконавець, називають системою команд виконавця.

Нині алгоритми найчастіше розробляють для їх реалізації на комп'ютері з необхідним програмним забезпеченням.

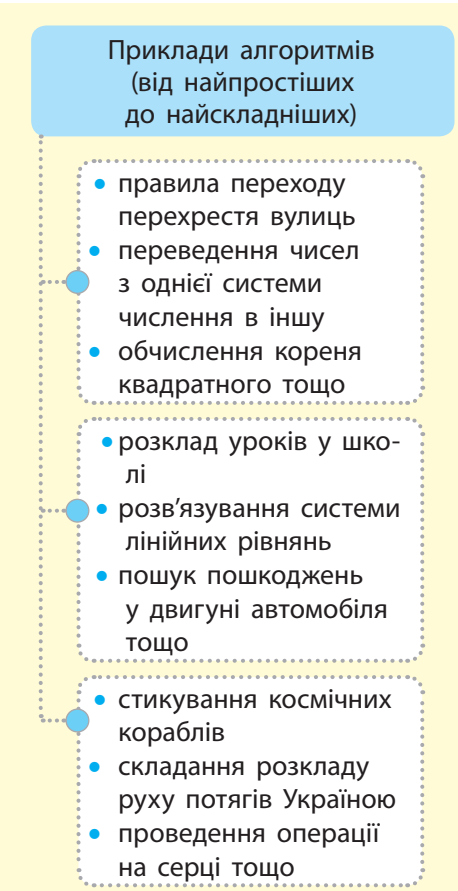


Рис. 8.1. Приклади застосування алгоритмів

Щоб алгоритм виконав своє призначення, він має відповідати певним вимогам (властивостям). Будь-який алгоритм має сукупність властивостей. Пригадаємо основні з них.

Властивість	Опис
Визначеність	В алгоритмі використовуються лише команди із системи команд виконавця. Команди повинні бути чітко сформульовані й не мати подвійного тлумачення, щоб виконавець алгоритму розумів їх однозначно
Дискретність	Усі команди мають виконуватися покроково. Перехід до чергової команди може відбутися тільки після завершення попередньої
Результативність	Виконання алгоритму має завершитися за скінченну кількість кроків отриманням кінцевого результату, за умови що вхідні дані належать області допустимих значень. Якщо вхідні дані виходять за область допустимих значень, то алгоритм може не завершитися виконанням або видати неправильний результат. У разі повторного виконання алгоритму для одних і тих самих вхідних даних послідовність виконання команд і отриманий результат мають бути однаковими
Масовість	Алгоритм призначений для розв'язування не однієї конкретної задачі, а певного класу однотипних задач. Вхідні дані в однотипних задачах можуть бути різними, проте вони не повинні виходити за межі допустимого діапазону
Формальність	Різні виконавці алгоритмів мають отримувати одні й ті самі кінцеві результати. Будь-яка команда, виконана багато разів одним або різними виконавцями для тих самих вхідних даних, завжди має видавати однаковий результат. Після виконання кожної команди виконавець алгоритму має знати, яку команду слід виконувати наступною

Зазвичай команди алгоритму виконуються послідовно. Якщо вони мають виконуватися непослідовно, то це зазначається в спеціальній команді окремо.

Алгоритм обчислення НСД двох натуральних чисел A і B (алгоритм Евкліда)

1. Порівняйте числа A і B. Якщо $A = B$, то перейдіть до п. 7, інакше — виконайте п. 2.
2. Якщо $A > B$, то перейдіть до п. 5, інакше — виконайте інструкцію п. 3.
3. Зменште значення B на значення A.
4. Перейдіть до п. 1.
5. Зменште значення A на значення B.
6. Перейдіть до п. 1.
7. Присвойте найбільший загальний дільник, який має значення A, змінній D.
8. Виведіть значення D.
9. Кінець.

Виконаємо наведений алгоритм (приклад 1). Ступінь деталізації алгоритму може бути різним, наприклад, скороченим, якщо визначається лише сутність розв'язування задачі, або розгорнутим, якщо детально описується обчислювальний чи інший процес. Деталізація алгоритму залежить від того, на якого виконавця розраховано алгоритм (приклад 2). Слід ураховувати, що кожний виконавець може виконувати чітко

Приклад 1.
Якщо $A = 119$, а $B = 51$, то $D = 17$.

Приклад 2.
Для дітей, які знають таблицю множення, сам запис $A*B$ однозначно визначає послідовність дій для отримання правильного результату. Для дітей, які знають правила додавання чисел, але не знають таблиці множення, множення двох натуральних чисел A і B можна описати у вигляді послідовності інструкцій операцій додавання.

визначену систему команд (приклад 3). Отже, кожний виконавець має власну систему команд, яку він спроможний виконати. Спроба виконати алгоритм іншим виконавцем, який має іншу систему команд, може завершитися безрезультатно.

Формальне виконання алгоритму означає, що під час виконання алгоритму виконавець не вдається в сутність задачі. Наприклад, якщо виконавець виконує команду множення числа a на число b, то не замислюється над тим, є це множенням двох дійсних чисел або обчислюється площа прямокутника. Тобто виконавець повністю абстрагується від змісту завдання, не вдається в сутність виконуваних команд і при цьому отримує правильний результат.

Приклад 3.
Один робот може виконувати команди переміщення прямо, наліво та направо на задану кількість кроків, а інший робот — ще й переміщуватися назад та виконувати стрибки.

Запитання для перевірки знань

1. Що називають алгоритмом?
2. Наведіть приклади найпростіших алгоритмів.
3. У якому порядку виконуються інструкції алгоритму?
4. Від чого залежить ступінь деталізації алгоритму?
5. Назвіть і поясніть основні властивості алгоритму.
6. Від чого залежить деталізація алгоритму?

Завдання для самостійного виконання

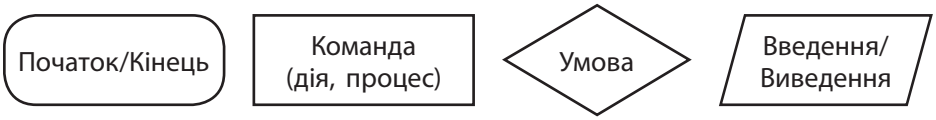
1. Опишіть алгоритм вашого шляху до школи.
2. Опишіть алгоритм визначення центра рівнобічного трикутника за допомогою циркуля та лінійки.
3. Опишіть алгоритм побудови бісектриси кута за допомогою циркуля та лінійки.

8.2. Способи подання алгоритмів

Які способи подання алгоритмів ви знаєте?

Існують різні способи подання алгоритмів — залежно від того, на якого виконавця орієнтовано алгоритм. Пригадаємо основні з них.

Алгоритм	Опис
Словесний	Описує алгоритм природною мовою, широко застосовується у повсякденному житті (наприклад, у вигляді інструкцій з експлуатації приладів, рецептів виготовлення ліків тощо). Інструкція складається з указівок, форма запису яких довільна. Головне, щоб вони були точними й зрозумілими всім користувачам. Цей спосіб є досить простим і доступним, проте опис алгоритмів часто є досить громіздким, а команди можуть сприйматися виконавцями неоднозначно

Алгоритм	Опис
Формульний (словесно-формульний)	Використовує природну мову, а також спеціальні символи (наприклад, хімічні формули, математичні вирази та ін.), застосовувані в певній науковій галузі. В алгоритмах часто використовують оператор присвоювання, який позначається двома крапками зі знаком дорівнює (:=). Зазвичай його записують у такій формі: <змінна> := <вираз> Під час виконання оператора спочатку обчислюється вираз, значення якого присвоюється змінній. Розв'яжемо, наприклад, квадратне рівняння $ax^2+bx+c=0$. 1. Уведемо значення змінних a, b, c. 2. Обчислимо дискримінант $d=b^2-4ac$. 3. Якщо $d<0$, то виконаємо п. 7, інакше — п. 4. 4. Знайдемо корені рівняння: $x_1:=\frac{-b-\sqrt{d}}{2a}$; $x_2:=\frac{-b+\sqrt{d}}{2a}$. 5. Виведемо значення x_1, x_2. 6. Перейдемо до п. 8. 7. Виведеться повідомлення **Дискримінант від'ємний. Розв'язання немає.** 8. Кінець.
Графічний	Передбачає подання алгоритму у вигляді геометричних фігур (блоків), з'єднаних стрілками (лініями зв'язку). Така організована послідовність блоків називається **блок-схемами**. Блок-схеми мають високу наочність. На рис. 8.2 зображено основні блоки, за допомогою яких створюються блок-схеми алгоритмів.  Рис. 8.2. Основні блоки блок-схеми

Розглянемо на прикладах подання алгоритму у вигляді блок-схеми.

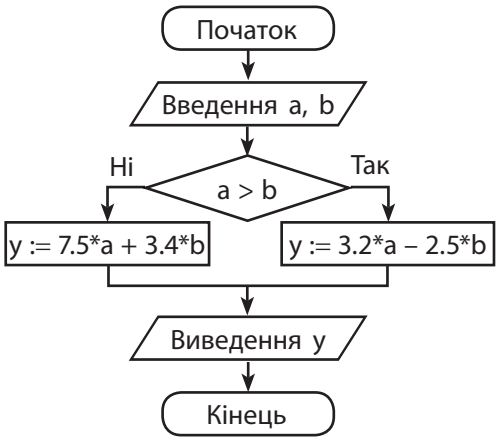


Рис. 8.3. Блок-схема алгоритму обчислення виразу

Приклад.

На рис. 8.3 зображено блок-схему алгоритму обчислення значення y:

$$\begin{cases} 3{,}2a-2{,}5b, \text{ якщо } a>b, \\ 7{,}5a+3{,}4b, \text{ якщо } a\le b. \end{cases}$$

У цьому прикладі значення виразу залежить від значень змінних a і b.

Якщо $a>b$ (гілка Так), обчислюється значення виразу $3{,}2\cdot a-2{,}5\cdot b$, інакше (гілка Ні) — значення виразу $7{,}5\cdot a+3{,}4\cdot b$.

Запитання для перевірки знань

1. Поясніть сутність словесно-формульного способу подання алгоритму.
2. Які переваги та недоліки має словесно-формульний спосіб подання алгоритмів?
3. Які основні графічні елементи використовуються в блок-схемах?
4. Які переваги та недоліки має графічний спосіб подання алгоритмів?

Завдання для самостійного виконання

1. Розробіть блок-схему алгоритму обчислення виразу:

$$y=\begin{cases}5,6x-b, \text{ якщо } x>0;\\ 2x, \text{ якщо } x\le 0.\end{cases}$$

2. Розробіть блок-схему алгоритму обчислення виразу:

$$y=\begin{cases}\left(a^3+b^2\right)\cdot c-1, \text{ якщо } a=0;\\ a^3+b^2, \text{ якщо } a\ne 0.\end{cases}$$

3. Дано три числа: a, b, c. Розробіть блок-схему алгоритму визначення найбільшого з них.

8.3. Базові алгоритмічні структури. Типи алгоритмів

Пригадайте, які є типи алгоритмів. Яка функція в табличному процесорі реалізує алгоритм розгалуження? Які вона має складові?

У процесі роботи з табличним процесором ви вже познайомились із фукціями табличного процесора для реалізації різного типу алгоритмічних структур. Розглянемо їх на прикладах.

В алгоритмах із лінійною структурою команди виконуються послідовно одна за одною (приклад 1).

Існують три базові алгоритмічні структури: слідування (лінійна), розгалуження, повторення (цикл). На їх основі можна побудувати алгоритми будь-якої складності.

Приклад 1.

На рис. 8.4 наведено приклад блок-схеми лінійного алгоритму обчислення значення виразу $(2\cdot a+b)\cdot(a-b)$.

Здійснюємо уведення значень змінних a і b, значення a множимо на 2, до нього додаємо значення b. Отриманий результат присвоюється змінній z, що є проміжним результатом. У наступному блоці він множиться на значення $a-b$ і присвоюється змінній y.

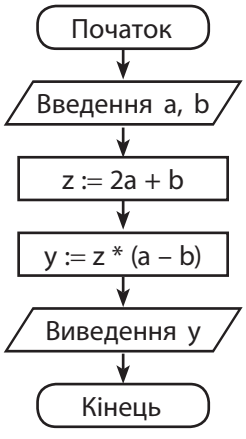


Рис. 8.4. Блок-схема лінійного алгоритму

В алгоритмах із розгалуженою структурою порядок виконання інструкцій (команд мови програмування) залежить від результату обчислення умовного виразу. Існує три види розгалужень: неповне, повне, за вибором. Блок-схеми перших двох показані у прикладі 2 (рис. 8.5).

Приклад 2. На рис. 8.4 наведено блок-схеми повного (а) і неповного (б) розгалужень.

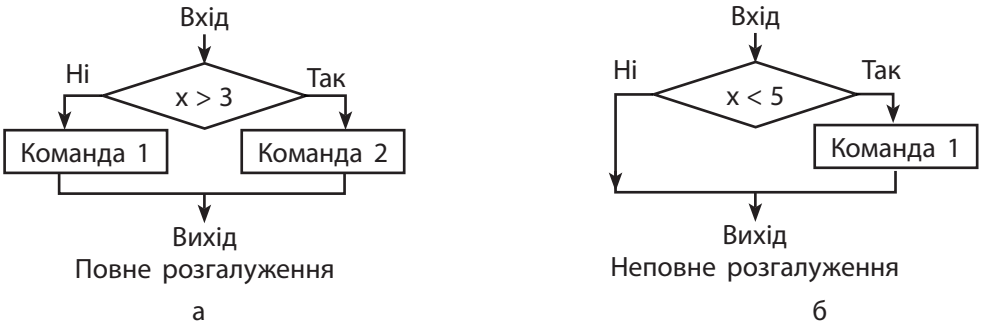


Рис. 8.5. Блок-схеми повного (а) і неповного (б) розгалуження

Приклад 3. Побудуємо блок-схему для обчислення y після введення значень x і a:

$$y = \begin{cases} ax, \text{ якщо } x > 0 \text{ і } a \geq 0; \\ a \cdot \sin x, \text{ якщо } x > 0 \text{ і } a < 0; \\ 0, \text{ якщо } x \leq 0. \end{cases}$$

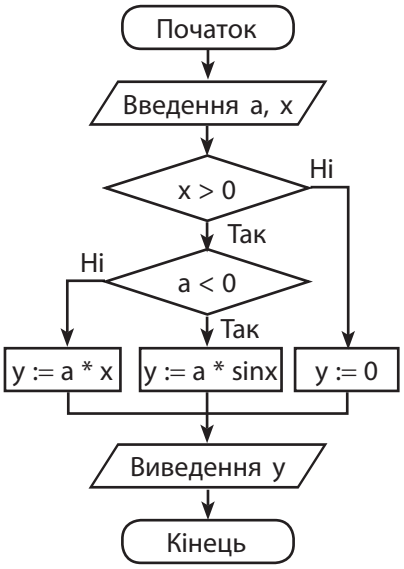


Рис. 8.6. Блок-схема алгоритму для обчислення y

У блок-схемах повного розгалуження залежно від результату обчислення умовного виразу виконується блок команд гілки Так або гілки Ні. Якщо значення x більше за 3, результатом умовного виразу є істина і виконується Команда 2, інакше (x менше або дорівнює 3) результатом є хибність і виконується Команда 1. Надалі виконуються інструкції, записані після оператора розгалуження.

У блок-схемах неповного розгалуження гілка для результату обчислення умовного виразу порожня — вона не містить жодних команд. Для випадку, коли $x < 5$, виконується Команда 1. А при x більше або дорівнює 5 одразу виконання переходить до команд, записаних після розгалуження.

У кожній гілці розгалужень можуть міститися інші розгалуження, що дозволяє реалізувати складені умови (приклад 3).

В алгоритмах із повторенням, або з циклічною структурою, одні й ті самі інструкції (команди) виконуються багаторазово для різних значень одних і тих самих змінних. На рис. 8.7 зображено блок-схему алгоритму із циклічною структурою (приклад 4).

В алгоритмах із повторенням можна виділити три частини: підготовчу, робочу і керівну.

Підготовча частина містить команди, що визначають початкові значення змінних (у нашому алгоритмі початкові значення — $i = 1$, $y = 1$ та введене значення x), робоча частина, або тіло циклу, — команди основних обчислень (тілом циклу в алгоритмі є команди $y := y * x$ та $i := i + 1$).

У керівній частині здійснюється перевірка умови ($i <= n$).

Повторювані під час виконання циклу дії називають тілом циклу, а кожне виконання тіла циклу — ітерацією.

Змінну i називають лічильником циклів, її значення визначає, скільки разів було виконано тіло циклу.

Приклад 4. На рис. 8.7 зображено блок-схему алгоритму із циклічною структурою для обчислення виразу $y = x^n$, де x — дійсне число, n — ціле.

Сутність алгоритму полягає в тому, що попереднє значення y послідовно множиться на x n разів. Так, якщо $x = 4$ і $n = 3$, то обчислення виконуватиметься так:

$$y = 1 * 4 = 4;$$

$$y = 4 * 4 = 16;$$

$$y = 16 * 4 = 64.$$

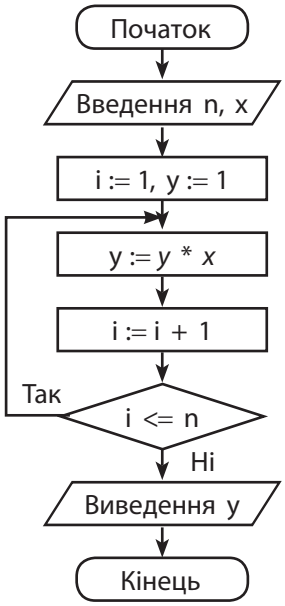


Рис. 8.7. Блок-схема алгоритму з повторенням (із циклічною структурою)

Запитання для перевірки знань

1. Назвіть базові структури алгоритмів.
2. Поясніть, для чого використовують лічильник циклів.
3. Що таке тіло циклу?
4. З яких частин складаються алгоритми з повторенням?

Завдання для самостійного виконання

1. Розробіть алгоритм обчислення $y = 6a^3 - 7$ у словесно-формульному поданні.
2. Розробіть блок-схему алгоритму обчислення y:

$$y = \begin{cases} 8{,}1a + 4a^2, \text{ якщо } a < 3; \\ 9{,}75 - 0{,}125a, \text{ якщо } a \geq 3. \end{cases}$$

3. Дано послідовність n натуральних чисел. Розробіть блок-схему алгоритму обчислення її суми.
4. Складіть блок-схему алгоритму обчислення s, де $s = x + x^2 + x^3 + \ldots + x^n$.

Практична робота № 17

ТЕМА. Способи подання алгоритмів. Базові алгоритмічні структури. Типи алгоритмів

ЗАВДАННЯ: розробити алгоритми на основі базових алгоритмічних структур у словесно-формульній та графічній формах.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Розробіть у словесно-формульній формі алгоритм свого шляху до школи.
2. Розробіть алгоритм у словесно-формульної формі обчислення часу руху автобуса зі Львова до Києва із середньою швидкістю v км/год з двома зупинками по 30 хв.
3. Розробіть у словесно-формульній формі алгоритм обчислення y:

$$y = 6{,}125a + \left(b^2 - 3c\right).$$

4. Розробіть алгоритм обчислення y:

$$y = \begin{cases} 5{,}4a + 2{,}2b, & \text{якщо } a > b; \\ 17{,}5ab - 3{,}1b, & \text{якщо } a \le b. \end{cases}$$

5. Розробіть у словесно-формульній формі алгоритм обчислення y:

$$y = \begin{cases} \sqrt{2a+b}, & \text{якщо } a > b; \\ 8{,}1a^2 - 5{,}7b, & \text{якщо } a \le b. \end{cases}$$

6. Дано послідовність чисел: 3, 5, 7, 9, 11, 13. Розробіть алгоритм обчислення суми цих чисел.
7. Розробіть блок-схему алгоритму обчислення суми n членів спадної геометричної прогресії із заданими значеннями її першого члена і знаменника.

Зробіть висновки про переваги та недоліки різних способів подання алгоритмів.

4. Виберіть спосіб подання алгоритму самостійно.

6. Подайте алгоритм у словесно-формульній формі.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 8

9. Мови програмування

9.1. Етапи розв'язування задач із використанням комп'ютера

Поміркуйте, чи є розроблення алгоритмів і програм та їх подальша реалізація на комп'ютері творчим процесом.

Якщо задачу подано у вигляді математичних формул, то вона вже є формалізованою.

У кожного розробника є свої стиль і методи роботи. Існують і загальні принципи, яких варто дотримуватися. Розв'язування задач за допомогою комп'ютера відбувається у два етапи.

Етап. 1 Підготовка задач для розв'язання

1. З'ясування сутності задачі. Розробник має чітко зрозуміти умову, перелік вхідних даних, кінцеві результати.
2. Формалізація задачі. Формалізувати задачу — означає побудувати її математичну модель і описати формулами.
3. Вибір методу розв'язування задачі. Метод має забезпечити розв'язування задачі з тим, щоб час її реалізації був якомога меншим, а обсяг задіяної пам'яті мінімальним.
4. Розробка алгоритму. Головне — забезпечити правильність, наочність і зрозумілість алгоритму. Алгоритм може розроблятися у спрощеному чи детальному вигляді. Для складних задач часто розробляються обидва.
5. Розробка програми. Розробити програму — означає описати алгоритм мовою програмування. Запис алгоритму здійснюється точно і формалізовано, це дозволяє автоматизувати деякі процеси її реалізації на комп'ютері.

Етап 2. Виконання алгоритму на комп'ютері

1. Введення програми в комп'ютер. На персональному комп'ютері програма вводиться за допомогою текстового редактора, вбудованого в середовище програмування. Вона відображається на екрані монітора, тому помилки введення можна відразу виправляти.
2. Транслювання програми. Транслювання — це процес перетворення програми у двійковий код з одночасним пошуком помилок у програмі, які називаються синтаксичними. Є два види транслювання програми: компіляція і інтерпретація. У процесі компілювання вся програма одночасно перетворюється на машинний код. У процесі інтерпретування кожна команда перетворюється у двійковий код і відразу виконується.
3. Налагодження програми. У процесі налагодження програм слід виявити й усунути помилки в програмі. Синтаксичні помилки, що порушують конструкції та правила мови програмування, виявляють транслятори. Вони виникають внаслідок використання недозволених знаків операцій, наявності зайвих дужок тощо (приклад 1).
4. Тестування програми. Тестування — це процес багаторазового виконання програми для різних даних. Воно здійснюється з метою виявлення діапазону даних, для яких можна отримати неправильні результати, і оцінювання ефективності програми (приклад 2).

Процес перетворення програми у двійковий код виконується за допомогою спеціальних програм, які називають трансляторами.

Компілятори функціонують швидше від інтерпретаторів, але помилки видаються після завершення транслювання програми. Інтерпретатор видає повідомлення після перетворення кожної команди, що досить зручно для програміста.

Приклад 1. У записі $y = 2 - (a/b + c)$ синтаксична помилка полягає у тому, що є зайва дужка. Інші помилки в програмі, які називають логічними, транслятор не виявляє.

Приклад 2. Якщо замість $y = a - b$ записано $y = a + b$, то таку помилку транслятор не виявляє. Її має виявити сам програміст.

Запитання для перевірки знань

1. Назвіть основні етапи розв'язування завдань із використанням комп'ютера.
2. Які дії виконує транслятор?
3. Які помилки виявляє транслятор?
4. Наведіть приклад синтаксичної помилки в програмі.
5. Для чого тестують програми?
6. Які переваги й недоліки компіляторів?

9.2. Мова програмування як один зі способів описання алгоритмів. Класифікація мов програмування

З якими мовами програмування ви ознайомилися в попередніх класах?

Вважають, що першим пристроєм із програмним керуванням був ткацький верстат, побудований Жозефом Марі Жаккаром у 1804 році. Верстат здійснив революцію в ткацькій промисловості: Жаккар віднайшов можливість за допомогою перфокарт програмувати візерунки на тканинах.

Початок розвитку історії мов програмування припадає на XIX ст., коли англійський учений Чарльз Беббідж розробив механічну обчислювальну машину, програму для якої написала Ада Лавлейс. Мови програмування в сучасному розумінні почали розвиватися фактично з появою ЕОМ.

Мова програмування (англ. programming language) — це штучна мова, створена для розроблення програм, які призначено для виконання на комп'ютері.

Як і будь яка інша мова, мова програмування містить такі складові: алфавіт, синтаксис і семантику (рис. 9.1).

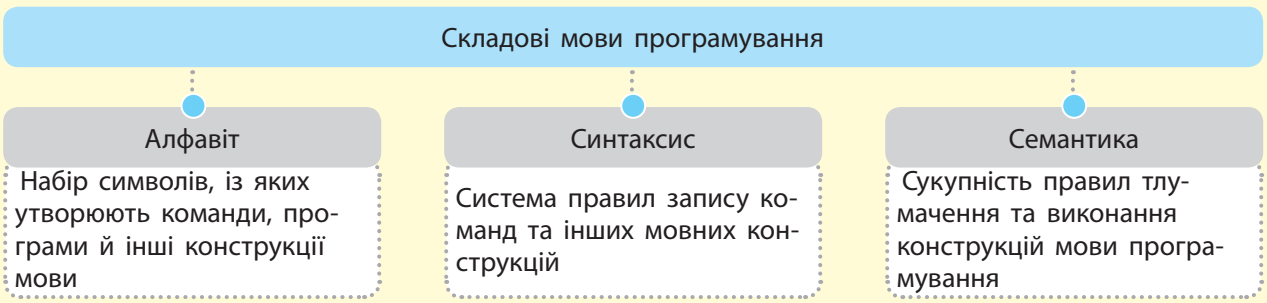


Рис. 9.1. Складові мови програмування

Нині розповсюдженою мовою низького рівня є Асемблер.

Однією з перших мов високого рівня, що набула широкого визнання серед програмістів світу, була мова Fortran.

Кожна мова програмування має власний словник — набір ключових (зарезервованих) слів, які мають строго визначене призначення і які не можна використовувати в інших цілях. Основні складові мови програмування забезпечують абсолютно точний запис алгоритму для його реалізації на комп'ютері.

Нині існують десятки тисяч різних мов програмування, які умовно класифікуються за кількома основними ознаками (рис. 9.2, с. 181).

- За ступенем залежності від апаратних засобів розрізняють мови низького, високого і надвисокого рівнів.

Мови програмування низького рівня (машинно-орієнтовані) — це мови, у яких команди та дані орієнтуються на архітектуру комп'ютера. Такі мови орієнтовані на конкретний тип комп'ютера і враховують його особливості. Практично кожний тип комп'ютера мав власну мову програмування низького рівня.

