

# 9

# ІНФОРМАТИКА



Бондаренко О. О., Ластовецький В. В.,  
Пилипчук О. П., Шестопапов Є. А.

## Інформатика

«Інформатика»

підручник для 9 класу закладів загальної середньої освіти

Видавництво «Ранок»

9

# ІНФОРМАТИКА

Підручник для 9 класу  
закладів загальної середньої освіти

Харків  
Видавництво «Ранок»

# Зміст

Передмова	5
РОЗДІЛ 1. Програмне забезпечення та інформаційна безпека	
§ 1. Програмне забезпечення комп'ютера	9
§ 2. Встановлення та видалення програм	15
§ 3. Архівування даних. Архіватори	20
<i>Практична робота 1. Архівування та розархівування даних</i>	24
§ 4. Шкідливі програми та захист комп'ютера	25
§ 5. Комп'ютерні віруси. Антивірусні програми	28
§ 6. Захист комп'ютерних систем. Безпека в інтернеті	32
§ 7. Зберігання даних	38
<i>Практична робота 2. Налаштування параметрів безпеки в середовищі браузера</i>	42
РОЗДІЛ 2. Тривимірна графіка	
§ 8. Основні поняття тривимірної графіки	45
§ 9. Тривимірний редактор Blender	52
§ 10. Робота з об'єктами	59
<i>Практична робота 3. Робота з тривимірними примітивами</i>	65
§ 11. Редагування форми об'єкта. Модифікатори	66
<i>Практична робота 4. Моделювання витисканням</i>	72
§ 12. Матеріали та освітлення. Рендеринг	73
§ 13. Анімація ключовими кадрами	77
§ 14. Робота з текстовими об'єктами	81
<i>Практична робота 5. Тривимірна анімація</i>	86
РОЗДІЛ 3. Опрацювання табличних даних	
§ 15. Типи посилань на клітинки в Excel	89
<i>Практична робота 6. Розв'язування задач на обчислення</i>	93
§ 16. Логічні функції	95
§ 17. Математичні та статистичні функції	99
§ 18. Параметри сторінки. Друкування таблиці	102
§ 19. Упорядкування даних. Умове форматування	106
<i>Практична робота 7. Використання математичних, логічних та статистичних функцій табличного процесора. Умове форматування</i>	111



§ 20. Розширені фільтри. Проміжні підсумки	113
§ 21. Діаграми і графіки	116
<i>Практична робота 8. Упорядкування даних у таблицях.</i> Автоматичні та розширені фільтри	119
РОЗДІЛ 4. Системи керування базами даних	
§ 22. Бази даних і системи керування ними	123*
§ 23. Реляційна база даних	129
§ 24. Заповнення та зв'язки таблиць	137
§ 25. Впорядкування таблиць	144
§ 26. Запити	151
<i>Практична робота 9. Робота з таблицями і запитам</i>	157
§ 27. Форми	161
§ 28. Звіти	167
РОЗДІЛ 5. Алгоритми та програми	
§ 29. Поняття одновимірного масиву. Списки	176
§ 30. Функції та методи списків	180
§ 31. Опрацювання елементів списку	184
§ 32. Алгоритми впорядкування масиву	188
§ 33. Об'єкти класу Listbox	192
§ 34. Візуалізація елементів табличної величини	199
<i>Практична робота 10. Складання і виконання алгоритмів</i> знаходження сум і кількостей значень елементів табличних величин за заданими умовами	208
§ 35. Поняття двовимірного масиву. Вкладені списки	210
<i>Практична робота 11. Робота з двовимірними масивами</i> у програмі з графічним інтерфейсом	216
§ 36. Опрацювання двовимірних масивів даних	218
§ 37. Поняття складності алгоритмів	223
Комп'ютерний словник	228
Алфавітний покажчик	233

\* Починаючи зі с. 121 нумерацію сторінок вказано за авторським оригіналом.

# ДОРОГІ ДЕВ'ЯТИКЛАСНИКИ ТА ДЕВ'ЯТИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 9 класу. У 5–8 класах ви вивчали основи об'єктно-орієнтованого програмування та розробки застосунків мовою Python, ознайомилися з поштовими службами інтернету, етикетом електронного листування, онлайн-новими перекладачами і хмарними сервісами, опанували табличний процесор — програму для опрацювання даних, поданих у вигляді електронних таблиць.

У цьому навчальному році на вас теж очікує чимало цікавого та корисного, а цей підручник буде вашим надійним помічником. Підручник складається з п'яти розділів. Пригадаймо, як із ним працювати.

На початку кожного розділу вміщено матеріали рубрики «*Повторюємо*». Вони допоможуть згадати відомості, які ви вивчали з цієї теми в попередніх класах і які будуть необхідні для засвоєння нового матеріалу.

Розділ складається з параграфів і практичних робіт за темами розділу. Кожен параграф містить теоретичні відомості, приклади практичного застосування знань, вправи, питання для самоперевірки.

*Питання для самоперевірки* допоможуть з'ясувати, чи достатньо мірою ви опанували вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером.

*Вправи* складаються із завдань теоретичного і практичного спрямування. Вправи з позначкою ► містять набір окремих завдань, а вправи з позначкою ►► — завдання з описом покрокового виконання. Виконуючи ці завдання, ви набудете навичок роботи за комп'ютером.

*Практичні роботи* дають вам змогу розвинути дослідницькі навички. Щоб успішно виконати кожну з них, ви маєте повторити матеріал, вивчений на попередніх уроках, — і тоді зможете застосувати навички роботи з певними програмними засобами.

Оцінити свої знання та вміння вам допоможе *комп'ютерне тестування* в онлайн-режимі з автоматичною перевіркою результату.

У тексті підручника використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Знайдіть відповідь в інтернеті



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



Дайте відповідь на питання для самоперевірки



Виконайте тестове завдання

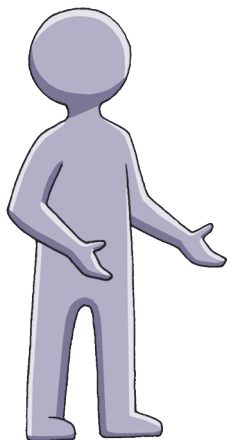
# РОЗДІЛ 1

## ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ТА ІНФОРМАЦІЙНА БЕЗПЕКА



- § 1. Програмне забезпечення комп'ютера
  - § 2. Встановлення та видалення програм
  - § 3. Архівування даних. Архіватори
- Практична робота 1. Архівування та розархівування даних
- § 4. Шкідливі програми та захист комп'ютера
  - § 5. Комп'ютерні віруси. Антивірусні програми
  - § 6. Захист комп'ютерних систем. Безпека в інтернеті
  - § 7. Зберігання даних
- Практична робота 2. Налаштування параметрів безпеки в середовищі браузера

## ПОВТОРЮЄМО



У повсякденному житті ви використовуєте багато різноманітних програм, які виконуються на персональному комп'ютері або смартфоні, керують роботою пральної машини, телевізора, роутера тощо.

Як відомо, *комп'ютерна програма* — це послідовність команд, призначених для виконання комп'ютером, і комп'ютер не працюватиме без програми.

Саме завдяки розмаїттю програм комп'ютер стає корисним: ви набираєте текст у *текстовому редакторі*, виконуєте малюнки за допомогою *графічного редактора*, готуєте презентації у середовищі *редактора презентацій*, граєте в комп'ютерні ігри, переглядаєте відео, слухаєте музику, шукаєте інформацію в інтернеті тощо.

Ви знаєте, що програми і дані в комп'ютері можуть зазнати атаки *шкідливих програм*, а щоб уникнути збитків, користувачі та користувачки мають дотримуватися *інформаційної безпеки*.

1. Що таке комп'ютерна програма?
2. Наведіть приклади використання різних програм.
3. Як захистити комп'ютер від шкідливих програм?
4. Як запобігти впливу інформації шкідливого змісту?
5. Як захиститися від загроз під час листування?
6. Зайдіть у свою електронну скриньку та вилучіть непотрібні листи.



У цьому розділі на вас чекає докладне знайомство з програмним забезпеченням комп'ютера. Ви також більше дізнаєтеся про захист комп'ютера та даних від різноманітних небезпек.

# § 1. Програмне забезпечення комп'ютера

Комп'ютер — чудовий винахід людства, в якому тісно поєднано апаратну і програмну складові. Він є основою будь-якої сучасної інформаційної системи. Проте навіть найдосконаліше комп'ютерне обладнання не може працювати без програм.



**Програмне забезпечення** — сукупність програм, які призначені для забезпечення роботи інформаційної системи.

Існує *системне, прикладне та службове* програмне забезпечення.

## Системне програмне забезпечення

**Системне програмне забезпечення** (системні програми) призначене для керування роботою складових комп'ютера та обміну даними між ними. Серед системних програм особливе місце належить **операційним системам (ОС)**. Ви вже знаєте, що ОС — це комплект програм, призначених для керування ресурсами, обчислювальними процесами комп'ютера, а також для забезпечення взаємодії з користувачем.

Після увімкнення комп'ютера ОС завантажується автоматично. Розглянемо типи ОС за деякими ознаками (рис. 1.1).



Рис. 1.1

До **основних функцій ОС** належить таке:

- введення-виведення даних, запуск і зупинка програм тощо;
- забезпечення роботи пристроїв введення-виведення;
- завантаження програм в оперативну пам'ять та їх виконання;
- керування розподілом оперативної пам'яті між програмами;
- керування доступом до даних на зовнішніх носіях;
- забезпечення інтерфейсу користувача;
- підтримка файлової системи тощо.

1 На персональних комп'ютерах використовують ОС Windows, Linux, macOS (рис. 1.2, а). Для мобільних пристроїв призначені ОС Android, Windows Mobile, iOS (рис. 1.2, б) та ін.



Рис. 1.2

### ► Сімейство ОС Windows

Вже декілька десятиліть на комп'ютерах найпопулярнішими є ОС сімейства Windows, які розробляє американська компанія Microsoft.

2 Windows 10 — комерційна (див. далі) багатозадачна локальна ОС із графічним інтерфейсом. Існують варіанти різної розрядності для персонального комп'ютера (ПК), планшетів, смартфонів тощо.

### ► Сімейство ОС Linux

Linux — це ОС різноманітного призначення, побудовані на основі однойменного ядра. Використовуються на інтернет-серверах, персональних та автомобільних комп'ютерах, телевізорах.

ОС сімейства Linux постачаються у вигляді *дистрибутивів*, тобто збірок, які включають не тільки ОС, але й набір застосунків. Більшість дистрибутивів Linux вільні, проте існують і комерційні.

3 Debian — популярний вільний багаточільовий дистрибутив, який став основою для інших дистрибутивів: Ubuntu і Linux Mint — для ПК; SteamOS — для ігрової платформи Steam Box тощо. Rocks Cluster Distribution — дистрибутив Linux для суперкомп'ютерів. OpenWrt — ОС на ядрі Linux для бездротових Wi-Fi роутерів.

На вебсайті <https://distrowatch.com/> можна ознайомитися зі стислими оглядами різноманітних дистрибутивів вільних ОС.

## ► ОС Android

На ядрі Linux ґрунтується також ОС Android, розроблена компанією Google для мобільних телефонів і планшетів. Саме під керуванням різних версій ОС Android працюють більшість смартфонів у світі.

## Драйвери

Взаємодію комп'ютера із зовнішніми пристроями забезпечують спеціальні програми — **драйвери**.

4 Під час підключення до комп'ютера нового принтера потрібно, щоб користувач із відповідними привілеями (наприклад, із правами адміністратора) встановив на комп'ютер драйвер принтера саме цієї моделі (рис. 1.3). Після встановлення в комп'ютер додаткового відеоадаптера, для користування ним також доведеться встановити його драйвер.

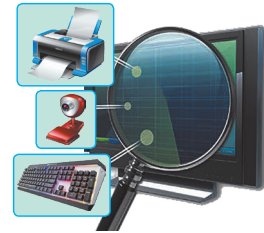


Рис. 1.3

Слід установлювати драйвер, призначений саме для встановленої на комп'ютері ОС. Драйвер для іншої ОС найпевніше працюватиме неправильно або не встановиться взагалі.

Драйвер може постачатися разом із пристроєм (наприклад, на компакт-диску), проте більшість виробників обладнання розміщують драйвери на своїх сайтах. Крім того, часто ОС, виявивши новий пристрій, підбирає драйвер із власної бази. Такий вибір не завжди найкращий, варто встановити найновішу версію драйвера від виробника.

## Прикладне програмне забезпечення

**Прикладне програмне забезпечення** (прикладні програми або застосунки) призначене для розв'язування задач користувача. Прикладні програми поділяють за призначенням. Розглянемо таблицю:

Прикладні програми	Призначення
Програми загального призначення	Різні галузі діяльності (наприклад, опрацювання текстів, малюнків, таблиць тощо)
Програми спеціального призначення	Певні галузі діяльності або конкретні підприємства (наприклад, створення відеоефектів, моделювання інтер'єрів тощо)
Інструментальні програми	Створення інших програм



## Службове програмне забезпечення

**Службове програмне забезпечення** (службові програми) призначене для обслуговування апаратної і програмної складових комп'ютера.

До службових належать програми для архівування файлів, форматування та перевірки цілісності носіїв даних тощо.



**Форматування носія даних** — процес розмічання області зберігання даних.

Форматування виконується перед першим використанням диска або у випадку виявлення серйозних збоїв у його роботі. У багатьох ОС цей процес можна здійснити за допомогою контекстного меню носія. Під час форматування наявні на носії дані стають недоступними.

У ході форматування також можна перевірити цілісність носія.

## Ліцензії на програмне забезпечення

Комп'ютерні програми є результатом праці багатьох людей, тому їх не можна використовувати без дозволу авторів (власників).



**Ліцензія на програмне забезпечення** — це угода, яка надає користувачу право встановити його і використовувати.

Закон про авторське право забороняє виконувати над програмами дії, не дозволені ліцензією (модифікувати, копіювати, розповсюджувати тощо). Існують різні види ліцензій на програмне забезпечення:

Вид ліцензії	Особливості використання програм
Комерційна ( <i>commercial</i> )	Власницькі програми (для їх використання необхідно купувати ліцензію)
Некомерційна ( <i>non-commercial use</i> )	Програми, безкоштовні для некомерційного використання (наприклад, спрощена версія комерційної програми)
Умовно-безкоштовна ( <i>shareware</i> )	Програми, якими можна користуватися безкоштовно за певних умов (низка комерційних програм надається для безкоштовного ознайомлення протягом певного періоду)
Вільна ( <i>free software</i> )	Програми, які дозволено копіювати, модифікувати, розповсюджувати практично без обмежень

5 Однією з найпопулярніших ліцензій на вільне програмне забезпечення є GNU GPL, яка надає користувачам широкий обсяг прав.

## Сумісність програмного забезпечення

ОС можуть значно відрізнятись одна від одної, тому здебільшого програма, розроблена для однієї ОС (наприклад, Windows 10), не працюватиме або працюватиме неправильно в середовищі іншої ОС (наприклад, Linux). У таких випадках кажуть про несумісність програми й ОС.

Для налаштування сумісності програми з різними версіями ОС Windows іноді достатньо вибрати в контекстному меню команду Власності → Сумісність та потрібну версію ОС.

Перед тим як купити і/або встановити програму, слід обов'язково звернути увагу на вимоги до комп'ютера та ОС.

**6** Якщо спробувати встановити тривимірний редактор Blender (з яким ви ознайомитеся далі) версії 2.93 або вище на комп'ютері з ОС Windows 7, з'явиться повідомлення про несумісність:

«This application is only supported on Windows 8.1, Windows Server 2016 or higher»

(Цей застосунок підтримується тільки Windows 8.1, Windows Server 2016 або вищою).

Програми віртуалізації (наприклад, VirtualBox корпорації Oracle) дозволяють створювати і запускати на комп'ютері так звані віртуальні машини (рис. 1.4). На віртуальну машину можна встановити ОС, що відрізняється від ОС, завантаженої на комп'ютері. Після запуску вона працює як повноцінний комп'ютер, а її робота відображається в окремому вікні. Для роботи використовується частина ресурсів (оперативної пам'яті, процесорного часу), тому це сповільнює роботу основної ОС.

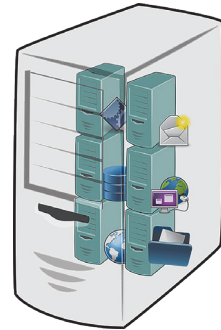


Рис. 1.4

## Питання для самоперевірки



1. Що таке програмне забезпечення комп'ютера?
2. Що таке «операційна система»? Наведіть приклади.
3. Яке призначення прикладного програмного забезпечення?
4. Яке призначення службового програмного забезпечення?
5. Які бувають ліцензії на програмне забезпечення?
6. Що таке «сумісність програм»?

## Вправа 1



- Розгляньте діаграму, яка ілюструє розвиток дистрибутивів Linux <https://bit.ly/3bgmE2o>.
- Створіть текстовий документ з назвою Вправа 1 і, спираючись на діаграму з п. 1, дайте в ньому відповіді на такі запитання.
  - Які дистрибутиви стали основою для найбільшої кількості інших, які існують нині? Назвіть принаймні три.
  - Чи існують дистрибутиви, які від моменту виникнення розвиваються донині, але не стали основою для інших дистрибутивів? Якщо так, то назвіть декілька.
  - Які дистрибутиви створено на основі початкового дистрибутиву ОС Android?
- Доберіть програмний засіб і побудуйте діаграму (карту знань) «Дистрибутиви на основі ОС Android» (рис. 1.5). Покажіть взаємозв'язки між дистрибутивами, а також включіть короткий опис кожного з них. При цьому скористайтесь вікіпедією і, за потреби, онлайнним перекладачем.



Рис. 1.5

*Підказка.* Побудувати карту знань можна за допомогою, наприклад, векторного графічного редактора. Проте існують спеціалізовані програми, за допомогою яких це робиться легше й швидше. На ПК можна скористатися вільною програмою Freeplane. Додаток Coggle Collaborative Mind Maps до вебглядача Google Chrome дозволяє створювати карти знань і працювати над ними колективно. Такі застосунки існують і для смартфонів.

- З'ясуйте, чи можна відновити дані з помилково відформатованого флешносія. Додайте коментар у файл Вправа 1.

## Комп'ютерне тестування



Виконайте тестове завдання 1 з автоматичною перевіркою результату.



## § 2. Встановлення та видалення програм

Перед початком використання програми на комп'ютері треба налагодити її взаємодію з ОС, тобто *встановити програму* на комп'ютер.

### Встановлення програм



Процес встановлення програми на комп'ютер називають також **інсталяцією програми**.

Під час встановлення програми відбувається таке:

- у постійну пам'ять комп'ютера записуються її складові (виконувані файли, файли зображень, звуків, документації тощо);
- вносяться необхідні записи до файлів ОС;
- в головному меню ОС (в ОС Windows — у меню Пуск) створюються команди для запуску програми тощо.

Особливості встановлення програм залежать від ОС, а також від конкретної програми. Багато ОС підтримують декілька способів встановлення програм.

Серед способів встановлення програм розглянемо:

- використання *встановлювальних (інсталяційних) файлів*;
- встановлення зі *сховища*;
- використання *переносних програм*.

### Встановлювальні файли

Розробник програми може поширювати її у вигляді встановлювального файлу. Це файл (виконуваний або спеціального формату), який потрібно відкрити на комп'ютері, після чого розпочнеться діалог щодо встановлення програми. В ОС Windows це може бути файл формату EXE або MSI, в ОС Linux — DEB або RPM тощо.

1

Для встановлення в ОС Windows програми Blender потрібно:

- 1) на вебсторінці [www.blender.org/download/](http://www.blender.org/download/) завантажити встановлювальний файл (наприклад, blender-2.76b-windows64.msi);
- 2) відкрити файл — при цьому запуситься програма Windows Installer, яка вестиме подальший діалог з користувачем (перехід до наступного кроку діалогу — кнопка Next);

- 3) погодитися з умовами ліцензії GNU GPL, поставивши позначку I accept the terms in the License Agreement (рис. 2.1);

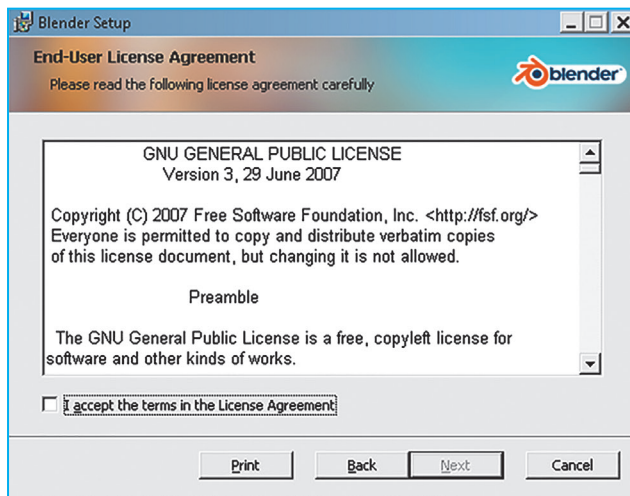


Рис. 2.1

- 4) вибрати папку для встановлення або погодитися із запропонованою;  
 5) натиснути кнопку Install — розпочнеться встановлення, по завершенні якого слід закрити вікно, натиснувши кнопку Finish. Після цього програму можна запустити за допомогою ярлика, який з'явився в головному меню ОС.