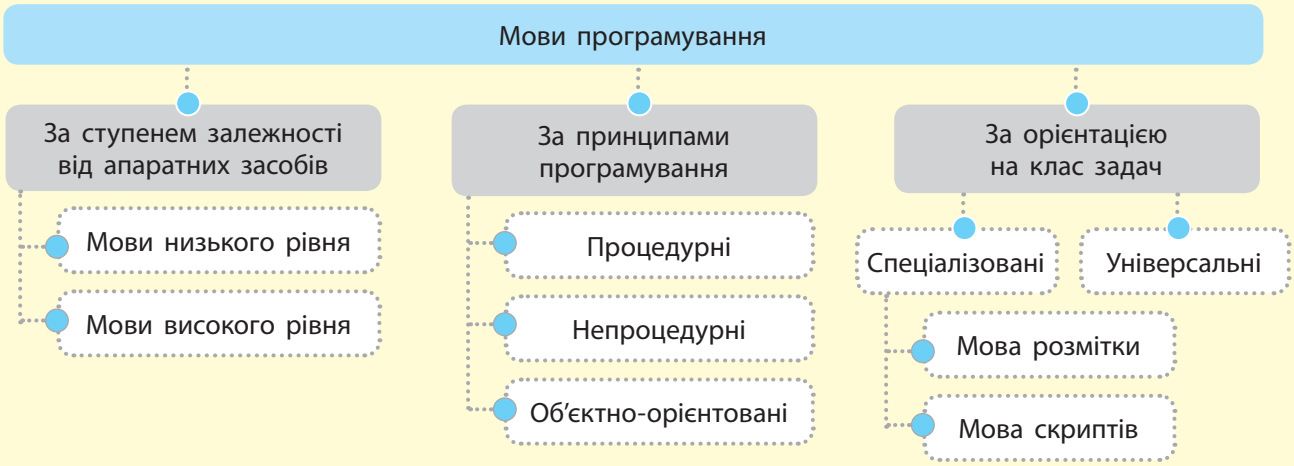


Рис. 9.2. Класифікація мов програмування

Мови програмування високого рівня (машинно-незалежні) — мови, на яких програми можуть використовуватися на комп'ютерах різних типів і які є більш зрозумілими людині, ніж мови низького рівня. У мовах високого рівня команди не залежать від апаратних особливостей.

Програми, написані цими мовами, можуть виконуватися на будь-якому комп'ютері.

Мови надвисокого рівня мають механізми абстрактного описування задачі та вбудовані засоби їх розв'язування. Такими мовами є, наприклад, Prolog, APL.

- За принципами програмування мови поділяються на процедурні, непроцедурні та мови об'єктно-орієнтованого програмування.

Процедурні мови дозволяють складати програми, у яких описується послідовне змінення стану комп'ютера, тобто значення комірок пам'яті, стану процесора й інших пристроїв. Вони маніпулюють даними в покроковому режимі, використовуючи послідовні інструкції. У процедурних мовах витримано чітку структуризацію програм, тому їх ще називають мовами структурного програмування.

До непроцедурних мов належать мови високого рівня абстракції. Вони є досить ефективними для розв'язування задач пошуку даних у великих масивах і завдань, процес розв'язування яких неможливо описати точно. Такими завданнями є, наприклад, переклад тексту з однієї мови на іншу, розпізнавання образів.

Мови об'єктно-орієнтованого програмування містять конструкції, що дозволяють створювати об'єкти, класи, методи. Нині ці мови є найбільш поширеними. Вони орієнтовані на складний клас задач і фактично витіснили процедурні мови програмування.

На різних етапах розвитку комп'ютерної техніки популярністю користувалисярізні мови програмування.



- До мов високого рівня належать: Pascal, Delphi, C#, C++, Java, Python.
- До процедурних мов належать Fortran, Algol, Pascal, Basic та ін.
- До об'єктно-орієнтованих мов належать Java, Python, C++ та ін.
- До універсальних мов належать Pascal, C++, C#, Python, Java.

- За орієнтацією на клас задач мови програмування поділяють на універсальні та спеціалізовані.

Універсальні мови орієнтовані на програмування різноманітних обчислювальних, логічних та інших задач.

Спеціалізовані мови враховують специфіку предметної галузі. Особливими класами спеціалізованих мов є мови скриптів і мови розмітки. Мови скриптів призначені для створення невеликих допоміжних програм. Мови розмітки містять шаблони й правила описання вмісту, структури та форми електронних документів. Наприклад HTML дозволяє виконувати розмітку гіпертекстового документа. Не всі з перелічених мов у класичному розумінні є мовами програмування.

Найпопулярнішою мовою скриптів нині є JavaScript, яка використовується для створення динамічних компонентів на вебсторінках.

HTML є мовою розмітки гіпертексту, її часто помилково називають мовою програмування.

Матеріал нашого підручника зорієнтовано на роботу з мовою програмування Python, яка підтримує об'єктно-орієнтований і процедурний методи програмування.

Наведемо її стислу характеристику.

- Мова Python підтримується всіма операційними системами і дозволяє розв'язувати складні математичні задачі, створювати графічні зображення, розробляти веб-сайти, працювати з реляційними базами даних.
- Мова Python має потужну стандартну бібліотеку, яку користувач може розширювати власними бібліотеками й бібліотеками інших користувачів. Наприклад, розширення .NumPy містить реалізацію різноманітних математичних обчислень, модуль tkinter дає змогу реалізувати графічний інтерфейс користувача.
- Програми можуть розроблятися в консольному режимі (такі програми мають розширення .py) і з графічним інтерфейсом (програми мають розширення .pyw).
- Python має алфавіт, який складається із символів, розташованих на клавіатурі, ключові (зарезервовані) слова, наприклад and, break, літерали (константи), оператори, вбудовані функції та інші елементи. Однакові ідентифікатори на різних регістрах сприймаються як різні імена. Так, Delta і delta є різними ідентифікаторами.
- Програму мовою Python можна створювати й редагувати за допомогою будь-якого редактора, наприклад Notepad++, Eclipse, PythonWin та ін.



Нідерландський програміст Гвідо ван Россум створив мову програмування і назвав її Python — на честь скетч-серіалу «Літаючий цирк Монті Пайтона» (англ. *Monty Python's Flying Circus*). Існують версії для Linux, Windows, MacOS.

Запитання для перевірки знань

1. Назвіть основні ознаки класифікації мов програмування.
2. Як мови поділяються за принципом програмування?
3. Назвіть найсучасніші універсальні мови програмування.
4. Які спеціалізовані мови програмування нині є найпоширенішими?
5. Чому процедурні мови були витіснені об'єктно-орієнтовананими мовами?
6. Які переваги та недоліки мають мови низького і високого рівнів?

9.3. Середовище програмування

Які середовища програмування ви використовували в попередніх класах?

Для зручного розроблення програм існують спеціальні засоби — середовища (системи) програмування, що забезпечують весь цикл роботи з програмою — від розробки до виконання й отримання необхідних результатів.

Середовище програмування — це комплекс програмних засобів, які призначено для автоматизації процесу підготовки та виконання програм користувача.

Python інтегрується з іншими мовами, наприклад C, C++, Java. Це означає, що програміст може використовувати у власних програмах програми на інших мовах.

Середовище програмування містить низку різних програм (рис. 9.3). Розглянемо таблицю:

Програма	Призначення
Редактор тексту	Забезпечує введення, записування та читання програми із файлу
Транслятор	Перетворює програму у двійковий код — об'єктний модуль. Містить посилання на інші модулі, що зберігаються в бібліотеці середовища програмування
Редактор зв'язків	Збирає всі зазначені модулі в єдиний модуль і створює двійковий код, що виконується. Такий код може виконуватися без транслятора
Налагоджувач	Допомагає виявити помилки в програмі і здійснити її тестування

Для програм, розроблених мовами об'єктно-орієнтованого програмування (ООП), після компіляції отримується проміжний код (байт-код), який за допомогою спеціального програмного забезпечення перетворюється в машинний.

Далі ми будемо використовувати середовище IDLE, яке підтримує режим компілювання й інтерпретування програм. Програмний код мовою Python можна виконувати в інтерактивному режимі та режимі виконання файлів коду.

На етапі вивчення мови Python особливо корисним для виконання окремих команд є інтерактивний режим IDLE. Запустити його можна за допомогою команд Пуск → Усі програми → Python 3.4 → IDLE (Python 3.4GUI — 32 біти). Відкриється вікно, зображене на рис. 9.4. Після знака запрошення (>>>) можна вводити команди.



Рис. 9.3. Програми середовища програмування

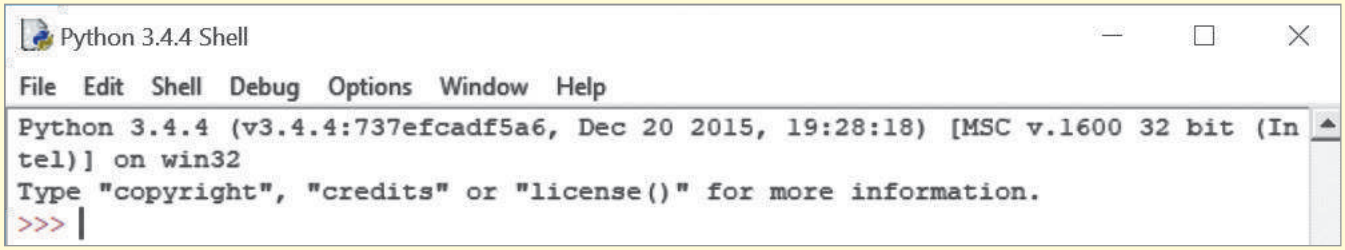


Рис. 9.4. Вікно інтерпретатора IDLE

У прикладі 1 подано виконання найпростіших команд. Після введення кожної команди потрібно натискати Enter.

В одному рядку можна вводити більш ніж одну команду й відокремлювати їх крапкою з комою (приклад 2). Одну команду можна розміщувати у кількох рядках (приклад 3). Розглянемо приклад 4 виконання найпростішого коду в інтерактивному режимі.

Приклад 1.

```
>>> 5 + 7
12
>>> y = 3.4 + 5.9
>>> y
9.3
>>> y = 56.7 / 3
>>> y
18.900000000000002
```

Приклад 2.

```
>>> 54 + 27; 13.4 + 32
81
45.4
```

Приклад 3.

Команду 203 / 4 + 5 – 2 * 4 можна записати так:

```
>>> 203 / 4 + 5 –\
        2 * 4
47.75
```

Приклад 4.

Нехай програмний код повинен виводити на екран повідомлення, що містить два рядки:

Я — учень
8 класу.

Вивести рядок повідомлення на екран можна за допомогою оператора print. Код програми міститиме дві команди:

```
print("Я учень")      # Перший рядок
print("8 класу.")     # Другий рядок
```

Тут знак # — це коментар, який пояснює сутність команди й на виконання коду не впливає.

Після знака запрошення уводимо перший рядок коду і натискаємо клавішу Enter. Знак запрошення переміститься на наступний рядок, після якого уводимо другу команду і натискаємо клавішу Enter. На рис. 9.5 відображено динаміку цього процесу.

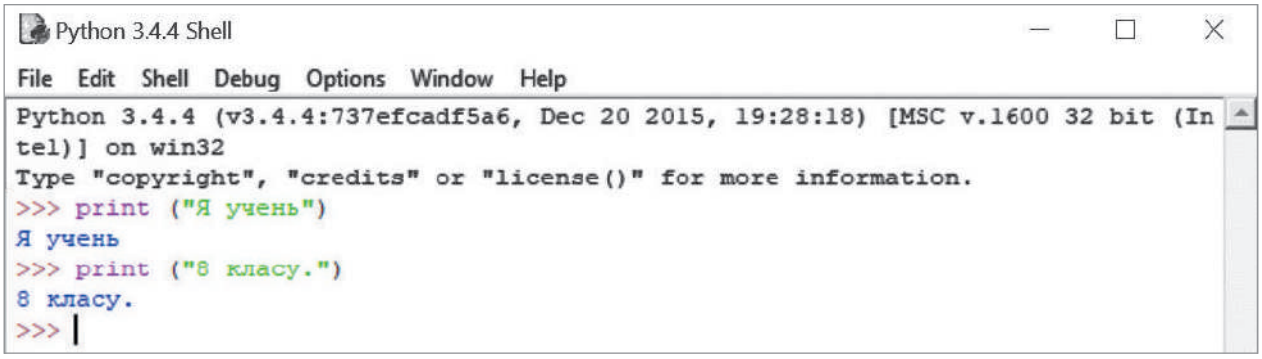


Рис. 9.5. Динаміка виконання коду в інтерактивному режимі

Виконати код можна за допомогою команд Run → Run Module, унаслідок чого отримаємо $y = 11.456666666666667$.

Режим виконання файлів програмного коду. В інтерактивному режимі програмний код не зберігається. Для повторного виконання коду його потрібно увести заново. Код, збережений у файлі, можна виконувати багаторазово. Якщо у коді є синтаксичні помилки, програма не виконується, а повідомлення про помилки будуть виведені на екран.

Для закриття коду потрібно виконати команду File → Close. Для відкриття файлу слід виконати команду File → Open… і у вікні, що відкриється, вибрати папку Програми, потім знайти файл Pr_10.01 і натиснути кнопку Відкрити.

Алгоритм уведення програмного коду

Крок 1

У середовищі IDLE виконайте команди **File** → **New File.** У вікно **Untitled**, що відкриється, введіть код. На рис. 9.6 наведено код обчислення значення *y*: $y = \dfrac{\dfrac{23{,}4+8}{2}+3\cdot(35{,}5-8)}{3\cdot(6-2{,}5)}$.

```
*Untitled*
File  Edit  Format  Run  Options  Window  Help
a = (23.4+8)/2+3*(35.5-8)      # Обчислення чисельника
b = 30/(6-2.50)                # Обчислення знаменника
y = a/b                        # Обчислення виразу
print "y = ", y)               # Виведення результату
```

Рис. 9.6. Код обчислення арифметичного виразу

Крок 2

Збережіть код: виконайте команду **File** → **Save As...** У результаті відкриється вікно **Збереження файлу.** У цьому вікні відкрийте, наприклад, папку **Програми.**

Крок 3

Уведіть ім'я файлу, наприклад Pr_10.01, і натисніть кнопку **Зберегти.** У результаті файл буде збережено у вигляді, який наведено на рис. 9.7.

```
Pr_10.01.py - C:/Users/Віктор/Desktop/ПІДРУ_8_погл/Програми/Pr_10.01.py (3.4.4)
File  Edit  Format  Run  Options  Window  Help
a = (23.4+8)/2+3*(35.5-8)      # Обчислення чисельника
b = 30/(6-2.50)                # Обчислення знаменника
y = a/b                        # Обчислення виразу
print "y = ", y)               # Виведення результату
```

Рис. 9.7. Файл PR_10.01, збережений у папці **Програми**

Запитання для перевірки знань

1. Для чого призначене середовище програмування?
2. Які складові середовища програмування?
3. Як запустити інтерактивний режим IDLE?
4. Для чого призначений транслятор?
5. Які функції виконує редактор зв'язків?
6. Наведіть приклади виконання команд в інтерактивному режимі.
7. Порівняйте інтерактивний режим і режим виконання файлів програмного коду.

Завдання для самостійного виконання

1. Обчисліть в інтерактивному режимі: (17 + 5) / 3.3.
2. Обчисліть в інтерактивному режимі: 21.7 + 5 і 32 / 2.75.
3. Обчисліть в інтерактивному режимі у двох рядках: 139/8 + 12.5 – 3(11.6 + 2.2).

Завдання для самостійного виконання

4 Виведіть в інтерактивному режимі на екран два рядки: Мова Python
є гарною мовою.

5 Розробіть, збережіть і виконайте код обчислення значення y:

$$y = 3{,}75 + 3 \cdot \left(\frac{24{,}8}{3} + \frac{2}{9} \right).$$

Переконайтеся у правильності результату.

6 Розробіть, збережіть і виконайте код обчислення значення y:

$$y = \frac{43{,}7 + 11{,}5}{3 + 3{,}3} + 4{,}1 \left(\frac{66{,}4}{39{,}5 - 12{,}4} + \frac{2}{4} \right).$$

Переконайтеся у правильності результату.

9.4. Поняття об'єкта, властивості, події та обробники подій. Проєкт і його структура

Пригадайте, що таке об'єкт.

Об'єкт в об'єктно-орієнтовоному програмуванні — це предмет, поняття, явище, які подані у цифровій формі і належать до певного класу, а також характеризуються своїми властивостями і поведінкою.

Приклад 1. Учень — це абстрактне поняття, яке може бути класом, а учень Сидоренко Іван або учениця Іваненко Катерина — конкретне поняття, тобто є об'єктом.

Властивості об'єкта визначаються його атрибутами, які називають полями даних, а його поведінка — функціями їх опрацювання — методами.

Під час виконання програми об'єкти зазвичай зберігаються в оперативній пам'яті. Всі методи опрацювання об'єктів зберігаються поза межами об'єктів. Отже, об'єкти групуються в однотипні класи. Клас — це абстрактне поняття, а об'єкт — конкретне (приклад 1).

Кожний об'єкт має ім'я, яке записується як звичайний ідентифікатор, наприклад obj, man. Кожна властивість об'єкта має певне значення (приклад 2).

Приклад 2. Властивість автомобіля color може мати такі значення: чорний, білий, червоний; властивість номер класу об'єкта учень Сидоренко Іван може мати значення 1, 2 та ін.

Подія в програмуванні — це спосіб взаємодії між об'єктами. Кожна подія розпізнається комп'ютером і опрацьовується відповідним обробником подій.

Події можуть бути згенеровані системою, наприклад у разі ділення на нуль, а також користувачем або іншими способами (приклад 3)

Приклад 3. Після натискання у банкоматі кнопки Баланс викликається відповідний обробник подій, у результаті чого на екрані можна отримати власний баланс.

У мові Python є значна кількість вбудованих об'єктів відповідних класів, які не потрібно створювати програмісту самому, а лише слід правильно використовувати. Так, об'єктами є цілі числа, які належать класу int, дійсні числа (клас float), рядки (клас str), списки, кортежі та інші. Самі класи теж є об'єктами.

Програміст у мові Python може не тільки використовувати вбудовані об'єкти, а й створювати об'єкти. У такому разі на початку програмного коду створюється клас.

Найпростіша структура створення класа:

class <ім'я класу>:
    <тіло класу>

Наприклад,

class Car:
    <тіло класу>

У тілі класу зазначаються методи, які опрацьовуватимуть дані програмного коду.

Загальна структура створення об'єктів:

<ім'я об'єкта> = <ім'я класу>([параметри])

Наприклад, Audi = Car(). Як бачимо, тут параметри оголошення об'єкта не обов'язкові. Методи оголошуються як звичайні функції за допомогою інструкції def.

Структура оголошення методу:

def <ім'я методу>(self, [інші параметри]):

До одного методу можна звертатися з багатьох об'єктів класу. Під час звернення до методу в параметр self автоматично передається ім'я об'єкта, із якого здійснюється звернення до цього методу, а в інші параметри передаються значення відповідних аргументів.

Об'єктами у мові Python є також такі графічні елементи, як кнопки (клас Button), перемикачі (клас Radiobutton), прапорці (клас Checkbutton) та інші, які в деяких мовах програмування, наприклад у мові Delphi, називають компонентами.

Отже, поведінка об'єкта визначається можливостями методу, до якого здійснюється звернення із певного об'єкта.

Приклад 4. На рис. 9.8 наведено код обчислення площі й периметра прямокутного трикутника за значенням його катетів. У коді створюється клас, два об'єкти і два методи.

```
# Код, у якому створюється клас, два методи і два об'єкти
class Pr_10_02:                                    # Клас з іменем Pr_10_02
    def func1 (self, a1, a2):                      # Метод обчислення площі
        surface = 0.5*a*b                          # Обчислення площі трикутника
        print ('Площа = ', surface)                # Виведення значення площі
    def func2 (self, a1, a2):                      # Метод обчислення периметра
        perimeter = a + b + math.sqrt (a*a + b*b)  # Обчислення периметра
        return  (perimeter)                        # Повернення результата
import math                                        # Імпортування модуля math
a = float (input ('Катет a = '))                   # Введення значення катета a
b = float (input ('Катет b = ',))                  # Введення значення катета b
obj1 = Pr_10_02 ()                                 # Створення об'єкта obj1
obj2 = Pr_10_02 ()                                 # Створення об'єкта obj2
obj1.func1 (a, b)                                  # Звернення до методу func1
print ('Периметр = ', obj2.func2 (a,b))            # Звернення до методу func2
```

Рис. 9.8. Код обчислення площі й периметра прямокутного трикутника

У коді обчислення площі трикутника здійснюється за допомогою методу func1, а його периметра — за допомогою методу func2. Обидва методи мають параметри self, a1 і a2.

У процесі звернення до методу func1 із об'єкта obj1 параметру self передається ідентифікатор об'єкта obj1, параметру a1 передається значення аргументу a, параметру a2 — значення аргументу b. Обчислене значення площі трикутника присвоюється змінній surface, яке за допомогою оператора print виводиться на екран.

Звернення до методу func2 здійснюється із команди print('Периметр =', obj2.func2(a, b)). Параметру self цього методу передається ідентифікатор об'єкта obj2, а параметрам a1 і a2 — ті самі значення катетів a і b, що й у процесі звернення до методу func1.

Для обчислення периметра необхідно використовувати функцію кореня квадратного (sqrt), яка міститься у модулі math. Щоб скористатися цією функцією модуль math імпортується в програму за допомогою команди import math.

Обчислене значення периметра у методі func2 за допомогою команди return(perimeter) повертається в команду, що викликала цей метод, тобто у команду print, і виводиться на екран.

У коді введення значення катета *a* здійснюється за допомогою команди a = float(input('Катет = ')). У команді використовується оператор input. Цей оператор призупиняє виконання коду й очікує введення з клавіатури необхідного значення.

Після введення значення слід натиснути клавішу Enter, у результаті уведене значення буде присвоєно змінній a і виконання коду буде продовжено. Аналогічно здійснюється уведення значення катета b.

Усі дані, що вводяться з клавіатури, сприймаються як рядковий тип (тип str). Навіть у випадку, якщо буде введено ціле число, в дійсності воно матиме тип str. Для перетворення його у дійсний тип у команді використано функцію float.

Результат виконання програми:

```
Катет a = 4
Катет b = 3
Площа =  6.0
Периметр =  12.0
```



Катерина Логвинівна Ющенко — засновниця української школи програмування, авторка однієї з перших у світі мов програмування високого рівня, член-кореспондентка НАН України.

Програма мовою Python, її називають проєктом, складається з модулів. Модуль — це будь-який файл із програмним кодом. Кількість таких модулів не обмежена. Один модуль може бути вкладений в інший, тобто застосовується багатоієрархічна структура модулів. Модулі можуть групуватися в пакети.

Для використання одного модуля з іншого його потрібно підключити (імпортувати) в цей модуль.

Отже, основою ООП є клас і об'єкт. Клас — це складний тип даних із певним набором змінних (полів) і функцій (методів) опрацювання значень, що зберігаються у цих змінних. Під час кожного звернення до класу створюється новий об'єкт — екземпляр класу. Тому можна отримати необмежену кількість екземплярів на основі цього класу.

Можна використовувати модулі як розроблені самостійно, так і наявні у стандартній бібліотеці мови. Один із модулів є головним, з нього запускається проєкт на виконання. Щоб запустити один модуль з іншого, його необхідно підключити до останнього (імпортувати).

Запитання для перевірки знань

1. Поясніть сутність терміна «об'єкт» у програмуванні.
2. Наведіть приклади властивостей та значень об'єктів.
3. Поясніть сутність події та обробника подій у програмуванні.
4. Які особливості мають об'єкти у мові Python?
5. Як оголошуються об'єкти і класи у мові Python?
6. Яку структуру має проєкт мовою Python?

Практична робота № 18

ТЕМА. Робота у середовищі програмування

ЗАВДАННЯ: виконати в інтерактивному режимі середовища IDLE найпростіші команди і код в режимі виконання файлів програмного коду.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки

1. Обчисліть в інтерактивному режимі середовища IDLE:
   а) 156.5 + 21.6; б) 378 : 5; в) 584 * 13.
2. Обчисліть в інтерактивному режимі вирази 48 – 21 / 3.5 і 263 * 5 – 4.1. Обидва вирази запишіть в одному рядку.
3. Обчисліть в інтерактивному режимі значення виразу
   753 – 3.1 * (54 + 32.5) + 77.4 / 6.
   Вираз запишіть у двох рядках.
4. Обчисліть в інтерактивному режимі:
   y = 28.6 + 41 * 3.5.
5. Обчисліть в інтерактивному режимі:
   y = (96.7 + 8.25 * 3.7) / 6.2.
6. Розробіть код обчислення значення виразу
$$\frac{112{,}6+85\cdot 4{,}2}{(96{,}7+8{,}25\cdot 3{,}7)\cdot 2{,}1}$$
7. Уведіть код, збережіть його, виконайте.
8. Розробіть код із привітанням з днем учителя. Уведіть, збережіть і виконайте код.
9. Розробіть код обчислення значення виразу:
$$\frac{78{,}5-(43{,}8+52{,}6)\cdot 32}{\dfrac{45{,}7+23{,}9}{3{,}2}}.$$
10. Уведіть, збережіть і виконайте код.
11. Закрийте файл коду. Завантажте його і виконайте.

Зробіть висновок про доцільність використання інтерактивного режиму та переваги режиму виконання файлів.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 9

7. Доведіть, що отриманий результат є правильним.

8. Переконайтеся, що код функціонує правильно.

10. Доведіть, що код функціонує правильно.

11. Переконайтеся, що отримано той самий результат.

10. Лінійні алгоритми

10.1. Дані, змінні, константи. Прості типи даних

Пригадайте, які типи даних ви використовували в попередніх класах.

Комп'ютер може опрацьовувати цифрові, текстові, графічні, звукові дані тощо.

Спочатку всі дані перетворюються у двійковий код, а потім зберігаються й опрацьовуються. Для відображення або відтворення даних вони, навпаки, перетворюються із двійкового коду в текст, звук тощо.

Дані в інформатиці — це відомості, подані у формі, придатній для їх збереження й опрацювання в комп'ютері.

Дані мають тип, який визначає множину значень і операцій, що можуть виконуватися над ними. Тип визначає також структуру даних і обсяг пам'яті, необхідний для їх зберігання. Розглянемо, як поділяють типи даних (рис. 10.1).

У Python існують базові об'єктні типи (вбудовані в саму мову) і типи, які розробляє програміст.

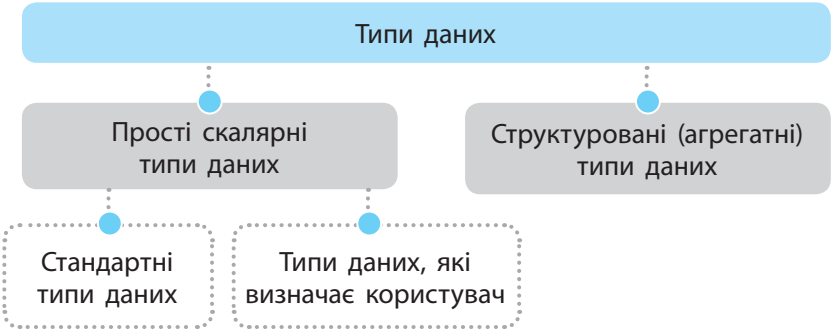


Рис. 10.1. Класифікація типів даних

У мові Python використовується динамічна типізація даних. Це означає, що не потрібно оголошувати тип даних, він визначається за значенням самих даних. Усі типи даних у Python подаються у формі об'єкта певного класу. Наприклад, цілі числа визначаються класом int.

Під час виконання програми дані можуть змінюватися або бути незмінними, тобто в програмуванні існують змінні й константи.