! З міркувань безпеки інсталювати програми за допомогою встановлювального файлу або зі сховища (див. далі) у більшості ОС можуть тільки користувачі та користувачки, які мають права адміністратора комп'ютера.

- 2 В ОС Windows, якщо виконано вхід в обліковий запис з обмеженими правами, можна запустити встановлювальний файл від імені адміністратора. В контекстному меню файлу слід вибрати відповідну команду, а потім у діалоговому вікні — ім'я користувача чи користувачки з правами адміністратора і ввести пароль.

! Якщо є декілька варіантів отримання встановлювального файлу, варто надавати перевагу тому, який пропонує розробник програми (наприклад, сайту розробника). Це допоможе уникнути файлів, змінених зловмисниками.

## Сховище програм

Розробники багатьох ОС пропонують не лише власне ОС, а й певний набір програмного забезпечення, яке можна встановити додатково. Ці програми зберігаються у сховищі в інтернеті, доступ до якого здійснюють за допомогою спеціальної програми.

- 3 У сучасних версіях ОС Windows доступ до сховища забезпечує програма Microsoft Store, в ОС Linux Mint — Керування програмами, а в ОС Android — Play Маркет (рис. 2.2).

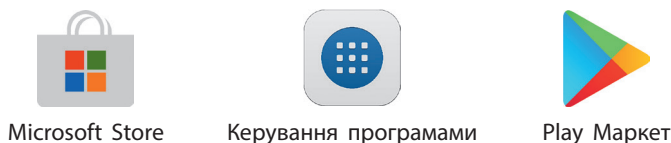


Рис. 2.2

Перевагою такого сховища є те, що вміщені в нього програми розробники перевіряють на сумісність з ОС. Це зменшує ймовірність виникнення проблем під час користування програмами. Проте, у сховищі часто містяться не найновіші версії програм, тому, якщо ОС це дозволяє, встановлювати нові версії доводиться іншим способом.

## Переносні програми

Якщо людина не планує користуватися програмою довго або не має прав адміністратора, потрібних, щоб її встановити, для ознайомлення з програмою можна скористатися її переносним варіантом.



**Переносна** (або **портативна**) **програма** (англ. *portable application* — переносний застосунок) — це програма, для запуску якої на комп'ютері не потрібне встановлення.

Для користування переносною програмою достатньо скопіювати на комп'ютер або флешнакопичувач папку з файлами програми, відшукати в ній потрібний виконуваний файл і запустити. Після цього можна для запуску програми зробити ярлик на робочому столі.

Не всі розробники надають переносні варіанти своїх програм.

- ! Якщо скопіювати папку програми, встановленої на комп'ютері за допомогою встановлювального файлу, і перенести на інший комп'ютер, то копія може виявитися непрацездатною.

4 На сторінці завантаження тривимірного редактора Blender для ОС Windows розробники пропонують такі варіанти завантаження (рис. 2.3):

- встановлювальний файл (Installer);
- переносна програма (Portable);
- посилання на сховище програм (Microsoft Store).

Для інших ОС, для яких випускається Blender, також існують різні варіанти отримання програми.

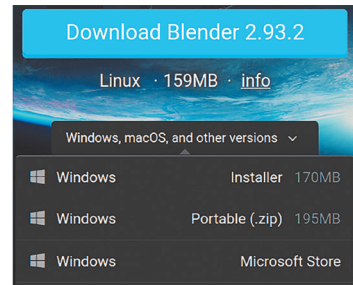


Рис. 2.3

## Видалення програм

Програму, яка більше не потрібна на комп'ютері, варто з нього видалити. Це вивільнить пам'ять, яку займають файли програми, а також усуне інші зміни, зроблені в ОС під час встановлення.


Прийнявши рішення про видалення програми, не слід зразу видаляти пов'язані з програмою папки, файли і ярлики. Натомість потрібно запустити системну програму для видалення інстальованих програм або програму деінсталяції конкретної програми. А потім у діалоговому режимі видалити програму з комп'ютера. При цьому з ОС буде вилучено зміни, зроблені під час встановлення програми (команди головного меню, асоціації файлів, файли та ярлики тощо).



Процес видалення програми з комп'ютера називають також **деінсталяцією програми**.

5 В ОС Windows 10 програма для керування встановленими програмами запускається з Панелі керування командою Програми та компоненти. Щоб видалити одну з програм, слід у вікні Програми та компоненти вибрати зі списку її назву і натиснути Видалити.

6 В ОС Linux Mint для видалення програми слід запустити програму Керування програмами, в якій потрібно:

- 1) натиснути кнопку меню  і вибрати Показати встановлені програми;
- 2) натиснути рядок відповідної програми;
- 3) натиснути кнопку Вилучити;
- 4) у діалоговому вікні ввести пароль, який підтверджує права адміністратора, і натиснути Гаразд.

- 7 Щоб видалити програму в ОС MIUI (розробка корпорації Xiaomi на основі Android), слід на її значку виконати довге натискання і вибрати з меню, що з'явиться, команду Видалити. Після цього потрібно підтвердити згоду на видалення програми та всіх її даних.

У багатьох ОС видаляти програми можна тільки маючи підвищені привілеї, наприклад, права адміністратора. Для видалення переносної програми достатньо видалити файл або папку з її файлами.

### Питання для самоперевірки



1. Що таке інсталяція та деінсталяція програм?
2. Які є способи встановлення програми на комп'ютер?
3. Що таке встановлювальний (інсталяційний) файл?
4. Які переваги та недоліки використання сховища програм?
5. Які програми називають переносними?
6. У чому перевага використання встановлювальних файлів, наданих розробником програми?
7. Чому, за потреби видалити встановлену на комп'ютері програму, не слід просто видаляти її папку?

### Вправа 2



- Установити на комп'ютер програму-архіватор PeaZip.
- 1) Відкрийте офіційну сторінку програми <https://peazip.github.io/>
  - 2) Залежно від ОС, встановленої на комп'ютері, натисніть PeaZip Linux або PeaZip Windows.
  - 3) Поцікавтесь у вчителя чи вчительки, який саме варіант потрібно звантажити (32- чи 64-бітовий), натисніть його назву, а потім натисніть Download PeaZip (Звантажити PeaZip).
  - 4) Запустіть установлювальний файл від імені адміністратора.
  - 5) У діалоговому вікні підтвердьте намір встановити програму.
  - 6) Перевірте, чи з'явилась команда для запуску програми в меню Пуск, і запустіть програму. Продемонструйте результат учителю і закрийте встановлену програму.

### Комп'ютерне тестування



Виконайте тестове завдання 2 з автоматичною перевіркою результату.





## § 3. Архівування даних. Архіватори

Людина, яка користується смартфоном, знає, що час від часу перед нею постає проблема нестачі пам'яті для програм і даних. Виявляється, у багатьох випадках можна зробити так, щоб файли займали обсяг пам'яті значно менший від їх звичного обсягу.



**Архівування даних** — процес стиснення даних, що зберігаються у файлах.

Результатом архівування є **архів** — файл, який містить у стисненому вигляді один або більше файлів. Архівування даних дозволяє заощаджувати дисковий простір, швидше передавати файл за допомогою інтернету, захищати дані паролем тощо.

### Принцип архівування даних

Поширеним методом стиснення даних є знаходження в них фрагментів, які повторюються, та їх кодування коротшими кодами. Такими фрагментами можуть бути пікселі растрового зображення, які мають однаковий колір, повторювані слова чи символи в тексті тощо.

1 Код растрового зображення (див. § 3, 8 клас), яке має великі одноколірні ділянки, містить довгі послідовності однакових чисел: 5 4 **1 1 1 1 1 1 1 1** 6 6 **1 1** 6 6 6 6 6 6 6

Побудуємо на його основі послідовність пар чисел, у яких перше число (виділене синім кольором) означає кількість повторень, а друге — код кольору, який повторюється:

**1** 5 **1** 4 **8** 1 **2** 6 **2** 1 **8** 6

Цю послідовність можна прочитати так: одна п'ятірка, одна четвірка, вісім одиниць і т. д.

Отже, послідовність із 22 чисел ми замінили коротшою послідовністю з 12 чисел, за якою легко відновити початкову.

2 Описаний у прикладі 1 спосіб архівування легко реалізувати у вигляді програми мовою Python, робота якої зрозуміла з коментарів:

```
# Вводимо рядок чисел і перетворюємо його на список
code = list(map(int,input().split()))
```

P = 0

# Позиція у списку: починаємо з 0

```

while P<len(code)-1:           # Цикл до кінця списку
    K = P                       # Допоміжна змінна

    # Шукаємо кінець чергової послідовності
    while code[P+1] == code[P]: # Поки наступне число таке саме,...
        P = P+1                 # переходимо далі
        if P == len(code)-1:    # Якщо раптом список закінчився,...
            break                # зупиняємо цикл

    # Опрацьовуємо знайдену послідовність
    K = P-K+1                    # Обчислюємо кількість однакових
    print(K, code[P], end = ' ')
    P = P+1                      # Переходимо до наступного числа

if P<len(code):                # Якщо залишилось одне число,...
    print(1, code[P])           # то опрацьовуємо його

```

Зразок вхідних і вихідних даних для перевірки програми наведено вище в прикладі 1.



**Ступінь стиснення даних ( $K$ )** — це відношення обсягу стиснених даних ( $V_a$ ) до обсягу початкових (нестиснених) даних ( $V_0$ ), яке зазвичай виражають у відсотках  $K = (V_a / V_0) \cdot 100\%$ .

У прикладі 1 ступінь стиснення становить  $K = (12 / 22) \cdot 100\% \approx 55\%$ . Ступінь стиснення залежить від типу файлу та методу стиснення.

- ! Файли деяких форматів (наприклад, JPG, MP3) майже не вдається стиснути, оскільки вміщені в них дані стискаються під час створення файлу.

3

У текстах українською мовою разом із розділовими знаками і цифрами використовується близько 60 символів. Для стиснення тексту замість 8-бітового кодування можна використати 6-бітове (оскільки  $2^6 = 64 > 60$ ). Ступінь стиснення в такому разі становитиме  $K = (6 / 8) \cdot 100\% = 75\%$ , тобто буде заощаджено 25% пам'яті.



**Розархівування (видобування) даних** — процес повернення даних, записаних в архіві, до початкового стану.

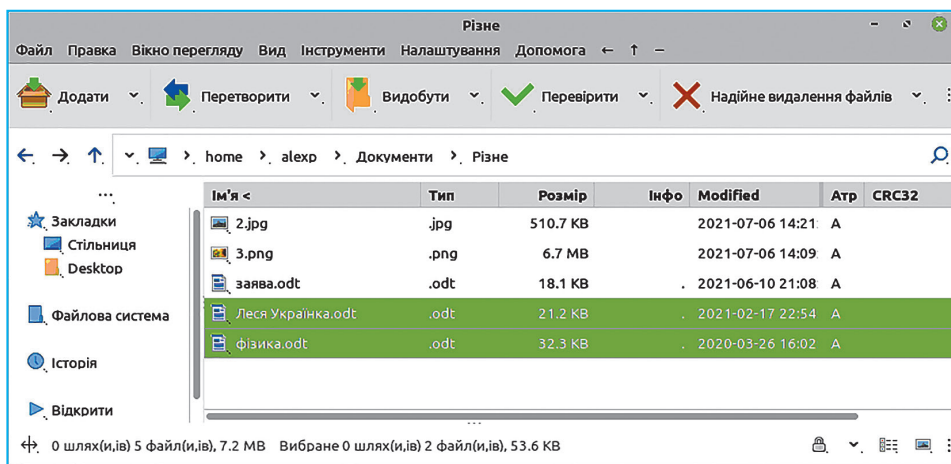
**Багатотомний архів**, тобто розділений на декілька файлів (томів), створюють тоді, коли один файл великого обсягу ускладнює подальше використання архіву, наприклад не дозволяє зберегти його на носіях меншого обсягу, надіслати електронною поштою тощо.

**Архів із саморозархівуванням** (SFX-архів, від англ. *self extracting* — самовидобування) — це виконуваний файл, який після запуску автоматично розархівується. Такі архіви доцільно використовувати, якщо на комп'ютері отримувача не встановлено відповідного архіватора.

## Архіватори

Операції архівування та розархівування здійснюються за допомогою службових програм — **архіваторів**.

Серед сучасних архіваторів можна назвати безкоштовні PeaZip (багатофункціональний) і 7-Zip, умовно безкоштовні WinRAR і WinZIP.



### ▶ Архіватор PeaZip

Архіватор PeaZip випускається для ОС Linux, Windows та ін. Він дозволяє опрацьовувати архіви більше ніж 15 різних форматів, а також різних типів: звичайні, багатотомні, із саморозархівуванням.

Для створення архіву потрібно:

- 1) запустити програму і в робочому полі вікна відкрити папку з файлами для архівування;
- 2) вибрати один або більше файлів (декілька файлів вибирають, утримуючи клавішу Ctrl);
- 3) натиснути кнопку Додати — відкриється сторінка для налаштування параметрів архіву (папки для збереження, формату, типу, розміру тому тощо);
- 4) зробити потрібні налаштування і натиснути кнопку Так.

Після цього в зазначеній папці з'явиться архівний файл із вибраною назвою і розширенням, відповідним формату.

Для **видобування файлів з архіву** потрібно:

- 1) запустити програму і в робочому полі вікна відкрити архів;
- 2) вибрати один або більше файлів;
- 3) натиснути кнопку Видобути — відкриється сторінка для налаштування параметрів (вибору папки для збереження файлів тощо);
- 4) зробити потрібні налаштування і натиснути кнопку Так.

У зазначеній папці з'являться файли, видобуті з архіву.

Крім операцій архівування та розархівування, сучасні архіватори дозволяють виконувати й інші дії: наприклад, переглядати вміст архіву, додавати файли до архіву, вилучати окремі файли з архіву, перевіряти цілісність архівного файлу тощо.


### Питання для самоперевірки



1. Що таке архівування і розархівування даних?
2. Опишіть, яким способом можна зменшити обсяг даних, не втративши їх.
3. Поясніть зв'язок між двома рядками чисел із прикладу 1.
4. Що називають ступенем стиснення даних?
5. Який ступінь стиснення має архів обсягом 51,4 КіБ, якщо початковий обсяг файлів становить 5,9 МіБ?
6. Наведіть декілька прикладів архіваторів.
7. Як передати архів обсягом 50 МіБ через службу миттєвих повідомлень, якщо вона дозволяє передавати файли обсягом до 10 МіБ?

### Вправа 3



1. Випробуйте програму з прикладу 2.
  2. Змініть програму з прикладу 2 так, щоб дані для архівування вводилися з текстового файлу, а результат виводився в інший текстовий файл (див. § 34 в підручнику для 8 класу).
- 
3. Складіть програму для розархівування даних із послідовності чисел, виведеної програмою з прикладу 2.

### Комп'ютерне тестування

Виконайте тестове завдання 3 з автоматичною перевіркою результату.





## Практична робота 1

### Архівування та розархівування даних

**Завдання:** здійснити за допомогою програми-архіватора стиснення даних в архіви різних типів, виконати розархівування, порівняти ступінь стиснення даних.

**Обладнання:** комп'ютер зі встановленим архіватором, папка Archive з файлами для архівування.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

1. Запустіть текстовий процесор. Визначте загальний обсяг файлів папки Archive, наданої вчителем. Створіть новий документ і запишіть у нього отримані дані.
2. Створіть звичайний архів папки. Визначте обсяг цього архіву та запишіть дані в документ.
3. Визначте ступінь стиснення архіву. Додайте результат у документ.
4. Створіть багатотомний архів папки із трьох томів приблизно однакового розміру. Знайдіть загальний обсяг трьох томів архіву і допишіть дані в документ.
5. Створіть саморозпаковуваний архів папки. Визначте обсяг архіву і запишіть результат у документ. Видобудьте з архіву папку, змінивши її назву.
6. Додайте в документ таблицю результатів і заповніть її даними, отриманими під час виконання попередніх пунктів.

Вид архіву	Обсяг файлу	Коефіцієнт стиснення, %
Звичайний		
Багатотомний		
Саморозпаковуваний		

Збережіть файл з назвою Практична робота 1.

Заверште роботу за комп'ютером.

**Зробіть висновок:** як за допомогою архіватора створити звичайний, багатотомний, саморозпаковуваний архіви; чому, на вашу думку, архівні файли різних типів мають різний обсяг.

## § 4. Шкідливі програми та захист комп'ютера

Відтоді, як комп'ютер з наукових лабораторій вийшов у широкий світ, пройшло багато років. Комп'ютер став доступним кожному. Але, на жаль, не всі люди використовують його з добрими намірами, що спричинило появу різних шкідливих програм.

### Класифікація загроз

Шкідливі програми, залежно від мети зловмисника, можуть мати різне призначення, а для їх проникнення на комп'ютер та маскування застосовуються різноманітні технології (рис. 4.1).



Рис. 4.1

Існують програми, які дозволяють **віддалено керувати комп'ютером** і, завдяки цьому, використовувати його без відома власника для своїх потреб (розсилання спаму, проведення атак на інші комп'ютери тощо). Такий комп'ютер іноді називають *зомбі*.

**1** Для створення (**майнінгу**, від англ. *mining* — видобування) криптовалюти (наприклад, біткоїнів) потрібно виконувати багато комп'ютерних обчислень. Якщо на комп'ютер потрапила шкідлива програма-криптомайнер, то вона може запускатися і виконувати обчислення, наприклад, тоді, коли на комп'ютері ніхто не працює. Комп'ютер працюватиме доволі інтенсивно, споживаючи, без відома власника чи власниці, багато електроенергії.

- 2 Людина, яка має контроль над багатьма комп'ютерами, може організувати атаку на відмову в обслуговуванні (DoS-атака, від англ. *Denial-of-service* — відмова в обслуговуванні), за якої багато комп'ютерів — так званий **ботнет** — одночасно звертаються до атакованої системи (наприклад, до сайту установи), надсилаючи безліч запитів, яких вона не встигає опрацювати. Корисні запити «губляться» в потоці і можуть залишитися невиконаними.

**Програми-вандали** мають на меті пошкодити дані, що зберігаються на комп'ютері, або встановлені на ньому корисні програми.

Якщо зловмисник, отримавши доступ (можливо, через мережу) встановить на комп'ютер **шпигунську програму**, він може заволодіти секретною інформацією і спричинити неприємні наслідки.

- 3 «Клавіатурний шпигун» (англ. *keylogger*, від *key* — клавіша і *logger* — реєструвальний пристрій) — це програма або пристрій, що реєструє всі натиснення клавіш на клавіатурі комп'ютера, зберігаючи зібрані дані у файлі або надсилаючи власнику через мережу. Така програма корисна, якщо потрібно забезпечити посилений контроль за використанням комп'ютера. А зловмисник завдяки їй може отримати дані, які вводилися з клавіатури.

### Антишпигунські програми

Для усунення із системи шпигунської програми її потрібно спочатку виявити. Для цього застосовують так звані **антишпигунські програми** (англ. *antispyware*). Такі програми перевіряють різні місця в ОС та застосунках, де можуть «переховуватись» небажані програми.

- 4 Власницька (є обмежена безплатна версія) програма SpyHunter дозволяє знаходити й видаляти шпигунські та інші шкідливі програми.

Багато антивірусів мають антишпигунські компоненти.

### Надійне видалення даних

Іноді конфіденційна інформація може потрапити до інших осіб без зловмисних зусиль з їхнього боку. Розглянемо можливі варіанти.

- 5 Перед продажем ноутбука з нього видалили папки із власними документами і фотографіями. Проте під час видалення файлу засобами ОС записані в ньому дані залишилися на диску і їх можна відновити за допомогою спеціальних програм.

Уникнути описаної у прикладі 5 проблеми допоможе застосування програми для надійного видалення даних. Відповідні засоби має, наприклад, розглянутий раніше архіватор PeaZip (рис. 4.2).