Змінна в програмуванні — це певна ділянка пам'яті (комірка пам'яті), яка позначається іменем (ідентифікатором) і у якій зберігається значення певного типу, наприклад число, слово тощо.

Ідентифікатором не можуть бути ключові слова, наприклад, print, input, in, for, import та інші, а також назви функцій (sqrt, sum та ін.).

Ідентифікатор змінної складається з латинських букв, цифр і знаків підкреслення.

Першим символом в імені не може бути цифра (приклад 1).

Однакові імена на різних регістрах сприймаються як різні ідентифікатори. Так, Masa і masa є різними ідентифікаторами.

Приклад 1.
Правильні ідентифікатори:
mas_1, V, beta.
Неправильні ідентифікатори:
23ab, 45, a-m, cd?a.

Недоцільно користуватися дуже довгими та дуже короткими іменами. Бажано вибирати осмислені імена, наприклад швидкість руху можна позначити змінною v, значення суми — змінною Suma.

Простими вбудованими типами даних у мові Python є цілі числа (int), дійсні числа (float), логічні дані (bool) (приклад 2). Інколи до простих відносять і рядки символів (тип str), хоча в дійсності це складний тип даних (приклад 3).

Для присвоювання значень змінним використовується оператор присвоювання, який у мові Python позначається знаком дорівнює (=).

Структура команди присвоювання:

<ім'я змінної> = <вираз>

Команда виконується у такому порядку: обчислюється значення виразу, яке потім присвоюється змінній (приклад 4). Найпростішим виразом може бути константа.

Приклад 2.
Прості типи даних:

Цілі числа (int)	231, 77
Дійсні числа (float)	31.75, 5.25
Логічні дані (bool)	True, False

Приклад 3.
Рядки: "Україна", "система".

Приклад 4.

```
>>> suma = 429          # Змінна suma має тип int і значення 429
>>> a_1 = 321.76        # Змінна a_1 має тип float і значення 321.76
>>> slov = «принтер»    # Змінна slov має тип str і значення принтер
>>> cd = False          # Змінна cd має тип bool і значення False
```

Зазначимо, що після виконання оператора присвоювання у змінній зберігається не сам об'єкт (значення), а лише посилання на нього, тобто адреса комірки пам'яті, у якій зберігається об'єкт.

Приклад 5.

```
>>> a_1 = a_2 = [32, 55]      # Створюються ніби два об'єкти
>>> a_1, a_2                  # Виведення значень a_1 і a_2
([32, 55], [32, 55])          # Двічі виводяться значення одного об'єкта
```

У прикладі 5 обидві змінні мають одну й ту саму адресу пам'яті. Тому реально створюється один об'єкт, значення якого виводяться двічі. Щоб переконатися в цьому, змінимо одне зі значень об'єкта a_2 (нумерація елементів у мові починається з нуля) і перевіримо значення об'єктів:

```
>>> a_2[0] = 100          # Зміна значення нульового елемента
                          # об'єк та a_2
>>> a_1, a_2              # Виведення значень a_1 і a_2
([100, 55], [100, 55])    # Значення змінних a_1 і a_2 однакові,
                          # хоча значення a_1 не змінювалося
```

День програміста святкують у 256-й день року (у високосний рік це 12 вересня, а в невисокосний — 13 вересня). Як ви думаєте, чому обрано саме цей день? Вибір пояснюється тим, що це число символічне, воно тісно пов'язане з комп'ютерами, але не асоціюється з конкретними особами чи кодами спеціальностей. Число 256 відповідає кількості символів, які можна подати за допомогою одного байта.

Алан Кертіс Кей — американський інформатик, відомий своїми працями в галузі ООП, президент дослідного інституту В'юпоїнта, ад'юнкт-професор інформатики в Каліфорнійському університеті (США). Кей уважав, що найкращий спосіб спрогнозувати майбутнє — винайти його.

Як бачимо, значення a_1 не змінювалося, а об'єкти також мають однакові значення, тому що вони мають однакову адресу. Для того щоб об'єкти мали різні значення, необхідно присвоювати значення окремо кожному з них.

Для перевірки того, чи посилаються змінні на один і той самий об'єкт, слугує оператор is. Цей оператор повертає значення True, якщо змінні посилаються на один об'єкт, і значення False, якщо інакше (приклад 6).

Приклад 6.

```
>>> a_1 = a_2 = [56, 20]    # Перевірка посилання змінних на один
>>> a_1 is a_2              # і той самий об'єкт
True
```

Одним оператором присвоювання можна присвоїти значення декільком змінним. У такому разі змінні й значення відокремлюються комою одне від одного (приклад 7).

Приклад 7.

```
>>> a_1, a_2 = 183, 57
>>> a_1, a_2
(183, 57)
```

Кількість значень в операторі присвоювання може бути більшою за кількість змінних. У такому разі в переліку змінних використовується символ зірочка (*). Змінна, перед якою розташований цей символ, буде містити усі зайві значення (приклад 8).

Приклад 8.

```
>>> a_1, a_2, *a_3 = (8, 23, 4, 32, 5)      # Змінна a_3 набуде значень 4, 32, 5
>>> a_1, a_2, a_3
(8, 23, [4, 32, 5])
>>> a_1, *a_2, a_3 = (7, 2, 13, 25)         # Змінна a_2 набуде значень 2 і 13
>>> a_1, a_2, a_3
(7, [2,13], 25)
```



Американська розробниця ПЗ Радія Джой Перлман стала першою, хто почав навчати програмуванню дітей молодшого віку. Вона розробила дитячу версію навчальної робототехнічної мови LOGO (названу TORTIS).

Тип даних, що зберігаються у змінних, можна перевірити за допомогою функції type(ім'я змінної) (приклад 9).

Приклад 9.

```
>>> a_1 = 171
>>> type(a_1)    # Визначення типу змінної a_1
<class 'int'>    # Змінна a_1 посилається на тип int
```

Для кожного конкретного типу даних існує строго визначений набір операцій, які можуть виконуватися над ним. Наприклад, для даних типів int і float можна виконувати арифметичні операції.

Спроба виконати, наприклад, операцію додавання цілого числа й рядка:

```
>>> 43+"25"
```

призведе до виведення повідомлення про синтаксичну помилку.

Для перетворення одного типу даних в інший у мові Python застосовуються спеціальні функції. Далі розглянемо основні з них.

Функція	Опис	Приклад
bool([об'єкт])	Перетворює об'єкт в логічний тип. З прикладу видно: якщо об'єкт дорівнює нулю або порожній, генерується значення False, інакше — значення True	>>> bool(0), bool(21), bool([9, 13]), bool(""), bool("біт") (False, True, True, False, True)
int([об'єкт [, <система числення>]])	Перетворює об'єкт у ціле число. Система числення, у яку буде перетворюватися об'єкт, може бути десятковою, вісімковою, шістнадцятковою. За замовчуванням — десяткова система	>>> int(93.35), int("25") (93, 25)
float(ціле число або рядок)	Перетворює ціле число або рядок у число дійсного типу	>>> float(41), float("45.25") (41.0, 45.25)
str(об'єкт)	Перетворює об'єкт у рядок	>>> str(23), str([55, 7]) ('23', '[55, 7]')

Запитання для перевірки знань

1. Поясніть сутність динамічної типізації даних.
2. Назвіть прості типи вбудованих даних мови Python.
3. За допомогою якого оператора змінним присвоюються значення?
4. Що називають даними в програмуванні?
5. За якими ознаками класифікують дані?
6. Які обмеження існують на ідентифікатори змінних?
7. Що називають змінною в програмуванні?

Завдання для самостійного виконання

1. Присвойте двом змінним значення 67.34 і 31. Ідентифікатори змінних виберіть самостійно.
2. Трьом змінним присвойте одним оператором присвоювання значення 76, 3216, 563.
3. Доведіть, що після виконання оператора
   >>> a_1 = a_2 = a_3 = [23, 5, 101]
   змінні будуть посилатися на один об'єкт.
4. Спробуйте виконати оператор
   >>> a_1, a_2, a_3 = (65, 24).
   Проаналізуйте повідомлення, яке ви отримаєте. Знайдіть і видаліть помилку в цьому операторі.
5. Виконайте оператор
   >>> a_1, *a_2, a_3 = (9, 14, 23, 18, 213).
   Доведіть, що результат, який ви отримали, правильним.
6. Запишіть оператор присвоювання так, щоб змінна a_1 набула значення 43, змінна a_2 — значення 231 і змінна a_3 — значення 8 і 83.

10.2. Арифметичні операції і вирази. Стандартні підпрограми



Які основні складові містить арифметичний вираз і який існує пріоритет виконання арифметичних операцій у виразі?

Пригадаємо, що до арифметичних операцій у мові Python належать операції додавання (+), віднімання (–), множення (*) і ділення (/).

Якщо арифметичні операції виконуються над цілим і дійсним числами, то ціле число буде спочатку перетворене на дійсний тип, а потім виконуватиметься операція над дійсними числами. Результатом операції у цьому випадку буде число дійсного типу. Результатом операції ділення завжди буде число дійсного типу.

Окрім арифметичних операцій, у мові Python застосовуються такі операції:

Поняття виразу в програмуванні відповідає поняттю виразу в математиці. Залежно від типу отриманого результату розрізняють вирази арифметичні, логічні, рядкові та ін.

Операція	Позначення	Приклад
Цілочислове ділення (без остачі)	//	У результаті виконання операції: 22.0//3.0 отримаємо результат 7.0
Ділення за модулем (остача від ділення)	%	У результаті виконання операції 14%3 отримаємо результат 2
Піднесення до степеня	**	У результаті виконання операції 10**3 отримаємо результат 1000

У мові Python використовуються арифметичні оператори з присвоюванням:

Операція	Позначення	Приклад
Збільшення значення змінної на вказану величину	+=	x += 23 (еквівалентно x = x+23)
Зменшення значення змінної на вказану величину	–=	x –= 19 (еквівалентно x = x–19)
Множення значення змінної на вказану величину	*=	x *= 6 (еквівалентно x = x*6);
Ділення значення змінної на вказану величину	/=	x /= 7 (еквівалентно x = x/7).

Кожний оператор може виконувати операції над строго визначеними для нього типами операндів. Залежно від типу об'єктів, над якими виконуються операції, оператори групуються в арифметичні, логічні, порівняння, присвоювання тощо.

Об'єкти, над якими виконуються арифметичні операції, називають **операндами**.

Над операндами можна виконувати операції порівняння. Оператори порівняння порівнюють значення об'єкта, який розташовано ліворуч від оператора, зі значенням об'єкта, який розташовано праворуч від цього оператора.

Якщо задана умова виконується, повертається значення True, інакше — False.

Склад, позначення і приклади використання операцій порівняння наведено в таблиці:

Позначення	Назва	Пояснення	Приклад
==	Дорівнює	Якщо значення операндів однакові, то повертається значення True, інакше — False	>>> 33 == 9 False
!=	Не дорівнює	Якщо значення операндів неоднакові, то повертається значення True, інакше — False	>>> 24 != 7 True
>	Більше	Якщо значення операнда зліва більше за значення операнда справа, то повертається значення True, інакше — False	>>> 13 > 41 False
<	Менше	Якщо значення операнда зліва менше за значення операнда справа, то повертається значення True, інакше — False	>>> 5 < 8 True
>=	Більше або дорівнює	Якщо значення операнда зліва більше або дорівнює правому, то повертається значення True, інакше — False	>>> 9 >= 9 True
<=	Менше або дорівнює	Якщо значення операнда зліва менше або дорівнює правому, повертається значення True, інакше — False	>>> 17 <= 5 False

Арифметичні вирази складаються із арифметичних операцій, операндів, функцій і круглих дужок, наприклад: $y = 4.5*a + (a*b - 3.1*c) / 2.2$.

Операції в арифметичному виразі виконуються з урахуванням їх пріоритету, а операції, що мають однаковий пріоритет, — у порядку їх розміщення (зліва направо).

У першу чергу виконуються операції в круглих дужках. Кожній дужці, що відкривається, має відповідати дужка, що закривається.

Приклад 1. На рис. 10.2 зображено код обчислення значення y, де $y = \dfrac{3,4(2,1a+4b)}{2,5}$.

```
a = 23                                  #23 присвоюється змінній a
b = int (input ('Увести значення b '))  #Введення значення b
y = 3.4*(2.1*a + 4*b)/2.5               #Обчислення виразу
print ('y = ',y)                        #Виведення результату
```

Рис. 10.2. Код обчислення арифметичного виразу

Тут input — оператор введення. Він призупиняє виконання коду й очікує введення значення з клавіатури. Після введення значення слід натиснути клавішу Enter, і виконання коду продовжується до повного завершення.

Усі дані, що вводяться з клавіатури, є типом str. Тому вони перетворені на тип int.

Результат виконання програми:

```
Увести значення b 4
y =  87.44800000000001
```

Для опрацювання числових даних у мові Python є стандартні підпрограми, оформлені у вигляді функцій:

Функція	Опис	Приклад
round(<число>[, <кількість знаків після коми>]))	Повертає число із зазначеною кількістю цифр після коми	>>> round(8.234, 1) 8.2
abs(число)	Повертає абсолютне значення	>>> abs(–86) 86

Функція	Опис	Приклад
pow(<число>, <ступінь>)	Повертає число в степені	>>> pow(10, 3) 1000
max(<числа через кому>)	Повертає максимальне значення	>>> max(13, 45, 7) 45
min(<числа через кому>)	Повертає мінімальне значення	>>> min(8, 3, 6) 3
sum([<числа>], <початкове значення>)	Повертає суму чисел і початкового значення	>>> sum([3, 5], 13) 21

Приклад 2. На рис. 10.3 зображено код обчислення y, де $y = \frac{\left(3a^3 + 2|b|\right)}{b^2} - a$, із використанням вбудованих у мову функцій опрацювання чисел.

```
a = int (input ('Увести значення a '))      #Введення значення a
b = float (input ('Увести значення b '))    #Введення значення b
y = (3*pow(a,2) + 2*abs(b))/pow(b,2) - a    #Обчислення виразу
print ('y = ', round (y,2))                 #Виведення результату
```

Рис.10.3. Код обчислення арифметичного виразу з використанням функцій

Результат виконання програми:

```
Увести значення a 3
Увести значення b 2
y =  18.25
```

Крім описаних функцій, у мові Python існують і інші функції для роботи з числами, які містяться у модулі math. Цей модуль містить також деякі константи, наприклад, pi, e. Для використання констант і функцій цього модуля його необхідно імпортувати у програму за допомогою інструкції import math. Розглянемо найуживаніші функції модуля math:

Функція	Опис	Приклад

sqrt()	Корінь квадратний	>>> import math
		>>> math.sqrt(77)
		8.774964387392123

log10()	Логарифм десятковий	>>> math.log10(25) 1.3979400086720377
ceil()	Найближче більше ціле	>>> math.ceil(29.3) 30
floor()	Найближче менше ціле	>>> math.floor(3.71) 3
fmod()	Остача від ділення	>>> math.fmod(17, 3) 2.0
factorial()	Факторіал числа	>>> math.factorial(6) 720

Після того як Гвідо ван Россум розробив мову Python (приблизно у 1991 році), він виклав її в інтернет. Мова Python сподобалася програмістам і почала вільно поширюватися. Таким чином, до розробки приєдналося вже співтовариство програмістів. У середньому кожні 2 чи 2,5 року з'являється нова версія мови.

Функції для роботи з випадковими числами містяться в модулі random. Цей модуль також необхідно імпортувати в програму за допомогою команди import random. Розглянемо найуживаніші функції модуля random:

Функція	Опис	Приклад

random.random()	Генерує випадкове число від 0.0 до 1.0	>>> import random
		>>> andom.random()
		0.5432200164843041

uniform(початок, кінець)	Генерує дійсне випадкове число у діапазоні від «початок» до «кінець»	>>> random.uniform(1, 10) 5.342462615004567
randint(початок, кінець)	Генерує ціле випадкове число у діапазоні від «початок» до «кінець»	>>>random.randint(3, 12) 8
choice(послідовність)	Вибирає з послідовності (рядка, списку) випадковий елемент	>>>random.choice ("клавіатура") 'в'

Приклад 3. На рис. 10.4 наведено код обчислення y, де $y = a^3 + \sqrt{3ab + 4a}$, із використанням функцій модулів math і random. Отриманий результат множиться на ціле випадкове число в діапазоні від 2 до 10 і від нього береться найближче ціле.

```
import math                                  #Імпортування модуля math
import random                                #Імпортування модуля random
b = 4                                        #Присвоювання значення змінній
a = float (input ('Увести значення a '))     #Введення значення a
y = pow (a, 3) + math.sqrt (3*a*b + 4*a)     #Обчислення виразу
c = random.randint (2, 10)                   #Генерування випадкового числа
print ('Випадкове число = ',c)               #Виведення випадкового числа
y = y * c                                    #Множення значення на випадкове число
print ('y = ', y)                            #Виведення результату множення
y = math.floor (y)                           #Найближче ціле
print ('y = ', y)                            #Виведення найближчого цілого
```

Рис.10.4. Код обчислення значення виразу з використанням модулів math і random

Результат виконання програми:

```
Увести значення a 3
Випадкове число =  7
y =  237.49742261192856
y =  237
```

Запитання для перевірки знань

1. Наведіть приклад ділення чисел без остачі.
2. Поясніть порядок виконання оператора присвоювання.
3. Як виконуються операції порівняння?
4. Які складові містять арифметичні вирази?
5. Як виконується оператор **input**?
6. Які основні функції містить модуль **math**?
7. Які основні функції містить модуль **random**?

Завдання для самостійного виконання

1. В інтерактивному режимі обчисліть корінь квадратний із 67.
2. В інтерактивному режимі обчисліть 6^4.
3. Розробіть код обчислення кореня квадратного із суми чисел 54,7 і 326,57.
4. Генеруються два цілих випадкових числа в діапазоні від 5 до 12. Розробіть код обчислення їх суми і кореня квадратного із суми.
5. Розробіть код обчислення значення *y*, де $y=\frac{\sqrt{a^3+2,5b}}{0,5(a+b)}$, і ділення його на ціле випадкове число в діапазоні від 3 до 7.
6. Розробіть код обчислення суми трьох випадкових чисел у діапазоні від 7 до 15 і визначення їх середнього значення.

10.3. Структуровані вбудовані типи даних та їх опрацювання

Пригадайте, як поділяються типи даних у будь-якій мові програмування.

Нині Python — одна з найпопулярніших мов програмування, яка охоплює нові сфери застосування і останні 5 років входить до п'ятірки найзатребуваніших технологій.

До структурованих типів даних, убудованих у мову Python, належать списки, кортежі, діапазони, множини, словники. Рядки теж є структурованими, але внаслідок широкого розповсюдження їх часто відносять і до простих. Ми розглядатимемо списки і словники, а також спеціальні функції та методи, застосовувані для їх опрацювання.

Пригадаємо, що усі типи даних у мові Python є об'єктами певних класів. Класи містять методи — програми, що виконують ту чи іншу функцію.

Метод викликається для конкретного об'єкта. Для виклику методу спочатку вказується об'єкт, потім крапка, за якою зазначається ім'я методу:

<об'єкт>.<ім'я методу>

10.3.1. Списки

Пригадайте, які списки ви використовували у повсякденній діяльності. Спробуйте сформулювати означення списку.

Список — це певна сукупність об'єктів будь-якого типу у квадратних дужках, які відокремлюються один від одного комою.

Список є одним з основних типів даних (приклад 1). У списках можна змінювати значення його елементів, збільшувати і зменшувати кількість елементів, здійснювати пошук потрібних елементів і впорядковувати їх. Отже, список є об'єктом, що змінюється.

Списки можуть бути вкладені в об'єкти інших типів.

Різновидом списків є масив. Різниця між списком і масивом полягає у тому, що у масивах значення елементів можуть бути лише одного типу, наприклад тільки цілі числа, тільки рядки, а у списках — будь-якого типу. Для того щоб зручніше орієнтуватися, списки позначатимемо ідентифікатором lst.

У мові Python є значна кількість операцій, функцій і методів опрацювання списків. Далі розглядаються лише основні з них.

Списки бувають одновимірні й багатовимірні.

Одновимірні списки. Позиція елемента у списку задається індексом, який починається з нуля. Списки можна створювати простим перерахуванням елементів списку в квадратних дужках (приклад 2).

Розглянемо операції, які можна виконувати над списками:

Приклад 1. [48, "клавіатура", 29.5]. Тут об'єктами списку є ціле число, рядок і дійсне число.

Приклад 2.

```
>>> lst = ["море", 1, 22, "5"]
>>> lst
['море', 1, 22, '5']
```

Щоб звернутися до елемента списку, необхідно у квадратних дужках зазначити індекс елемента (нумерація елементів у списку починається з нуля)	>>> lst = [557, 40, «принтер», 77]
	>>> lst [2]
	
	'принтер'
		# Виведення другого елемента

Значення елементів списку можна змінювати шляхом присвоювання їм нових значень	>>> lst = [44, 33, 22, 11]
	>>> lst[2] = 55
	>>> lst
	[44, 33, 55, 11]
		# Зміна значення другого елемента
		# Виведення списку

Значення можна одночасно присвоїти декільком спискам	>>> lst1, lst2, lst3 = [6, 12, 9]
	
	>>> lst1, lst2, lst3
	([6, 12, 9], [6, 12, 9], [6, 12, 9])
		# Значення 6, 12, 9 присвоюються
		# 3 спискам
		# Виведення значень списків
		# (значення списків однакові)

Операція зрізу дозволяє отримати певний діапазон значень, він має формат: <ім'я списку>[початок: кінець: крок]. Усі параметри є необов'язковими	>>> lst = [1, 2, 3, 4, 5, 6]
	>>> lst[1 : 4]
	
	[2, 3, 4]
		# Виділення елементів з 1 до 3 включно

Об'єднання списків виконується за допомогою операції конкатенації	>>> lst1 = [11, 12, 13, 14]
	>>> list2 = [55, 56, 57]
	>>> lst1+list2
	[11, 12, 13, 14, 55, 56, 57]
		# Об'єднання списків

Наведемо найуживаніші функції опрацювання списків:

len() — визначає довжину списку	>>> lst = [2, 34, 16, 19]
	>>> len(lst)
	4
		# Довжина списку

sample() — призначена для отримання списку з випадковими числами у заданому діапазоні. Вона міститься у модулі random і має таку структуру: sample(range(діапазон), кількість елементів)	>>> import random
	>>> random.sample
	(range(60), 6)
	[32, 44, 5, 33, 17, 2]
		# Імпортування модуля random
		# Шість випадкових чисел
		# у діапазоні 0–60

max() і min() — призначені відповідно для визначення максимального і мінімального значень у списку	>>> lst = [22, 16, 3, 19, 35, 26]
	>>> max(lst), min(lst)
	
	(35, 3)
		# Виведення максимального
		# і мінімального елементів

choice() — міститься у модулі random і слугує для вибору випадкового елемента зі списку	>>> import random >>>random.choice([54, 2.25, 19, 54.3, 7.6, 25]) ‘19’	# Імпортування модуля random # Вибір випадкового елемента
list() — перетворює рядок у список	>>> list("підручник") [‘п’, ‘і’, ‘д’, ‘р’, ‘у’, ‘ч’, ‘н’, ‘и’, ‘к’]	# Перетворення рядка у список

Розглянемо основні методи опрацювання списків на наведених прикладах:

append (об’єкт) — додає один об’єкт у кінець списку	>>> lst = [77, "abc", 66]
	>>> lst.append("sbc")
	
	>>> lst
	[77, ‘abc’, 66, ‘sbc’]
		# Список
		# Додавання елемента sbc у кінець
		# списку
		# Виведення списку

extend() — додає кілька елементів у кінець списку	>>> lst = [65, "abc", 88]
	>>> lst.extend(["bas", 99])
	>>> lst
	[65, ‘abc’, 88, ‘bas’, 99]
		# Список
		# Додавання списку у список
		# Виведення списку

Insert (індекс, об’єкт) — додає один об’єкт у вказану позицію списку	>>> lst = [3, "bit", 5]
	>>> lst.insert(1, "file")
	
	>>> lst
	[3, ‘file’, ‘bit’, 5]
		# Список
		# Додавання значення file у 1-шу позицію
		# списку
		# Виведення списку

pop(індекс) — видаляє елемент зі списку за вказаним індексом. Видалити елемент зі списку можна також за допомогою оператора del[індекс]	>>> lst = [17, 9, "and", 55]
	>>> lst.pop(1)
	9
	>>> lst
	[17, ‘and’, 55]
		# Список
		# Виведення видаленого елемента
		
		# Виведення списку після видалення
		# елемента

remove(значення) — видаляє зі списку перший елемент, який містить вказане значення	>>> lst = [13, 5, "paint", 7, "paint", 66]
	>>> lst.remove("paint"); lst
	[13, 5, 7, 'paint', 66]
		
		# Видалення елемента paint
		# із другої позиції

clear() — видаляє зі списку усі елементи		

index(значення [, початок [, кінець]]) — повертає індекс елемента, який має вказане значення.
За замовчуванням параметрів початок і кінець пошук елемента буде виконуватися від початку до кінця списку
	>>> lst = [23, 71, 15, 19, 8, 50]
	>>> lst.index(19)
	
	3
		
		# Виведення індексу
		# елемента зі значенням 19

count (значення) — призначений для визначення кількості елементів із вказаним значенням. Якщо елемент відсутній у списку, повертається значення 0	>>> lst = [2, 5, 5, 7, 13, 5, 6]
	>>> lst.count(5)
	
	3
		
		# Виведення кількості
		# елементів із значенням 5

sort() — слугує для сортування елементів списку. Він має таку загальну структуру: sort([key = None][, reverse = False]).
Як бачимо, параметри є необов'язковими. За замовчуванням сортування виконується за зростанням значень елементів з урахуванням регістра. Для сортування за спаданням слід вказати другий параметр таким: reverse = True. Відзначимо, що метод sort() перетворює старий список у новий
	>>> lst = [7, 99, 20, 13, 88, 26, 25]
	>>> lst.sort(reverse = True)
	
	>>> lst
	[99, 88, 26, 25, 20, 13, 7]
		# Список
		
		# Сортування у порядку
		# зменшення значень
		# Виведення
		# упорядкованого списку

sorted(список[, reverse = False]) — призначений для сортування списку і збереження старого	>>> lst = [65, 39, 19, 22, 7, 15]
	>>> sorted(lst)
	[7, 15, 19, 22, 39, 65]
	>>> lst
	[65, 39, 19, 22, 7, 15]
		# Початковий список
		# Повернення
		# упорядкованого списку
		# Повернення початкового
		# списку

Особливості багатовимірних списків. У багатовимірних списках кожна група елементів списку береться у квадратні дужки, а весь список також міститься у квадратних дужках. У прикладі 3 наведено, як можна створити двомірний список.

Для наочності краще записувати списки по рядках (приклад 4).

Для звернення до елемента багатовимірного списку потрібно вказати у квадратних дужках усі його індекси. Так, звернутися до першого елемента першої групи двовимірного списку можна так:

```
>>> lst[1][1]
'мова'
```

Звертатися до усіх або частини елементів списку можна за допомогою операторів циклу for і while, а також функції range(), які розглядаються далі.

Приклад 3.

```
>>> lst = [[3, 2, 1], [13, "мова", 20], [7, 6, 5]]
```

Приклад 4.

```
>>> lst = [
    [3, 2, 1],
    [13, "мова", 20],
    [7, 6, 5]]
>>> lst
[[3, 2, 1], [13, 'мова', 20], [7, 6, 5]]
```

Приклад 5.

Дано список [17, 44, 5, 21, 22, 38, 9]. Розробити код, за допомогою якого вилучається зі списку елемент із максимальним значенням, після цього на другу позицію вставляється число 19, упорядковується новий список у порядку зростання значень його елементів і виводиться на екран.

Код реалізації алгоритму наведено на рис. 10.5.

```
lst = [17, 44, 5, 21, 22, 38, 9]          #Створення списку
a = max (lst)                             #Визначення максимального елемента
print ('Максимальний елемент = ', a)      #Виведення максимального елемента
lst.remove (a)                            #Видалення максимального елемента
print ('Список після видалення ', lst)    #Виведення списку
lst.insert (2,19)                         #Вставлення у список елемента
print ('Список після вставлення ', lst)   #Виведення списку
lst.sort ()                               #Упорядкування списку
print ('Упорядкований список ', lst)      #Виведення списку
```

Рис. 10.5. Код опрацювання списку

Результат виконання коду:

```
Максимальний елемент =  44
Список після видалення  [17, 5, 21, 22, 38, 9]
Список після вставлення  [17, 5, 19, 21, 22, 38, 9]
Упорядкований список  [5, 9, 17, 19, 21, 22, 38]
```

Приклад 6.

Розробити код, за допомогою якого створюється список із шести випадкових цілих чисел у діапазоні від 0 до 30, визначається мінімальне значення і видаляється зі списку. На місце мінімального значення вставляється число 777, а у кінець списку додається число 999. Після цього список упорядковується і виводиться на екран.

Код реалізації наведено на рис. 10.6.

```
import random                                  #Імпортування модуля random
lst = random.sample (range (30),6)             #Створення списку випадкових чисел
print ('Список випадкових чисел ',lst)         #Виведення списку
a = min (lst)                                  #Мінімальне значення списку
print ('Мінімальне значення ',a)               #Виведення мінімального значення
i = lst.index (a)                              #Індекс мінімального значення
print ('Індекс мінімального елемента', i)      #Виведення індексу
lst.pop (i)                                    #Видалення мінімального значення
print ('Список після видалення',lst)           #Виведення списку
lst.insert (i, 777)                            #Вставлення числа в і-ту позицію
print ('Список після вставлення', lst)         #Виведення списку
lst.append (999)                               #Додавання у кінець списку 999
print ('Список після додавання ', lst)         #Виведення списку
lst.sort ()                                    #Упорядкування списку
print ('Упорядкований список ',lst)            #Виведення списку
```

Рис. 10.6. Код створення й опрацювання списку

Результат виконання коду:

```
Список випадкових чисел  [12, 23, 25, 17, 7, 11]
Мінімальне значення  7
Індекс мінімального елемента 4
Список після видалення [12, 23, 25, 17, 11]
Список після вставлення [12, 23, 25, 17, 777, 11]
Список після додавання  [12, 23, 25, 17, 777, 11, 999]
Упорядкований список  [11, 12, 17, 23, 25, 777, 999]
```

Запитання для перевірки знань

1. Що називається списком?
2. Як можна замінити значення елемента списку?
3. Як здійснюється об'єднання списків?
4. Для чого слугують методи append(); remove()?
5. Поясніть сутність методу insert().
6. За допомогою яких методів виконується сортування елементів списку?
7. Поясніть порядок створення списку випадкових чисел.

Завдання для самостійного виконання

1. Знайдіть в інтерактивному режимі мінімальне і максимальне значення у списку [45, 23.5, 76, 29, 41].
2. В інтерактивному режимі у списку ["and", 44, "not", 77, 23] вилучіть другий елемент і на його місце вставте елемент "or".
3. Дано список [19, 41, 25, 41, 5, 41, 41]. Розробіть код визначення в ньому кількості чисел, рівних 41, і упорядкування списку за спаданням його значень.
4. Розробіть код, за допомогою якого виконується сортування списку [57, 29, 3, 87, 37, 46, 69] зі збереженням початкового списку.
5. Розробіть код, за допомогою якого створюється список із семи випадкових чисел в діапазоні від 5 до 15, відшукуються максимальне і мінімальне значення, вилучаються зі списку, після чого список упорядковується і виводиться на екран.
6. Розробіть код, за допомогою якого у списку [13, 29, 88, 7, 65, 22, 55, 123] визначається індекс числа 88 і на його місце вставляється число 111, після чого у кінець списку додається число 11.

10.3.2. Словники

Пригадайте, якими словниками ви користувалися. Для чого вони призначені?

Словник — це набір об'єктів будь-якого типу, доступ до яких здійснюється не за допомогою індексу, а за допомогою ключа. Ключем можуть бути числа, рядки або кортежі. Значення елементів словника можна змінювати.

Словники можна створювати у різний спосіб. Найпростіший (приклад 1) — шляхом перелічення у фігурних дужках ключів і значень за структурою:

{ключ: значення, ключ: значення, …, ключ: значення}

Словник можна створити за допомогою функції dict() (приклади 2, 3), один із форматів якої є таким:

dict(ключ1 = значення1, …, ключN = значенняN)

Приклад 1.

```
>>> d = {"a1": 21, "a2": 33}
>>> d
{'a2': 33, 'a1': 21}
```

Словники не є послідовностями, тому такі загальні операції, як отримання зрізу, конкатенації та інші, вони не підтримують.

Приклад 2.

```
>>> d = dict(a1 = 7, a2 = 3, a3 = 8)      # Створення словника
>>> d                                     # Виведення словника
{'a2': 3, 'a3': 8, 'a1': 7}
```

Приклад 3.

```
>>> d = dict({"a1": 15, "a2": 17, "a3": 18}) # Створення словника
>>> d                                        # Виведення словника
{'a2': 17, 'a3': 18, 'a1': 15}
```

Значення елементів словника можна змінювати.

Для звернення до елемента словника потрібно зазначити у квадратних дужках його ключ (приклад 4). Якщо елемент із ключем у словнику відсутній, то генерується виняток KeyError.

Приклад 4.

```
>>> d = {1: "list", "ab": "random", (1, 4): "randint"}      # Словник
>>> d[1], d["ab"], d[(1, 4)]                                # Звернення
                                                            # до елементів словника
('list', 'random', 'randint')
```



Швейцарець Ніклаус Вірт — жива легенда в світі програмування. Це блискучий інженер і глибокий дослідник, який у 1984 році був удостоєний премії Тюрінга.

Наявність ключа у словнику можна перевірити за допомогою оператора in (приклад 5). Якщо ключ у словнику є, генерується значення True, інакше — False.

Приклад 5.

```
>>> d = {1: "file", "bsa": 5}      # Словник
>>>"bsa" in d                     # Перевірка наявності у словнику
                                  # ключа bsa
True
```

Змінити значення елемента словника або додати новий елемент можна за допомогою його ключа (приклад 6).

Приклад 6.

```
>>> d = {"ab": 25, "c": 100}       # Словник d
>>> d["ab"] = 200                  # Зміна значення, що має ключ "ab"
>>> d[2] = "import"                # Додавання нового елемента
>>> d                              # Виведення нового словника
{2: 'import', 'c': 100, 'ab': 200}
```

Кількість ключів у словнику визначається за допомогою функції len().

Видалити елемент зі словника можна за допомогою оператора del, вказавши його ключ у квадратних дужках (приклад 7).

Приклад 7.

```
>>> d = {"abs": "монітор", "dok": "принтер", "sba"": 25} # Словник d
>>> del d["dok"]          # Видалення елемента з ключем dok
>>> d                     # Виведення нового словника
{'abs': 'монітор', 'sba': 25}
```



Мову Python відрізняє швидкість і простота скриптів. Разом із набором доступних бібліотек для роботи з мережами і файлами це робить її незамінним помічником системного адміністратора.

Розглянемо основні методи роботи зі словниками:

get(ключ) — повертає значення, яке відповідає ключу. Якщо ключ відсутній, то повертається значення None	>>> d = {"p": 50, 60: "байт", "sk": "біт"}
	>>> d.get("p"), d.get("sk"), d.get(70)
	(50, "біт", None)
		# Словник
		
		# Повернення значень
		# словника, елемента з ключем
		# 70 немає

keys() — дозволяє отримати всі ключі словника. Можна використати для виведення всіх елементів словника з ключами за допомогою оператора циклу for. Елементи словника виводяться у довільному порядку, оскільки словники є неупорядкованими структурами. Якщо потрібно вивести елементи словника упорядкованими за значенням ключа, то слід скористатися функцією sorted()	>>> d = {"ab": 5, "ba": 9, "cd": 11}
	>>> d.keys()
	dict_keys(['ba', 'cd', 'ab'])	

update() — додає елементи у словник. Одна з найпростіших структур методу така: update(словник)	>>> d = {1: "a", 2: "b"}
	>>> d.update({3: "c", 4: "d"})
	
	>>> d
	{1: 'a', 2: 'b', 3: 'c', 4: 'd'}
		# Словник d
		# Додавання у словник нових
		# елементів
		# Виведення нового словника

values() — повертає об'єкт dict_values, який містить усі значення словника	>>> d = {"a": 5, "b": 6}
	>>> d.values()
	dict_values([6, 5])
	>>> list(d.valyes())
	[6, 5]
		
		# Об'єкт dict_values()
		# Отримано список значень

pop(ключ[, значення за умовчанням]) — вилучає зі словника елемент з указаним ключем і повертає його значення. Якщо ключ відсутній, то повертається значення другого параметра, а якщо відсутній ключ і не вказаний другий параметр, то генерується виняток KeyError	>>> d = {"a": 5, "b": 6, "c": 7}
	>>> d.pop("b")
	6
	>>> d
	{'c': 7, 'a': 5}
		# Словник d
		# Видалення елементів
		# з указаним ключем
		# Виведення нового словника

Далі на підтримку викладеного розглянемо приклади 8 і 9.

Приклад 8.

Створити словник, у якому ключами є прізвища видатних українських спортсменів і спортсменок: Яна Клочкова, Сергій Бубка, Андрій Шевченко, а їх значеннями є рік народження спортсмена і вид спорту відповідно: 1982, плавчиня; 1963, легкоатлет; 1976, футболіст. Визначити рік народження і вид спорту Сергія Бубки та Яни Клочкової, перевірити наявність у словнику прізвищ Андрія Шевченка і Руслана Пономарьова. Вивести дані Андрія Шевченка.

Код реалізації наведено на рис. 10.7.

```
d = {"Яна Клочкова":"1982, плавчиха", "Сергій Бубка":"1963, легкоатлет",\
     "Андрій Шевченко":"1976, футболіст"}
print ('Дані Сергія Бубки: ', d["Сергій Бубка"])
print ('Дані Яни Клочкової: ', d["Яна Клочкова"])
print ('Чи є прізвище Андрія Шевченка ',"Андрій Шевченка" in d)
print ('Чи є прізвище Руслана Помарьова ', "Руслан Пономарьов" in d)
print ('Дані Андрія Шевченка: ', d.get ('Андрій Шевченко'))
```

Рис. 10.7. Код створення й опрацювання словника

Результат виконання програми:

```
Дані Сергія Бубки:  1963, легкоатлет
Дані Яни Клочкової:  1982, плавчиха
Чи є прізвище Андрія Шевченка  True
Чи є прізвище Руслана Помарьова  False
Дані Андрія Шевченка:  1976, футболіст
```

Приклад 9.

За розкладом у неділю з аеропорта «Бориспіль» вилітають рейси: 1010 до Лондона, 1030 до Мюнхена, 525 до Афін, 1035 до Мюнхена, 673 до Лондона. Однак рейс 673 скасовано, а оголошено додатковий рейс 1075 до Будапешта. Розробити код, за допомогою якого виводиться змінений розклад рейсів, загальна кількість рейсів, список міст, до яких вилітають авіарейси, і кількість рейсів до Мюнхена.

Код реалізації наведено на рис. 10.8.

```
d = {1010:'Лондон', 1030:'Мюнхен', 525:'Афіни',\
     1035:'Мюнхен', 673:'Лондон'}                    #Створення словника
del d [673]                                          #Видалення рейсу 673
d.update ({1075:"Будапешт"})                         #Додавання рейсу 1075
print ('Номери рейсів до міст :',)                   #Повідомлення
print (d)                                            #Виведення рейсів і міст
print ('Кількість рейсів у неділю',len (d))          #Усього рейсів у неділю
dl = list (d.values ())                              #Виділення міст
print ('Рейси до міст ',dl)                          #Виведення міст
print ('Рейсів до Мюнхена ',dl.count ("Мюнхен"))     #Рейсів до Мюнхена
```

Рис. 10.8. Код створення словника і його опрацювання

Результат виконання програми:

```
Номери рейсів до міст :
{1010: 'Лондон', 1075: 'Будапешт', 1030: 'Мюнхен', 1035: 'Мюнхен', 525: 'Афіни'}
Кількість рейсів у неділю 5
Рейси до міст  ['Лондон', 'Будапешт', 'Мюнхен', 'Мюнхен', 'Афіни']
Рейсів до Мюнхена  2
```

Запитання для перевірки знань

1. Як оголошуються словники?
2. Як можна перевірити наявність ключа у словнику?
3. Як можна видалити елемент зі словника?
4. Для чого призначений метод **get()**?
5. Яку функцію виконує метод **keys()**?
6. Для чого призначений метод **update()**?
7. Поясніть сутність методу **values()**.

Завдання для самостійного виконання

1. В інтерактивному режимі створіть словник, елементами якого є: "Костенко": "зріст 173", "Варламов": "зріст 169", "Берзко": "зріст 165", "Петренко": "зріст 170". Визначте дані про Петренка і кількість елементів у словнику.
2. Микола зібрав 3,4 кг грибів, Настя — 4 кг, Тетяна — 3,3 кг, Петро — 5,2 кг. Розробіть код створення словника, елементами якого є наведені дані. Визначте, чи збирав гриби Іван; кількість грибів, зібраних Настею; усі імена учнів і учениць, які збирали гриби.
3. Розробіть код створення словника, ключами і значеннями якого є назви комунальних послуг та їх ціни: споживання газу — 51 грн, опалення — 1350 грн, електроенергія — 152 грн, користування телефоном — 112 грн. До створеного словника додайте послугу: холодна вода — 63 грн. Визначте ціну за електроенергію і телефон. Обчисліть загальну суму послуг.
4. Розробіть код створення словника, елементами якого є імена та прізвища видатних українських лікарів і лікарок і роки їхнього життя: Микола Амосов (1913–2002), Софія Окуневська-Морачевська (1865–1926), Олександр Богомолець (1881–1946), Володимир Філатов (1875–1956). Додайте у список: Микола Скліфосовський (1836–1904). Перевірте наявність у словнику Володимира Філатова, визначте роки життя Олександра Богомольця.
5. У шкільній бібліотеці зберігаються підручники з різних шкільних предметів. Їх кількість така: з фізики — 95, біології — 40, історії України — 60, математики — 70. Розробіть код створення словника з цих даних. Видаліть зі словника дані про підручник із біології, додайте підручник з інформатики — 60. Визначте наявність підручників з історії України та хімії.

10.4. Використання текстових файлів для введення тавиведення даних

Під час виконання програмного коду ви вводили дані за допомогою клавіатури і виводили на екран монітора. Чому, на вашу думку, інколи виникає потреба використовувати ще й текстові файли?

Порядок використання файлів такий: файл необхідно відкрити, прочитати дані з файлу або записати дані у файл, закрити файл.

Мова Python підтримує введення-виведення даних у текстові файли і двійкові файли (графічні, аудіо- й відеофайли).

Файли відкриваються за допомогою функції open().

Загальна структура функції open():

open(<'шлях до файлу»>, [<mode>][, вид кодування])

Параметр mode може мати такі значення:

r (Read)	Відкриття файлу для читання (діє за замовчуванням)
w (Write)	Відкриття файлу для запису
a (Append)	Відкриття файлу для дозапису (дані додаються у кінець файлу)
b	Відкриття файлу у двійковому режимі
t	Відкриття файлу у текстовому режимі (діє за замовчуванням)
+	Відкриття файлу для читання й запису

Режими відкриття можуть об'єднуватися: rb — читання у двійковому режимі, rt — читання у текстовому режимі. За замовчуванням діє режим rt.

Параметр вид кодування вказує назву коду, у якому здійснюється робота. Для цього зазначається: encoding = 'код', наприклад encoding = 'UFT-8'.

Для роботи з файлами у мові Python існують основні методи close() — закриває файл та write() — записує дані у файл. Аргументом методу write() є символьний рядок, який записується у файл (приклад 2).

Приклад 1.
Відкрити текстовий файл, який зберігається, приміром, на робочому столі, можна в режимі читання за допомогою команди:

```
>>> f = open('C:\\Users\\Віктор\\Desktop\\file_1.txt')
```

Подвійний слеш тут застосовано тому, що кодування здійснено в Unicode.

Приклад 2.

```
f.write('монітор'\n)
f.write('операційна система»)
```

У результаті виконання цих команд будуть записані два символьні рядки.

Приклад 3. У файл file_5.txt записати рядок "keys, dict, union" і зберегти його на робочому столі. Динаміку процесу відкриття файлу та запису в нього рядка зображено на рис. 10.9.

```
>>> f = open ('C:\\Users\\Віктор\\Desktop\\file_5.txt', 'w')
>>> f.write ("keys, dict, union")
17
>>> f.close ()
```

Рис. 10.9. Динаміка процесу відкриття файлу та запису в нього рядка

Звернемо увагу на те, що після запису у файл рядка автоматично видається кількість символів у рядку. Знайдемо на робочому столі файл file_5 і відкриємо його.

Вміст файлу показано на рис. 10.10.

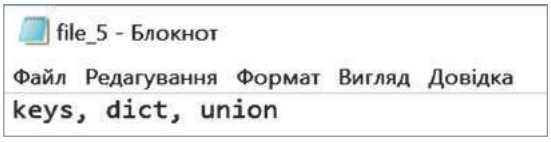


Рис. 10.10. Вміст файлу file_5.txt

Існує також низка інших методів.

- Метод writelines() — отримує послідовність символьних рядків, наприклад списку, кортежу та інших, і записує усі елементи послідовності у файл (приклад 4).

Приклад 4. Дано список: 'max', 'import', 'remove'. Послідовність дій запису списку у файл file.txt на робочому столі зображено на рис. 10.11.

```
>>> lst = ['max', 'import', 'remove']
>>> f = open ('C:\\Users\\Віктор\\Desktop\\file.txt', 'w')
>>> f.writelines (lst)
>>> f.close ()
```

Рис. 10.11. Послідовність дій запису списку у файл

Створений файл наведено на рис. 10.12.

Звернемо увагу на те, що хоча у списку містяться три рядки, у файлі вони подані одним рядком.



Рис. 10.12. Файл, створений за допомогою методу writelines()

- Метод read() — зчитує всі дані з текстового файлу і повертає їх як один символьний рядок (приклад 5).

Приклад 5. Створимо текстовий файл file_1.txt такого вмісту:

1 2 3 4 5
7 8 9 10 11
12 13 14 15

і збережімо його на робочому столі. Динаміку процесу відкриття й читання цього файлу зображено на рис. 10.13.

```
>>> f = open ('C:\\Users\\Віктор\\Desktop\\file_1.txt', 'r')
>>> a = f.read ()
>>> print (a)          #Виведення файлу
1 2 3 4 5
7 8 9 10 11
12 13 14 15
>>> f.close ()         #Закриття файлу
```

Рис. 10.13. Динаміка процесу відкриття й читання текстового файлу

- Метод readline() під час кожного виклику зчитує з файлу черговий рядок і повертає його як символьний рядок.

Окремі рядки тексту у файлі мають розділятися символом '\n', він увійде до результату, що повертає readline() (приклад 6).

- Метод readlines() зчитує дані з файлу, розділяє їх на окремі символьні рядки й повертає список із цих рядків (символ '\n' присутній у кожному символьному рядку).

Приклад 6. Створимо, наприклад, текстовий файл file_3.txt, зображений на рис. 10.14, і збережемо його на робочому столі.

Динаміку процесу читання файлу показано на рис. 10.15.

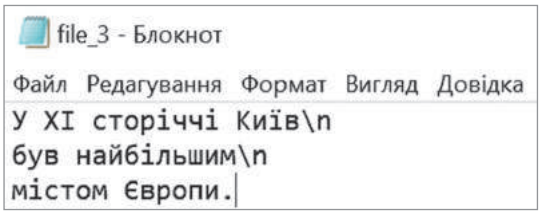


Рис. 10.14. Текстовий файл file_3.txt

```
>>> f = open
('C:\\Users\\file_3.txt', 'r')
>>> a = f.readline ()
>>> print (a)
У XI сторіччі Київ\n

>>> a = f.readline ()
>>> print (a)
був найбільшим\n

>>> a = f.readline ()
>>> print (a)
містом Європи.
```

Рис. 10.15. Динаміка процесу читання файлу file_3.txt

Запитання для перевірки знань

1. На які типи поділяються файли для запису і читання даних?
2. За допомогою якої функції відкривається файл?
3. Які існують режими читання й запису даних у файл?
4. За допомогою яких методів можна читати дані з файлу?
5. За допомогою яких методів можна записувати дані у файл?
6. Наведіть приклад відкриття файлу для запису й читання.

Завдання для самостійного виконання

1. Створіть текстовий файл **file1**, який містить цифри 0, 2, 4, …, 14, і збережіть його на робочому столі. Прочитайте цей файл за допомогою методу **read()**.
2. Запишіть у текстовий файл **file2** за допомогою методу **write()** речення: «Побутує думка, що для українців сало є своєрідним символом національної культури». Збережіть його на робочому столі.
3. Дано послідовність рядків 'біт', 'файл', 'байт'. Використайте метод **writelines()** для їх запису у файл **file3.txt**. Файл збережіть.
4. Створіть текстовий файл **file4.txt** «П'ять українських об'єктів внесені до переліку Світової спадщини ЮНЕСКО» і збережіть його у будь-якій папці. Прочитайте файл за допомогою методу **readline()**.
5. Створіть текстовий файл **file5.txt** і запишіть речення: «Через територію України проходить чотири з десяти європейських транспортних коридорів», збережіть його у будь-якій папці. Прочитайте файл за допомогою методу **readlines()**.

Практична робота № 19

ТЕМА. Розроблення лінійних алгоритмів та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити лінійні алгоритми та програми їх реалізації, виконати налагодження програм.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Розробіть блок-схему алгоритму обчислення y:

$$y = \frac{2\sqrt{3a} + 3{,}5\left(a^3 - c\right)}{1{,}6a + \sqrt{bc}}.$$

2. Розробіть код обчислення значення виразу за умови, що значення змінних a, b, c вводяться з клавіатури.
3. Уведіть код в комп'ютер, знайдіть і усуньте помилки введення. Збережіть файл коду.
4. Виконайте код для значень змінних: $a = 4.2$; $b = 3.5$; $c = 2$.
5. Усуньте синтаксичні помилки в коді, якщо вони є.
6. Виконайте код і доведіть, що він функціонує правильно.
7. Виконайте код для різних значень змінних a, b, c.

7. Переконайтеся, що отримані результати є правильними.

8. Розробіть блок-схему алгоритму обчислення площ рівностороннього трикутника зі стороною a, площі вписаного у трикутник кола і їх різниці.
9. Розробіть код реалізації алгоритму, уведіть у комп'ютер і збережіть файл коду.
10. Виконайте налагодження коду.

    Виконайте код для різних значень сторони трикутника і переконайтеся, що результати є правильними.

10. Доведіть, що код виконується правильно.

11. Із Києва у Харків виїхав автобус і рухався без зупинок із середньою швидкістю 45 км/год. Одночасно із Харкова у Київ виїхав мотоцикліст і рухався без зупинок із середньою швидкістю 55 км/год. Через 4 год 50 хв вони зустрілися.

    Розробіть блок-схему алгоритму визначення відстані між Києвом і Харковом.

12. Розробіть код реалізації алгоритму. Уведіть і збережіть код у файлі. Виконайте код.

    Зробіть висновок, чому необхідно здійснювати виконання коду для різних значень вхідних даних.

12. Доведіть, що код виконується правильно.'

Практична робота № 20



ТЕМА. Розробка лінійних алгоритмів та їх реалізація у вигляді програм з використанням текстових файлів

ЗАВДАННЯ: розробити лінійні алгоритми та програми їх реалізації, виконати налагодження програм.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Розробіть у словесно-формульній формі алгоритм обчислення значення y:

$$y = \frac{32{,}425\sqrt{3{,}1a^2 + 2{,}25b} - 8{,}2ab}{9{,}1c\left(\sqrt{a + b^2 - 2{,}1ac}\right)}.$$

Значення змінних a, b, c уводяться з клавіатури.

2. Створіть код реалізації алгоритму. Уведіть код, виконайте його налагодження.

2. Доведіть, що результат виконання є правильним.

3. Запишіть отриманий результат у файл file_10 власної папки.

3. Переконайтеся, що результат збережено правильно.

4. Створіть текстовий файл, що містить речення: «В Україні 68 % населення міське, 32 % — сільське. Середня густота населення — 81 житель на квадратний кілометр».

Збережіть набраний текст у файлі file_11 власної папки.

5. Відкрийте і прочитайте файл file_11 за допомогою методів open() і read().

5. Переконайтеся, що файл прочитано правильно.

6. Уведіть текст прислів'я: «Людина змінюється у житті, а насправді життя змінює людину». Збережіть прислів'я у текстовому файлі file_12 власної папки.

7. Прочитайте створений файл за допомогою методу readlines(). Переконайтеся, що файл прочитано правильно. Закрийте файл.

8. Розробіть блок-схему алгоритму обчислення значення y:

$$y = 54{,}6a^3 + \frac{43\sqrt{7{,}3a + b^2}}{2{,}1\left(3a - 1{,}6b + \sqrt{c}\right)} - 0{,}65\left(a^2 + 2\left(a - c\right)\right).$$

Значення a, b, c уведіть з клавіатури.

9. Створіть код реалізації алгоритму.

10. Уведіть код і виконайте його налагодження.

10. Доведіть, що код виконується правильно.

11. Запишіть результат у файлі file_13 власної папки.

12. Переконайтеся, що дані у файлі збережено правильно.

Зробіть висновок про доцільність використання текстових файлів для запису й читання з них даних.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 10

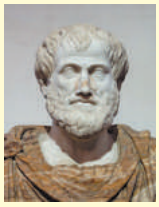
11. Елементи алгебри логіки

11.1. Висловлювання. Логічні змінні, операції і вирази

Пригадайте, з якими логічними змінними й логічними операціями ви ознайомилися у шостому і сьомому класах.

Висловлювання є одним із фундаментальних понять логіки — науки про закони мислення. У логіці застосовується термін «міркування», який означає особливу форму мислення людини. Темін «міркування» частіше використовують у традиційній логіці, а термін «висловлювання» — у математичній логіці.

Висловлювання — це речення, про яке можна говорити, що воно істинне або хибне. Лише розповідне речення може бути висловлюванням.



Основоположником логіки як науки є давньогрецький учений і філософ Аристотель (384–322 рр. до н. е.)