- ✗ Швидке видалення
- ✗ Видалення з затиранням нулями (Alt + Del)
- ✗ Надійне видалення файлів (Ctrl + Del)

Затирання нулями вільного простору  
Надійне очищення вільного простору

Рис. 4.2

- **Швидке видалення** не відрізняється від видалення засобами ОС і не зачіпає записаних у файлі даних.
- **Видалення з затиранням нулями** передбачає перезапис місця на носії, зайнятого файлом, послідовністю нулів.
- Під час **надійного видалення** файлу на зайняте ним місце декілька разів поспіль записуються випадкові дані. Такий спосіб видалення даних потребує виконання багатьох операцій запису на носій, що зменшує його термін служби.

### Питання для самоперевірки



1. З якою метою створюють шкідливі програми?
2. Які є способи проникнення шкідливих програм на комп'ютер?
3. Для чого використовують програми-криптомайнери?
4. Опишіть принцип роботи «клавіатурного шпигуна».
5. Як працюють антишпигунські програми?
6. Коли слід застосовувати надійне видалення даних із носія?

### Вправа 4



- ▶▶ Створити презентацію для виступу про шкідливе програмне забезпечення на основі інформації з інтернету.
- 1) Запустіть редактор презентацій, виберіть шаблон оформлення.
  - 2) На титульному слайді зазначте тему й дані про себе.
  - 3) Створіть слайд про комп'ютерні віруси та захист від них.
  - 4) Створіть слайд про мережеві хробаки та захист від них.
  - 5) Створіть слайд про троянські програми та захист від них.
  - 6) На останньому слайді розмістіть посилання на джерела матеріалу презентації. Збережіть презентацію з назвою Вправа 4.

### Комп'ютерне тестування



Виконайте тестове завдання 4 з автоматичною перевіркою результату.





## § 5. Комп'ютерні віруси. Антивірусні програми

1961 року науковці з фірми Bell Telephone Laboratories (США) створили ігрову програму Дарвін, у якій влаштовували в пам'яті комп'ютера «змагання» між програмами-«організмами». Кожен «організм» намагався створювати власні копії і знищувати «організми-конкуренти». Нині програми, здатні до самокопіювання, ми називаємо комп'ютерними вірусами.

### Комп'ютерні віруси



**Комп'ютерний вірус** — це програма, яка копіює та приєднує свій програмний код до інших програм і може негативно впливати на їх виконання та роботу комп'ютера в цілому.

Програма, до якої приєднався вірус, стає «зараженою», і заражає інші. Через деякий час програми перестають працювати або працюють неправильно, робота сповільнюється, а з часом комп'ютер може повністю втратити працездатність.

### Класифікація вірусів

Сьогодні існує десятки тисяч різноманітних вірусів. Розглянемо, як їх можна класифікувати за деякими ознаками:

За середовищем існування	<b>Файлові</b> — заражають виконувані файли і допоміжні програми
	<b>Завантажувальні</b> — заражають завантажувальний сектор диска
	<b>Макровіруси</b> — заражають файли документів, які підтримують роботу макросів (вбудованих програм).
	<b>Мережеві</b> — розповсюджуються мережею
За зовнішнім виглядом	<b>Звичайні</b> — програмний код вірусу видно на диску
	<b>Невидимі (замасковані)</b> — код вірусу не видно на диску
	<b>Поліморфні</b> — програмний код вірусу видозмінюється
За результатами діяльності	<b>Безпечні</b> — не виконують шкідливих дій, крім свого поширення і виконання безпечних дратівливих ефектів
	<b>Небезпечні</b> — виконують шкідливі дії, що призводять до втрати інформації і руйнування обчислювальної системи

Як немає ліків від усіх хвороб, так немає програмного засобу, який би гарантував виявлення і знищення всіх вірусів, але боротися з вірусами допомагають *антивірусні програми*, які ще називають *антивірусами*.

## Антивірусні програми

Є десятки антивірусів, які можна класифікувати так:

Антивірус	Опис
Детектори	Виявляють файли, заражені вірусами
Лікарі (фаги)	«Лікують» заражені програми або диски, видаляючи із заражених програм вірусний код
Ревізори	Спочатку запам'ятовують стан програм і дисків, а потім порівнюють їх поточний стан з попереднім і повідомляють про виявлені невідповідності
Фільтри	Перехоплюють ті звернення до системи, які віруси використовують для розмноження і заподіяння шкоди

Сучасні антивірусні програми здебільшого є комплексними застосунками, тобто використовують для боротьби з вірусами декілька з перелічених у таблиці варіантів. До складу більшості ОС такі програми не входять, тому їх необхідно встановлювати окремо.

- 1 До популярних антивірусів належать такі програми, як Zillya!, Avira, Bitdefender, AVG, Avast, ESET та ін. (рис. 5.1).

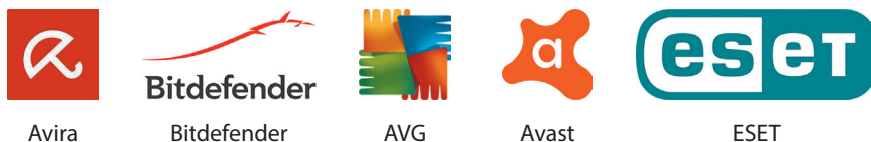


Рис. 5.1

- 2 У складі ОС Windows останніх версій постачається антивірусна програма Microsoft Defender (назва декілька разів змінювалась).

Під час встановлення антивірусна програма здебільшого налаштовується на автоматичний запуск і далі постійно контролює підозрілі дії на комп'ютері. Надалі користувач або користувачка може самостійно вимикати і вмикати такий режим роботи програми.

Примусово перевірити певний файл чи папку на віруси можна командою контекстного меню.

### ▶ Антивірус Zillya!

Антивірус українського виробництва Zillya! (рис. 5.2) захищає комп'ютер, на який установлений, від вірусів, троянських, шпигунських та інших шкідливих програм. Його якість підтверджено сертифікатом Державної служби спеціального зв'язку та захисту інформації України.

Антивірусна програма розробляється від 2009 року для ОС Windows, а від 2016 року — ще й для ОС Android (рис. 5.3) (від 2018 — під назвою Zillya! Internet Security&Scanner for Android). Для користувачів, яким потрібен тільки базовий захист, випускається безплатна версія.

Отримати Zillya!, а також дізнатися більше про різні його версії можна на сайті програми <https://zillya.ua/>.

Ніколи не можна бути впевненим, що на комп'ютері всі віруси знешкоджено, адже вірус спочатку з'являється, розповсюджується і лише через деякий час для його знешкодження розробляється антивірусна програма. Крім того, багато вірусів мають «інкубаційний період» — спочатку непомітно поширюються і лише через деякий час дають про себе знати. Отже, краще запобігати зараженню вірусами, ніж їх лікувати.



Рис. 5.2

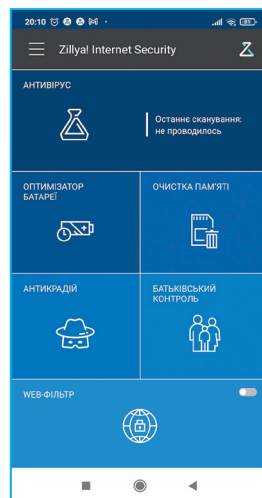


Рис. 5.3

- 3 На сайті <https://www.virustotal.com> можна перевірити підозрілий файл одночасно за допомогою більше ніж 60 антивірусів.

## Запобігання (профілактика) зараженню вірусами

Механізм передавання «інфекції» у комп'ютерів подібний до розповсюдження епідемії серед людей.

Ось кілька порад, щоб запобігти зараженню через носії даних.

- Остерігайтесь використання власних записуваних носіїв на чужих комп'ютерах.
- Якщо уникнути «контакту» з чужим комп'ютером не вдається, то після роботи перевіряйте свої носії на наявність вірусів.
- Не дозволяйте стороннім особам зі своїми носіями працювати на вашому комп'ютері.

- Якщо використання чужих носіїв уникнути не вдається, то перед використанням обов'язково перевірте їх на віруси.
- Уникайте неліцензованих програм, які часто є носіями вірусів.
- Не відключайте захисні програми: антивірус та брандмауер.
- Періодично робіть антивірусну перевірку комп'ютера.
- Періодично оновлюйте версії ОС, антивіруса та вірусної бази.

### Питання для самоперевірки



1. Якою особливістю комп'ютерні віруси відрізняються від інших програм?
2. Що означає вислів «заражена програма»?
3. Які віруси називають поліморфними?
4. Чим відрізняються антивірусні програми «детектори» і «лікари»?
5. Як перевірити, чи не заражений файл, якщо на комп'ютері немає антивіруса, але є доступ до інтернету?
6. Що слід робити, щоб уникнути зараження комп'ютера вірусом?

### Вправа 5



- ▶▶ Навчитися працювати з антивірусною програмою.
- 1) Запустіть текстовий процесор і створіть новий документ.
  - 2) Визначте, яка антивірусна програма встановлена на вашому комп'ютері, уведіть її назву у створений документ.
  - 3) Визначте дату останнього оновлення антивірусної програми і запишіть у документ.
  - 4) Визначте дату останньої перевірки комп'ютера антивірусною програмою та запишіть у документ.
  - 5) За допомогою контекстного меню або вікна антивірусної програми перевірте на віруси один із файлів (папок). Зробіть скриншот та вставте його в документ.
  - 6) Збережіть файл документа з назвою Вправа 5 у відповідній папці. Заверште роботу за комп'ютером.

### Комп'ютерне тестування



Виконайте тестове завдання 5 з автоматичною перевіркою результату.



## § 6. Захист комп'ютерних систем. Безпека в інтернеті

Із проникненням інформаційних технологій в усі сфери людської діяльності зростає потреба в захисті інформації. Адже нині від надійності роботи комп'ютерної системи може залежати результат хірургічної операції, посадка літака, робота атомної електростанції тощо.

### Загрози безпеці даним

Загрози безпеці та пошкодження даних в інформаційних системах можна класифікувати за чинниками, що їх спричиняють (рис. 6.1).



Рис. 6.1

Далі розглянемо захист від дій, які може спричинити людина.

### Етичні й правові основи інформаційної безпеки

Нині в суспільстві інформаційна безпека відіграє провідну роль.



**Інформаційна безпека** — це захищеність даних та інформаційної системи від випадкового або навмисного пошкодження та несанкціонованого доступу.

Засоби і методи підтримки інформаційної безпеки мають різне призначення. Ознайомимось з ними детальніше (рис. 6.2).



Рис. 6.2

Згадаймо вже відомі вам морально-етичні правила інформаційної безпеки (рис. 6.3).

- Не використовуйте неліцензійне програмне забезпечення.
- Робіть посилання на використаний ресурс.
- Використовуйте антивірусні програми.
- Створюйте надійні паролі під час реєстрації акаунтів.
- Не розголошуйте приватну інформацію стороннім особам.
- Обмежуйте доступ до комп'ютера сторонніх осіб.
- Будьте ввічливими під час інтернет-спілкування.
- Не відкривайте листи й файли, отримані від незнайомих.
- Не надсилайте СМС-повідомлення для отримання «безкоштовних» подарунків тощо.



Рис. 6.3

» В Україні прийнято низку законів та постанов щодо забезпечення інформаційної безпеки: «Про захист інформації в інформаційно-телекомунікаційних системах», «Про державну таємницю», «Про захист персональних даних», «Про авторське право та суміжні права» та ін.

- ! Незаконне втручання в роботу комп'ютерів і мереж, у розповсюдженні вірусів тягне за собою кримінальну відповідальність (ст. 361 Кримінального кодексу України).

## Загрози, що виникають під час роботи в інтернеті

Розглянемо основні типи шкідливих програм та загроз.

Загроза	Опис
Хробаки (черви)	Програми, які самостійно поширюються мережею, не «інфікуючи» інших файлів
Троянські програми	Програми, що поширюються під виглядом нешкідливих програм та виконують несанкціоновані дії: викрадають інформацію (паролі, рахунки тощо), передають її злочинцям через інтернет, самостійно відкривають сайти для зміни рейтингів, хакерських атак тощо
Скрипт-віруси	Програми, що потрапляють на комп'ютер через електронну пошту, маскуючись під документи
Дропери	Виконувані файли, які самі не є вірусами, але призначені для встановлення шкідливих програм
Боти	Програми, які дають можливість зловмиснику таємно керувати вашим комп'ютером
Шпигунські й рекламні програми	Програми, що зазвичай встановлюються на комп'ютер разом із неліцензованими програмами й збирають конфіденційну інформацію або демонструють нав'язливу рекламу
Фішинг	Вид інтернет-шахрайства: виманювання конфіденційної інформації через підробні сайти, які копіюють сайти відомих банків, інтернет-магазинів тощо, або за допомогою спаму

## Брандмауери

Для запобігання інтернет-загрозам між комп'ютером і мережею встановлюють перешкоди — міжмережеві екрани (брандмауер від нім. *Brandmauer*; файрвол від англ. *Firewall* — «вогняна стіна»).



**Брандмауер** — це технічний пристрій (маршрутизатор, роутер тощо) або програмний засіб для контролю даних, що надходять до комп'ютера через мережу.

Брандмауери допомагають запобігти проникненню на комп'ютер зловмисників або шкідливих програм, але не запобігають витоку персональної інформації та завантаженню користувачами вірусів.

» ОС Windows має вбудований персональний брандмауер. Щоб увімкнути і налаштувати його, слід вибрати команду: Пуск → Панель керування → брандмауер Windows.

## Засоби браузера, призначені для гарантування безпеки

Сучасні браузери мають вбудовані засоби захисту від низки інтернет-загроз.

1 До популярних нині браузерів належать Mozilla Firefox, Safari, Google Chrome, Opera, Microsoft Edge (рис. 6.4).



Mozilla Firefox



Safari



Google Chrome



Opera



Microsoft Edge

Рис. 6.4

На багатьох комп'ютерах, телефонах і планшетах встановлюють браузер Google Chrome, який: попереджає про відкриття сайта із загрозою фішингу або шкідливих програми; ізолювано відкриває вебсторінки, що в разі загрози приводить до закриття лише однієї шкідливої сторінки; дозволяє скасувати збереження конфіденційних даних; надає можливість керувати показом спливних вікон.

Можна налаштувати й інші дозволи або обмеження для інформаційного наповнення вебсторінок (анімація, малюнки тощо) або посилянь на сайти (місцезнаходження тощо).

Захист даних може забезпечуватися за допомогою електронно-цифрового підпису, що ідентифікує автора електронного документа.



**Електронний підпис** — це особливий код, цифровий аналог звичайного підпису, яким можна скріпити всі електронні документи, що гарантуватиме їх цілісність і секретність.

Для створення власного електронного цифрового підпису потрібен підтверджувальний сертифікат, який особа отримує після подання постачальнику підписів заяви із зазначенням персональних даних.

## Захищені сайти

Для забезпечення інформаційної безпеки недостатньо захистити дані на комп'ютері-клієнті або сервері — зловмисник може перехопити дані під час обміну ними по каналах зв'язку. Захист даних під час передавання забезпечує протокол захищеного зв'язку https.



**Захищений сайт** — це сайт, який використовує для обміну даними протоколи захищеного зв'язку.



- 2 Підтримка захищених каналів зв'язку потрібна сайтам, які працюють із персональною інформацією користувачів. Це можуть бути урядові сайти, сайти банків, інтернет-магазинів тощо. Під час відкриття такого сайту в адресному рядку відображається *пиктограма замка* і позначка одного з *протоколів захищеного зв'язку*, наприклад `https://` (рис. 6.5), що свідчить про захищеність каналу зв'язку цим сайтом.

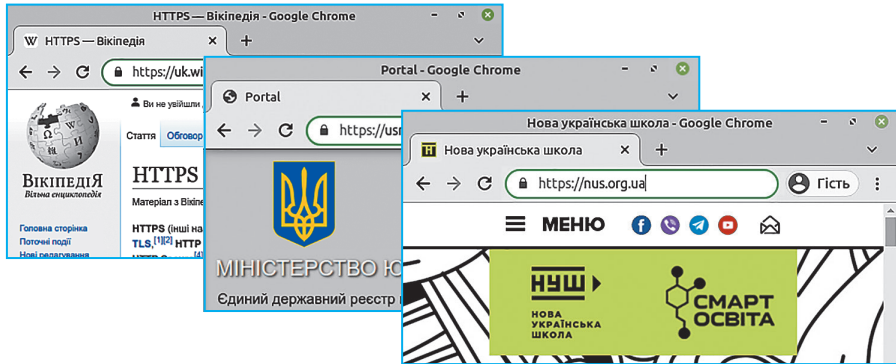


Рис. 6.5

## Захист від спаму

Крім зазначеного, існують непрямі загрози, наприклад спам.

**Спам** — це небажана пошта переважно рекламного характеру. Спам може містити посилання на небезпечні сайти, привабливі пропозиції перерахувати гроші на певні рахунки тощо.

Спамери збирають адреси потенційних «клієнтів» спеціальними програмами-роботами; найнадійніший спосіб захисту від спаму — не дати можливість роботам упізнати вашу електронну адресу.

Наведемо кілька порад, як захиститися від спаму.

- Не публікуйте свою адресу й телефони на сайтах.
- Якщо адресу доводиться публікувати, то закодуйте її на кшталт `r_a_n_o_c(a)_i_u_a` або подайте у вигляді картинки, що дозволить розпізнати адресу тільки людині.
- Іноді доцільно створити додаткову поштову скриньку для реєстрації в службах, які можуть розповсюджувати спам.
- Не реагуйте на спам (не переходьте за посиланнями тощо), бо такі дії можуть підтвердити спамерам активність електронної скриньки, що збільшить кількість спаму на вашу адресу.

Програмні засоби боротьби зі спамом — **спам-фільтри** — можуть бути складовими антивірусних програм або послугою поштових серверів. Багато поштових серверів дозволяють користувачам налаштувати власні фільтри й правила обробки поштових надходжень на основі набору символів з адреси відправника, фрагментів теми листа тощо.

### Питання для самоперевірки



1. Які загрози існують при роботі в інтернеті?
2. Яке призначення брандмауерів?
3. Які засоби захисту від інтернет-загроз є у браузерів?
4. Що таке захищені сайти?
5. Що таке фішинг; спам?
6. Назвіть відомі вам способи захисту від спаму.

### Вправа 6



- Навчитися працювати з поштою, брандмауером та із захищеними сайтами.
- 1) Запустіть текстовий процесор і створіть новий документ.
  - 2) Визначте, чи увімкнено брандмауер Windows на вашому комп'ютері. Зробіть скриншот вікна налаштування брандмауера та збережіть зображення вікна в документі.
  - 3) Запустіть браузер та відкрийте один із захищених сайтів. Зробіть скриншот вікна та збережіть зображення в документі.
  - 4) Запишіть у документ пояснення, чому, на вашу думку, знайдений сайт використовує захищений канал зв'язку.
  - 5) Відкрийте свою поштову скриньку та позначте лист в одній із папок як спам.
  - 6) Відкрийте папку Спам, зробіть скриншот вікна з вмістом папки та збережіть зображення в документі.
  - 7) Відновіть переміщений до папки Спам лист як Не спам і перевірте його появу у вихідній папці.
  - 8) Збережіть файл документа з назвою Вправа 6. Заверште роботу за комп'ютером.

### Комп'ютерне тестування



Виконайте тестове завдання 6 з автоматичною перевіркою результату.



## § 7. Зберігання даних

Надійність сучасних засобів зберігання інформації полягає у надзвичайно ефективних способах передавання й копіювання даних, що робить розповсюджену інформацію практично незнищеною. Сьогодні тривають роботи щодо збільшення довговічності і самих носіїв даних.

» 2013 року вчені Інституту проблем реєстрації інформації Національної академії наук України розробили технологію зберігання інформації на оптичних сапфірових дисках (рис. 7.1), які здатні зберігати дані впродовж десятків тисяч років. На цю роботу їх надихнули таблички шумерів, які зберігають інформацію вже понад 4 тисячі років.



Рис. 7.1

### Безпечне зберігання даних на комп'ютері

Серед заходів безпеки перше місце займає особиста організованість людини і відповідальне ставлення до інформації, що зберігається на ПК.

#### ► Захист конфіденційних даних

Якщо ваш комп'ютер використовується кількома особами, то для захисту конфіденційних даних слід зробити власний обліковий запис адміністратора, встановити пароль і зберігати файли у папках власної бібліотеки.

Щоб захистити інформацію, можна заархівувати її із заданням пароля. Це також доцільно використовувати і при листуванні.

#### ► Захист даних від пошкодження

Для збереження інформації та її захисту від пошкодження варто розділити жорсткий диск на кілька логічних розділів.

1 Зазвичай жорсткий диск ділять на диски (розділи) C, D і т. д. (рис. 7.2). На диску C встановлюють ОС і програмне забезпечення. На решті дисків зберігають інші дані. Це сприяє збереженню даних у разі пошкодження системного диска, наприклад, вірусом.

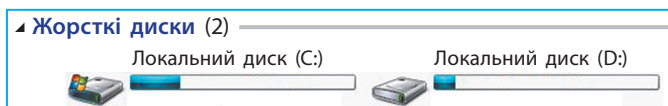


Рис. 7.2

Наведемо кілька порад для запобігання втраті важливих даних.

- Не зберігайте важливої інформації на системному диску, робочому столі, у папках власної Бібліотеки (Моя музика, Мої документи тощо), бо системний диск найчастіше зазнає впливу шкідливих програм.
- Інформацію, втрата якої є небажаною, зберігайте на зовнішніх накопичувачах: зовнішніх жорстких дисках, флешносіях, хмарних дисках, оптичних CD- і DVD-дисках тощо.

**!** Після збереження даних на флешносіях слід дотримуватися безпечного вилучення пристрою, щоб не втратити дані, які не встигли скопіюватись на носій із буфера запису.

- Для резервного копіювання файлів і можливості відновлення операційної системи користуйтеся утилітами ОС (див. далі).

## Резервне копіювання та відновлення даних

У складі багатьох ОС є службові програми для створення резервних копій даних жорсткого диска.

### ► Резервне копіювання

У Windows для **резервного копіювання** слід виконати такі дії:

- 1) відкрити вікно Архівация та відновлення даних, виконавши: Пуск → Всі програми → Обслуговування → Архівация та відновлення;
- 2) виконати команду Налаштувати резервне копіювання (або Змінити параметри). Вказати місце збереження резервних копій;
- 3) у вікні Налаштування архівації вказати спосіб вибору даних: Надати вибір Windows або Надати вибір мені → кнопка Далі;
- 4) при ручному способі задавання об'єктів для архівації (Надати вибір мені) потрібно позначати прапорцями, що саме архівуватиметься, та клацнути кнопку Далі;
- 5) установити або скасувати розклад автоматичного створення архівів: команда Змінити розклад;
- 6) натиснути кнопку Зберегти параметри та запустити архівацію.

### ► Відновлення даних

Для **відновлення файлів із резервних копій** виконують такі дії:

- 1) відкрити вікно Архівация та відновлення даних, виконавши: Пуск → Всі програми → Обслуговування → Архівация та відновлення;
- 2) натиснути кнопки Відновити мої файли → Огляд файлів;
- 3) знайти файли для відновлення і натиснути кнопку Додати файли;
- 4) після вибору файлів натиснути кнопку Далі;

- 5) вибрати місце для відновлення файлів: У вихідне місце або В наступне місце (вказавши місце для файлів);
- 6) натиснути кнопку Відновити.

## Точка відновлення Windows

Спостереження і запам'ятовування стану ОС та встановлених програм може відбуватись і в автоматичному режимі завдяки створенню так званих «точок відновлення».



**Точка відновлення Windows** (контрольна точка відновлення) — це збережена інформація про стан операційної системи і встановлених програм у певний час.

Точку відновлення доцільно створювати тоді, коли комп'ютер і програми працюють нормально. ОС створює точки відновлення системного диска автоматично після встановлення програм, зміни налаштувань тощо. Точку відновлення можна встановити в ручному режимі: Пуск → Панель керування → Система і безпека → Система → Захист системи → Далі → Створити (рис. 7.3) → надати опис точки відновлення → Створити.

Для відновлення видалених файлів можна скористатись функцією контекстного меню диска Відновити попередню версію диска, для якого був увімкнений захист системи. Далі слід вибрати: Налаштувати (рис. 7.3) → Відновити параметри системи та попередні версії файлів (рис. 7.4).

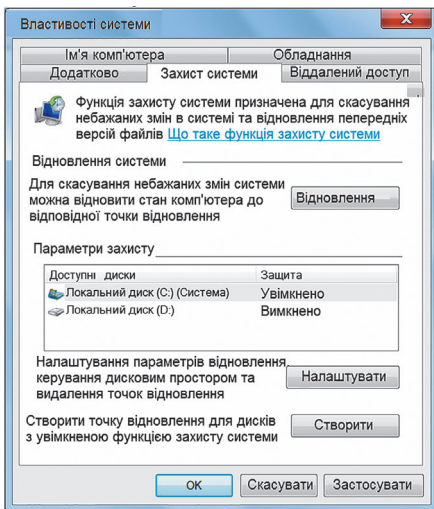


Рис. 7.3

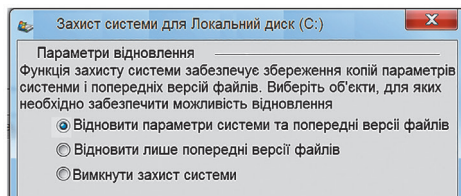


Рис. 7.4

Зазвичай стан комп'ютера доцільно повернути до певної точки відновлення у випадках погіршення роботи комп'ютера. Для цього треба виконати такі дії: Пуск → Всі програми → Стандартні → Службові → Відновлення системи → Далі → вибрати точку відновлення → Далі → Готово.

- 2 В ОС Linux Mint для збереження інформації про стан системи та подальшого відновлення стану призначена утиліта Timeshift. Порядок її використання подібний до описаної роботи з точками відновлення ОС Windows.

### Питання для самоперевірки



1. Чим зумовлена надійність сучасних засобів зберігання даних?
2. Назвіть способи безпечного зберігання даних на персональному комп'ютері.
3. Як виконати резервне копіювання даних системного диска?
4. Як відновити дані з резервних копій?
5. Що таке точка відновлення Windows?
6. Як видалити дані без можливості їх відновлення?

### Вправа 7



- Здійснити архівне копіювання та відновлення даних.
- 1) Запустіть текстовий процесор і створіть новий документ.
  - 2) Засобами ОС Windows зробіть резервну копію папки, запропонованої вчителем, і збережіть її на диску D.
  - 3) Відновіть зі створеної резервної копії файл, запропонований учителем, збережіть його у вашій папці Документи.
  - 4) Запишіть у документ назви логічних дисків, на які розбито жорсткий диск вашого комп'ютера.
  - 5) Визначте, чи увімкнено захист системи для дисків. Зробіть скриншот з інформацією про це та збережіть зображення у документі.
  - 6) Збережіть файл документа з назвою Вправа 7. Заверште роботу за комп'ютером.

### Комп'ютерне тестування



Виконайте тестове завдання 7 з автоматичною перевіркою результату.





## Практична робота 2

### Налаштування параметрів безпеки в середовищі браузера

**Завдання:** навчитись налаштовувати параметри браузера та поштової скриньки, що стосуються безпеки листування.

**Обладнання:** комп'ютер, з'єднаний з інтернетом, браузер Google Chrome.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

1. Запустіть браузер Google Chrome.
2. Налаштуйте конфіденційність користування браузером:
  - клацніть на значку у верхньому правому кутку браузера;
  - оберіть команду Налаштування;
  - внизу сторінки оберіть команду Показати розширені налаштування;
  - у розділі Конфіденційність зробіть потрібні налаштування.
3. Перевірте налаштування заборони показу спливних вікон: Меню → Налаштування → Показати розширені налаштування → Конфіденційність → Налаштування вмісту → Спливні вікна (Заборонити).
4. Налаштуйте (якщо не налаштовано) активацію захисту від фішингу та шкідливих програм (див. алгоритм п. 2).
5. Увімкніть, якщо не увімкнено, у браузері українську мову інтерфейсу (див. попередні алгоритми налаштувань).
6. Самостійно дослідіть налаштування вмісту сторінки браузера.
7. Зайдіть на свою поштову скриньку.
8. Додайте нову папку Шкільне для зберігання листів шкільного призначення.
9. Налаштуйте автоматичний підпис ваших листів, наприклад: З повагою, Ім'я Прізвище.
10. Позначте листи, отримані з невідомих джерел, як спам.
11. З'ясуйте в інтернеті, що таке файли cookies і навіщо їх видаляють. Створіть документ, уведіть відповідь і надішліть учителеві.
12. Заверште роботу за комп'ютером.

**Зробіть висновок:** які налаштування безпеки браузеру Google Chrome установлені за замовчуванням, а які потрібно налаштувати додатково.



# РОЗДІЛ 2

## ТРИВИМІРНА ГРАФІКА



§ 8. Основні поняття тривимірної графіки

§ 9. Тривимірний редактор Blender

§ 10. Робота з об'єктами

Практична робота 3. Робота з тривимірними примітивами

§ 11. Редагування форми об'єкта. Модифікатори

Практична робота 4. Моделювання витисканням

§ 12. Матеріали та освітлення. Рендеринг

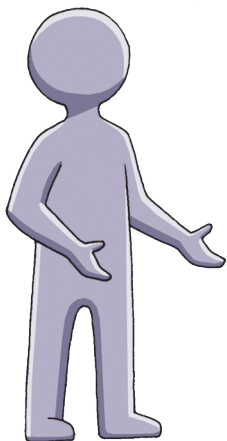
§ 13. Анімація ключовими кадрами

§ 14. Робота з текстовими об'єктами

Практична робота 5. Тривимірна анімація



## ПОВТОРЮЄМО



Раніше ви ознайомилися з основами *растрової* і *векторної* графіки. Ви знаєте, що в растровій графіці *числами кодуються* кольори *пікселів* — найменших крапок, з яких складається зображення. А векторне зображення складається з *ліній*, і саме їхні математичні властивості слід закодувати числами, щоб його створити і зберегти.

Ви створювали за допомогою *растрового* і *векторного графічних редакторів* різноманітні малюнки і знаєте, що зображення обох видів бувають *багатошаровими*. Ви знаєте, що файли формату GIF можуть містити рухоме зображення — *анімацію*.

Як ви могли помітити, в інтернеті, мультфільмах, рекламі трапляються *об'ємні зображення*, які можна оглянути *з різних боків*, а їхня поверхня нагадує справжні матеріали (деревину, пластик, скло тощо). Такі зображення дуже складно створити за допомогою згаданих графічних редакторів.

1. Які види комп'ютерної графіки ви вже вивчали?
2. Які особливості має растрова графіка?
3. Чим відрізняється векторна графіка від растрової?
4. Які програми призначені для опрацювання зображень? Які з них ви вивчали?
5. Які зображення складно створити за допомогою растрового та векторного графічних редакторів?



У цьому розділі ви ознайомитеся ще з одним видом комп'ютерної графіки — *тривимірною графікою*. Навчитися за допомогою *тривимірного редактора* моделювати об'єкти, створювати *анімацію* тощо.

## § 8. Основні поняття тривимірної графіки

Сучасні комп'ютери дозволяють моделювати просторові об'єкти. Пригадаємо рекламні відеоматеріали, спецефекти, комп'ютерні ігри.

### Тривимірна графіка

З такими видами комп'ютерної графіки, як растрова та векторна, ви вже ознайомилися. Векторні та растрові зображення є двовимірними, тобто пласкими. Ще одним видом є тривимірна графіка.



**Тривимірна графіка (3D-графіка)** — розділ комп'ютерної графіки, який присвячено методам створення зображень або відео шляхом моделювання об'ємних об'єктів у тривимірному просторі.

Існує багато програм для роботи з тривимірною графікою, що відрізняються одна від одної призначенням та можливостями. Умовно їх можна поділити на три групи. Розглянемо таблицю:

Призначення	Приклад	Опис
Художнє моделювання	 Blender  Autodesk Maya  Autodesk 3ds Max	Розробка реалістичних та фантастичних зображень і відеоматеріалів; дозволяють довільно змінювати форму об'єктів, добирати матеріали та змінювати їхні властивості (колір, текстуру, прозорість тощо), керувати рухами об'єктів та їх частин при розробці анімації
Технічне моделювання	 AutoCAD  FreeCAD	Розробка деталей з точним дотриманням розмірів, що є принциповим при підготовці конструкторської документації
Спеціалізовані програми	 Sweet Home 3D  RenderGold	Sweet Home 3D — моделювання розміщення меблів, дверей, вікон у кімнати та будинку; RenderGold — підготовка декоративних елементів: малюнки для кнопок на сайті, рамки для фотографій, візерунки для тла вітальної листівки

Щоб закодувати інформацію про форму просторового об'єкта (рис. 8.1), його поверхню потрібно уявно розбити на плоскі грані-многокутники (1) та зберегти дані про ці многокутники: координати ребер (2) і вершин (3), колір тощо. Це дає можливість обертати зображення на екрані, розглядати його з різних боків, моделювати освітлення тощо.

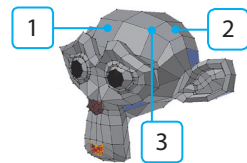


Рис. 8.1

Процес створення тривимірного зображення включає такі етапи:

- етап 1 — моделювання, під час якого створюються всі елементи, уточнюється їхня форма та взаємне розміщення;
- етап 2 — надання об'єктам кольору та текстур, налаштування освітлення;
- етап 3 — налаштування анімації у випадку розробки відеоматеріалу;
- етап 4 — вимальовування (рендеринг) зображення чи відеоматеріалу — так на основі моделі з урахуванням властивостей матеріалів і джерел світла формується кінцеве зображення (рис. 8.2).



Рис. 8.2

## Застосування тривимірних моделей



**Тривимірне моделювання** — процес створення 3D-моделі об'єкта, що є його візуальним об'ємним образом. **Тривимірна модель** — візуальний об'ємний прообраз певного об'єкта.

Змоделювати та змусити рухатися на екрані можна різні об'єкти, особливо ця технологія сподобалась художникам, дизайнерам і розробникам мультфільмів та ін. Давайте поміркуємо і наведемо приклади застосування тривимірних моделей.

- 1 Перший український 3D-анімаційний фільм «Микита Кожум'яка» вийшов у 2016 році. Фільм режисера Манука Депоєна було знято за мотивами однойменної казки письменника Антона Сіяніки. Над його виробництвом протягом двох років працювали декілька сотень людей. Пізніше з'явилися фільми «Викрадена принцеса: Руслан і Людмила» (2018), «Клара та чарівний дракон» (2019) та ін.

Технологія *доповнена реальність* додає штучні об'єкти до зображення на екрані, яке дає камера. Завдяки їй можна, навівши камеру на сторінку сучасного підручника біології, дослідити на екрані інтерактивну тривимірну модель, наприклад, кровоносної системи людини.

Тривимірне моделювання та анімація стали потужним помічником творців сучасного кіно. Відеоспецефекти застосовують не тільки для створення вражаючих або неможливих епізодів, а й для моделювання цілком буденних речей, наприклад, умеблювання кімнати.

2 Над спецефектами для фільму канадського режисера Джеймса Кемерона «Титанік» (1997 рік) працювало 17 студій. У побудованій моделі корабля не було носової частини, тому в епізодах використано її тривимірну модель. У багатьох кадрах ми бачимо змодельовану воду і меблі; під час зіткнення у воду «падають» тривимірні моделі шматків льоду. У 2012 році вийшла у світовий прокат 3D-версія фільму.

» Технологію, що називається *хромакей* (від англ. *chroma key* — колірний ключ), часто застосовують під час фото чи відеозйомки, на телебаченні тощо. При цьому акторів знімають на однотонному тлі, здебільшого зеленому чи синьому (рис. 8.3), а потім замінюють тло на відео змодельованою сценою.

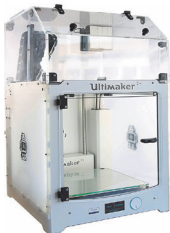


Рис. 8.3

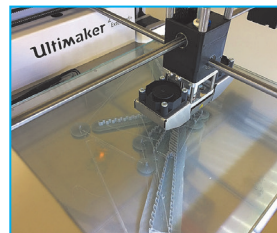
## Тривимірний друк

Застосування тривимірних моделей не обмежується «віртуальним світом». Протягом останніх десятиліть розвиваються **адитивні технології** (від англ. *add* — додавати), які полягають у створенні об'єктів поступовим додаванням матеріалу під керуванням комп'ютера.

Пристрій, який за тривимірною комп'ютерною моделлю виготовляє з певного матеріалу реальний об'єкт, називають **тривимірним принтером** (рис. 8.4, а), а сам процес виготовлення — **тривимірним друком** (рис. 8.4, б).



а



б

Рис. 8.4

З поширенням комп'ютерів адитивні технології надзвичайно розвинулись. Нині можливий тривимірний друк із найрізноманітніших матеріалів: різних сортів пластику, металів (зокрема й дорогоцінних), шоколаду тощо. Будівельні принтери дозволяють зі спеціальної суміші «друкувати» цілі будинки.

- 1 Влітку 2021 року в Амстердамі встановлено перший у світі сталевий міст, виготовлений методом тривимірного друку (рис. 8.5; світлина з вікісховища — [w.wiki/4Lwr](https://commons.wikimedia.org/wiki/File:Stoofbrug)). Мосту дали назву Стофбруг (нідерл. *Stoofbrug* — тушкований міст).



Рис. 8.5

Попри спільну назву, для тривимірного друку з різних матеріалів застосовуються різні фізичні та хімічні процеси.

- 2 Набули поширення тривимірні принтери, в яких поступово розплавляється тонкий пластиковий шнур і шар за шаром формується предмет. В інших випадках застосовується порошкоподібний або рідкий матеріал, який швидко полімеризується внаслідок опромінення лазером тощо.
  - 3 Від 2005 року триває розробка тривимірного принтера з відкритим кодом RepRap (англ. *Replicating Rapid Prototyper*). Однією з цілей проекту є можливість «самокопіювання» пристрою, тобто придатність для виготовлення всіх деталей, необхідних для виготовлення такого ж пристрою.
- » Технологію тривимірного друку одним із перших описав у своєму оповіданні 1945 року «Речі минають» (англ. *Things Pass By*) американський письменник-фантаст Маррі Лайнстер.

## Програма Sweet Home 3D

Програма Sweet Home 3D (рис. 8.3), поширювана за ліцензією GNU GPL, призначена для моделювання інтер'єру. Її, а також файл українського перекладу, можна завантажити з офіційного сайту [www.sweethome3d.com](http://www.sweethome3d.com). Тут описано версію 6.6. Після запуску програми на екрані з'являється головне вікно, робоче поле якого поділено на чотири ділянки (рис. 8.6).

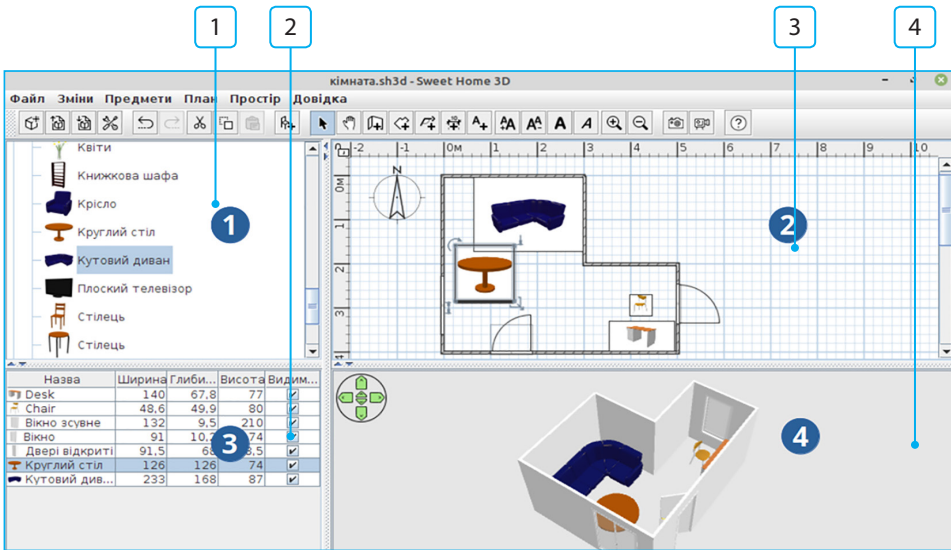


Рис. 8.6



1 — каталог предметів

3 — план

2 — перелік вибраних предметів

4 — просторовий вигляд

Щоб змінити розміри ділянок вікна, слід перетягти їхні внутрішні межі. У каталозі предметів (1) зібрано елементи, які можна використати для побудови плану. Потрібний елемент треба перетягти на план (3). Одночасно з цим у переліку вибраних предметів (2) з'явиться опис доданого елемента, а на просторовому вигляді (4) — зображення.

Для навігації в межах плану застосовують смуги прокручування, режим перетягування плану, який вмикається кнопкою  на панелі інструментів. Щоб побачити частину плану, його треба перетягти мишею, а щоб активувати режим вибору об'єктів, клацнути кнопку . Збільшити та зменшити масштаб плану можна кнопками із зображеннями лінз.