Не кожне речення може бути висловлюванням. Висловлюваннями не можуть бути оголошення, накази, поради та ін. Так, речення «Середня врожайність зернових у районі становить 30 ц/га» не є висловлюванням.

Висловлювання поділяються на прості і складні. Прості — це висловлювання, що не містять іншого іисловлювання; їх ще називають атомарними. Складні висловлювання утворюються з кількох простих за допомогою логічних сполучників заперечення, кон'юнкції, диз'юнкції та ін. (приклад 1).



Вагомий внесок у розвиток теорії логіки належить німецькому математику Готфріду Вільгельму Лейбніцу (1646–1716).

Приклад 1.

Просте висловлювання: «Два більше за одиницю».

Складене висловлювання: «Якщо a більше b і b більше c, то a більше c». Із прикладу випливає, що істинність складеного висловлювання визначається на основі істинності й хибності простих висловлювань.

Результати виконання логічних операцій заперечення, диз'юнкції і кон'юнкції над висловлюваннями:

Висловлювання		Заперечення	Кон'юнкція	Диз'юнкція
A	B	¬A	A&B	A∨B
Хибне	Хибне	Істинне	Хибне	Хибне
Хибне	Істинне	Істинне	Хибне	Істинне
Істинне	Хибне	Хибне	Хибне	Істинне
Істинне	Істинне	Хибне	Істинне	Істинне



Систему математичної логіки на базі логіко-математичної мови, яку називають алгеброю логіки, розробив британський математик і філософ Джордж Буль.

Алгебра логіки оперує логічними змінними й логічними функціями, які мають значення істинно або хибно. Ці значення ототожнюються відповідно до значень одиниці й нуля. Логічні функції можуть бути задані таблицями істинності або логічними формулами. Наприклад, логічними формулами трьох аргументів можуть бути такі:

$$f(x, y, z) = xy \vee yz;$$
$$f(x, y, z) = (x \vee y)(y \vee z).$$

Висловлювання та його заперечення не можуть бути одночасно істинними або одночасно хибними.

Приклад 3. Нехай маємо вираз ((x > y) or (y > z)) and not((x > 1) or (z > 1)).

Визначення значення виразу для значень змінних x = 1, y = 3, z = 2 пояснено на рис. 11.1.

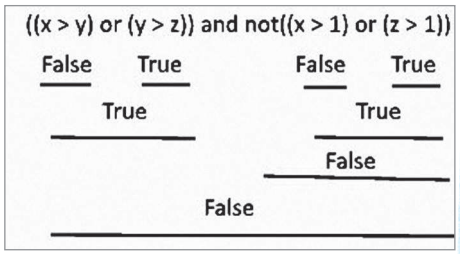


Рис. 11.1. Аналіз значення логічного виразу

Заперечення виконується над одним висловлюванням, яким у таблиці є висловлювання *A*. Заперечення має значення хибне, якщо значення *A* істинне, і навпаки.

Результатом операції кон'юнкції є значення істинне тоді і тільки тоді, коли обидва висловлювання істинні.

Результатом операції диз'юнкції є значення хибне тоді й тільки тоді, коли обидва висловлювання хибні.

Математична логіка — різновид загальної логіки, що вивчає закони математичного мислення, предметом її вивчення є математичні теорії у цілому. Математична логіка є формальною логікою, що вивчає процеси мислення (поняття, умовиводи, доведення, судження) з точки зору логічної структури, абстрагуючись від конкретного змісту.

У мові Python над логічними даними виконуються такі операції: заперечення, що позначається not (ні), диз'юнкції — or (або) і кон'юнкції — and (і).

Результати виконання логічних операцій наведено в таблиці:

x	y	not x	x or y	x and y
False	False	True	False	False
True	False	False	True	False
False	True	True	True	False
True	True	False	True	True

Із таблиці видно, що оператор not є унарним, тобто діє лише для одного операнда. Результатом його виконання є значення, протилежне значенню операнда x.

Результатом виконання оператора or є значення False лише в тому випадку, коли обидва операнди мають значення False.

Результатом виконання оператора and є значення True лише у випадку, коли обидва операнди мають значення True.

Логічні вирази складаються із змінних, операцій порівняння й логічних операцій.

Логічні вирази можуть мати круглі дужки, а їх результатом є True або False.

Розглянемо приклади 2 і 3.

Приклад 2. Логічний вираз x > 3 набуває значення True, якщо x набуває значення більше або дорівнює трьом, інакше — False.

Логічний вираз (x > 2) and (y = 7) набуває значення True, якщо x більше за 2 і y дорівнює 7, інакше — False.

Логічний вираз (x < 5) or (y > 3) набуває значення True, якщо x менше за 5 або y більше від 3, інакше — False.

Запитання для перевірки знань

1. Поясніть сутність висловлювання.
2. Яких значень набувають висловлювання?
3. Які типи речень не можуть бути висловлюваннями?
4. Яких значень набуває диз'юнкція над двома висловлюваннями?
5. Яких значень набуває кон'юнкція над двома висловлюваннями?
6. Наведіть приклад простого і складеного висловлювань.

Завдання для самостійного виконання

1. В інтерактивному режимі для x = 5 і x = 2 визначте значення виразу x > 3.
2. В інтерактивному режимі для x = 0.5 і y = 3 визначте значення виразу (x > 1) or (y == 2).
3. В інтерактивному режимі для x = 2 і y = 1 визначте значення виразу (x > y) and (x > 1).
4. Розробіть код обчислення значення виразу (x < y) or (x == z). Перевірте правильність виконання коду для різних значень змінних.
5. Розробіть код обчислення значення логічного виразу (x > y) and not (x < z). Перевірте правильність виконання коду для різних типів даних.
6. Розробіть код обчислення значення логічного виразу ((x > y) or (x = z)) and (y == z). Перевірте правильність виконання коду для різних значень змінних.
   Доведіть, що код виконується правильно.

11.2. Розроблення і обчислення значень логічних виразів

Спробуйте сформулювати речення, яке містить такі слова: якщо, то, інакше.

Розроблення й обчислення логічних виразів здійснюється на основі законів алгебри логіки.

Для булевої алгебри справедливими є такі рівняння:

$x \vee 0 = x$ $\quad x \cdot 0 = 0$ $\quad x \cdot \overline{x} = 0$

$x \vee 1 = 1$ $\quad x \cdot 1 = x$ $\quad x \vee \overline{x} = 1$

Розглянемо порядок обчислення значень логічних виразів на прикладах 1 і 2.

Приклад 1. Іван, Петро і Соломія спрогнозували результат Чемпіонату України з футболу у поточному році таким чином:

Іван: першими будуть «Карпати», другим — «Динамо».

Петро: першою буде «Зоря», четвертим — «Шахтар».

Соломія: другими будуть «Карпати», третім — «Динамо».

У кожної дитини, яка робила свій прогноз, правильним було лише одне висловлювання. З'ясувати, які місця в чемпіонаті реально посіли зазначені футбольні команди.

Хід виконання

Позначимо висловлювання змінними:
K1 — першими будуть «Карпати»;
Z1 — першою буде «Зоря»;
D2 — другим буде «Динамо»;
K2 — другими будуть «Карпати»;
D3 — третім буде «Динамо»;
Sh4 — четвертим буде «Шахтар».

Оскільки одне з висловлювань кожної дитини було правильним, то формально це можна записати у вигляді виразів:

$$K1 \vee D2 = 1,\ Z1 \vee Sh4 = 1,\ K2 \vee D3 = 1,$$

де одиниця відповідає значенню істинне. Очевидно, що і кон'юнкція всіх цих виразів також має значення істинне.

Математично це можна записати так:

$$(K1 \vee D2)\ \&\ (Z1 \vee Sh4)\ \&\ (K2 \vee D3) = 1.$$

Перетворимо цей вираз згідно з правилами математичної логіки:

$$(K1Z1 \vee K1Sh4 \vee D2Z1 \vee D2Sh4)\ (K2 \vee D3) = 1.$$

Значення K1Z1 є хибним, оскільки на першому місці не можуть бути дві команди.

Тому вираз можна записати так:

$$(K1Sh4 \vee D2Z1 \vee D2Sh4)(K2 \vee D3) = 1.$$

Також відповідно до правил математичної логіки можна записати:

$$K1Sh4K2 \vee K1Sh4D3 \vee D2Z1K2 \vee D2Z1D3 \vee D2Sh4K2 \vee D2Sh4D3) = 1.$$

Усі складові, які виділено жирним накресленням, є хибними.

Так, вирази K1Sh4K2, D2Z1D3 і D2Sh4D3 є хибними, оскільки одна й та сама команда (у нашому випадку «Карпати» і «Динамо») не можуть бути на двох різних місцях одночасно.

Вирази D2Z1K2 і D2Sh4K2 є хибними, оскільки на другому місці не можуть бути дві команди одночасно.

Залишається істинним вираз K1Sh4D3, що означає: перше місце посіли «Карпати», друге — «Зоря», третє — «Динамо», четверте — «Шахтар».

Приклад 2. Дано логічний вираз:
((x < y) or (y < z)) and ((x > 2) and not (z > 1)).

Код його обчислення подано на рис. 11.2.

```
x = float (input ('Уведення x '))        #Уведення значення x
y = int (input ('Уведення y '))          #Уведення значення y
z = float (input ('Уведення  '))         #Уведення значення z
f = ((x < y) or (y < z))\
    and ((x > 2) and not(z > 1))         #Обчислення значення виразу
print ('f = ', f)                        #Виведення результату
```

Рис. 11.2. Код обчислення логічного виразу

Наведено два варіанти виконання коду:

```
Уведення x 2
Уведення y 3
Уведення  4
f =  False
```

```
Уведення x 2.5
Уведення y 3
Уведення  0.5
f =  True
```

Запитання для перевірки знань

1. Чому дорівнює значення $x \vee 1$?
2. Чому дорівнює значення $x \vee \bar{x}$?
3. Чому дорівнює значення виразу $(x>1)(y<3)$ для $x=3$, $y=5$?

Практична робота № 21

ТЕМА. Побудова логічних виразів та їх обчислення
ЗАВДАННЯ: побудувати логічні вирази та розробити програми їх реалізації.
ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Сергій і Олексій домовилися у суботу покататися на лижах, якщо не буде хуртовини, температура повітря не більше від 7 м/сек.

   Формалізуйте сформульовану домовленість і розробіть для неї логічний вираз.
2. Розробіть код реалізації логічного виразу.
3. Доведіть, що отриманий результат є правильним.
4. Дано вираз: (x > 5) and (x > y). Розробіть код обчислення логічного виразу. Значення змінних x і y вводяться з клавіатури.
5. Виконайте код для значень x = 6, y = 8.
6. Виконайте код для інших значень змінних.
7. Булева функція трьох аргументів має формулу $f(x,y,z)=xy \vee yz \vee x\bar{z}$.

   Обчисліть значення функції для всіх наборів (значень) її аргументів. Заповніть наведену таблицю.
8. Розробіть код обчислення логічного виразу ((x > y) or not (y > z)) and (x > 2).
9. Виконайте код для значень змінних x = 3, y = 2, z = 5.
10. Виконайте код для інших значень змінних.

Зробіть висновок про те, який етап у розв'язуванні логічних адач є найскладнішим.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 11



5. Доведіть, що код виконується правильно.

6. Переконайтеся, що отримано правильний результат.

7.

x	y	z	f(x,y,z)
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

9, 10. Доведіть, що отримано правильний результат.

12. Алгоритми з розгалуженнями

12.1. Команда розгалуження. Умовний оператор мовою Python

Пригадайте, які види розгалужень ви вивчали у попередніх класах. Якими командами розгалужень користувалися?

Як ви вже знаєте, існують три види розгалужень: неповне, повне, вибір.

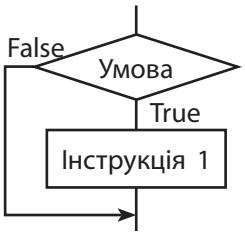


Рис. 12.1. Фрагмент блок-схеми неповного розгалуження

Неповне розгалуження

Пригадаємо, який вигляд має блок-схема неповного розгалуження, і розглянемо рис. 12.1.

Якщо умова має значення True, то виконується Інструкція 1, а якщо умова має значення False, то вона не виконується, а виконується інструкція, розташована безпосередньо за Інструкцією 1.

Цей вид розгалуження мовою Python реалізується командою умовного переходу такої структури:

```
if <логічний вираз>:
    <блок інструкцій>
```

Наведемо приклади з неповним розгалуженням (приклади 1 і 2).

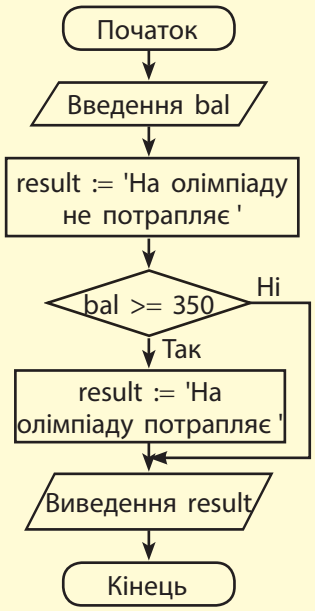


Рис. 12.2. Блок-схема алгоритму визначення

Приклад 1.

```
if x>6.5:
    y = 2*x–5; z = y+x/2.5
```

Якщо значення змінної x більше за 6.5, то виробляється значення True, і після цього виконується обчислення значення двох виразів.

Якщо значення x менше або дорівнює 6.5, то виробляється значення False і значення обох виразів не обчислюються.

Приклад 2. Для того щоб потрапити у збірну команду області для участі у Всеукраїнській олімпіаді з інформатики, Петрику потрібно набрати на обласній олімпіаді не менше ніж 350 балів. Чи потрапить Петрик на Всеукраїнську олімпіаду, якщо набере кількість балів, яка міститься у змінній bal?

Блок-схему алгоритму розв'язування наведено на рис. 12.2.

Код реалізації алгоритму подано на рис. 12.3.

```
#Програма визначення, чи потрапить Петрик на олімпіаду
bal = int (input ("Скільки балів набрав Петрик? "))  #Уведення кількості балів
result = "На олімпіаду не потрапляє"                 #Присвоювання значення змінній result
if bal >= 350:                                       #Перевірка умови
    result = "На олімпіаду потрапляє"                #Нове значення змінної result
print (result)                                       #Виведення результату
```

Рис. 12.3. Код визначення, чи потрапив Петрик на Всеукраїнську олімпіаду

Результат виконання коду:

```
Скільки балів набрав Петрик? 409
На олімпіаду потрапляє
```

Після завершення виконання команди умовного переходу виконується команда, розташована безпосередньо за нею.

Команда if <логічний вираз>: закінчується двома крапками, а команди в наступному рядку автоматично зсуваються на чотири позиції праворуч. У цьому рядку містяться дві команди, які відокремлюються одна від одної комою з крапкою.

Після останньої команди блока програміст сам має вводити наступну команду, перемістивши її на чотири позиції ліворуч.

Повне розгалуження

Пригадаємо, який вигляд має блок-схема повного розгалуження, і розглянемо рис. 12.4.

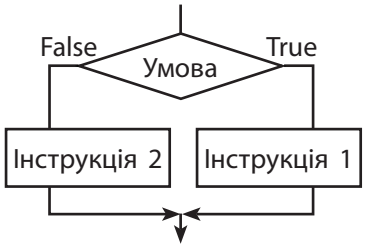


Рис. 12.4. Фрагмент блок-схеми повного розгалуження

Якщо умова має значення True, виконується Інструкція 1, інакше — виконується Інструкція 2.

У мові Python повне розгалуження реалізується командою такої структури:

```
if <логічний вираз>:
    <блок інструкцій 1>
else:
    <блок інструкцій 2>
```

Наведемо приклади алгоритму з неповним розгалуженням (приклади 3 і 4).

Приклад 3.

```
if x>2.3:
    y = 4.5*pow(3, 5); print("y = ", y)
else:
    y = 7.1*(21–pow(1.4, 2)); print("y = ", y)
```

У цьому прикладі, якщо значення змінної x більше за 2,3, обчислюється значення виразу $4{,}5 \cdot 3^5$ і виводиться на екран, інакше — обчислюється значення виразу $7{,}1 \cdot \left(21 - 1{,}4^2\right)$ і також виводиться на екран.

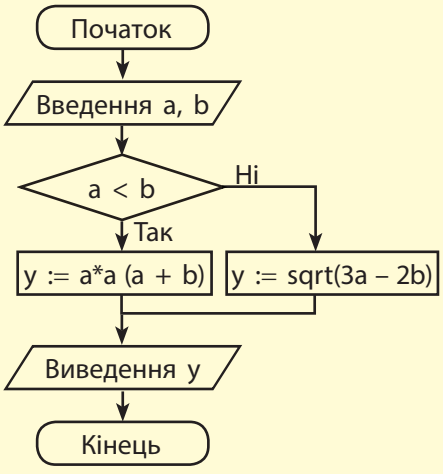


Рис. 12.5. Блок-схема алгоритму обчислення виразу (до прикладу 4)

Приклад 4.

Обчислити значення y:

$$y = \begin{cases} a^2(a+b), \text{якщо } a < b, \\ \sqrt{3a-2b}, \text{якщо } a \geq b. \end{cases}$$

Блок-схему алгоритму обчислення виразу наведено на рис. 12.5, код реалізації алгоритму — на рис. 12.6.

```
#Код з повним розгалуженням
import math                                  #Імпортування модуля math
a = float (input ("Значення a= "))           #Введення значення a
b = float (input ("Значення b= "))           #Введення значення b
if a < b:                                    #Якщо a менше b
    y = math.pow (a,2)*(a + b)               #Обчислення виразу
else:                                        #Якщо a не менше b
    y = math.sqrt (3*a - 2*b)                #Обчислення виразу
print ('y =',y)                              #Виведення результату
```

Рис. 12.6. Код обчислення виразу

Один з варіантів виконання коду:

```
Значення a = 6
Значення b = 3.5
y = 3.3166247903554
```

Приклад 5.

Визначити, у якій валюті слід зробити банківський внесок у розмірі 25 000 грн, щоб через рік отримати максимальну суму у гривнях. Відомо, що у гривнях відсоток річних дорівнює 11,5 %, а у доларах — 4 %. На момент вкладу курс на купівлю становить 27 грн, а за рік курс продажу долара прогнозується 28,6. Розробити алгоритм і програму визначення, у якій валюті вигідно зробити внесок.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. Обчислити суму через рік (вклад у гривнях): Sgrn_1 = 25000 + 0.11.5 * 25004.
3. Перевести гривні у долари (перед вкладом):
   dol = 25000/27.
4. Обчислити суму доларів через рік:
   Sdol = dol + 0.04 * dol.
5. Перевести долари у гривні:
   Sgrn_2 = 28.6 * Sdol.
6. Якщо Sgrn_1 >= Sgrn_2, то p = "Вигідніше у гривнях", інакше p = "Вигідніше у доларах".
7. Виведення значення p.
8. Кінець.

Код реалізації алгоритму наведено на рис. 12.7.

```
#Код з командою повного розгалуження
Sgrn_1 = 25000 + 0.115*25000          #Вклад у гривнях, отримана сума
dol = 25000/27                        #Переведення грн у долари
Sdol = dol + 0.04*dol                 #Вклад у доларах, отримана сума
Sgrn_2 = 28.6*Sdol                    #Переведення доларів у грн
if Sgrn_1 >= Sgrn_2:                  #Команда повного розгалуження
    p = "Вигідніше у гривнях"         #Присвоювання значення змінній
else:                                 #Інакше
    p = "Вигідніше у доларах"         #Присвоювання значення змінній
print (p)                             #Виведення результату
```

Рис. 12.7. Код визначення вигідного вкладу

Результат виконання коду: Вигідніше у гривнях.

Запитання для перевірки знань

1. Які існують види розгалужень?
2. Накресліть блок-схему неповного розгалуження.
3. Накресліть блок-схему повного розгалуження.
4. Яку структуру має команда неповного розгалуження мовою Python?
5. Яку структуру має команда повного розгалуження мовою Python?
6. Наведіть приклад команди неповного розгалуження мовою Python.
7. Наведіть приклад команди повного розгалуження мовою Python.

Завдання для самостійного виконання

1. Літак може здійснити посадку в аеропорту призначення, якщо хмарність буде не нижчою за 500 м. Значення для реальної хмарності вводиться з клавіатури.
   Розробіть алгоритм і програму визначення, чи здійснить літак посадку в аеропорту призначення.
2. Розробіть алгоритм і програму обчислення значення *y*:

$$y=\begin{cases} a^2-bx, \text{якщо } x>0, \\ a^2-\left(c+\dfrac{x}{b}\right), \text{якщо } x\le 0. \end{cases}$$

   Значення *a*, *b*, *c* і x уводяться з клавіатури.
3. В останньому турі Чемпіонату України з футболу грають «Шахтар» і «Динамо». Якщо перемагає «Динамо», то стає чемпіоном України, інакше — чемпіоном стає «Шахтар». Розробіть алгоритм і програму визначення, хто стане чемпіоном, якщо результат гри вводиться з клавіатури.
4. Дано рівносторонній трикутник зі стороною *a* і коло з радіусом *r*. Розробіть алгоритм і програму визначення: чи можна у трикутник вписати це коло. Значення сторони трикутника і радіуса кола вводяться з клавіатури.

5 Розробіть алгоритм і програму обчислення обчислення y:

$$y = \begin{cases} a^3 + \sqrt{3{,}1a + 2(a + 1 < 8b)}, & \text{якщо } a > b, \\ \sqrt{\sqrt{27a + b}}, & \text{якщо } a \le b. \end{cases}$$

6 Знайдіть в інтернеті відомості про площу та населення Черкаської і Рівненської областей. Розробіть алгоритм і програму визначення, у якій області густота населення більша.

12.2. Команда вибору

Як краще реалізувати розгалуження з багатьма гілками розгалужень?

Розгалуження з багатьма гілками можна реалізувати за допомогою команд повного і неповного розгалужень, але зручніше за допомогою команди вибору.

Команда вибору реалізує розгалуження за багатьма гілками. Приклад блок-схеми розгалуження за вибором із трьома гілками наведено на рис. 12.8.

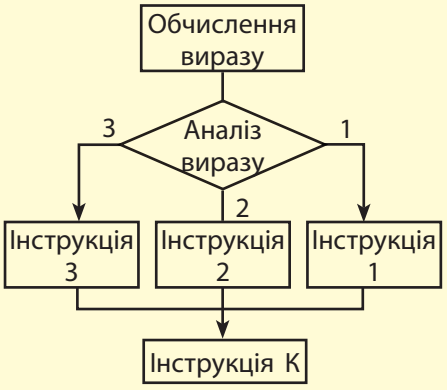


Рис.12.8. Фрагмент блок-схеми розгалуження з трьома гілками

Якщо значення виразу дорівнює 1, виконується Інструкція 1, якщо 2 — Інструкція 2, якщо 3 — Інструкція 3. У випадку якщо вираз не дорівнює жодному з перерахованих значень, то виконується Інструкція K.

Цей вид розгалуження у мові Python реалізується командою такої структури:

```
if <вираз> == <значення_1>:
    <команда_1>
elif <вираз> == <значення_2>:
    <команда_2>
    …
elif <вираз> == <значення_N>:
    <команда_N>
else:
    <команда_K>
```

Тут знак (==) — знак рівності. Якщо значення виразу дорівнює певному значенню, розташованому після знака (==), то виконується відповідна команда (приклад 1).

Приклад 1.

Якщо значення виразу дорівнює значенню_2, то виконується команда_2. Якщо значення виразу не дорівнює жодному з перерахованих значень, то виконується команда_K.

Наведемо приклади алгоритму з вибором (приклади 2 і 3).

Приклад 2.

У XXI ст. літні Олімпійські ігри відбулися в таких країнах: 2000 рік — в Австралії, 2004 рік — в Греції, 2008 рік — в Китаї, 2012 рік — у Великій Британії, 2016 рік — у Бразилії, 2020 рік — планувалися в Японії. За назвою країни визначити рік проведення в ній Олімпіади.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. Увести назву країни у змінну *a*.
3. Якщо a = Австралія, то вивести 2000 рік і перейти до п. 9, інакше — до п. 4.
4. Якщо a = Греція, то вивести 2004 рік і перейти до п. 9, інакше — до п. 5.
5. Якщо a = Китай, то вивести 2008 рік і перейти до п. 9, інакше — до п. 6.
6. Якщо a = Бразилія, то вивести 2016 рік і перейти до п. 9, інакше — до п. 7.
7. Якщо a = Японія, то вивести Планувалося у 2020 році і перейти до п. 9.
8. Якщо a = будь-якій іншій країні, то вивести У цій країні ігор не було.
9. Кінець.

Код реалізації алгоритму наведено на рис. 12.9.

```
#Код з шістьма гілками розгалужень
a = input ("Країна ? ")                    #Введення назви країни
if a == "Австралія":                       #Якщо введена країна Австралія
    print ("2000 рік")                     #Виведення
elif a == "Греція":                        #Якщо введена країна Греція
    print ("2004 рік")                     #Виведення
elif a == "Китай":                         #Якщо введена країна Китай
    print ("2008 рік")                     #Виведення
elif a == "Велика Британія":               #Якщо введена країна Велика Британія
    print ("2012 рік")                     #Виведення
elif a == "Бразилія":                      #Якщо введена країна Бразилія
    print ("2016 рік")                     #Виведення
elif a == "Японія":                        #Якщо введена країна Японія
    print ("Планувалися у 2020 році")     #Виведення
else:                                      #Якщо країна відсутня
    print ("У цій країні ігор не було")
```

Рис. 12.9. Код визначення року проведення Олімпійських ігор за назвою країни

Результат виконання коду:

```
Країна ? Греція
2004 рік
```

Приклад 3.

В Україні є кілька міст, у яких кількість мешканців перевищує 1 млн осіб. Візьмемо відомості про чотири з них: Київ — 2 965 000 осіб, Харків — 1 443 000, Одеса — 1 015 615, Дніпро — 1 003 033. Розробити код, за допомогою якого за номером міста виводиться його назва і кількість мешканців.

Код реалізації алгоритму наведено на рис. 12.10.

```
num = int (input ('Номер міста '))          #Уведення номера міста
if num == 1:                                #Якщо уведено місто 1
    print ('Київ, населення = 2965000')     #Виведення результату
elif num == 2:                              #Якщо уведено місто 2
    print ('Харків, населення = 1443000')   #Виведення результату
elif num == 3:                              #Якщо уведено місто 3
    print ('Одеса, населення = 1015615')    #Виведення результату
elif num == 4:                              #Якщо уведено місто 4
    print ('Дніпро, населення = 1003033')   #Виведення результату
else:                                       #Якщо уведений непередбачений
    print ('Відсутне')                      #Виведення результату
```

Рис. 12.10. Код із розгалуженням вибором

Результат виконання коду:

```
Номер міста 3
Одеса, населення = 1015615
```

Запитання для перевірки знань

1. Накресліть блок-схему команди вибору.
2. Яку структуру має команда вибору мовою Python?
3. Наведіть приклад команди вибору мовою Python.