Для здійснення **навігації у вікні тривимірного вигляду** потрібно:

- 1) обернути сцену відносно двох осей, пересуваючи мишу, утримуючи натиснутою ліву або праву кнопку;
  - 2) змінити масштаб зображення, обертаючи коліщатко миш
- Точніше керувати обертанням відносно осей дозволяють навігаційні стрілки, розташовані в лівому верхньому куті просторового вигляду (рис. 8.7).



Рис. 8.7

Властивості елементів, доданих до плану, можна змінити шляхом перетягування кутів прямокутника, що зображає елемент (рис. 8.8).



Рис. 8.8

Щоб **зберегти розроблений план**, вибирають команду меню Файл → Зберегти як... Вікно збереження файлу — стандартне. Результатом буде файл із розширенням .sh3d.

4 Створимо модель житлової кімнати розмірами  $3 \times 4,5$  м.

- 1) Вибираємо інструмент Створення стін і будуємо за його допомогою на плані прямокутник, контролюючи розміри за підказками, що з'являються біля вказівника миші. Побудову останньої сторони прямокутника завершуємо подвійним клацанням миші.
- 2) У каталозі предметів розгортаємо розділ Двері і вікна. Відшукавши у списку Двері вхідні, перетягуємо цей елемент списку на зображення на плані тієї стіни, де мають бути двері. Після відпускання кнопки миші на ділянках плану (рис. 8.9) і просторового вигляду з'являться зображення дверей.
- 3) У каталозі предметів розгортаємо розділ Житлова кімната. Послідовно додаємо, перетягуючи на план, стіл, диван, інші меблі. Після цього можна переглянути результат на ділянці просторового вигляду (рис. 8.10), обертаючи зображення.

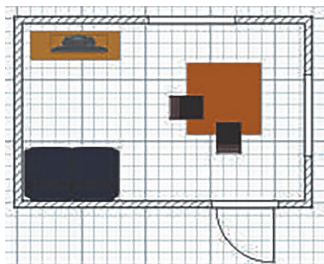


Рис. 8.9



Рис. 8.10

### Питання для самоперевірки



1. Порівняйте тривимірну, растрову та векторну графіку.
2. Як кодують інформацію про форму просторового об'єкта?
3. Які бувають програми для тривимірної графіки за призначенням?
4. Що таке «адитивні технології»?
5. Як працює тривимірний принтер?

### Вправа 8



1. Програма Sweet Home 3D дозволяє використовувати нові предмети, імпортувавши їх з файлів у форматах OBJ, DAE/Collada, KMZ або 3DS. У текстовому процесорі створіть новий документ і, скориставшись інтернетом, складіть і оформте таблицю з переліком програм, якими можна створювати такі файли. Збережіть документ у файлі Вправа 8\_1.

2. Користуючись довідковою системою програми Sweet Home 3D (клавіша F1), з'ясуйте, як змінити колір предмета; як додати на стіну малюнок; що відбудеться, якщо перетягнути новий предмет на місце, зайняте іншим предметом; як моделюють двоповерховий будинок.



3. Змоделюйте кабінет інформатики. Меблі та інше обладнання доберіть із доступних у програмі. За потреби змініть розміри та колір окремих предметів. Модель збережіть у файлі Вправа 8\_2.

4. У документі Вправа 8\_3 складіть словесний опис кімнати (наприклад, власної): зазначте її розміри, розташування та розміри дверей і вікон, перелік та розташування меблів.



5. Запропонуйте однокласнику чи однокласниці змоделювати кімнату за цим описом.



6. Перегляньте результат моделювання і з'ясуйте, чи відрізняється розташування елементів інтер'єру у змодельованій кімнаті від розташування елементів у кімнаті-прототипі. Якщо так, то як саме?

7. Додайте в документ скриншот моделі.

### Комп'ютерне тестування



Виконайте тестове завдання 8 з автоматичною перевіркою результату.





## § 9. Тривимірний редактор Blender

Для вивчення прийомів роботи з тривимірною графікою розглянемо Blender (рис. 9.1) — редактор художньої тривимірної графіки, який об'єднує засоби для побудови складних композицій, підготовки анімацій та відеоефектів і розробки комп'ютерних ігор.



Рис. 9.1

### Інтерфейс програми Blender

Blender розробляється спільнотою програмістів і поширюється за ліцензією GNU GPL. Тут описано версію 2.76b, звантажити яку можна з офіційного сайту програми на сторінці [download.blender.org/release/Blender2.76](http://download.blender.org/release/Blender2.76). (Починаючи з версії 2.77 зросли вимоги до комп'ютера, на який можна встановити Blender).

» Том Розендал (рис. 9.2) — нідерландський розробник програмного забезпечення, який у кінці 1990-х років розпочав розробку тривимірного редактора Blender. У 2002 році він створив і очолив некомерційну організацію Blender Foundation, яка оголосила в інтернеті збирання коштів для викупу Blender. За 7 тижнів було зібрано 100 000 євро. Це дозволило опублікувати програмний код Blender під вільною ліцензією і продовжити розробку спільнотою добровольців.



Рис. 9.2

Після запуску програми в центрі головного вікна виводиться вікно вітання, у якому, крім малюнка, є гіперпосилання на сайти з описом Blender та перелік останніх файлів, що редагувались. Якщо клацнути ім'я файлу, він відкриється для редагування. Кнопка Відновити попередній сеанс (Recover Last Session) корисна у випадку аварійного завершення попереднього сеансу роботи з програмою. Щоб вікно вітання зникло, слід клацнути за межами гіперпосилань.

Як бачимо з рис. 9.3, вікно складається з менших вікон (робочих просторів). Найбільшим є вікно 3D-вигляду, в якому видно сцену з кубом, лампою та камерою і панель інструментів. Угорі є вікно інформації (видно заголовки із головним меню), вікно дерева елементів сцени, вікно властивостей об'єктів і сцени, вікно керування ключами анімації (шкалу часу).

Особливість інтерфейсу полягає в тому, що тип, кількість та розміщення вікон користувач може легко змінити.

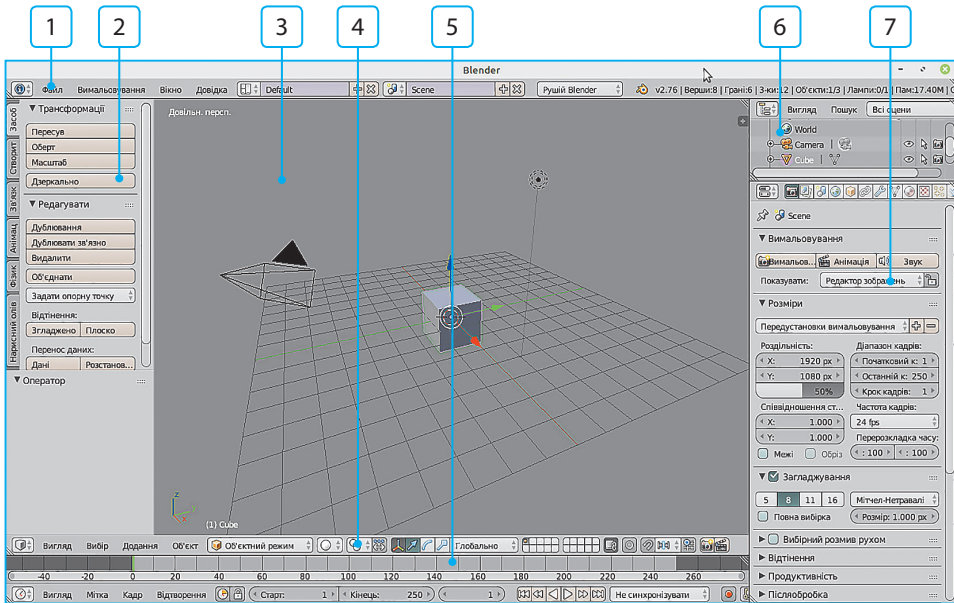


Рис. 9.3

- |                                |                            |
|--------------------------------|----------------------------|
| 1 — вікно інформації           | 5 — вікно шкали часу       |
| 2 — панель інструментів        | 6 — вікно дерева елементів |
| 3 — вікно 3D-вигляду           | 7 — вікно властивостей     |
| 4 — заголовок вікна 3D-вигляду |                            |

Розглянемо засоби керування вікнами (рис. 9.4).

У лівому нижньому та правому верхньому кутах вікна є позначки у вигляді заштрихованих кутів (1) (наведений на них вказівник набуває вигляду білого хрестика). Якщо одну з них перетягти вздовж межі вікна, то воно розділиться на два вікна. Якщо перетягти позначку в бік сусіднього вікна такого самого розміру, воно зникне, а розмір поточного вікна відповідно збільшиться.




Якщо клацнути маркер (2), то відкриється додаткова панель. Щоб сховати цю панель, слід перетягти її ліву межу до краю вікна.

Щоб змінити тип вікна, достатньо клацнути кнопку на початку його заголовка, а потім зі списку, що з'явиться, вибрати тип. Після цього вигляд вікна і піктограма на кнопці зміняться.



Рис. 9.4

Ознайомимося з деякими типами вікон та відповідними їм піктограмами. Розглянемо таблицю:

Піктограма	Тип вікна
	<b>3D-вигляд</b> (англ. <i>3D View</i> ) показує сцену, дозволяє розглянути сцену з різних боків, додавати, редагувати та вилучати об'єкти
	Вікно <b>Властивості</b> (англ. <i>Properties</i> ) призначене для переглядання та змінення властивостей як об'єктів, так і всієї сцени або анімації
	<b>Редактор UV/зображень</b> (англ. <i>UV/Image Editor</i> ) призначений для накладання на грані об'єктів малюнків, як завантажених з файлів, так і виконаних власноручно
	<b>Дерево елементів</b> (англ. <i>Outliner</i> ) дозволяє вибрати об'єкт, керувати видимістю об'єктів, блокувати від випадкових переміщень тощо
	Вікно <b>Користувацькі уподобання</b> (англ. <i>User Preferences</i> ) дозволяє налаштувати середовище програми. Наприклад, на вкладці <b>Теми</b> (англ. <i>Themes</i> ) можна обрати кольори для всіх елементів вікна програми, на вкладці <b>Система</b> (англ. <i>System</i> ) — змінити мову інтерфейсу тощо
	Вікно <b>Інформація</b> (англ. <i>Info</i> ) містить у заголовку головне меню та деякі інші засоби. У робоче поле виводяться службові повідомлення
	Вікно <b>Огляд файлів</b> ( <i>File Browser</i> ) надає доступ до дисків та папок для пошуку файлів
	<b>Шкала часу</b> ( <i>Timeline</i> ) містить засоби для налаштування та перегляду анімації

На різних етапах роботи зручно користуватися різними наборами (розкладками) вікон. Blender дозволяє зберігати такі набори, а потім — швидко переходити від одного до іншого. Для цього на інформаційній панелі є список з назвами готових розкладок (рис. 9.5): 1 — розкрити список; 2 — поточне значення; 3 — додати до списку; 4 — вилучити зі списку; 5 — поле пошуку — у ньому ліворуч від поточного значення є кнопка, що розкриває список. Достатньо клацнути назву певної розкладки, і вигляд вікна зміниться.

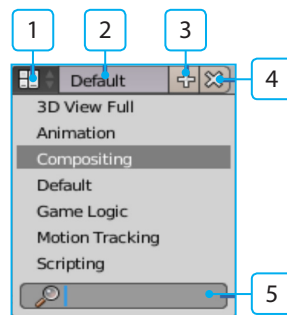


Рис. 9.5

Вважатимемо розкладку Default (англ. *типова*) (рис. 9.5) основною.

Щоб видалити поточну розкладку з проєкту або додати нову, потрібно натиснути відповідні кнопки. Щоб доданій розкладці надати зрозумілу назву, слід клацнути назву, що з'явилась автоматично, і ввести нову.

## Основні прийоми просторової навігації

Після запуску програми в ділянці 3D-вигляду (рис. 9.6) видно координатну сітку, камеру (1), лампу (3) і куб (2). Центр куба збігається з початком координат. Куб вибрано, тому від його опорної точки виходять три кольорові стрілки, напрями яких збігаються з напрямками координатних осей.

Масштаб зображення у вікні можна змінити, обертаючи коліщатко миші.

Щоб **вибрати (виділити) об'єкт**, достатньо клацнути його ПКМ, а щоб вибрати декілька — клацнути, утримуючи клавішу Shift.

Щоб **показати в центрі вікна вибрані об'єкти**, потрібно натиснути клавішу з крапкою на додатковому полі клавіш.

Щоб **змінити точку зору** на сцену, потрібно натиснути коліщатко або середню кнопку миші та пересунути її. У лівому нижньому куті вікна 3D-навігації є схематичне зображення координатних осей. При зміні точки зору воно обертається, вказуючи напрями глобальних осей координат. Якщо ж перед цим натиснути й утримувати клавішу Shift, то замість обертання сцени вона пересунеться без зміни напрямку осей.

**Клавіші додаткового поля** (надалі позначатимемо їх префіксом NUM) дозволяють швидко змінювати напрям погляду на сцену:

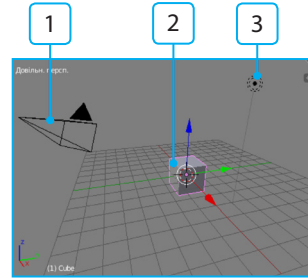


Рис. 9.6

Клавіша	Дія
NUM1, NUM3, NUM7	Встановлює вигляд спереду, зліва та зверху, відповідно
NUM2, NUM4, NUM6, NUM8	Обертають сцену на 15° відносно осей вікна
NUM0	Встановлює вигляд з камери, що дозволяє оцінити кінцевий вигляд композиції. При цьому елементи сцени, що не потраплять у кадр камери, зображуються затіненими
NUM5	Перемикає перспективну та паралельну проєкції

Розглянемо рис. 9.7. У *перспективній проєкції* (а) віддаленіші об'єкти виглядають меншими, порівняно з розміщеними ближче. Така проєкція сприймається природніше. У *паралельній проєкції* (б) спотворення розмірів немає, що дозволяє, зокрема, точніше розміщувати та масштабувати об'єкти (див. далі).

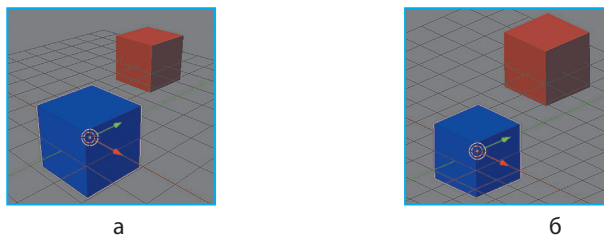


Рис. 9.7

Важливим елементом вікна 3D-вигляду є **просторовий курсор**, або **3D-курсор** (рис. 9.8). Саме там, де розташований 3D-курсор, з'являються об'єкти, що додаються до сцени.

Щоб **встановити 3D-курсор у певну точку**, достатньо її клацнути. Точніше положення 3D-курсора задається шляхом викликання меню клавішами Shift+S. У ньому, зокрема, є команди (рис. 9.9), що встановлюють курсор: у центр вибраного об'єкта чи групи об'єктів (Курсор до вибраного); у початок координат (Курсор до центра); у найближчий вузол тривимірної сітки (Курсор до сітки); у центр останнього з вибраних об'єктів (Курсор до активного).



Рис. 9.8

Прив'язка
Вибране до сітки
Вибране до курсора
<b>Курсор до вибраного</b>
Курсор до центра
Курсор до сітки
Курсор до активного

Рис. 9.9

» Грейс Мюррей Гоппер (рис. 9.10) — американська вчена в галузі комп'ютерних наук та контрадмірал військово-морських сил США, одна з перших програмістів комп'ютера Марк I. Популяризувала ідею машинезалежних мов програмування, що привело до розробки мови програмування високого рівня COBOL. На вшанування Грейс Гоппер Асоціацією обчислювальної техніки у 1971 році була заснована нагорода її імені.



Рис. 9.10

## Додавання об'єктів до сцени

Найпростіші композиції будують із тривимірних примітивів, що є об'єктами типу **сіть** (іноді кажуть *каркас* або *меш*, від англ. *mesh* — сітчастий) — ребра таких об'єктів справді утворюють сітчастий «каркас».

Щоб додати до сцени новий об'єкт, на панелі інструментів слід вибрати вкладку Створити, де в списку Каркас натиснути кнопку з назвою примітива (рис. 9.11). Після клацання на сцені в місці розташування просторового курсору з'явиться новий об'єкт. Його контур буде обведено яскравою лінією; тобто об'єкт вибрано для подальших операцій.

Так, якщо натиснути клавішу G, колір контуру зміниться і активується **режим переміщення**: досить пересунути мишу і клацнути, щоб змінити розташування об'єкта. Аналогічно, клавіша S вмикає **режим змінення розміру** (масштабування) об'єкта, а клавіша R — **режим обертання**. Детальніше про це далі.

Якщо утримувати натиснутою клавішу Ctrl, то операції виконуватимуться не плавно, а з прив'язкою до сітки, що дає змогу точніше розташувати об'єкти один відносно одного.

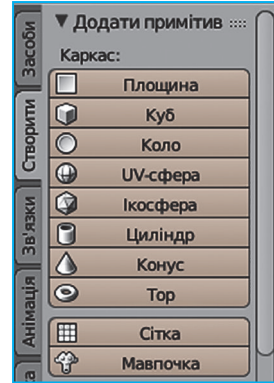


Рис. 9.11

## Робота з файлами

Щоб записати результат роботи на диск, у Blender слід вибрати команду меню Файл → Зберегти. У робочому полі вікна відкриється Оглядач файлів (рис. 9.12). Зліва у вікні можна вибрати диск, справа — папку, ввести назву файлу і клацнути кнопку Зберегти файл Blender — створено файл із розширенням .blend.

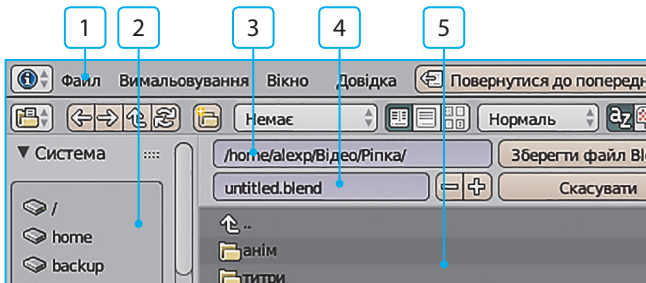


Рис. 9.12

- 1 — меню Файл
- 2 — доступні розташування
- 3 — активна папка
- 4 — ім'я файлу
- 5 — вміст активної папки

## Питання для самоперевірки



1. Чим відрізняється тривимірна графіка від растрової і векторної?
2. Як кодують інформацію про форму просторового об'єкта?
3. Які бувають програми для тривимірної графіки за призначенням?
4. Яка різниця між перспективною та паралельною проєкціями?
5. Що таке тривимірні примітиви?
6. Як можна перемістити, змінити розмір або повернути об'єкт?

## Вправа 9



► Ознайомитися із середовищем тривимірного редактора Blender.

- 1) Запустіть Blender і клацніть на малюнку у вікні привітання (воно має зникнути).
- 2) Увімкніть українську мову інтерфейсу і підказок (меню File → User Preferences → System → прапорець International Fonts → Language → вибрати мову: Українська → Інтерфейс → Підказки → Зберегти за замовчуванням).
- 3) Ознайомтесь і випробуйте основні засоби перегляду і керування об'єктами: зміну точки зору, масштабування, переміщення та повороти куба тощо. Видаліть куб (виділити куб правою кнопкою миші (ПКМ) → Delete → підтвердити видалення).



- 4) Додайте новий куб (Створити → Каркас → Куб). Увімкніть вигляд спереду (NUM1). Поверніть куб навколо осі Y (зелена вісь) ребром донизу (виділити куб ПКМ → клавіша R → клавіша Y → повернути куб (для точного повороту утримуйте клавішу Ctrl) → клацнути ліву кнопку миші) (рис. 9.12).

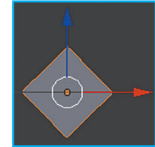


Рис. 9.12

- 5) Перемістіть куб вгору (синя стрілка), щоб він ребром лежав на координатній сітці.

Для точного переміщення увімкніть паралельну проєкцію (NUM5), збільште масштаб і утримуйте Ctrl.

- 6) Збережіть файл з назвою Вправа 9 у вказаному вчителем місці (папці). Заверште роботу за комп'ютером.

## Комп'ютерне тестування



Виконайте тестове завдання 9 з автоматичною перевіркою результату.





## § 10. Робота з об'єктами

Ви вже вмiєте швидко змiнювати розмiри та положення вибраних об'єктiв за допомогою клавіш G, S і R та миші. Розглянемо, як можна виконувати ці операції точніше, з'ясуємо, як стиснути або розтягти об'єкт уздовж певної осі. А також дізнаємось, як швидко змінити колір об'єкта.

### Координатні осі

З уроків математики вам відомо, що положення точки на числовій прямій можна задати одним числом — координатою точки (рис. 10.1). На площині для цього ж знадобиться 2 осі, тобто прямокутна система координат і, відповідно, 2 числа — абсциса й ордината точки (рис. 10.2).

Об'єкти тривимірної сцени розташовані в просторі, тому координати кожної точки задають трьома числами (рис. 10.3).

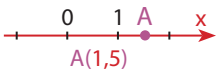


Рис. 10.1

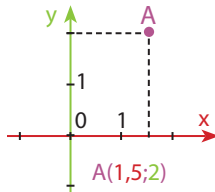


Рис. 10.2

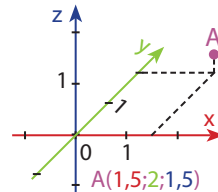


Рис. 10.3



Простір, у якому положення точки задається трьома координатами, називають **тривимірним**, а моделювання на комп'ютері об'єктів, розташованих у такому просторі, — **тривимірним моделюванням**.

Щоб закодувати числами форму тривимірного об'єкта, можна:






- визначити координати всіх вершин об'єкта;
- зазначити, які вершини сполучені між собою ребрами;
- зазначити, які ребра утворюють грані.

### Маніпулятори трансформацій

Для переміщення об'єктів ви вже користувалися стрілками, які виходять з опорної точки об'єкта. Це один із маніпуляторів трансформацій — маніпулятор переміщення.



У заголовку вікна 3D-вигляду є віджети для керування маніпуляторами трансформацій для вибраних об'єктів. Розглянемо таблицю.

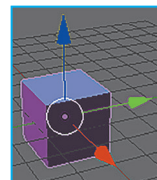
Віджет	Дія
	Вмикає або вимикає показ маніпуляторів на вибраних об'єктах (і решти віджетів для їх налаштування)
	Вмикає маніпулятор переміщення
	Вмикає маніпулятор обертання
	Вмикає маніпулятор масштабування
	Список для вибору напрямків осей для маніпуляторів

Розглянемо, які дії можна виконувати.


▶ **Переміщення об'єктів**

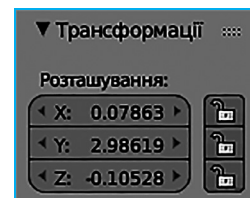
Ви вже випробували два способи переміщення об'єктів:

- Якщо вибрати об'єкти і натиснути клавішу G, колір підсвічення об'єктів зміниться і рухи миші зумовлюватимуть відповідні рухи об'єктів у площині, перпендикулярній до променя зору. Перемістивши об'єкти, слід завершити дію клацанням ЛКМ.
- Перетягування стрілок маніпулятора переміщення (рис. 10.4, а) зумовлює переміщення об'єктів у напрямку стрілки.



а

Ще один спосіб полягає в тому, щоб увести в поля групи Розташування на панелі Трансформації, яку відкривають маркером  (рис. 10.4, б) числові значення його нових координат. Об'єкт переміститься у відповідне місце простору.




б

Рис. 10.4

▶ **Масштабування об'єктів**

Змінити розміри об'єктів також можна різними способами.

Ви вже знаєте, що після натискання клавіші S можна, пересуваючи мишу, змінювати розміри об'єкта в цілому. Якщо увімкнути маніпулятор масштабування , то замість стрілок на координатних осях, що виходять з опорної точки об'єкта, з'являться

кубики (рис. 10.5). Унаслідок їх перетягування відбувається стиснення або розтягнення об'єкта вздовж відповідної осі.

Точно задати розміри об'єкта вздовж кожної з осей можна на панелі Трансформації, ввівши числові значення в поля групи Масштаб (рис. 10.6).

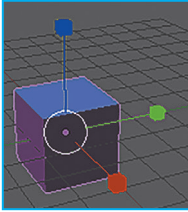


Рис. 10.5

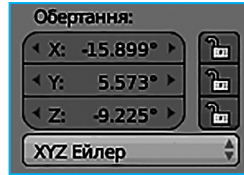



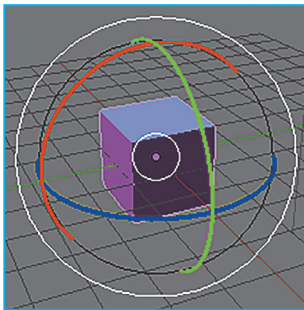
Рис. 10.6

### ► Маніпулятор обертання

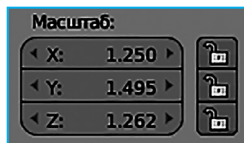
Подібно до двох попередніх діє й маніпулятор обертання, який вмикається кнопкою . Замість стрілок навколо об'єкта з'являються чотири кола різних кольорів (рис. 10.7, а). Перетягування будь-якої точки білого кола дозволяє повернути об'єкт навколо осі, перпендикулярної до променя зору (так само, як натискання відомої вам клавіші R), а червоного, зеленого і синього — навколо осей x, y і z відповідно.

**Задати кути повороту об'єкта** навколо кожної з осей можна на панелі Трансформації, ввівши числові значення в поля групи Обертання (рис. 10.7, б).

**!** Щоб увімкнути одночасно декілька маніпуляторів трансформації (рис. 10.8), перш ніж натиснути їхні кнопки, потрібно натиснути та утримувати клавішу Shift.



а



б

Рис. 10.7

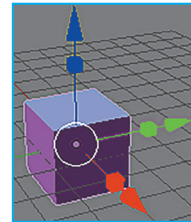





Рис. 10.8

Побудуємо з тривимірних примітивів модель сніговика.

- 1) Запустимо Blender і видалимо куб. Якщо тривимірний курсор зміщено, повернемо його в центр сцени (Shift + S → Курсор до центра).
- 2) Додамо UV-сферу (на панелі інструментів Створити → UV-сфера).
- 3) На вигляді спереду (NUM1) увімкнемо паралельну проєкцію (NUM5). У лівому верхньому куті вікна 3D-вигляду має з'явитися напис Спереду орто.
- 4) Увімкнемо маніпулятор масштабування  і трохи стиснемо об'єкт уздовж осі Z (синій кубик).
- 5) Увімкнемо маніпулятор переміщення  і перетягнемо UV-сферу вгору так, щоб вона «стояла» на осі X (червона лінія).
- 6) Для подальших дій увімкнемо два маніпулятори (див. рис. 9.8): утримуючи Shift, натиснемо .
- 7) Додамо до сцени ще дві UV-сфери, кожен спочатку трохи зменшимо (клавіша S), трохи стиснемо і перетягнемо вгору, формуючи «тіло» сніговика.
- 8) Додамо конус (майбутню морквину). Зменшимо його розміри (клавіша S), потім розтягнемо вздовж вертикальної осі. Для подальших операцій зручно увімкнути прозорість (клавіша Z; рис. 10.9, а).
- 9) Вимкнемо прозорість (клавіша Z) і розглянемо сніговика з різних боків (СКМ + рухи мишею; рис. 10.9, б).

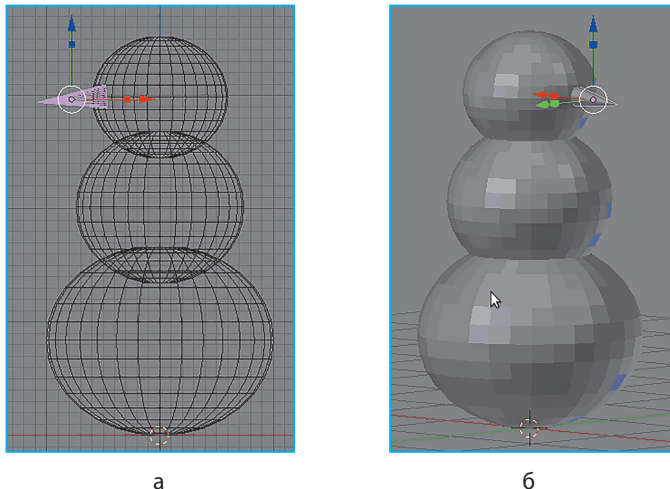




Рис. 10.9

## Змінення кольору об'єкта

Blender має розвинені засоби для налаштування матеріалів об'єктів, які дозволяють моделювати одноколірні матеріали, додавати до них процедурні текстури, накладати на грані растрові зображення з файлів, налаштовувати прозорість та блиск матеріалу тощо. Ці та інші властивості матеріалу кодуються у блоці даних матеріалу.

Щоб доданий до сцени **тривимірний примітив отримав блок даних матеріалу**, потрібно:

- 1) його виділити;
- 2) у вікні властивостей перейти на вкладку Матеріал ;
- 3) клацнути кнопку Новий або вибрати один із матеріалів зі списку  — у вікні властивостей з'являться засоби для налаштування матеріалу.

Тепер, якщо на панелі Розсіювання клацнути прямокутник зі зразком кольору, з'явиться панель, в якій можна вибрати колір матеріалу (рис. 10.10). Вибір зразу ж відбивається у вікні 3D-вигляду.

Докладніше про використання матеріалів ви дізнаєтеся з наступних параграфів.

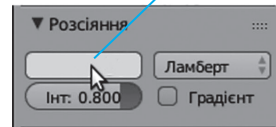
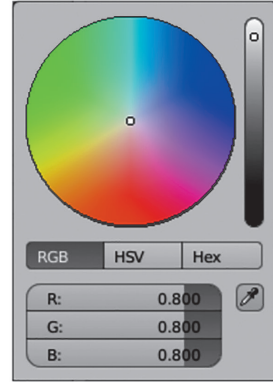


Рис. 10.10

Blender має вбудовану підтримку мови програмування Python. Це дозволяє скласти програму, яка після запуску побудує у вікні 3D-вигляду об'єкт, який складно або неможливо створити вручну. На рис. 10.11 показано куби різних розмірів, які утворюють просторову фрактальну структуру, побудовану за таким алгоритмом:

- 1) побудувати куб (на малюнку — синій);
- 2) навколо куба побудувати 26 кубів утричі менші (на рис. 10.11 червоні);
- 3) для кожного з менших кубів виконати п. 2.

Після першого виконання п. 3 маємо:  $26 \cdot 26 = 676$  (сірих кубів). Якщо побудову не зупинити, то на наступному кроці ми б дістали:  $26 \cdot 676 = 17\,576$  (ще менших кубів) і т. д.

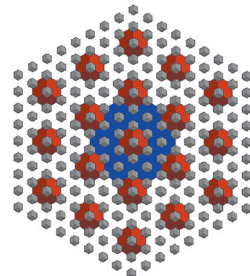


Рис. 10.11



Доктор фізико-математичних наук Катерина Ющенко (рис. 10.12) стояла біля витоків української школи програмування. Вона авторка однієї з перших у світі мов програмування високого рівня, в якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера. Ця мова використовувалась на комп'ютерах «Київ», «Дніпро», М20, БЕСМ, «Урал».



Рис. 10.12

### Питання для самоперевірки



1. Поясніть походження терміна «тривимірна графіка».
2. Як кодується положення точки в просторі?
3. Поясніть використання маніпуляторів трансформацій.
4. Як увімкнути одночасно два маніпулятори трансформацій?
5. Як додати до об'єкта новий матеріал?
6. Як змінити колір об'єкта?
7. Які можливості надає підтримка мови програмування Python?

### Вправа 10



► Побудувати за допомогою тривимірного редактора Blender модель сніговика (рис. 10.13).

- 1) Побудуйте модель сніговика за алгоритмом, наведеним у тексті параграфа.
- 2) Додайте ще декілька елементів, наприклад очі і рот (з тривимірних примітивів Ікосфера, Циліндр тощо).
- 3) Розфарбуйте сніговика, змінивши кольори окремих примітивів.
- 4) Додайте в сцену інші елементи: поверхню «снігу», декілька «снігових» куль різних розмірів тощо.
- 5) Збережіть файл з ім'ям snowman.blend.
- 6) Продемонструйте результат однокласникам і вчителю, обговоріть його. Заверште роботу на комп'ютері.

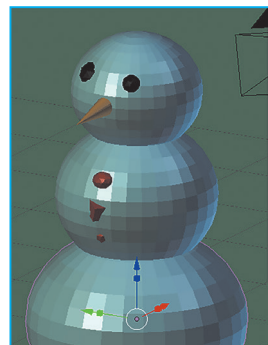


Рис. 10.13



### Комп'ютерне тестування



Виконайте тестове завдання 10 з автоматичною перевіркою результату.





## Практична робота 3

### Робота з тривимірними примітивами

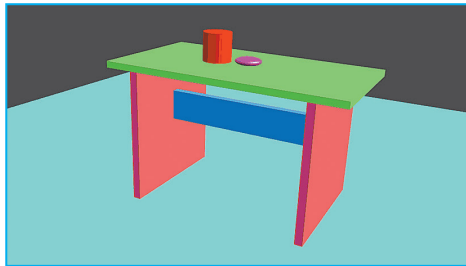
**Завдання:** змоделювати журнальний столик за допомогою тривимірних примітивів.

**Обладнання:** комп'ютер із тривимірним редактором Blender.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

1. Запустіть програму Blender.
2. Додайте на сцену куб (або скористайтесь наявним) і, змінюючи його розміри, створіть стільницю. Для перевірки її розташування в кадрі камери натисніть NUM0 і, за потреби, перемістіть стільницю в потрібний бік.
3. Додайте куб і створіть з нього першу опору. Щоб точно розташувати її відносно стільниці, користуйтеся виглядами спереду (NUM1), збоку (NUM3) і зверху (NUM7), масштабуванням сцени (коліщатко миші) та режимом прозорості (клавіша Z).
4. Створіть другу опору і перемичку між ними. Додайте на поверхню стола червоний циліндр і ще декілька предметів різної форми (тривимірні примітиви різних видів).
5. Надайте кольори частинам столика та предметам (див. рисунок).



6. Додайте площину (підлогу), доберіть її розташування, розміри та колір. Збережіть файл із назвою Практична робота 3. Заверште роботу за комп'ютером.

**Зробіть висновок:** як змоделювати предмет за допомогою тривимірних примітивів; наведіть приклади предметів, які ви тепер вмієте моделювати.

## § 11. Редагування форми об'єкта. Модифікатори

В основі тривимірних об'єктів лежить сіть із вершин, сполучених ребрами, яка й визначає їхню форму. Для побудови об'єктів складної форми потрібно навчитись редагувати сіть, тобто виконувати операції з вершинами, ребрами та утвореними з них гранями.

### Режим редагування

Якщо вказівник перебуває у вікні 3D-вигляду, то натискання клавіші Tab дозволяє перевести вибраний об'єкт у режим редагування (рис. 11.1). Для подальшої роботи кнопками в заголовку вікна слід вибрати, з чим саме будуть виконуватися операції: з вершинами (📍), ребрами (📏) чи гранями (📐). В останньому випадку кожна грань позначається квадратиком у центрі.

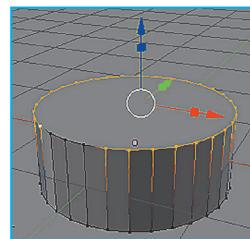


Рис. 11.1

Якщо утримувати натиснутою клавішу Shift, то можна увімкнути декілька режимів одночасно.

### Способи виділення елементів

Далі операції виконуються над уже виділеними елементами. З'ясуємо, якими способами можна виділяти вершини, ребра і грані.

- Окремі елементи виділяють так само, як і звичайні об'єкти — клацанням правою кнопкою (з клавішею Shift — декілька).
- Щоб виділити всі елементи, слід натиснути клавішу A (від англ. *all* — все). Повторне натискання знімає вибір зі всіх елементів.
- Якщо натиснути B (від англ. *box* — коробка), то біля вказівника з'являться вертикальна і горизонтальна штрихові лінії, які означають, що можна виділити елементи, охопивши їх у прямокутник.
- Натискання клавіші C вмикає режим виділення малюванням: біля вказівника з'являється коло, яке слід навести на потрібні елементи і клацнути або перетягти — елементи виділяться. Цю дію можна повторити декілька разів, змінюючи розмір кола коліщатком миші, то завершити слід натисканням клавіші Enter.



Виділені елементи можна переміщувати, обертати, масштабувати і видаляти так само, як ви це робили з об'єктами.

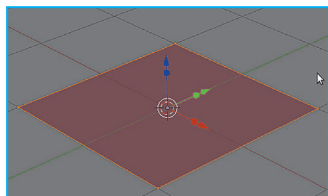
**!** Утримання натиснутою клавіші Shift дозволяє зняти виділення з раніше виділених елементів.

Якщо увімкнути **режим пропорційного редагування** (клавіша O), то деформація зачепить ще й елементи, сусідні з виділеними, зменшуючись із відстанню (див. приклад далі). Обертанням коліщатка при цьому регулюється відстань, на яку поширюється деформація.

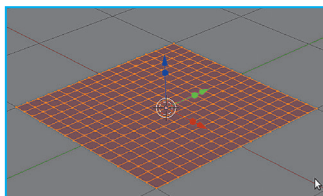
## Поділ ребер

Якщо сіті потрібно надати складної форми, її ребра і грані слід поділити на менші частини. Для цього вибирають потрібні елементи і на панелі інструментів на вкладці Засоби в групі Додавання вибирають команду Поділити — посередині виділених ребер з'являться нові вершини, а ребра, що їх сполучають, утворять нові грані.

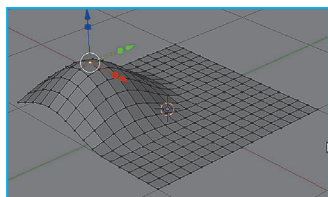
- 1** Виконаємо моделювання поверхні території (рис. 11.2). Для цього:
- 1) додамо площину і перейдемо в режим редагування (клавіша Tab). Всі вершини при цьому будуть виділеними (а);
  - 2) чотири рази поділимо вибране (б);
  - 3) виділимо одну з вершин (ПКМ), увімкнемо пропорційне редагування (клавіша O) і перемістимо її вгору (в);
  - 4) повторимо попередню дію декілька разів (г).



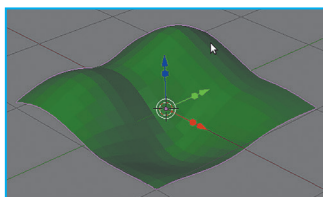
а



б



в



г

Рис. 11.2

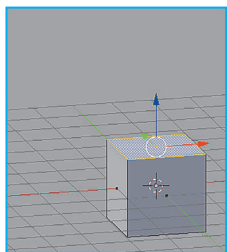


## Моделювання витисканням

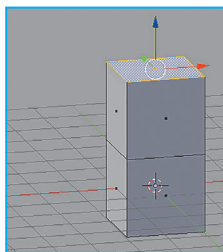
Існує ще один спосіб отримання об'єктів складної форми — **витискання** або **екструзія** (від англ. *extrusion* — витискання) виділених елементів.

2 Змоделюємо кактус, подібний до об'єктів в іграх, де ігровий світ будується з кубиків (рис. 11.3). Для цього виконаємо такі дії:

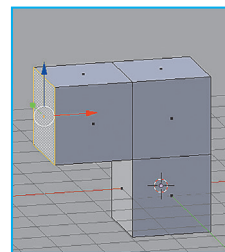
- 1) розпочнемо з куба: перейдемо в режим редагування (Tab) і увімкнемо режим роботи з гранями (☐);
- 2) виберемо верхню грань (а) і натиснемо клавішу E (від англ. *extrusion*) — з'явиться нова грань, «витиснута» з вибраної, яку можна переміщати рухом миші;
- 3) пересунемо нову грань вгору і клацнемо ЛКМ — створені грані додадуться до сіті (б);
- 4) повторимо подібні дії з різними гранями, формуючи «кактус» (в-г);



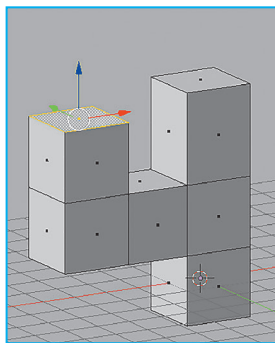
а



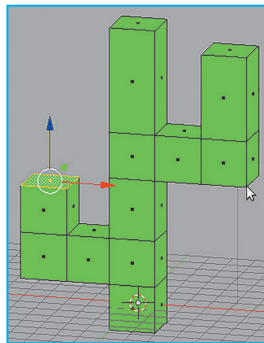
б



в



г



г

Рис. 11.3

## Модифікатори

Багато різноманітних операцій моделювання та редагування об'єктів доступні у вигляді **модифікаторів**.