Завдання для самостійного виконання

1. У журналі успішності класу четвертою є Карпенко Ніна, сьомим — Радченко Ігор, восьмою — Лісова Ганна, тринадцятим — Нестеров Павло. Розробіть код, за допомогою якого за номером учнів і учениць повідомляється їх прізвище і ім'я.
2. Розробіть код обчислення значення y:

$$y=\begin{cases} a^3+2,4\sqrt{5}, \text{ якщо } -a\neq 2 \text{ і } a\neq 4,\\ 3,5+4,23\sqrt{a}, \text{ якщо } a=2,\\ \dfrac{a}{1,7}+\sqrt{3+a}, \text{ якщо } a=4. \end{cases}$$

12.3. Вкладені розгалуження

Спробуйте сформулювати умову задачі, для розв'язування якої необхідно застосувати вкладені розгалуження.

У вкладених розгалуженнях одне розгалуження містить інше розгалуження. Існує багато варіантів вкладених розгалужень. Один із них наведено на рис. 12.11.

Алгоритми можуть складатися з декількох алгоритмічних структур, які можуть бути вкладені одна в одну.

Наведемо приклад 1 із вкладеним розгалуженням.

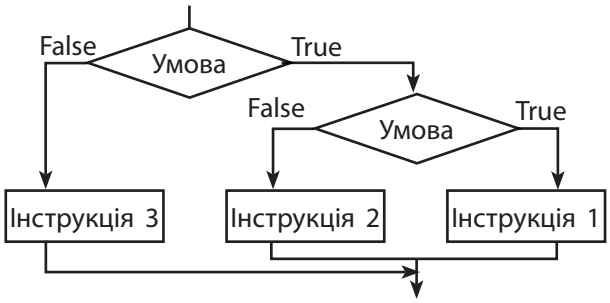


Рис. 12.11. Фрагмент блок-схеми вкладеного розгалуження

Приклад 1. На рис. 12.12 наведено фрагмент блок-схеми алгоритму обчислення значення y:

$$y = \begin{cases} 2a^2 - x, \text{якщо } x > 0 \text{ і } a > 1, \\ 3a^3 + 2x, \text{якщо } x > 0 \text{ і } a \le 1, \\ a + x, \text{якщо } x \le 0. \end{cases}$$

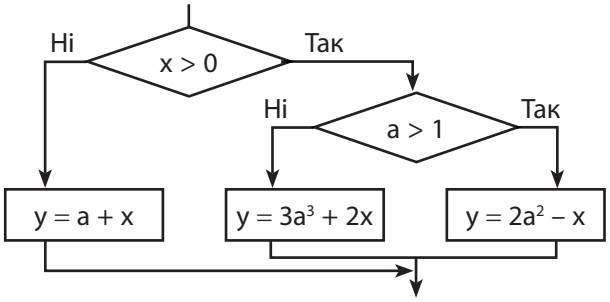


Рис. 12.12. Фрагмент блок-схеми алгоритму обчислення значення y

У мові Python ця структура алгоритму реалізується так:

```
if x>0:                 # Перший оператор if
    if a>1:             # Другий оператор if
        y = 2*pow(a, 2)–x
    else:               # Для другого оператора if
        y = 3*pow(a, 3)+2*x
    else:               # Для першого оператора if
    y = a+x
```

На рис. 12.13 подано код реалізації алгоритму, наведеного на рис. 12.12.

```
a = float (input ('Увести значення a '))   #Уведення значення a
x = float (input ('Увести значення x '))   #Уведення значення x
if x > 0:                                  #Якщо x більше нуля
    if a > 1:                              #Якщо a більше одиниці
        y = 2 * pow (a, 2) - x             #Обчислення виразу
    else:                                  #Якщо x більше нуля і a менше одиниці
        y = 3* pow (a, 3) + 2 * x          #Обчислення виразу
else:                                      #Якщо x менше або дорівнює нулю
    y = a + x                              #Обчислення виразу
print ('y= ', y)                           #Виведення результату
```

Рис. 12.13. Код обчислення значення виразу

Варіант результату виконання коду:

```
Увести значення a 3.75
Увести значення x 0.23
y =  27.895
```

На рис. 12.14 наведено ще один варіант вкладеного розгалуження.

Python входить до трійки найбільш значущих мов у сфері машинного навчання й аналізу великих обсягів даних. Як універсальна мова Python має застосування практично скрізь, навіть в ігровій індустрії.

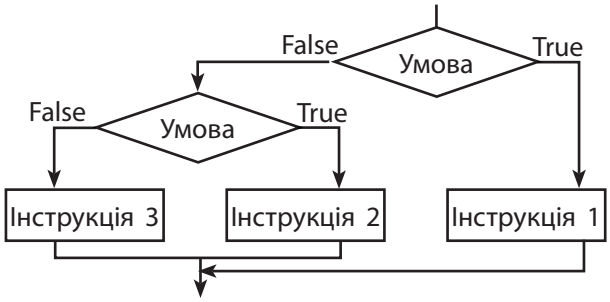


Рис. 12.14. Фрагмент блок-схеми алгоритму з вкладеним умовним переходом

Наведемо приклад із вкладеним розгалуженням за цим варіантом (приклад 2).

Приклад 2. На рис. 12.15 наведено фрагмент блок-схеми алгоритму обчислення значення y:

$$y = \begin{cases} 2{,}9(a-x), \text{якщо } x<1, \\ a^3+3{,}2\left(a^2+ax\right), \text{якщо } x\geq 1 \text{ і } a<2, \\ 3{,}5(4{,}8a-x), \text{якщо } x\geq 1 \text{ і } a\geq 2. \end{cases}$$

Код реалізації алгоритму наведено на рис. 12.16.

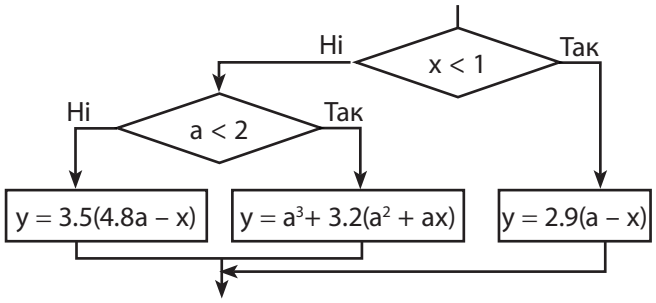


Рис. 12.15. Блок-схема фрагмента алгоритму обчислення виразу

```
a = float (input ('Увести a '))                    #Уведення значення a
x = float (input ('Увести x '))                    #Уведення значення x
if x < 1:                                          #Якщо x менше одиниці
    y = 2.9*(a - x)                                #Обчислення виразу
else:                                              #Якщо x не менше одиниці
    if a < 2:                          #Якщо x не менше одиниці і a менше двох
        y = pow(a, 3) + 3.2*(pow(a, 2) + a*x)      #Обчислення виразу
    else:                              #Якщо x не менше одиниці і a не менше двох
        y = 3.5*(4.8*a - x)                        #Обчислення результату
print ('y = ', y)                                  #Виведення результату
```

Рис. 12.16. Код обчислення арифметичного виразу з вкладеним розгалуженням

Результат виконання коду:

```
Увести a 1
Увести x 1
y =  7.4
```

В алгоритмах із розгалуженням досить зручно застосовувати логічні оператори or, and, not. Переконаємося в цьому на прикладі (приклад 3).

Приклад 3.

Розробити код обчислення значення y:

$$y=\begin{cases}\sqrt{a+b}-1{,}5b\cdot x, \text{ якщо } a>b \text{ і } x\geq 2,\\ a^2+2\left(a+\sqrt{b}\right), \text{ якщо } a\leq b \text{ або } x<2.\end{cases}$$

Код обчислення виразу наведено на рис. 12.17.

```
import math                                          #Імпортування модуля math
a = float (input ('Уведіть значення a '))            #Уведення значення a
b = float (input ('Уведіть значення b '))            #Уведення значення b
x = float (input ('Уведіть значення x '))            #Уведення значення x
if a > b and x >= 2:                                 #Якщо a > b і x >= 2
    y = math.sqrt (a + b) - 1.5*b*x                  #Обчислення виразу
else:                                                #Якщо a <= b або x < 2
    y = pow (a, 2) + 2*(a + math.sqrt (b))           #Обчислення виразу
print ( 'y = ', y)                                   #Виведення результату
```

Рис. 12.17. Код з оператором and у команді розгалуження

Результат виконання коду:

```
Увести значення a 3
Увести значення b 2
Увести значення x 2
y =  -3.76393202250021
```

Запитання для перевірки знань

1. Які розгалуження називають вкладеними?
2. Накресліть блок-схему одного з варіантів вкладеного розгалуження.
3. Як вкладене розгалуження реалізується у мові Python?
4. Наведіть приклад такої реалізації.

Завдання для самостійного виконання

1. З клавіатури вводяться значення трьох сторін трикутника. Розробіть алгоритм визначення, чи є трикутник рівнобедреним, і код реалізації алгоритму.
2. Для того щоб потрапити у збірну команду України з легкої атлетики, в останніх змаганнях у стрибках у довжину спортсменам необхідно стрибнути далі, ніж на 7,5 м. Перший спортсмен стрибнув на L1 м, а другий — на L2 м. Розробіть алгоритм і код реалізації визначення усіх варіантів зарахування або незарахування спортсменів у команду.
3. З клавіатури вводяться значення радіусів двох кіл *R* і *r* і сторона квадрата a. Розробіть алгоритм визначення, чи можна у перше коло вписати квадрат, а у квадрат вписати друге коло. Розробіть код реалізації алгоритму.
4. Розробіть алгоритм обчислення значення *y*:

$$y=\begin{cases}\sqrt{a+b}, \text{ якщо } a>2 \text{ і } b>3,\\ a^3+3{,}5b, \text{ якщо } a>2 \text{ і } b\leq 3,\\ 2a\sqrt{\sqrt{3a+2{,}5b}}, \text{ якщо } a\leq 2.\end{cases}$$

Розробіть код реалізації цього алгоритму.

12.4. Використання об'єктів мови Python для реалізації розгалужень

Пригадайте, які існують у мові Python прості й структуровані вбудовані типи об'єктів.

Пригадаємо, що у мові програмування Python все, що опрацьовується, є об'єктом: числа, списки, рядки, множини тощо. Навіть класи є об'єктами. Графічні кнопки, мітки, перемикачі, прапорці, текстові поля тощо теж. У деяких мовах, наприклад, у Delphi, вони називаються компонентами. Порядок роботи з таким типом об'єктів вивчається у 9 класі.

Розглянемо приклади програм, у яких для реалізації розгалужень використовуються списки, діапазони, множини й словники.

Приклад 1. Дано числа: 23, 4, 19, 11, 9, 16, 19, 4, 21, 9, 16, 19.

Створити список, елементами якого є ці числа.

Якщо число на четвертій позиції списку є більшим за число на восьмій позиції, то робиться зріз першої половини списку, упорядковується, у ній визначається максимальне й мінімальне значення. Інакше робиться зріз другої половини списку й виконуються аналогічні операції.

Код реалізації наведено на рис. 12.18.

```
import random                                          #Імпортування модуля random
lst = [23, 4, 19, 11, 9, 16, 19, 4, 21, 9, 16, 19]    #Створення списку
if lst [4] > lst [8]:                                  #Якщо перше більше другого
    lst1 = lst [0:6]                                   #Зріз першої половини
    print ('Перша половина, неупорядкована ',lst1)     #Виведення першої половини
    lst1.sort ()                                       #Сортування першої половини
    print ('Перша половина, упорядкована ',lst1)       #Упорядкована перша половина
    print ('Maxs = ',max (lst1),'Min = ',min (lst1))   #Мак. і мін. значення
else:                                                  #Інакше
    lst2 = lst [6:12]                                  #Зріз другої половини
    print ('Друга половина, неупорядкована ',lst2)     #Виведення другої половини
    lst2.sort ()                                       #Сортування другої половини
    print ('Друга половина, упорядкована ',lst2)       #Упорядкована друга половина
    print ('Maxs = ',max (lst2),'Min = ',min (lst2))   #Мак. і мін. значення
```

Рис. 12.18. Код опрацювання списку з розгалуженням

Результат виконання коду:

```
Друга половина, неупорядкована  [19, 4, 21, 9, 16, 19]
Друга половина, упорядкована  [4, 9, 16, 19, 19, 21]
Maxs =  21 Min =  4
```

Приклад 2. Підприємець планує вилетіти до Праги у неділю першого тижня січня, якщо на цей день не припадає Різдво і є рейс до Праги. Якщо виліт у першу неділю не відбудеться, він планує вилетіти у неділю другого тижня, якщо у цей день є рейс до Праги. Є календар на першу декаду січня і розклад рейсів на першу і другу неділі тижня. Коли підприємець вилетить до Праги?

Код реалізації алгоритму наведено на рис. 12.19.

```
d1 = {"четвер":3, "п'ятниця":4, "субота":5,\
      "неділя":6, "понеділок":7, "вівторок":8}     #Календар
d2 = {"Рим":967, "Мюнхен":1075, "Лондон":675,\
      "Мюнхен":1039, "Прага":1231, "Стамбул":897} #Розклад у першу неділю
d3 = {"Софія":1043, "Прага":1342, "Мюнхен":1054,\
      "Рим":995, "Берлін":1036, "Париж":1055}      #Розклад у другу неділю
p = 'Підприємець у заплановані дні не вилетить'   #Повідомлення
a = d1["неділя"]                                  #Число першої неділі
print ('У неділю чило ',a)                        #Виведення числа першої неділі
if (7 != a) and ("Прага" in d2):                  #Якщо не 7 і є рейс до Праги
    p = 'Вилетить у першу неділю'                 #Повідомлення
else:                                             #Інакше
    if "Прага" in d3:                             #Якщо у другу неділю є рейс
        p = 'Вилетить у другу неділю'             #Повідомлення
print (p)                                         #Виведення результату
```

Рис. 12.19. Код опрацювання словника з розгалуженням

Результат виконання коду:

```
У неділю чило  6
Вилетить у першу неділю
```

Запитання для перевірки знань

1. Розробіть код створення списку, елементами якого є вісім літер української абетки. Якщо в ньому є літера д, то видаліть її і додайте дві нові літери, відсутні у списку.
2. Розробіть код створення словника, ключами якого є назви навчальних предметів, які вивчаються у 8 класі, а значеннями — кількість уроків на тиждень. Визначте назви предметів, з яких кількість уроків на тиждень — чотири і два.

Практична робота № 22

ТЕМА. Розроблення алгоритмів із послідовними розгалуженнями та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити алгоритми з повним і неповним розгалуженнями і коди їх реалізації.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Одне й те саме тіло кидають вертикально вниз із висот h2 і h1 (h2 > h1). З висоти h2 тіло кидають із початковою швидкістю V0 = 6.8 м/сек, а з висоти h1 — із початковою швидкістю V0 = 0.

   Розробіть математичну модель визначення, з якої із цих висот тіло упаде на землю першим.

2. Розробіть блок-схему алгоритму розв'язування задачі.
3. Розробіть код реалізації алгоритму. Уведіть код і виконайте його налагодження.
4. Виконайте код.
5. Виконайте код для п'яти-шести різних значень V0 кидання тіла з висоти h2.
6. Побудуйте графік залежності часу падіння тіла від початкової швидкості його кидання.
7. Розробіть блок-схему алгоритму обчислення значення y:

$$y=\begin{cases}\sqrt{\sqrt{2{,}5\left(a^3+3b^2\right)}}, \text{якщо } a>2 \text{ і } b>1,\\ 2{,}35\left(a^3+3{,}75(a+b)\right), \text{якщо } a\le 2 \text{ або } b\le 1.\end{cases}$$

8. Розробіть код реалізації алгоритму обчислення.
9. Уведіть код і виконайте його налагодження.
10. Виконайте код для різних значень змінних.

Зробіть висновок: у яких випадках доцільно застосовувати команди неповного і повного розгалужень і логічні оператори у цих командах.

4. Доведіть, що отримано правильний результат.

9. Доведіть, що код виконується правильно.

10 Переконайтеся у правильності отриманих результатів.

Практична робота № 23

ТЕМА. Розроблення алгоритмів із вкладеними розгалуженнями та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити алгоритми з вкладеними розгалуженнями і коди їх реалізації.

ОБЛАДНАННЯ: комп'ютер з операційною системою Windows 7 і вище, з мовою Python і середовищем IDLE.



Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. О 9:00 з Києва до Дніпра виїхав автобус. Він рухався із середньою швидкістю v1 км/год і зробив дві зупинки тривалістю по 20 хв. Об 11:30 за тим самим маршрутом виїхав легковий автомобіль, який рухався без зупинок із середньою швидкістю v2 км/год (v2 > v1).

   Розробіть блок-схему алгоритму визначення, який автотранспортний засіб прибуде у Дніпро першим.

2. Розробіть код реалізації алгоритму та виконайте налагодження коду.

3. Виконайте код.

4. Визначте значення швидкостей руху транспорту, за яких автобус прибуде до міста Дніпро першим.

5. Розробіть алгоритма обчислення значення y:

$$y = \begin{cases} \dfrac{17{,}1\left(3{,}8a + \sqrt{5b}\right)}{3-b}, & \text{якщо } a > 0 \text{ і } b > 1, \\ \sqrt{7{,}5b\sqrt{a^3 + 2{,}3a}}, & \text{якщо } a > 0 \text{ і } b \le 1, \\ 26{,}3\left(2{,}4a^3 + 2{,}3a\right), & \text{якщо } a \le 0. \end{cases}$$

6. Розробіть блок-схему алгоритму обчисленння значення y.

7. Розробіть код реалізації алгоритму без використання в командах розгалуження логічних операторів.

8. Уведіть код і виконайте його налагодження.

9. Виконайте код.

10. Зробіть такі зміни в коді, щоб у командах розгалуження використовувалися логічні оператори.

11. Виконайте змінений код.

Зробіть висновок: який код є простішим — з використанням логічних операторів чи без них.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 12

3. Доведіть, що отримано правильний результат.

9. Доведіть, що код виконується правильно.

11. Переконайтеся у правильності отриманих результатів.

13. Алгоритми з повторенням

13.1. Команди повторення. Оператори циклу

Наведіть найпростіший приклад алгоритму із заздалегідь відомою та невідомою кількістю повторень.

Алгоритми з повторенням, або **циклічні**, — це алгоритми, у яких одні й ті самі інструкції можуть виконуватися багаторазово для різних значень змінних.

Серію інструкцій, що виконуються багато разів під час виконання циклу, називають тілом циклу.

Існують два основні види алгоритмів із повторенням.

Цикли з параметрами

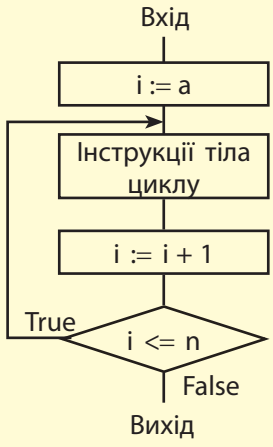


Рис. 13.1. Блок-схема циклу з параметром

Цикли із заздалегідь відомою кількістю повторень інструкцій тіла циклу ще називають циклами з параметрами. Найпростішу структуру таких циклів наведено на рис. 13.1. У ній змінна i — лічильник циклів, який набуває значень від a до n. Блок інструкції тіла циклу може містити одну або кілька інструкцій. Змінна n містить кінцеве значення змінної циклу, а змінна a — її початкове значення.

Наведена структура циклу з лічильником реалізується у багатьох мовах програмування. У Python лічильники у прямому сенсі не використовуються. Цикли із заздалегідь відомою кількістю циклів реалізуються оператором циклу for з об'єктом.

Команда повторення з оператором for має таку структуру:

```
<створення об'єкта>
for <змінна> in <об'єкт> :
    <блок інструкцій тіла циклу>
```

де: <об'єкт> — може бути рядок, список, кортеж, діапазон, словник тощо, які підтримують реалізацію циклу.

Оператор for перебирає всі значення об'єкта й присвоює їх змінній. Спочатку вибирається перше значення об'єкта і присвоюється змінній, потім друге, третє значення і так далі до останнього значення (приклад 1).

Приклад 1. На рис. 13.2 наведено приклад, у якому об'єктом є список із чотирма значеннями. Спочатку змінна a набуває значення 5, до якого додається число 7, потім вона набуває значення 7, до якого також додається число 7, і так далі.

```
>>> slp = [5, 7, 10, 15]      #Список
>>> for a in slp:
        print ((7 + a), end = " ")

12 14 17 22
```

Рис. 13.2. Приклад використання оператора for

У команді циклу з оператором for досить часто застосовується вбудована функція range(), яка генерує послідовність цілих чисел у заданому діапазоні.

Структура функції range():

range([початок], <кінець>[, крок]).

Обов'язковим є лише параметр кінець. Якщо інші параметри не зазначено, то генерується послідовність цілих чисел від 0 до значення кінець (приклад 2).

Приклад 2. На рис. 13.3 наведено приклад використання функції range() в команді циклу з оператором for.

```
>>> s = 23
>>> for i in range (5, 21, 3):
        s = s + i
        print (s, end = " ")


28 36 47 61 78 98
```

Рис. 13.3. Приклад використання функції range() в операторі for

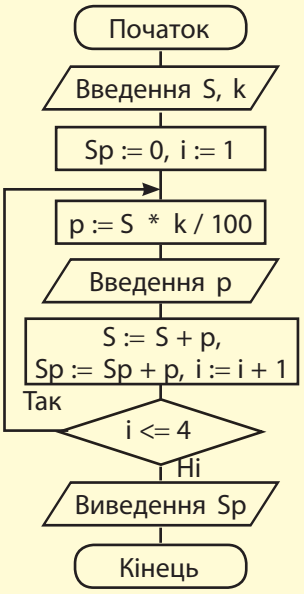


Рис. 13.4. Блок-схема алгоритму обчислення прибутку від вкладу

Приклад 3. У банк покладено на чотири роки S гривень під k відсотків річних. Визначити суму прибутку за кожний рік і загальну суму прибутку.

Використаємо такі змінні:

i — значення поточного року,

p — прибуток за поточний рік,

S — сума вкладу,

Sp — загальна сума прибутку.

Блок-схему алгоритму розв'язування задачі наведено на рис. 13.4, код реалізації алгоритму — на рис. 13.5.

```
#Код із заздалегідь відомою кількістю циклів
S = int (input ("Сума вкладу = "))          #Введення початкової суми вкладу
k = int (input ("Відсоток річних = "))      #Введення відсотка річних
Sp = 0                                      #Початкова сума прибутку
for i in range (4):                         #Цикл із функцією range ()
    i = i + 1                               #Збільшення значення поточного року
    p = S * k/100                           #Обчислення прибутку за поточний рік
    print ("За ",i, " рік", p)              #Виведення прибутку за поточний рік
    S = S + p                               #Обчислення поточної суми вкладу
    Sp = Sp + p                             #Обчислення загальної суми прибутку
print ("Загальна сума прибутку = ", Sp)     #Виведення загальної суми прибутку
```

Рис. 13.5. Код обчислення прибутку від вкладу

Результат виконання коду:

```
Сума вкладу = 40000
Відсоток річних = 11
За  1  рік 4400.0
За  2  рік 4884.0
За  3  рік 5421.24
За  4  рік 6017.5764
Загальна сума прибутку =  20722.8164
```

Цикли з умовою

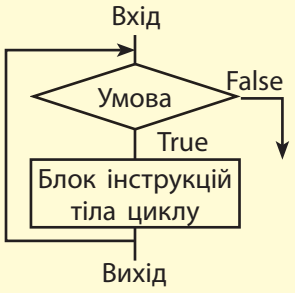


Рис. 13.6. Блок-схема циклу з передумовою

Циклами із заздалегідь невідомою кількістю повторень ще називають циклами з умовою. Умова може бути на початку циклу (цикли з передумовою) і в кінці (цикли з післяумовою).

На рис. 13.6 наведено блок-схему циклу з передумовою. Як бачимо, умова перевіряється до початку виконання блоку інструкцій тіла циклу. Це означає, що інструкції тіла циклу можуть бути не виконані жодного разу. Блок інструкцій тіла циклу може містити одну або кілька інструкцій. Вони виконуються доти, доки умова має значення True.

Цикли з передумовою реалізуються оператором while.

Структура оператора while:

```
<початкове значення>
while <перевірка початкового значення>:
    <блок команд тіла циклу>
    <зміна початкового значення>
```

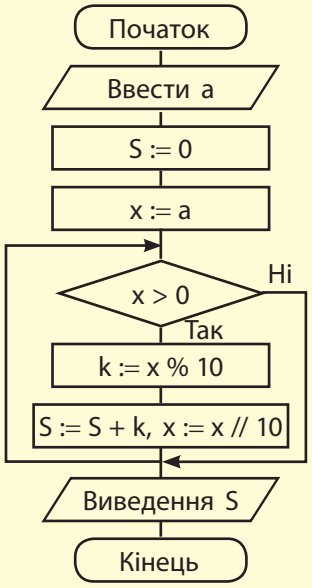


Рис. 13.8. Блок-схема алгоритму обчислення суми цифр

Результатом перевірки початкового значення може бути True або False.

Якщо результат перевірки має значення True, то виконуються команди тіла циклу, змінюється початкове значення змінної, після чого керування передається оператору while.

Якщо результатом перевірки є значення False, то буде виконана перша команда, розташована після тіла циклу.

Приклад 4. На рис. 13.7 наведено приклад використання оператора while. Тут змінна і набуває початкового значення 1, і потім на кожному циклі обчислюється i2, отримане значення виводиться на екран і далі збільшується на одиницю. Як тільки значення і буде більшим або дорівнюватиме 12, виконується значення False і здійснюється вихід із циклу.

```
>>> i = 1
>>> while i <= 12:
        print (pow (i, 2), end = " ")
        i = i + 2

1 9 25 49 81 121
```

Рис. 13.7. Приклад використання оператора while

Приклад 5. З клавіатури уводиться ціле десяткове число довільної розрядності. Визначити суму цифр цього числа.

На рис. 13.8 наведено блок-схему алгоритму обчислення. Основна ідея алгоритму полягає в тому, що початкове значення числа ділиться на 10. У результаті отримаємо остачу й цифру молодшого розряду. На наступному кроці остача також ділиться на 10, у результаті чого отримаємо цифру наступного розряду і чергову остачу. Цей процес продовжується доти, доки поточна остача залишається більшою за нуль.

У блок-схемі використано такі змінні: a — початкове значення десяткового числа, S — сума цифр числа, k — значення поточної цифри, x — поточне значення цілої частини числа

Код реалізації алгоритму наведено на рис. 13.9.

```
#Код з циклом з передумовою
a = int (input ("Увести число : "))   #Введення числа
S = 0                                 #Початкова сума цифр
x = a                                 #Змінна x набуває значення введеного числа
while x > 0:                          #Поки ціла частина більше за 0
    k = x % 10                        #Остача від ділення числа на 10
    S = S + k                         #Обчислення суми цифр
    x = x // 10                       #Ціла частина від ділення числа 10
print ("Сума цифр = ", S)             #Виведення результату
```

Рис. 13.9. Код обчислення суми цифр десяткового числа

Результат виконання коду:

```
Увести число : 6390175010
Сума цифр =  32
```

На рис. 13.10 наведено блок-схему циклу з післяумовою. У таких алгоритмах спочатку виконуються інструкції тіла циклу, а потім перевіряється умова.