Додавання та використання модифікаторів розглянемо на двох прикладах.

3 Додамо до моделі кактуса з прикладу 2 модифікатор Поділ поверхні. Для цього вийдемо з режиму редагування і виконаємо такі дії:

- 1) у вікні Властивості перейдемо в розділ Модифікатори (🔧);
- 2) натиснемо Додати модифікатор;
- 3) виберемо зі списку, що розкрився, модифікатор Поділ поверхні. У вікні властивостей з'явиться панель налаштування модифікатора (рис. 11.4), а вигляд об'єкта зміниться, відповідно до встановлених у ньому значень (рис. 11.5, а).

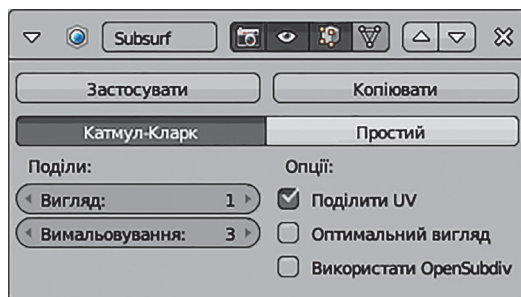
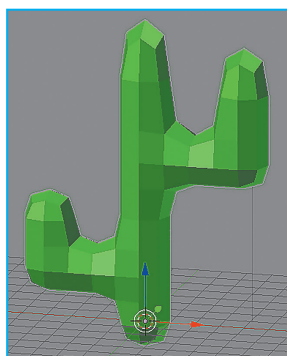
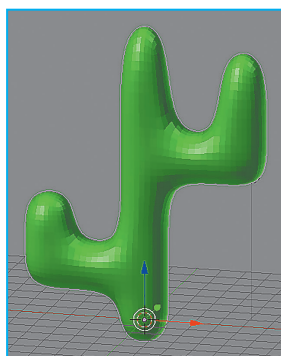


Рис. 11.4







а



б

Рис. 11.5

Розгляньмо панель налаштування модифікатора:

Віджет	Призначення	Віджет	Призначення
	Перемикач використання модифікатора під час рендерингу		Переміщення модифікатора вгору або вниз у списку доданих до об'єкта модифікаторів
	Перемикач показу дії модифікатора у вікні 3D-вигляду		Видалення модифікатора з об'єкта

Після видалення модифікатора з об'єкта його дія скасовується.

Якщо результат дії модифікатора повністю влаштовує, то можна натиснути кнопку Застосувати — форма об'єкта зміниться відповідно до параметрів модифікатора, а сам модифікатор зникне з об'єкта.

**!** Для зменшення обсягу обчислень результат дії модифікатора у вікні 3D-вигляду може показуватися спрощеним. Так, на рис. 11.4 (див. приклад 3) видно, що для вікна 3D-вигляду виконується один поділ поверхні (рис. 11.5, а). А для кінцевого зображення буде виконано три поділи (рис. 11.5, б).

Модифікатор Дзеркало дозволяє прискорити розробку, якщо об'єкт має містити симетричні частини. Після додання модифікатора з'являються дзеркальні копії об'єкта, кількість яких залежить від того, які осі позначено на панелі модифікатора. При цьому для редагування доступним є лише початковий об'єкт (рис. 11.6).

«Дзеркала» розташовуються перпендикулярно до відповідних осей і проходять через опорну точку початкового об'єкта або іншого, зазначеного в полі Об'єкт-дзеркало на панелі модифікатора (рис. 11.7).

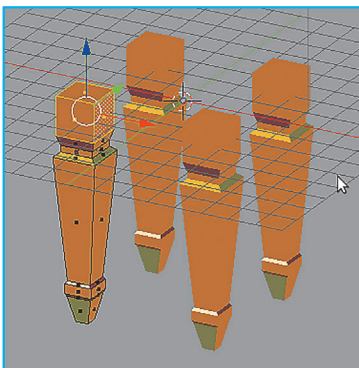


Рис 11.6

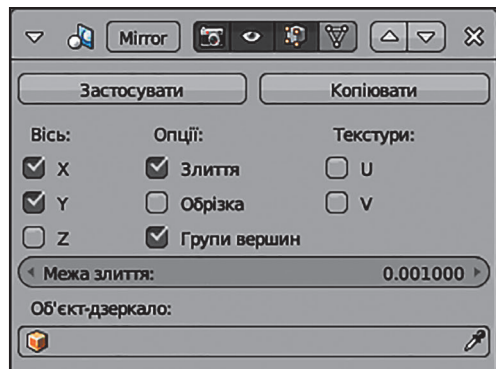


Рис 11.7

4

Щоб змоделювати 4 ніжки для табуретки, нам потрібно:

- 1) змоделювати одну ніжку;
- 2) додати модифікатор Дзеркало і перейти в режим редагування (Tab);
- 3) виділити всі вершини (A) і перемістити їх від опорної точки, яка залишиться на місці.

Після цього з'являться дзеркальні копії.

Після натискання кнопки Застосувати всі дзеркальні копії буде додано до початкового об'єкта, тобто утвориться єдиний об'єкт.

### Питання для самоперевірки



1. Які можливості доступні в режимі редагування сіті?
2. Якими способами можна виділити елементи сіті?
3. Як і для чого виконують поділ ребер?
4. Опишіть процес витискання.
5. Як додати до об'єкта модифікатор?
6. Для чого призначений модифікатор Поділ поверхні?
7. Як діє модифікатор Дзеркало?
8. Яка дія кнопки Застосувати на панелі модифікатора?

### Вправа 11



1. Побудуйте модель поверхні за алгоритмом із прикладу 1. Збережіть її у файлі terrain.blend.
2. На цій самій сцені побудуйте модель кактуса за алгоритмом із прикладу 2. Розмістіть її на створеній раніше поверхні. Збережіть файл.
3. Відкрийте файл snowman.blend і створіть для «сніговика» нерівну снігову поверхню.
4. Змоделюйте ніжки для табуретки за алгоритмом із прикладу 4. Додайте до табуретки кругле сидіння. Збережіть файл tabouret.blend.



### Комп'ютерне тестування



Виконайте тестове завдання 11 з автоматичною перевіркою результату.





## Практична робота 4

### Моделювання витисканням

**Завдання:** перетворити циліндр на модель горщика для кімнатних рослин (рис. 1) за допомогою витискання вершин.

**Обладнання:** комп'ютер зі встановленим тривимірним редактором Blender, файл Практична робота 3.

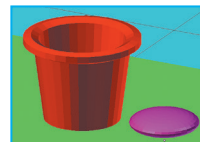


Рис. 1

### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

1. Запустіть програму Blender. Відкрийте файл Практична робота 3.
2. Виберіть циліндр і налаштуйте перегляд так, щоб він був у центрі екрана (клавіша з крапкою на додатковому полі) на вигляді спереду (NUM1). Для точнішого керування процесом вимкніть перспективу (NUM5).
3. Перейдіть у режим редагування (клавіша Tab). Виберіть вершини верхньої основи циліндра.
4. Збільште вибраний фрагмент (клавіша S, перетягування, клацнути ЛКМ), щоб отримати форму зрізаного конуса.
5. Витисніть вибрані вершини вгору (клавіша E, перетягування, клацнути ЛКМ) і збільште фрагмент (див. п. 4).
6. Повторіть операцію декілька разів, щоб отримати форму, подібну до наведеної на рисунку. Останнє витискання виконайте вниз, щоб отримати порожнисту посудину.

Для точнішого контролю товщини стінок перемикайте режим прозорості (клавіша Z; див. рис. 2).

Збережіть файл із назвою Практична робота 4. Заверште роботу за комп'ютером.

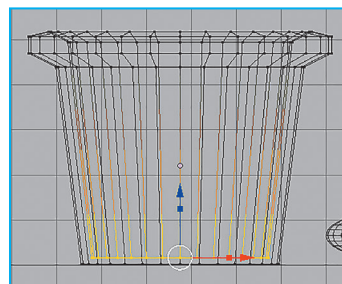


Рис. 2

**Зробіть висновок:** як виконується і які можливості дає операція витискання.

## § 12. Матеріали та освітлення. Рендеринг

Ви вже вмієте змінювати колір об'єкта. Тепер ознайомимося докладніше з налаштуванням матеріалів у Blender, а також із рендерингом.

### Рендеринг

Кінцевою метою процесу тривимірного моделювання у Blender є зображення або відеофрагмент, у якому враховано всі налаштування сцени, матеріалів тощо.



**Рендеринг (вимальовування)** — процес побудови зображення з урахуванням усіх налаштувань.

Для керування рендерингом призначений розділ Вимальовування у вікні властивостей.

Розглянемо панель Розміри в цьому розділі (рис. 12.1). Розміри зображення в пікселях залежать від значень у полях X і Y та розташованого під ними регулятора відсотків. Регулятор зменшує розміри, щоб швидше отримувати проміжні зображення, оскільки сцена з багатьма складними налаштуваннями може вимальовуватися дуже довго. Для кінцевого рендерингу слід встановити значення 100 %.

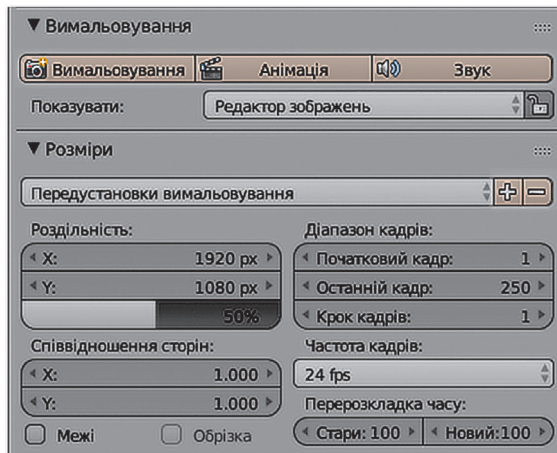



Рис. 12.1

Для того щоб **запустити рендеринг**, слід натиснути кнопку Вималювання або клавішу F12. Решта параметрів переважно стосуються відео і будуть розглянуті пізніше.

## Процедурні текстури

Імітувати різноманітні матеріали дозволяють так звані *процедурні текстури*. При цьому до матеріалу додається згенероване за певним алгоритмом зображення (у відтінках сірого), яке керує змішуванням двох кольорів: власного кольору матеріалу та кольору текстури.

- 1 Додамо до матеріалу куба текстуру деревини. Для цього:
  - 1) встановимо основний колір матеріалу світло-коричневий;
  - 2) у вікні властивостей відкриємо розділ Текстури  і натиснемо кнопку Новий, щоб створити блок текстурних даних;
  - 3) клацнемо список Тип і виберемо Деревина (рис. 12.2);
  - 4) на панелі Деревина налаштуємо параметри текстури (рис. 12.3);
  - 5) на панелі Розкладка задамо параметри накладання та розмірів текстури (рис. 12.4);

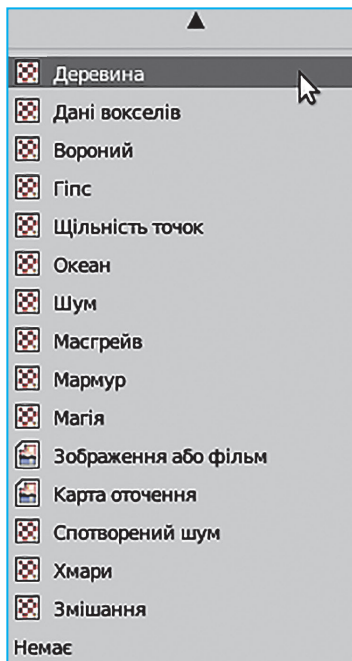


Рис. 12.2

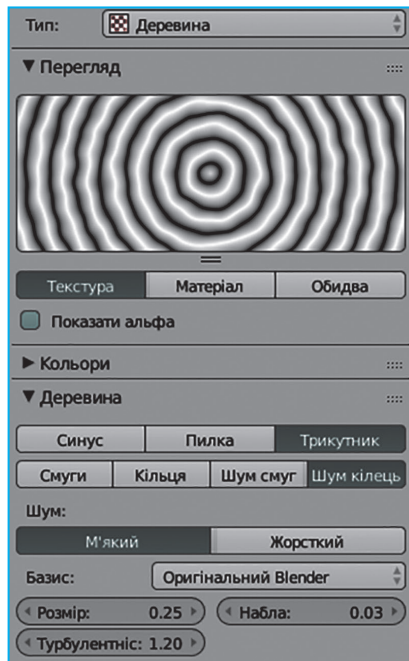


Рис. 12.3



- 6) на панелі Вплив виберемо для текстури колір, світліший від кольору матеріалу;
- 7) для перегляду кінцевого результату рендерингу натиснемо F12 (рис. 12.5).

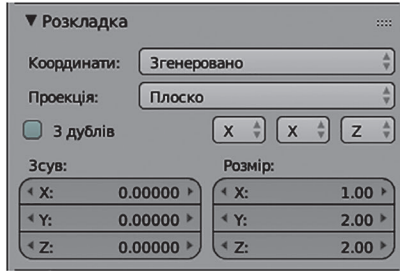


Рис. 12.4

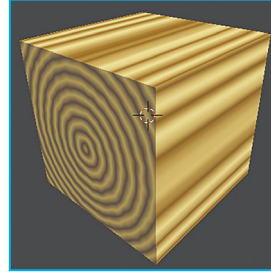


Рис. 12.5

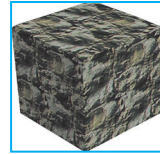
## Графічні текстури

На грані сітей можна накладати графічні текстури із файлів форматів PNG, JPG тощо (рис. 12.6, а):

- 1) підготувати растрові зображення для текстур за допомогою графічного редактора;
- 2) під час додавання текстури зі списку типів текстур (див. рис. 12.2) вибрати Зображення або фільм;
- 3) уточнити спосіб накладання текстури на панелі Розкладка (див. рис. 12.4). Так, для об'єкта у формі паралелепіпеда зі списку Проекція слід вибрати варіант Куб.



а



б

Рис. 12.6

## Освітлення

Невдало підібране освітлення може зіпсувати художній задум.

У початковій сцені є одне точкове джерело світла. Щоб змінити його параметри, слід його вибрати і на панелі властивостей відкрити розділ Дані (рис. 12.7). Основними параметрами тут є тип джерела світла (Точка, Сонце, Прожектор, Півсфера, Площина) і його енергія.

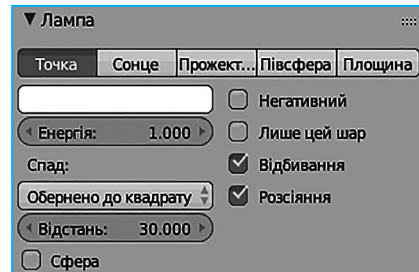


Рис. 12.7



- 2 На рис. 12.8 показано результат рендерингу сцени з циліндром, освітленим різними джерелами світла: точка (а), сонце (б), півсфера (в).

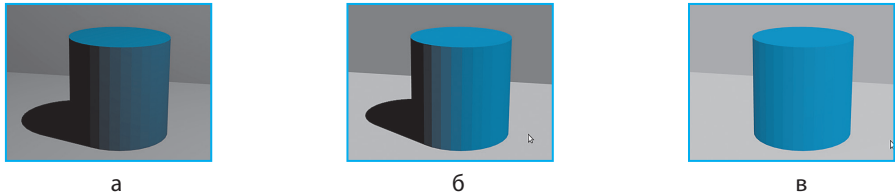


Рис. 12.8

**!** За потреби до сцени можна додати два і більше джерел світла.

Як бачимо, реалістичні текстури та вдале освітлення є важливими елементами сцени і значно покращують її сприйняття.

### Питання для самоперевірки



1. Що таке рендеринг?
2. Які налаштування слід зробити перед рендерингом сцени?
3. Що таке процедурна текстура?
4. Як створюють і використовують графічні текстури?
5. Як створити матеріал із текстурою деревини?
6. Джерела світла яких видів підтримує Blender?

### Вправа 12



▶▶ Дослідити властивості різних джерел світла.

- 1) Відкрийте файл `snowman.blend`. Відкрийте текстовий процесор з новим документом.
- 2) На вигляді зверху (NUM7) розташуйте джерело світла так, як показано на рис. 12.9. Переконайтесь, що воно має тип Точка. Зробіть скриншот і додайте в текстовий документ.
- 3) Виконайте рендеринг (F12). Зробіть скриншот, додайте в текстовий документ і зазначте тип джерела (Точка).

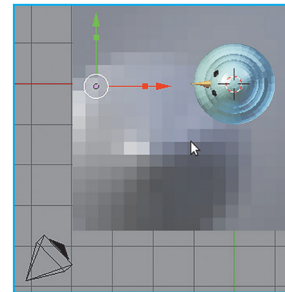


Рис. 12.9



- 4) Змініть тип джерела світла на Сонце. Виконайте рендеринг і додайте в документ скриншот із відповідним підписом.

- 5) Поверніть джерело світла Сонце в інший бік і додайте скриншот із підписом.
- 6) Дослідіть так само джерело світла Півсфера. Збережіть документ із назвою Вправа 12.



### Комп'ютерне тестування

Виконайте тестове завдання 12 з автоматичною перевіркою результату.



## § 13. Анімація ключовими кадрами

Blender дозволяє розробляти не тільки нерухомі (статичні) зображення, але й має потужні засоби для створення анімаційних фільмів.

Як відомо, ефект анімації виникає, якщо послідовно демонструвати з достатньою частотою нерухомі зображення.

Розглянемо основні принципи і налаштування анімації у Blender.

### Шкала часу

Для налаштування анімації активно використовується часова шкала (рис. 13.1), яка типово розташована внизу вікна Blender (див. рис. 8.8, 5).

На шкалі часу числами (0, 20, 40...) позначено номери кадрів. Світло-сірий колір показує діапазон кадрів, які буде опрацьовано під час рендерингу. Спочатку це 250 кадрів, що за частоти 25 кадрів на секунду дасть відео тривалістю 10 секунд.



Рис. 13.1

Щоб **перейти на певний кадр**, потрібно клацнути на шкалі поблизу його номера або ввести номер кадру в поле

## Ключові кадри

Одним із методів анімації, доступних у Blender, є метод ключових кадрів. На певних кадрах, які називають **ключовими**, об'єкт розташовують так, як потрібно, і фіксують це розташування. Для проміжних кадрів положення обчислюється автоматично.

Ключові кадри розрізняють за тим, який саме параметр вони фіксують: координати, кут повороту, розмір об'єкта вздовж однієї з осей тощо.

1 На сцені з розташованими поряд червоним і зеленим кубами налаштуємо анімацію за таким сценарієм: куби нерухомі; червоний куб піднімається; перелітає над зеленим; опускається з іншого боку (рис. 13.2).

- 1) Перейдемо на 20-й кадр, клацнувши на часовій шкалі відповідне місце, і виберемо червоний куб.
- 2) Коли вказівник перебуває у вікні 3D-вигляду, натиснемо клавішу I — з'явиться меню (рис. 13.3).



Рис. 13.2

Вставляння ключового кадру	
<b>Розташування</b>	I
Обертання	I
Розмір	I
РозташОберт	I
РозташМасштаб	I
РозташОбертМасштаб	I
ОбертМасштаб	I

Рис. 13.3

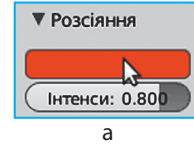
- 3) Оскільки куб має рухатися поступально, тобто без обертання, то виберемо тип Розташування. Це означає, що на перших 20 кадрах куби будуть нерухомими.
- 4) Перейдемо на 40-й кадр, пересунемо червоний куб угору, натиснемо I і знову додамо ключовий кадр для розташування.
- 5) Повторимо попередні кроки ще двічі: на 80-му і 95-му кадрах.
- 6) Перейдемо на 1 кадр і запустимо попередній перегляд анімації (сполучення клавіш Alt+A) — червоний куб перелетить над зеленим і опуститься біля нього.

Так само додають ключові кадри для інших трансформацій (див. рис. 12.3). Наприклад, якщо об'єкт має одночасно рухатись, обертатись і змінювати розміри, то слід вибрати тип ключового кадру РозташОбертМасштаб (розташування, обертання, масштабування).

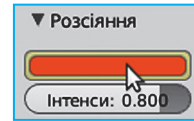
Як вже було сказано, під час анімації можуть змінюватися не тільки розташування і розміри об'єкта, але й інші його властивості: колір, прозорість, енергія джерела світла тощо.