Якщо умова має значення True, то виконання інструкцій тіла циклу продовжується. Щойно умова набуде значення False, виконання інструкцій тіла циклу закінчується й керування передається першому оператору, розташованому за оператором циклу.

У мові Python відсутній оператор, який безпосередньо реалізує такий варіант циклу. Його можна реалізувати такою конструкцією оператора while з оператором break:

```
while True:
    <блок команд тіла циклу>
    if <умова>: break
```

Оператор break перериває виконання циклу, якщо виконується деяка умова (приклад 6).

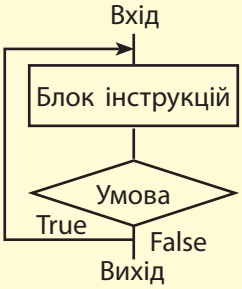


Рис. 13.10. Блок-схема циклу з післяумовою

Приклад 6. Список містить числа 2, 5, 7, 9, 15. З'ясувати, на якій позиції знаходиться число 7.

Код реалізації алгоритму наведено на рис. 13.11.

```
lst = [2, 5, 7, 9, 15]      #Створення списку
i = 0                       #Початковий номер елемента у списку
while True:                 #Оператор циклу while
    if lst [i] == 7:        #Елемент на i-тій позиції дорівнює 7?
        break               #Цикл переривається, якщо значення елемента дорівнює 7
    i = i + 1               #Збільшення номера елемента у списку
print ('На позиції',i)      #Виведення номера позиції, елемент на якій дорівнює 7
```

Рис. 13.11. Код реалізації циклу з післяумовою

Результат виконання коду: На позиції 2.

У командах реалізації циклів часто використовується оператор continue.

На відміну від break, оператор continue не перериває цикл, а повертає на початок циклу, що надає можливість перейти до наступної ітерації циклу до завершення виконання всіх команд усередині циклу (приклад 7).

Приклад 7. Генерується послідовність цілих чисел у діапазоні від 3 до 40 з кроком 3. Вивести на екран усі числа, крім чисел у діапазоні від 14 до 31.

Код реалізації алгоритму наведено на рис. 13.12.

```
for i in range (3, 40, 3):    #Генерування послідовності чисел
    if 14 < i < 31:           #Якщо число більше 14 і менше 31
        continue              #Повернення на початок циклу
    print (i, end = "  ")     #Виведення заданої послідовності чисел
```

Рис. 13.12. Код із використанням у циклі оператора continue

Результат виконання коду: 3 6 9 12 33 36 39.

Запитання для перевірки знань

1. Які алгоритми називають алгоритмами з повторенням?
2. Які існують види алгоритмів із повторенням?
3. Для чого призначений оператор break?
4. Для чого призначений оператор continue?
5. Поясніть сутність виконання команди повторення з оператором for.
6. Наведіть приклад команди повторення з оператором for і функцією range.

Завдання для самостійного виконання

1. Розробіть код таблиці множення на 7.
2. Розробіть код виведення окремо кожного символу слова «оператор».
3. Розробіть код обчислення значення виразу $a+2a+4a+6a+8a+10a+12a$. Значення a вводиться з клавіатури.
4. Початкове значення спадної геометричної прогресії дорівнює 12, а її знаменник — 0,5. Розробіть код обчислення суми членів прогресії, значення яких більше за 0,5.
5. Розробіть код обчислення значення виразу $1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n}$. Значення n вводиться з клавіатури.
6. З висоти 400 м тіло кидають вертикально вниз. Розробіть код визначення такої швидкості кидання, за якої тіло падає на землю не пізніше ніж через 2,1 с.
7. Запишіть структуру команди повторення з оператором while.

13.2. Поєднання повторення і розгалуження

Спробуйте навести приклади, у яких в алгоритмах із повторенням використовується розгалуження.

На практиці нам часто зустрічаються алгоритми, у яких тіло циклу містить команди розгалуження. Приклад блок-схеми алгоритму введення даних наведено на рис. 13.14 (с. 240).

Якщо дані відповідають певній умові, то вони опрацьовуються, інакше — ні. Процес введення й опрацювання даних здійснюється доти, доки не виконається умова кінця циклу.

Алгоритми з поєднанням повторення й розгалуження можуть бути досить складними: одна умова містить декілька інших умов, один цикл може бути вкладений в інший. Далі наведено приклади програм реалізації алгоритмів, у яких поєднується повторення й розгалуження (приклади 1 і 2).

Приклад 1. Сім випускників і випускниць ліцею вступали на факультет кібернетики й набрали із 200 можливих такі суми балів: 180, 165, 156, 177, 159, 187, 181. Скільки осіб вступили на факультет, якщо прохідний бал дорівнює a (уводиться з клавіатури)? Блок-схему алгоритму розв'язування наведено на рис. 13.15, код реалізації на рис. 13.13.

```
#Код з циклом, усередині якого є умова
a = int (input ("Прохідний бал = "))        #Увести прохідний бал
lst = [180, 165, 156, 177, 159, 187, 181]   #Створення списку
k = 0                                       #Початкова кількість учнів
for i in range (7):                         #Цикл з оператором for
    if lst [i] >= a:                        #Умова у середині циклу
        k = k + 1                           #Збільшення кількості учнів
print ("Поступило ", k," учнів")            #Виведення результату
```

Рис. 13.13. Код визначення кількості учнів і учениць, які вступили на факультет

Результат виконання коду:

```
Прохідний бал = 158
Вступило  6  учнів
```

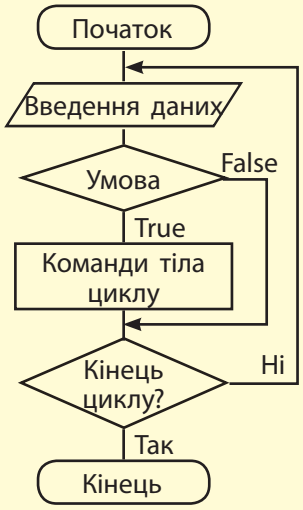


Рис. 13.14. Блок-схема алгоритму, у якому поєднуються повторення і розгалуження)

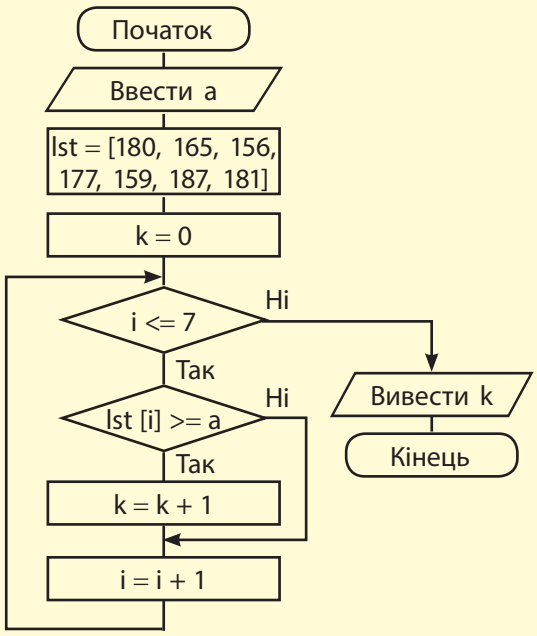


Рис. 13.15. Блок-схема алгоритму визначення кількості учнів, які вступили на факультет (до прикладу 1)

Приклад 2. У діапазоні цілих чисел 5, 6, 7, 8, ..., 73, 74, 75 визначити всі числа, кратні 3, та їх суму, крім чисел, що знаходяться в діапазоні від 31 до 61.

Блок-схему алгоритму розв'язування наведено на рис. 13.18.

Код реалізації алгоритму подано на рис. 13.16.

```
#Код з циклом, усередині якого є розгалуження і оператор continue
x = 5                          #Початкове число
S = 0                          #Початкове значення суми чисел
print ("Числа, кратні 3 :")    #Повідомлення
while (x < 75):                #Цикл з оператором while
    x = x + 1                  #Чергове значення числа
    if 31 < x < 61:            #Якщо число знаходиться в діапазоні від 31 до 61
        continue               #Перехід на початок циклу
    if (x % 3) == 0:           #Якщо число кратне 3
        S = S + x              #Обчислення суми чисел
        print (x, end = " ")   #Виведення числа, що кратне 3
print ()                       #Перехід на новий рядок
print ("Сума чисел = ", S)     #Виведення суми чисел
```

Рис. 13.16. Код визначення чисел, кратних 3, та їх суми

Результат виконання коду:

```
Числа, кратні 3 :
6 9 12 15 18 21 24 27 30 63 66 69 72 75
Сума чисел =  507
```

Цей код можна спростити, якщо використати в команді розгалуження логічні оператори not і and. Код реалізації того самого завдання наведено на рис. 13.17.

```
File  Edit  Format  Run  Options  Window  Help
#Код з циклом, усередині якого є розгалуження
x = 5                                   #Початкове число
S = 0                                   #Початкове значення суми чисел
print ("Числа, кратні 3 :")             #Повідомлення
while (x < 75):                         #Цикл з оператором while
    x = x + 1                           #Чергове значення числа
    if not(31<x<61) and ((x % 3) == 0): #x не в дапазоні 31-61 і ділиться на 3
        S = S + x                       #Обчислення суми чисел
        print (x, end = " ")            #Виведення числа, яке кратне 3
print ()                                #Перехід на новий рядок
print ("Сума чисел = ", S)              #Виведення суми чисел
```

Рис. 13.17. Код із логічними операторами в команді розгалуження

У результаті виконання коду отримаємо той самий результат, що й у попередньому коді.

Запитання для перевірки знань

1. Розробіть алгоритм і код визначення, чи є серед імен дівчат вашого класу хоча б одне ім'я Валентина.
2. Відомий зріст кожного хлопця вашого класу. Розробіть алгоритм і код визначення, скільки хлопців мають зріст, вищий за 165 см.
3. Генерується 17 цілих випадкових чисел у діапазоні від 6 до 37. Розробіть алгоритм і код обчислення суми чисел, кратних двом, які не потрапляють у діапазон від 10 до 16.
4. Розробіть алгоритм і код визначення у слові **multimedia** усіх букв, код яких більший за 104, але менший від 117.
5. Лотерейний квиток має дев'ятицифровий номер. Він вважається щасливим, якщо в ньому є не менше від двох цифр 7. Розробіть алгоритм і код визначення, чи є квиток, який ви придбали, щасливим.
6. У банк покладено 500 000 грн під 6 % річних. Щороку з рахунку знімається 20 000 грн. Розробіть алгоритм і код визначення, через яку мінімальну кількість років сума вкладу становитиме 750 000 грн.

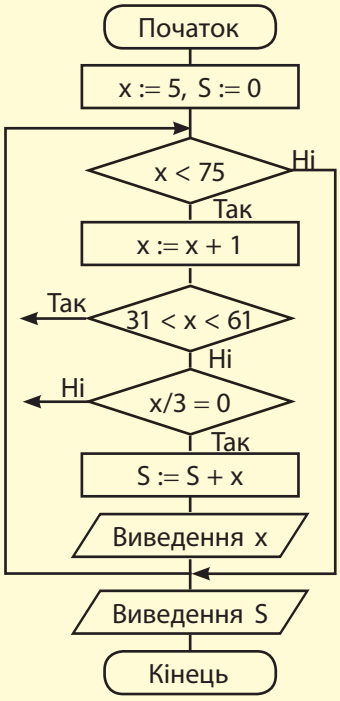


Рис. 13.18. Блок-схема алгоритму визначення чисел, кратних 3, та їх суми (до прикладу 2)

13.3. Числові і рекурентні послідовності

Ви вже вивчали числові послідовності, зокрема вбудовані. Пригадайте назви цих послідовностей.

Python дуже популярна для написання серверної частини веб-сайтів для мобільних і веб-застосунків та побудови різноманітних сервісів.

Пригадайте з повсякденного життя, які об'єкти для подальшого користування доречно пронумерувати. Наведіть приклади. Так, це можуть бути дні тижня, місяці, номери шкіл, будинків тощо. Як ви знаєте, послідовність у Python — це об'єкт, який підтримує звернення до його елемента за допомогою цілочислових індексів і дозволяє отримати його довжину за допомогою функції len().

Усі вбудовані послідовності підтримують операцію зрізу та перевірку входження елемента у послідовність (in), а також методи count() і index(). Крім того, для кожної послідовності існує багато інших методів і функцій їх опрацювання.

Числові послідовності

Елементами числових послідовностей є цілі або дійсні числа. Числовими послідовностями мови Python можуть бути списки, кортежі й діапазони (приклади 1 і 2).

Приклад 1.

Дано n чисел: 23, 44, 9, 24, 8, 15, 36, 7, 12, 28. Розробити алгоритм і код обчислення суми чисел послідовності, кратних 4.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. Створити числову послідовність lst довжиною n.
3. S := 0 # Початкова сума чисел
4. i := 0.
5. Якщо lst[i] кратне 4, то п. 6, інакше — п. 7.
6. S := S+lst[i].
7. i := i+1.
8. Якщо i ≤ n, то п. 5, інакше — п. 9.
9. Виведення значення S.
10. Кінець.

Код реалізації алгоритму наведено на рис. 13.19.

```
#Код обчислення суми чисел послідовності, кратних 4
lst = [23, 44, 9, 24, 8, 15, 36, 7, 12, 28]     #Створення списку
S = 0                                           #Початкове значення суми
for a in lst:                                   #Цикл перегляду послідовності
    if a % 4 == 0:                              #Якщо число ділиться на 4
        S = S + a                               #Обчислення суми чисел
print ('Сума = ', S)                            #Виведення суми чисел
```

Рис. 13.19. Код обчислення суми чисел послідовності, кратних 4

Результат виконання коду: 152.

Приклад 2. Розробити алгоритм і код формування числової послідовності, елементами якої є:

перший елемент — 1*2;
другий елемент — 2*3;
третій елемент — 3*4;
i-тий елемент — (i – 1)*i.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. Створити порожню послідовність lst.
3. Увести кількість елементів послідовності n.
4. i := 1.
5. a := i*(i+1).
6. Приєднати значення a до lst.
7. i := i + 1.
8. Якщо i ≤ n, то п. 5, інакше — п. 9.
9. Вивести послідовність lst.
10. Кінець.

Код реалізації алгоритма на рис. 13.20.

```
#Код формування числової послідовності
lst = []                          #Створення порожньої послідовності
n = int (input ('Увести n '))     #Уведення кількості елементів послідовності
for i in range (1, n):            #Цикл формування послідовності
    a = i * (i + 1)               #Обчислення значення чергового елемента
    lst.append (a)                #Формування послідовності
print (lst)                       #Виведення послідовності
```

Рис. 13.20. Код формування числової послідовності

Результат виконання коду:

```
Увести n 9
[2, 6, 12, 20, 30, 42, 56, 72]
```

Рекурентні послідовності

Рекурентні послідовності — це такі послідовності, у яких наступний член обчислюється через значення попередніх членів.

Найвідомішими рекурентними послідовностями є:

- значення членів арифметичної прогресії, у якій значення наступного члена обчислюється за допомогою формули $a_{n+1} = a_n + 1$;
- значення членів геометричної прогресії, у якій значення наступного члена обчислюється за допомогою формули $a_{n+1} = a_n \cdot q$;
- значення факторіалу, які обчислюються за формулою $a_{n+1} = a_n (n+1)$.

Класичним прикладом рекурентної послідовності є числа Фібоначчі: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55. У цій послідовності перший і другий елементи дорівнюють 1, а значення кожного наступного — сумі значень двох попередніх, тобто його значення обчислюється за допомогою формули: $a_n = a_{n-1} + a_{n-2}$.

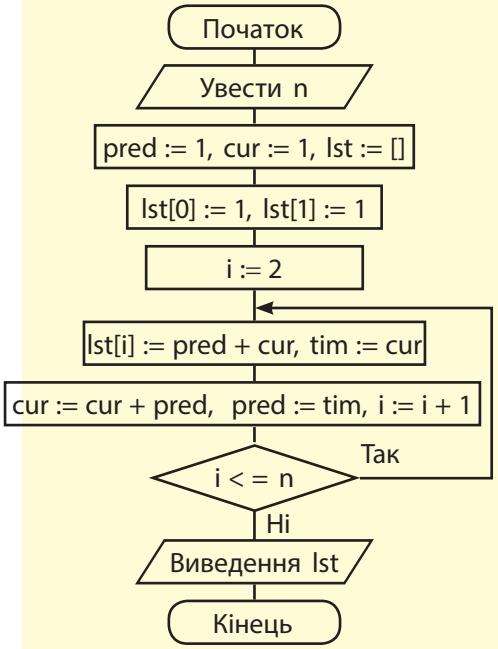


Рис. 13.21. Алгоритм створення чисел Фібоначчі

На рис. 13.21 зображено блок-схему алгоритму обчислення чисел Фібоначчі. В алгоритмі використані такі позначення: pred — попереднє значення числа, cur — поточне значення числа, lst — числова послідовність, i — лічильник чисел Фібоначчі, n — кількість чисел.

Код реалізації алгоритму наведено на рис. 13.22.

```
#Код створення послідовності чисел Фібоначчі
n = int(input("Кількість чисел  = ")) #Введення кількісті чисел фібоначчі
pred = 1                              #Перше число
cur = 1                               #Друге число
lst = []                              #Порожня послідовність
lst.append (1)                        #Формування нульового елемента в lst
lst.append (1)                        #Формування першого елемента в lst
for i in range (2, n):                #Цикл створення послідовн. чисел фібоначчі
    lst.append (pred + cur)           #Формування чергового елемента
    tim = cur                         #Тимчасове збереження поточного елемента
    cur = cur + pred                  #Нове значення поточного числа фібоначчі
    pred = tim                        #Нове значення попереднього числа
print ("Послідовність фібоначчі ")    #Повідомлення
print (lst, end = " ")                #Виведення послідовності чисел фібоначчі
```

Рис. 13.22. Код створення чисел Фібоначчі

Результат виконання коду:

```
Кількість чисел  = 13
Послідовність фібоначчі
[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233]
```

Запитання для перевірки знань

1. Які вбудовані об'єкти мови Python є послідовностями?
2. Які послідовності є числами Фібоначчі?
3. Які операції й методи обов'язково підтримують послідовності мови Python?
4. Які послідовності мови Python називають змінними і незмінними?
5. Наведіть формули обчислення членів арифметичної і геометричної прогресій.
6. Які послідовності називають рекурентними?

Завдання для самостійного виконання

1. Розробіть код створення послідовності довжиною n, елементами якої є цілі випадкові числа в діапазоні від 7 до 49. Упорядкуйте послідовність за зменшенням чисел.
2. Знайдіть в інтернеті відомості про кількість населення кожного району області, у якій ви мешкаєте. Розробіть код створення числової послідовності, елементами якої є кількість населення кожного району. Визначте район із найбільшою кількістю населення. Упорядкуйте послідовність у порядку зменшення кількості населення.
3. Розробіть код створення числової послідовності, елементами якої є значення n членів геометричної прогресі зі знаменником q і першим членом a. Обчисліть суму членів прогресії.
4. Розробіть код створення числової рекурентної послідовності, елементами якої є кожне значення факторіалу n. Обчисліть суму значень членів послідовності.
5. Розробіть код створення послідовності довжиною n, елементами якої є цілі випадкові числа від 13 до 50. Видаліть повторювані числа, впорядкуйте їх за зростанням.

13.4. Вкладені цикли

Пригадайте, як розробляти прості алгоритми з повторенням і алгоритми, у яких поєднується повторення з розгалуженням.

Поміркуємо, у яких випадках виникає потреба застосовувати алгоритми з вкладеними циклами.

Вкладені цикли — це такі цикли, що містяться в іншому циклі.

Цикл, який входить до складу іншого циклу, називають внутрішнім, а цикл, який містить інший цикл, — зовнішнім.

На рис. 13.23 наведено фрагмент блок-схеми алгоритму з вкладеним циклом. У ході виконання цього алгоритму змінна S набуває таких значень:

3 + 4 = 7,
3 + 7 = 10,
3 + 10 = 13,
3 + 13 = 16,
5 + 4 = 9,
5 + 7 = 12,
5 + 10 = 15 і т. д.

Код реалізації алгоритму наведено на рис. 13.24.

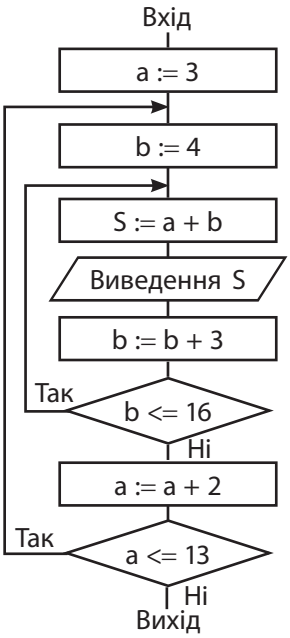


Рис. 13.23. Фрагмент блок-схеми алгоритму з вкладеним циклом

Результат виконання коду:

```
#Код із зовнішнім і внутрішнім циклами
for a in range (3, 12, 2):              #Зовнішній цикл
    print ()                            #Перехід на новий рядок
    for b in range (4, 16, 3):          #Внутрішній цикл
        S = a + b                       #Обчислення суми чисел
        print (S, end = " ")            #Виведення отриманої суми
```

```
7 10 13 16
9 12 15 18
11 14 17 20
13 16 19 22
15 18 21 24
```

Рис. 13.24. Код із зовнішнім і внутрішнім циклами

Приклад 1.

Розробити код визначення простих чисел у діапазоні від 8 до 50.

Ідея алгоритму визначення простих чисел у заданому діапазоні досить проста. Береться число N і ділиться на дільник (diln) 2. Якщо число ділиться без остачі, то це означає, що число N не просте.

Далі значення N збільшується на одиницю і також починає ділитися на 2. Якщо воно не ділиться на 2, то збільшується значення дільника на одиницю. Останнім значенням дільника, на яке слід ділити число N, щоб з'ясувати простоту числа, є значення N/diln.

Код реалізації такого підходу до визначення простих чисел зображено на рис. 13.25.

```
#Код із зовнішнім і внутрішнім циклами
N = 8                              #Початкове значення діапазону чисел
print ('Прості числа')             #Повідомлення
while (N < 50):                    #Доти, доки N менше 50
    diln = 2                       #Початкове значення дільника
    while (diln <= (N/diln)):      #Доти, доки дільник менше або дорівнює N/diln
        if not(N % diln):          #Якщо число не просте (N кратне diln)
            break                  #Вихід із внутрішнього циклу, якщо N не просте
        diln = diln + 1            #Збільшення дільника
    if (diln > N/diln):            #Якщо дільник більше N/diln
        print (N, end = " ")       #Виведення простого числа
    N = N + 1                      #Чергове значення числа
```

Рис. 13.25. Код визначення простих чисел

Результат виконання коду:

```
Прості числа
11 13 17 19 23 29 31 37 41 43 47
```

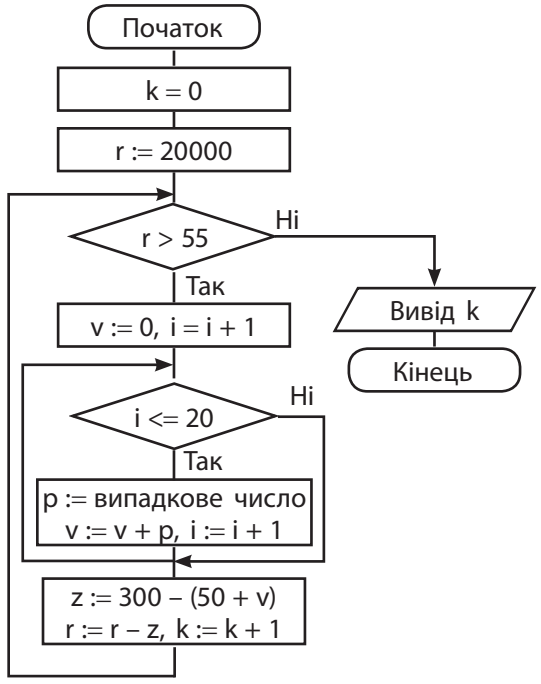


Рис.13.26. Блок-схема алгоритму визначення кількості банок огірків

Приклад 2. Автоматизована лінія консервування ніжинських огірків діє так: у контейнер наливають 10 кг розчину з приправами, у банку масою 50 г кладуть 20 огірків, масою від 8 до 12 г кожний. Визначається маса банки з огірками і доливається у банку така маса розчину, щоб загальна її маса дорівнювала 300 гр. Щойно в контейнері залишиться менше ніж 55 г розчину, робота лінії припиняється. Скільки банок огірків буде виготовлено?

Масу кожного огірка можна вважати випадковою величиною в межах 7–12 г. Блок-схему алгоритму розв'язування задачі наведено на рис. 13.26.

У блок-схемі використано такі позначення:

r — поточна маса розчину в контейнері;
v — поточна маса огірків у банці;
i — лічильник кількості огірків у банці;
k — кількість виготовлених банок огірків;
p — випадкова маса одного огірка;
z — маса розчину, який доливають у банку.

Код реалізації алгоритму наведено на рис. 13.27.

```
#Код з вкладеним циклом
import random                          #Імпортування модуля random
k = 0                                  #Початкова кількість банок огірків
r = 20000                              #Початкова маса розчину в грамах
while (r > 55):                        #Зовнішній цикл
    v = 0                              #Початкова маса огірків у банці
    for i in range (20):               #Внутрішній цикл
        p = random.randint (7,12)      #Генерування випадкової маси огірка
        v = v + p                      #Збільшення маси огірків у банці
    z = 300 - (v + 50)                 #Маса розчину, що доливається у банку
    r = r - z                          #Зменшення маси розчину в контейнері
    k = k + 1                          #Збільшення кількості склянок огірків
print ("Кількість банок = ",k)         #Виведення кількості виготовлених банок
```

Рис. 13.26. Код визначення кількості банок огірків

Результат виконання коду:

```
Кількість банок = 335
```

Запитання для перевірки знань

1. Які цикли називають вкладеними?
2. Накресліть блок-схему з вкладеним циклом.
3. У яких випадках доцільно застосовувати вкладені цикли?
4. Наведіть приклад алгоритму з вкладеним циклом.

Завдання для самостійного виконання

1. Дано 12 прямокутників, сторона a яких має значення 4, 6, 8 і 10, а сторона b — 6, 9 і 12. Розробіть алгоритм і код обчислення площ усіх прямокутників.
2. Розробіть алгоритм і код обчислення таблиць множення на 4, 5, 6 і 7.
3. Розробіть алгоритм і код визначення кількості цифр у числах: 64 649, 351, 8 376 771 905, 67 044, 55 004 329.
4. На уроці фізкультури учні та учениці об'єдналися у три команди: по чотири хлопця і три дівчини. Кожний із учнів пробігає 50 м із фіксацією часу в секундах. Розробіть алгоритм і код визначення команди-переможниці за середнім часом подолання дистанції.
5. На стадіоні довжина першої бігової доріжки становить 400 м. Спортсмен біжить дистанцію 10 000 м, а тренер фіксує час подолання кожного кола. Потім він обчислює середній час бігу спортсмена за кожні 2 км і робить необхідні висновки. Розробіть алгоритм і код обчислення середнього часу подолання кожного з двох кілометрів.

13.5. Використання об'єктів мови програмування для реалізації циклів

Пригадайте назви типів об'єктів, вбудованих у мову Python

Пригадаємо, що у мові Python все, що опрацьовується, є об'єктом, у тому числі списки, числа, словники, кнопки, перемикачі та ін. Наведемо приклади використання списків для реалізації циклів (приклади 1 і 2).

Приклад 1. Дано список цілих чисел: 5, 23, 135, 432, 15, 5, 355, 6725, 14, 75. Визначити усі числа, у яких у молодшому розряді є цифра 5.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. lst = [5, 23, 135, 432, 15, 5, 355, 6725, 14, 75].
3. k := 0 # Початкова кількість чисел, у яких в молодшому розряді є цифра 5
4. a := чергове число зі списку lst, починаючи з нульової позиції.
5. b := a%10.
6. Якщо b = 5, то п. 7, інакше — п. 9.
7. k := k+1.
8. Виведення числа a.
9. Якщо не перебрано всі значення списку lst, то п. 4, інакше — п. 10.
10. Якщо k = 0, то п. 11, інакше — п. 12.
11. Виведення таких чисел немає і перехід до п. 13.
12. Виведення значення k.
13. Кінець.

Код реалізації алгоритму наведено на рис. 13.28.

```
#Код визначення чисел, у яких в нульовому розряді є цифра 5
lst = [5, 23, 135, 432, 15, 5, 355, 6725, 14, 75] #Створення списку
k = 0                                             #Початкова кількість чисел
print ('Числа, у яких останньою є цифра 5')       #Повідомлення
for a in lst:                                     #Цикл перегляду списку
    b = a % 10                                    #Цифра молодшого розряду
    if b == 5:                                    #Якщо останньою є цифра 5
        k = k + 1                                 #Збільшення кількості чисел
        print (a, end = " ")                      #Виведення числа з цифрою 5
print ()                                          #Перехід на новий рядок
if k == 0:                                        #Якщо числа з 5 відсутні
    print ('Таких чисел немає')                   #Виведення результату
else:                                             #Якщо є числа з цифрою 5
    print ('Таких чисел = ', k)                   #Виведення результату
```

Рис. 13.28. Код визначення чисел, у яких у нульовому розряді є цифра 5

Результат виконання коду:

```
Числа, у яких останньою є цифра 5
5 135 15 5 355 6725 75
Таких чисел =  7
```

Приклад 2. У діапазоні цілих чисел від 31 до 73 із кроком 3 знайти ті, які у результаті ділення на 4 мають залишок 1 або 3.

Алгоритм розв'язування задачі у словесно-формульній формі

1. Початок.
2. Створити діапазон a чисел від 31 до 73 із кроком 3.
3. b := чергове значення числа з діапазону a, починаючі з нульової позиції.
4. Якщо (b%4) = 1 або (b%4) = 3, то п. 5, інакше — п. 6.
5. Вивести число b.
6. Якщо не перебрано всі числа діапазону a, то п. 3, інакше — п. 7.
7. Кінець.

Код реалізації алгоритму наведено на рис. 13.29.

```
#Код визначення в діапазоні чисел, які після ділення на 4 мають остачу 1 або 3
a = range (31,73,3)                                  #Діапазон чисел
print ('Числа, які після ділення на 4 мають остачу 1 або 3 ')  #Повідомлення
for b in a:                                          #Цикл перегляду чисел діапазону
    if (b % 4) == 1 or (b % 4) == 3:                 #Якщо число має в остачі 1 або 3
        print (b, end = " ")                         #Виведення чисел
```

Рис. 13.29. Код визначення чисел, які після ділення на 4 мають остачу 1 або 3

Результат виконання коду:

```
Числа, які після ділення на 4 мають остачу 1 або 3
31 37 43 49 55 61 67
```

Завдання для самостійного виконання

1. Розробити алгоритм і код його реалізації для визначення усіх чотирирозрядних чисел, сума цифр яких дорівнює 15.
2. Фермерське господарство закінчило рік із такими показниками: прибуток від продажу ячменю — 28 900 грн, прибуток від продажу пшениці — 80 500 грн, прибуток від продажу соняшнику — 75 000 грн, прибуток від продажу кукурудзи — 90 000 грн. Обчислити загальну суму прибутку господарства. Розробити алгоритм і код його реалізації.
3. Розробити алгоритм і код його реалізації для визначення всіх трирозрядних чисел, у яких немає двох і трьох однакових цифр.

Практична робота № 24

ТЕМА. Розроблення алгоритмів із послідовними повтореннями та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити алгоритми з послідовними повтореннями і програмні коди їх реалізації.

ОБЛАДНАННЯ: комп'ютер з ОС Windows 7 і старше, середовище програмування IDLE.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Відомо зріст кожного з дев'яти учасників шкільної баскетбольної команди. Розробіть алгоритм визначення середнього зросту баскетболістів.

3. Переконайтеся, що отримано правильний результат.

6. Переконайтеся, що отримано правильний результат.

9. Виконайте код і доведіть, що отримано правильний результат.

12. Виконайте код і доведіть, що отримано правильний результат.

2. Розробіть код реалізації алгоритму.
3. Уведіть код, усуньте всі помилки й виконайте його.
4. Розробіть алгоритм визначення усіх дворозрядних десяткових чисел, які у сумі з тими самими цифрами, але записаними у зворотному порядку, дають повний квадрат (наприклад, $92+29=121$, $\sqrt{121}=11$).
5. Розробіть код реалізації алгоритму.
6. Уведіть код, налагодьте його і виконайте.
7. Вводиться послідовність цілих чисел. Введення закінчується, якщо робиться спроба увести число 0. Розробіть алгоритм обчислення кількості уведених чисел і всіх чисел, які діляться на 3 без остачі.
8. Розробіть код реалізації алгоритму.
9. Уведіть код і здійсніть його налагодження.
10. Вводиться ціле n-розрядне десяткове число. Розробіть алгоритм визначення максимальної цифри у числі й номерів усіх позицій, на яких розташована ця цифра.
11. Розробіть код реалізації алгоритму.
12. Уведіть код, здійсніть його налагодження.

Зробіть висновок: у яких випадках для реалізації циклічних алгоритмів доцільно використовувати оператори for і while.

Практична робота № 25

ТЕМА. Розроблення рекурентних алгоритмів та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити рекурентні алгоритми й програмні коди їх реалізації.

ОБЛАДНАННЯ: комп'ютер з ОС Windows 7 і старше, середовище програмування IDLE.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Дано послідовність: 3, 4.5, 6.75, 8.925, ... Розробіть блок-схему алгоритму обчислення значення восьмого члена.
2. Розробіть код реалізації алгоритму.
3. Уведіть код, усуньте всі помилки і виконайте його.
4. Дано послідовність 2, 3, 5, 9, 17, ... Розробіть блок-схему алгоритму обчислення суми десяти перших її членів.
5. Розробіть код реалізації алгоритму.
6. Уведіть код, налагодьте його і виконайте.
7. Розробіть блок-схему алгоритму обчислення суми n членів послідовності: 1, 2^1, 2^2, 2^3, ..., 2^n.
8. Розробіть код реалізації алгоритму.
9. Уведіть код і здійсніть його налагодження.

3. Доведіть, що отримано правильний результат.

6. Доведіть, що отримано правильний результат.

9. Виконайте код і доведіть, що отримано правильний результат.

10 Розробіть блок-схему алгоритму визначення мінімальної кількості членів послідовності $\frac{1}{2^0}, \frac{1}{2^1}, \frac{1}{2^2}, \frac{1}{2^3}, \ldots, \frac{1}{2^n}$, сума значень яких більша за 1,892.

11 Розробіть код реалізації алгоритму.

12 Уведіть код і здійсніть його налагодження.

Зробіть висновок про особливості створення і реалізації рекурентних алгоритмів.

12. Виконайте код і доведіть, що отримано правильний результат.

Практична робота № 26

ТЕМА. Розроблення алгоритмів із вкладеними повтореннями та їх реалізація у вигляді програм

ЗАВДАННЯ: розробити алгоритми з вкладеними повтореннями та програмні коди їх реалізації.

ОБЛАДНАННЯ: комп'ютер з ОС Windows 7 і старше, середовище програмування IDLE.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1 Розробіть блок-схему алгоритму обчислення всіх значень y, де $y = 2(n+1)+3(k+2)$, для $n = 1, 2, 3$ і $k = 3, 4, 5, 6$.

2 Розробіть код реалізації алгоритму.

3 Уведіть код, усуньте всі помилки і виконайте його.

3. Переконайтеся, що отримано правильний результат.

4 *Старовинна задача*. Скільки можна купити биків, корів та телят, якщо вартість одного бика становить 10 крб, однієї корови — 5 крб, за одне теля платять по 0,5 крб і якщо на 100 карбованців потрібно купити 100 голів.

5 Розробіть алгоритм розв'язування задачі.

6 Розробіть код реалізації алгоритму. Уведіть код, налагодьте його і виконайте.

6. Переконайтеся, що отримано правильний результат.

7 Генеруються випадкові цифри, у діапазоні від 2 до 9. З цих цифр створюються чотири послідовності, кожна довжиною 6.

8 Розробіть алгоритм визначення послідовності з максимальною сумою її цифр.

9 Розробіть код реалізації алгоритму. Уведіть код, налагодьте і виконайте його.

9. Доведіть, що отримано правильний результат.

10 Дано послідовність чисел 75, 77, 79, 81, 83. Розробіть алгоритм визначення для кожного числа цієї послідовності того дільника, на який воно ділиться без остачі.

11 Розробіть код реалізації алгоритму.

12 Уведіть код, налагодьте і виконайте його. Доведіть, що отримані результати є правильним.

Зробіть висновок про можливі варіанти реалізації алгоритмів із вкладеними повтореннями.

Виконайте тестове завдання з автоматичною перевіркою результату

Тест 13

Комп'ютерний словник

Архітектура комп'ютера — набір дисциплін, які описують функціональність, організацію та реалізацію комп'ютерних систем.

Висловлювання — це речення, про яке можна говорити, що воно істинне або хибне.

Вкладені цикли — цикли, що містяться в іншому циклі.

Глибина кольору (бітова глибина) — кількість бітів (обсяг пам'яті) для зберігання й подання кольору під час кодування одного пікселя графіки або відеозображення.

Деінсталювання — процес видалення встановлених програм з комп'ютера.

Декодування — процес відновлення змісту закодованої інформації.

Дефрагментація дисків — процес, при якому здійснюється перезапис фрагментів файлів для їх розміщення в суміжних секторах диска з метою підвищення швидкості зчитування даних з жорсткого диска комп'ютера.

Дискретизація — процес вимірювання, зберігання та опрацювання значень аналогового сигналу через однакові проміжки часу, які називаються частотою дискретизації.

Ергономіка — наука, яка вивчає робочі процеси з метою створення оптимальних умов праці, що сприяє підвищенню її продуктивності.

Заповнювачі — точкові контейнери в макетах слайдів, які містять такі відомості про назви, як заголовки, текст, таблиці, діаграми, графічні елементи SmartArt, зображення, картинка з галереї, відео та звуки.

Змінна — певна ділянка пам'яті (комірки пам'яті), яка позначається іменем (ідентифікатором), у якій зберігається значення певного типу, наприклад число, слово тощо.

Зміст документа — структура, створена з форматованих заголовків із зазначенням сторінок їх розміщення в документі.

Код — система правил для перетворення форми даних.

Кодек — програма кодування відео, яка використовує алгоритми стиснення відеоданих.

Кодування — процес замінення однієї форми даних на іншу, зручну для передавання, опрацювання та зберігання за допомогою деякого коду.

Колірна модель — математична модель опису подання кольорів у вигляді послідовності чисел, кожне з яких називається колірним компонентом.

Колонтитул — напис, який розміщують у верхньому або нижньому полі сторінки.

Конвертор — комп'ютерна програма, яка перетворює дані, подані у файлі, з одного формату в інший.

Ліцензійна угода (ліцензія) — основний документ, який визначає правила використання програмного забезпечення, права та зобов'язання користувача програмного забезпечення, та яка додається до придбаного продукту у вигляді друкованого або електронного документа.

Макет — набір наповнювачів для подання контенту на слайді, що визначає, які об'єкти будуть представлені на слайді і яким чином розташовані.

Макрос — набір команд і інструкцій у вигляді єдиної команди з назвою для автоматичного виконання завдання.

Мультимедіа — комп'ютеризована технологія, яка поєднує різні способи подання даних на одному носієві.

Непозиційна система числення — система числення, у якій кількісний еквівалент кожної цифри визначається тільки цифрою і не залежить від її позиції у записі числа.

Операнд — об'єкт, над яким виконуються арифметичні операції.

Оцифрування звуку — процес перетворення звукових даних в аналоговому вигляді у закодований цифровий сигнал.

Подкаст — цифровий медіа-файл або низка таких файлів, які розповсюджуються інтер-

нетом для відтворення на портативних медіапрогравачах чи персональних комп'ютерах.

Позиційна система числення — система числення, у якій кількісний еквівалент кожної цифри у числі залежить не тільки від її значення, а й від позиції в записі числа.

Програмне забезпечення — сукупність програм, які призначені для функціонування комп'ютера, узгодження роботи апаратної складової та розв'язування різноманітних завдань.

Проєкт — файл, який містить відомості про порядок розташування та час відтворення аудіо й відеокліпів, відеопереходи, відеоефекти, назви, титри тощо.

Просторова дискретизація — перетворення аналогової форми подання графічних даних на цифрову.

Рекурентні послідовності — послідовності, у яких наступний член обчислюється через значення попередніх членів.

Розділ — частина документа з форматуванням, яке відрізняється від форматування решти документа.

Середовище програмування — комплекс програмних засобів, які призначено для автоматизації процесу підготовки та виконання програм користувача.

Система числення — сукупність правил запису чисел за допомогою символів (цифрових знаків) і виконання операцій над ними.

Слайдова презентація — послідовність слайдів — окремих екранних сторінок, кожна з яких може містити текстові, графічні, відео та аудіооб'єкти, анімаційні об'єкти, а також гіперпосилання.

Словник — набір об'єктів будь-якого типу, доступ до яких здійснюється не за допомогою індексу, а за допомогою ключа.

Список — певна сукупність об'єктів будь-якого типу у квадратних дужках, які відокремлюються один від одного комою.

Стиль — набір параметрів форматування, який має назву й визначає вигляд текстового або табличного фрагмента документа і встановлюється одночасно для автоматичної зміни всіх параметрів вигляду поточного фрагмента.

Структура документа — схема розміщення складових документа.

Умовне форматування — автоматичне форматування клітинок на основі встановлених правил для візуальної інтерпретації та аналізу значень клітинок.

Фільтрування списку — процес, в результаті якого для перегляду доступні записи, які задовольняють критеріям фільтрації. Решта записів (рядків таблиці) просто приховані.

Форматування диска — процес програмної розмітки області зберігання даних для будь-яких електронних носіїв інформації.

Формула — вираз, який складається з констант, функцій і посилань на клітинки, сполучених знаками математичних операцій.

Шаблон — зразковий документ, який зберігається у файлі з розширенням .dotx.

Шина — сукупність ліній (провідників на материнській платі), якими обмінюються інформацією компоненти і пристрої комп'ютера.

Алфавітний покажчик

А

Анімаційний ефект 132
Архівування 64
Архітектура комп'ютера 33

В

Відеохостинг 121
Висловлювання 213
Вкладені цикли 243

Г

Глибина кольору 27

Д

Дані 190
Декодування 18
Дискретизація 30
Дистрибутив 61

Е

Ергономіка 117

З

Змінна 190
Зміст документа 84

І

Інсталювання 60

К

Кодування 18
Колірна модель 26
Колонтитул 82
Конвертація файлів 111
Конвертор 99

Л

Ліцензійна угода 57
Лічильник циклу 177

М

Макет слайда 19
Макрос 89
Мова програмування 178
Мультимедіа 95

Н

Нейрон на мережа 100
Непозиційна система числення 4

О

Операнд 194
Оцифрування звуку 30, 96

П

Пам'ять комп'ютера 37
— внутрішня
— зовнішня
Подія 186
Позиційна система числення 5
Програмне забезпечення 56
Просторова дискретизація 25

Р

Рекурентна послідовность 241
Розділ 74

С

Середовище програмування 181
Система числення 4
Словник 203
Список 198
Стиль 78
Стиснення даних 63
Структура документа 81

Т

Тригер 133

У

Умовне форматування 155

Ф

Фільтрування списків 164
Формула 146

Ц

Центральний процесор 35

Ш

Шаблон 78
Шина 34

Зміст

Передмова ... 3

РОЗДІЛ 1. Математичні основи обчислювальної техніки

1.1. Поняття системи числення. Позиційні і непозиційні системи числення ... 4
1.2. Переведення чисел із десяткової системи числення в довільну і навпаки ... 7
1.3. Двійкова та шістнадцяткова системи числення. Опрацювання систем числення, основою яких є степінь двійки ... 11
1.4. Арифметичні операції у двійковій і шістнадцятковій системах числення ... 13
Практична робота № 1. Опрацювання чисел у різних системах числення ... 17

РОЗДІЛ 2. Кодування даних

2.1. Опрацювання даних як інформаційний процес. Кодування та декодування повідомлень Одиниці вимірювання довжини двійкового коду ... 18
2.2. Кодування чисел у комп'ютері ... 21
2.3. Кодування символів ... 23
2.4. Кодування графічних даних. Поняття колірної схеми ... 25
Практична робота № 2. Опрацювання колірних моделей у векторному графічному редакторі ... 29
2.5. Кодування звукових даних ... 30
Практична робота № 3. Кодування даних різного формату в комп'ютері ... 32

РОЗДІЛ 3. Комп'ютер як універсальний пристрій для опрацювання даних

3.1. Архітектура комп'ютера. Процесор, його будова та призначення. ... 33
3.2. Пам'ять комп'ютера, її види ... 37
3.3. Будова й алгоритм роботи ЕОМ за фон Нейманом ... 41
3.4. Пристрої введення-виведення даних. Пристрої, що входять до складу мультимедійного обладнання ... 44
3.5. Технічні характеристики складових комп'ютера. Визначення значень властивостей комп'ютера ... 48
Практична робота № 4. Конфігурування комп'ютера під потребу ... 51
3.6. Історія засобів опрацювання інформаційних об'єктів ... 51
3.7. Програмне забезпечення комп'ютера ... 56
3.8. Класифікація, основні функції та складові операційної системи ... 58
3.9. Інсталювання програмного забезпечення. Інсталяція та деінсталяція компонентів ОС ... 60
3.10. Архівування даних. Стиснення даних, види стиснення даних. Архіватори. Типи архівів. Операції над архівами. ... 63
Практична робота № 5. Створення архівів та операції над ними ... 65
3.11. Форматування та копіювання дисків. Дефрагментація пристроїв пам'яті з файловими системами, встановлення розкладу її проведення ... 66

РОЗДІЛ 4. Створення та опрацювання текстових документів

4.1. Створення та форматування списків у текстовому документі ... 69
4.2. Створення та форматування колонок, таблиць у текстовому документі ... 74
4.3. Стильове оформлення документів ... 78
4.4. Схема документа. Колонтитули ... 81
4.5. Автоматизоване створення змісту та покажчиків ... 84
4.6. Опрацювання складного текстового документа. Макроси ... 87
4.7. Налаштування параметрів роботи середовища текстового процесора ... 90
Практична робота № 6. Створення текстового документа, що містить об'єкти різних типів ... 92
Практична робота № 7. Використання стилів для оформлення текстових документів. Структура та зміст документа ... 93
Практична робота № 8. Автоматизоване створення покажчиків. Макроси ... 94

РОЗДІЛ 5. Створення та опрацювання об'єктів мультимедіа

5.1. Поняття мультимедіа. Кодування аудіо- та відеоданих ... 95

5.2. Програмне забезпечення для опрацювання об'єктів мультимедіа ... 98
5.3. Захоплення аудіо й відео, створення аудіо- і відеофрагментів ... 101
5.4. Побудова аудіо- й відеоряду. Додавання до відеокліпу ефектів ... 104
5.5. Налаштування часових параметрів аудіо- та відеоряду. Засоби перетворення аудіо- й відеоформатів ... 109
5.6. Сервіси для роботи з аудіо- й відеоданими та публікування їх в інтернеті. Подкаст ... 112
Практична робота № 9. Захоплення та конвертування аудіо (відео) даних ... 114
Практична робота № 10. Створення відеокліпу. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду ... 115

РОЗДІЛ 6. Мультимедійні презентації
6.1. Етапи розробки презентації. Критерії оцінювання презентації ... 116
6.2. Макети слайдів. Стильове оформлення слайдів презентації. Елементи дизайну презентацій. Використання організаційних діаграм у презентаціях ... 119
6.3. Проєктування та розробка розгалужених презентацій. Гіперпосилання і елементи керування в презентаціях ... 126
6.4. Елементи анімації в презентації ... 129
6.5. Додавання відеокліпів, звукових ефектів і мовного супроводу до слайдової презентації. Вбудовані та зв'язані об'єкти в презентаціях ... 135
6.6. Керування показом презентації ... 139
Практична робота № 11. Проєктування та розробка розгалужених презентацій за визначеними критеріями. Використання вбудованих та зв'язаних об'єктів у презентації ... 141
Практична робота № 12. Розробка презентацій з елементами анімації, відеокліпами, звуковими ефектами та мовним супроводом ... 142

РОЗДІЛ 7. Технології опрацювання числових даних у середовищі табличного процесора
7.1. Типи величин у текстовому процесорі Уведення даних ... 143
7.2. Формули. Опрацювання формул ... 146
7.3. Обчислювальні алгоритми в середовищі табличного процесора ... 148
7.4. Призначення й використання основних функцій табличного процесора ... 151
Практична робота № 13. Розв'язування задач на обчислення ... 154
7.5. Умовне форматування ... 154
Практична робота № 14. Умовне форматування ... 158
7.6. Створення та налагодження діаграм ... 158
7.7. Упорядковування даних у таблицях. Автоматичні та розширені фільтри ... 163
Практична робота № 15. Упорядкування даних у таблицях. Автоматичні та розширені фільтри ... 166
7.8. Проміжні підсумки. Створення форм уведення-виведення ... 167
Практична робота № 16. Створення форм уведення-виведення ... 170

РОЗДІЛ 8. Основи алгоритмізації та програмування
8.1. Поняття алгоритму. Властивості алгоритмів. ... 171
8.2. Способи описання алгоритмів ... 173
8.3. Базові алгоритмічні структури. Типи алгоритмів. ... 175
Практична робота № 17. Способи подання алгоритмів. Базові алгоритмічні структури. Типи алгоритмів ... 178

9. Мови програмування
9.1. Етапи розв'язування задач із використанням комп'ютера ... 178
9.2. Мова програмування як один зі способів описання алгоритмів. Класифікація мов програмування ... 180
9.3. Середовище програмування ... 183
9.4. Поняття об'єкта, властивості, події та обробники подій. Проєкт і його структура ... 186
Практична робота № 18. Робота у середовищі програмування ... 189

10. Лінійні алгоритми
10.1. Дані, змінні, константи. Прості типи даних ... 190
10.2. Арифметичні операції і вирази. Стандартні підпрограми ... 194

10.3. Структуровані вбудовані типи даних та їх опрацювання 198
10.4. Використання текстових файлів для введення та виведення даних 208
Практична робота № 19. Розроблення лінійних алгоритмів та їх реалізація у вигляді програм 211
Практична робота № 20. Розроблення лінійних алгоритмів та їх реалізація у вигляді програм з використанням текстових файлів 212

11. Елементи алгебри логіки
11.1. Висловлювання. Логічні змінні, операції і вирази 213
11.2. Розроблення і обчислення значень логічних виразів 215
Практична робота № 21. Побудова логічних виразів та їх обчислення 217

12. Алгоритми з розгалуженнями
12.1. Команда розгалуження. Умовний оператор мовою Python 218
12.2. Команда вибору 222
12.3. Вкладені розгалуження 224
12.4. Використання об'єктів мови Python для реалізації розгалужень 228
Практична робота № 22. Розроблення алгоритмів із послідовними розгалуженнями та їх реалізація у вигляді програм 230
Практична робота № 23. Розроблення алгоритмів із вкладеними розгалуженнями та їх реалізація у вигляді програм 231

13. Алгоритми з повторенням
13.1. Команди повторення. Оператори циклу 232
13.2. Поєднання повторення і розгалуження 237
13.3. Числові і рекурентні послідовності 240
13.4. Вкладені цикли 243
13.5. Використання об'єктів мови програмування для реалізації циклів 246
Практична робота № 24. Розроблення алгоритмів із послідовними повтореннями та їх реалізація у вигляді програм 247
Практична робота № 25. Розроблення рекурентних алгоритмів та їх реалізація у вигляді програм 248
Практична робота № 26. Розроблення алгоритмів із вкладеними повтореннями та їх реалізація у вигляді програм 249

Комп'ютерний словник 250

Алфавітний покажчик 252

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	у кінці року
1				
2				
3				
4				
5				

Навчальне видання

РУДЕНКО Віктор Дмитрович
РЕЧИЧ Наталія Василівна
ПОТІЄНКО Валентина Олександрівна

«ІНФОРМАТИКА»

Підручник для 8 класу з поглибленим вивченням інформатики закладів загальної середньої освіти

Рекомендовано
Міністерством освіти і науки України

Видано за рахунок державних коштів.
Продаж заборонено

Підручник відповідає Державним санітарним нормам і правилам «Гігієнічні вимоги до друкованої продукції для дітей»

Редактор *Л. А. Каюда*. Художнє оформлення *В. І. Труфена*.
Технічний редактор *А. В. Пліско*. Коректор *Н. В. Красна*

Підписано до друку 28.05.2021. Формат 84x108/16.
Папір офсетний. Гарнітура Шкільна. Друк офсетний.
Ум. друк. арк. 26,88. Обл.-вид. арк. 25,5.
Наклад 5 650 прим. Зам. № 3445.

ТОВ Видавництво «Ранок»,
вул. Кібальчича, 27, к. 135, Харків, 61165.
Свідоцтво суб'єкта видавничої справи ДК № 5215 від 22.09.2016.
Адреса редакції: вул. Космічна 21-а, Харків, 61165.
E-mail: office@ranok.com.ua. Тел. (057) 719-48-65, тел./факс (057) 719-58-67

Підручник надруковано на папері українського виробництва

Надруковано у друкарні ТОВ «Фактор-Друк»,
вул. Саратовська, 51, Харків, 61030.
Свідоцтво суб'єкта видавничої справи ДК № 5496 від 23.08.2017.
Тел. +38 (057) 717-51-85. E-mail: office@druk.factor.ua

Інформатика

З ПОГЛИБЛЕНИМ ВИВЧЕННЯМ ІНФОРМАТИКИ

Особливості підручника:

- Теоретичний матеріал, адаптований до дворівневої форми подання
- Основні алгоритми створення програм мовою Python
- Вправи практичного спрямування
- Різнорівневі запитання: репродуктивні, дослідницько-пошукові, творчі
- Приклади застосування інформаційних технологій у різних галузях

Інтернет-підтримка дозволить:

- Здійснити онлайн-тестування за кожною темою
- Ознайомитися з додатковими матеріалами до уроків

ВИДАВНИЦТВО РАНОК

ISBN 978-617-09-6960-6

2458 0922

9 786170 969606

Інтернет-підтримка