2 Додамо для рухомого куба (див. приклад 1) анімацію кольору: нехай його колір під час руху поступово зміниться на синій.

- 1) Виберемо червоний куб і перейдемо на 20-й кадр, клацнувши на часовій шкалі відповідне місце.
- 2) Відкриємо панель властивостей матеріалу і наведемо вказівник на зразок кольору (рис. 13.4, а).
- 3) Натиснемо клавішу з англійською буквою I — навколо зразка кольору з'явиться жовта рамка, яка означає, що для цього кольору створено ключовий кадр.
- 4) Перейдемо на 85-й кадр, змінимо колір на синій, наведемо вказівник на зразок кольору і знову натиснемо I.



а




б

Рис. 13.4

Тепер, під час перегляду анімації один з кубів не тільки перелітатиме через інший, але й змінюватиме при цьому колір. Так само можна додавати анімацію для інших властивостей об'єктів.

## Рендеринг анімації

Розглянемо, як створити відеоролик з анімацією. Для цього слід:

- 1) на панелі Вимальовування (див. рис. 12.1) увести номери початкового і кінцевого кадрів;
- 2) на панелі Вивід (рис. 13.5) вибрати формат відео (наприклад, H.264 і колірний режим (BW — відтінки сірого, RGB — кольоровий);
- 3) клацнути кнопку  і вибрати папку для збереження відеоролика;
- 4) на панелі Вимальовування натиснути кнопку Анімація.

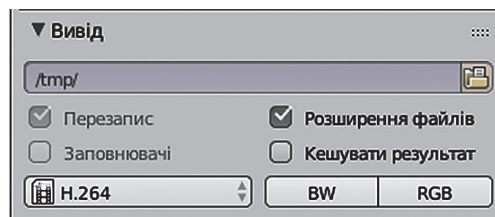


Рис. 13.5

Розпочнеться рендеринг, за яким можна спостерігати на екрані, і через певний час у зазначеній папці з'явиться створений відеоролик.

Отже, щоб отримати відеоролик, потрібно змодельовати об'єкти, налаштувати анімацію і виконати рендеринг.

### Питання для самоперевірки



1. Як досягається ефект анімації?
2. Для чого призначена шкала часу?
3. Поясніть принцип анімації ключовими кадрами.
4. Які типи ключових кадрів ви знаєте?
5. Як зробити, щоб яскравість джерела світла поступово зменшувалась?
6. Які основні параметри для рендерингу анімації?

### Вправа 13



▶▶ Створити анімаційний відеоролик.

- 1) Запустіть Blender. Додайте ще один куб і налаштуйте анімацію за алгоритмом із прикладу 1.
- 2) Додайте анімацію кольору одного з кубів за алгоритмом із прикладу 2.
- 3) На панелі Вимальовування задайте розмір кадру камери  $720 \times 576$  пікселів. Налаштуйте за допомогою ключових кадрів рух камери так, щоб вона поступово наближалася до кубів протягом перших 2 с.
- 4) Додайте площину — поверхню, на якій розміщені куби. Налаштуйте анімацію її кольору на свій смак.
- 5) Налаштуйте джерело світла так, щоб його енергія від 100-го до 120-го кадру спадала приблизно вдвічі.
- 6) Виконайте рендеринг анімаційного ролика (від 1 до 120 кадру) з якістю 25 % у файл з назвою Вправа 13.  
Перегляньте результат і, за потреби, змініть налаштування. Виконайте рендеринг з якістю 100 %.



### Комп'ютерне тестування



Виконайте тестове завдання 13 з автоматичною перевіркою результату.



## § 14. Робота з текстовими об'єктами

Тривимірні об'єкти, які містять написи, дають нам змогу отримувати цікаві ефекти.

### Створення текстового об'єкта



Для створення напису слід додати до сцени об'єкт типу Текст (на панелі Додати примітив у групі Інші). За замовчуванням він містить слово Text.

Якщо перейти в режим редагування (Tab), то можна змінити текст подібно до того, як це робиться в текстових редакторах.

! При роботі з текстом у вибраному шрифті може не виявитись українських літер. Тоді при натисканні клавіш букви не з'являтимуться, але коди символів у пам'яті збережуться. Після вибору іншого шрифту букви можуть з'явитись.

### Властивості текстового об'єкта

Вибрати шрифт для текстового об'єкта і зробити інші налаштування можна у вікні Властивості в розділі Дані (F).

Для вибору шрифту потрібно на панелі Шрифт клацнути кнопку  і в діалоговому вікні відшукати і вибрати файл зі шрифтом. Надалі для інших об'єктів цей шрифт буде доступним у списку шрифтів .

1 Текстовий об'єкт (рис. 14.1) утворений з кривих (це окремий тип об'єктів) і є плоским (а). Щоб він став об'ємним, у вікні властивостей на панелі Геометрія слід задати рівні витиснення (б) і обведення (фаски) (в). Параметр Роздільність визначає рівень згладжування фаски (г).

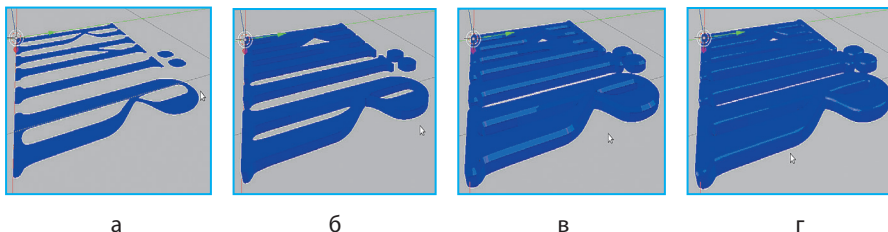


Рис. 14.1

## Конвертування текстових об'єктів

Якщо з текстовим об'єктом потрібно працювати, як зі звичайним сітвовим, його слід конвертувати (від англ. *convert* — перетворити).

Для конвертування вибраного об'єкта слід вибрати команду меню Об'єкт → Конвертувати в → Каркас з кривої/мета/поверхні/тексту (або натиснути Alt+C і вибрати команду) (рис. 14.2). Форма об'єкта після цього не зміниться, проте в режимі редагування (рис. 14.3) видно, що тепер це сітвовий об'єкт і його неможливо редагувати, як текстовий.

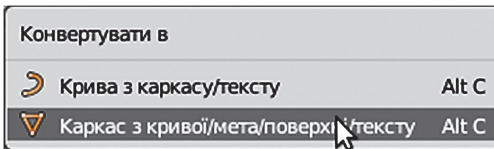


Рис. 14.2

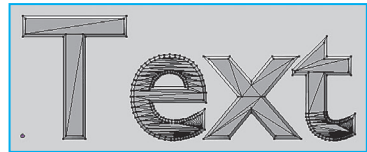


Рис. 14.3

- ! Через особливості алгоритму після конвертування окремі вершини можуть мати дублі, що ускладнює подальші операції із сіттю. Щоб усунути дублювання, слід виділити всі вершини (A), а потім вибрати команду меню Каркас → Вершини → Вилучити дублі.

Результатом конвертування є об'єкт, складений із декількох частин. Якщо є потреба працювати з окремими літерами (наприклад, налаштувати їх анімацію незалежно), об'єкт **розділяють**. Для цього потрібно:

- 1) виділити об'єкт і перейти в режим редагування (Tab);
- 2) викликати меню Каркас → Вершини → Відокремити або натиснути P;
- 3) вибрати один варіант розділення (рис. 14.4):

- Вибране — в новий об'єкт відокремитися виділена частина поточного об'єкта;
- За матеріалом — нові об'єкти утворюються з тих частин поточного, які мають однаковий матеріал;
- На окремі частини — кожна зв'язна частина сіті (наприклад, кожна літера) стане окремим об'єктом.

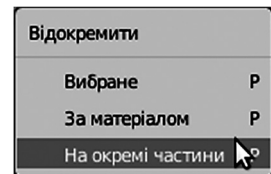


Рис. 14.4

- ! Опорні точки всіх утворених об'єктів збігаються з опорною точкою початкового об'єкта.

Для подальших маніпуляцій з об'єктом його опорну точку можна перемістити. Для цього слід, вибравши варіант зі списку, Задати опорну точку на вкладці Засоби панелі інструментів (рис. 14.5) («Геометрією» в цих командах названо сукупність вершин об'єкта).

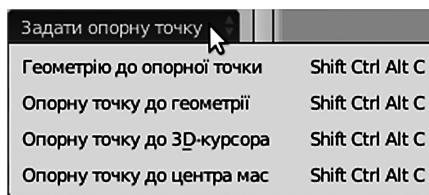


Рис. 14.5

Щоб об'єднати два і більше сітвових об'єктів, їх слід виділити і вибрати команду меню Об'єкт → Об'єднати (Ctrl+J).

2 Для конвертування і розділення на окремі літери об'єкта зі словом Київ (див. рис. 14.1 з прикладу 1), послідовність дій може бути такою:

- 1) виділити об'єкт, натиснути Alt+C і вибрати Каркас з кривої/мета/поверхні/тексту;
- 2) перейти в режим редагування, натиснути P і вибрати На окремі частини;
- 3) перейти в об'єктний режим, вибрати три об'єкти, що утворюють літеру «і» (крапки і паличку), і натиснути Ctrl+J;
- 4) виділити літери і вибрати команду Опорну точку до геометрії.

## Додатки до Blender

Значно розширити можливості користувачів і користувачок дозволяють додатки до Blender. Існують дуже різні додатки: одні реалізують прості покращення інтерфейсу, інші призначені для швидкого створення дуже складних об'єктів.

3 Наприклад, є додатки для:

- моделювання болтів і гайок з різними параметрами (рис. 14.6)
- моделювання рослин;
- додання до сцени моделі Сонця і неба;
- показу в кутку вікна 3D-вигляду інформації про натиснуті клавіші та кнопки миші тощо.

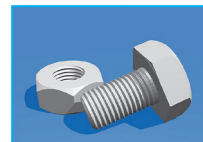


Рис. 14.6

Багато додатків постачається разом із програмою: частину з них створили й офіційно підтримують розробники програми, інші розроблені користувачською спільнотою. Людина, яка володіє програмуванням (наприклад, мовою Python), може ознайомитися з документацією Blender і почати створювати й поширювати власні додатки.



Щоб увімкнути, наприклад, додаток Import Images as Planes, який дозволяє перенести на сцену вибрані фотографії, розмістивши їх на площинах, потрібно:

- 1) відкрити вікно командою Файл → Користувацькі уподобання... і перейти на вкладку Додатки (рис. 14.7);

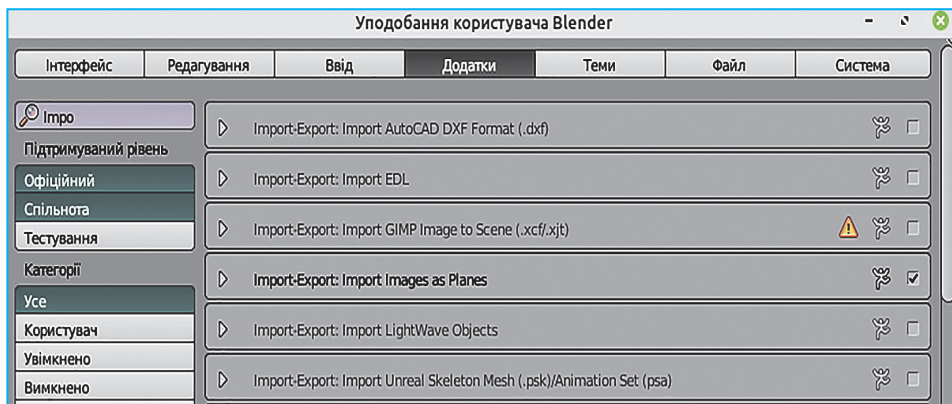


Рис. 14.7

- 2) розпочати набирати в полі пошуку назву додатку — список автоматично фільтруватиметься. Кнопка ▶ перед назвою додатку розкриває панель з його описом;
- 3) у кінці рядка з назвою додатку поставити позначку і закрити вікно.

Після цього в меню Файл → Імпортувати з'явиться команда Images as Planes. Якщо викликати її, вибрати в діалоговому вікні файли зображень і підтвердити вибір, на сцені з'являться примітиви типу Площина з вибраними зображеннями. Далі з ними можна працювати, як з іншими об'єктами, наприклад зробити анімацію.


### Питання для самоперевірки



1. Як додати в сцену текстовий об'єкт?
2. Як змінити текст у текстовому об'єкті?
3. Під час набирання тексту не видно українських літер. Що робити в такому випадку?
4. Які параметри визначають вигляд текстового об'єкта?
5. Хто і для чого створює додатки до Blender?
6. Опишіть послідовність увімкнення додатку.

## Вправа 14



1. Розділіть вікно 3D-вигляду по вертикалі на два вікна. Змініть тип правого вікна на Текстовий редактор.
2. Натисніть у заголовку цього вікна кнопку  **Новий** і наберіть декілька рядків тексту, наприклад уривок вірша М. Рильського:

Благословен той день і час,  
Коли прослалась килимами  
Земля, яку сходив Тарас...

3. Виберіть Редагувати → Текст в 3D-об'єкт → Один об'єкт на рядок.
4. Якщо українських букв не видно, виберіть у вікні Дерево елементів перший текстовий об'єкт (Текст) і виберіть для нього шрифт, у якому є українські літери. Зробіть це саме для решти текстових об'єктів.
5. Додайте витискання і фаску до текстових об'єктів (рис. 14.8).

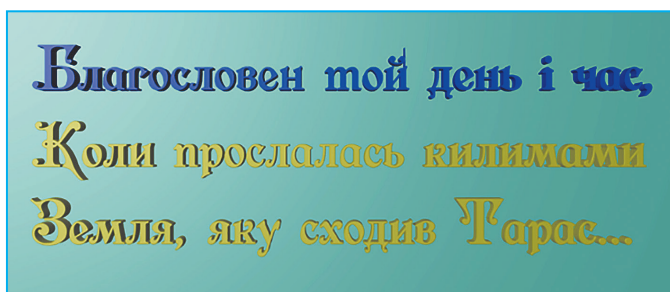


Рис. 14.8

6. Змініть колір текстових об'єктів. Розташуйте камеру так, щоб вони гарно розмістились у кадрі. Збережіть файл із назвою Вправа 14.

## Комп'ютерне тестування



Виконайте тестове завдання 14 з автоматичною перевіркою результату.





## Практична робота 5

### Тривимірна анімація

**Завдання:** змоделювати сцену з графічних примітивів і текстового об'єкта, налаштувати анімацію та створити відеоролик.

**Обладнання:** комп'ютер із тривимірним редактором Blender.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

1. Запустіть Blender. Видаліть куб.
2. Додайте до сцени текстовий об'єкт (наприклад, із власним ім'ям, написом 9 клас). Доберіть шрифт з українськими літерами.
3. На панелі Геометрія задайте витиснення і глибину обведення.
4. Конвертуйте текстовий об'єкт у сіть і розділіть на об'єкти.
5. Перемістіть опорну точку кожного символу до «геометрії».
6. Доберіть для кожного символу колір.
7. Встановіть камеру так, щоб напис був посередині кадру.
8. Підберіть розташування і тип джерела світла.
9. Виділіть перший символ і налаштуйте анімацію, щоб він протягом першої секунди (за частоти 25 кадрів/с) прилітав на своє місце з-за меж кадру. Для цього:
  - а) додайте на 25 кадрі ключовий кадр для розташування;
  - б) перейдіть на 1-й кадр, перемістіть об'єкт за межі кадру та додайте ще один ключовий кадр для розташування.Перегляньте результат, натиснувши Alt+A.
10. Налаштуйте анімацію решти букв: нехай деякі з них прилітають обертаючись, змінюють розміри тощо.
11. Налаштуйте анімацію кольору так, щоб після того, як усі літери займуть свої місця, кожна протягом 1 с змінила колір.
12. Налаштуйте параметри рендерингу відео: початковий кадр — 1, кінцевий — після останнього ключового кадру, частота кадрів — 25 кадрів/с, формат відео — H.264. Виконайте пробний рендеринг з розміром 25%. Перегляньте результат, виправте недоліки. Виконайте кінцевий рендеринг з якістю 100%. Обговоріть результат.

**Зробіть висновок:** яка послідовність розробки відеоролика у тривимірному редакторі Blender.

# РОЗДІЛ 3

## ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ



§ 15. Типи посилань на клітинки в Excel

Практична робота 6. Розв'язування задач на обчислення

§ 16. Логічні функції

§ 17. Математичні та статистичні функції

§ 18. Параметри сторінки. Друкування таблиці

§ 19. Упорядкування даних. Умове форматування

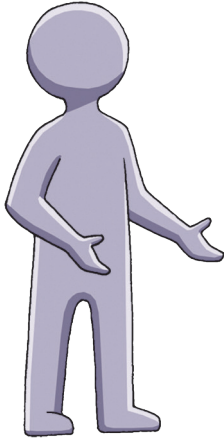
Практична робота 7. Використання математичних, логічних та статистичних функцій табличного процесора. Умове форматування

§ 20. Розширені фільтри. Проміжні підсумки

§ 21. Діаграми і графіки

Практична робота 8. Упорядкування даних у таблицях. Автоматичні та розширені фільтри

## ПОВТОРЮЄМО



Ви вже ознайомилися з табличним процесором Excel — прикладною програмою для опрацювання даних в електронних таблицях. Документ Excel називається *робочою книгою* і складається з аркушів. Аркуш містить *електронну таблицю*, що складається із рядків і стовпців. На перетині рядків і стовпців електронної таблиці знаходяться *клітинки*. Кожна клітинка має адресу, що складається із назви стовпця і номера рядка. Клітинкам і діапазонам клітинок можна надавати імена.

Клітинки можуть містити *дані різного типу*: текстові, числові тощо. Дані в клітинках можна використовувати для обчислень за допомогою *формул*. Будь-яка формула починається знаком «=» (дорівнює) і може містити адреси клітинок, функції, знаки арифметичних і логічних операцій.

1. Як називають програму опрацювання даних в електронних таблицях?
2. Як записується адреса клітинки?
3. Чи можуть збігатися адреси різних клітинок?
4. Які дані можуть містити клітинки таблиці?
5. Яке призначення формул і функцій?
6. З якого знака починається запис формули?



Формули в Excel можуть містити вбудовані функції, з деякими з них ви вже ознайомилися, деякі вивчите в цьому розділі. Ви розглянете методи опрацювання даних в електронних таблицях, навчитесь їх упорядковувати.

## § 15. Типи посилань на клітинки в Excel

Під час розв'язування прикладних задач виникає потреба в опрацюванні даних, що містяться в кількох клітинках. Так, щоб знайти найбільше або найменше число, розрахувати середнє арифметичне тощо, слід опрацювати множину чисел у певному діапазоні клітинок.

### Виділення діапазонів клітинок

У тих випадках, коли потрібно опрацювати сукупність чисел: відшукати найбільше чи найменше, розрахувати середнє арифметичне або суму тощо, вказують посилання не на одну клітинку таблиці, а на певний діапазон клітинок.



**Діапазон** — це сукупність клітинок, які можна опрацювати як єдине ціле.

Залежно від способу задання, розрізняють два види діапазонів: *зв'язні* і *незв'язні*.

**Зв'язний діапазон** позначається адресами лівої верхньої і правої нижньої клітинок або двома заголовками рядків (стовпців), відокремленими двокрапкою. *Клітинку* можна вважати зв'язним діапазоном.

- Наприклад, на рис. 15.1 A1:C5 — прямокутний діапазон;  
A:B — діапазон з усіх клітинок стовпців A і B;  
2:4 — діапазон з усіх клітинок рядків 2, 3 і 4.

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				

Рис. 15.1

Для виділення зв'язного прямокутного діапазону потрібно при натиснутій лівій кнопці миші протягнути вказівник від однієї кутової клітинки діапазону до діагонально протилежної (незалежно від напрямку руху вказівника).

Для виділення зв'язного діапазону стовпців або рядків потрібно протягнути вказівник з натиснутою лівою кнопкою миші по заголовках стовпців або рядків.

**Незв'язним** називається діапазон, який складається з кількох зв'язних діапазонів.

Щоб **виділити незв'язний діапазон**, потрібно, утримуючи натиснутою клавішу Ctrl, послідовно виділити декілька зв'язних діапазонів. Незв'язний діапазон позначають посиланнями на окремі його частини, відокремлюючи їх крапкою з комою: A1:C5;B7 (див. рис. 15.1).

Для **скасування виділень** слід клацнути на будь-якій клітинці.

- ! Зв'язний діапазон можна копіювати і переміщувати. Для незв'язних діапазонів такі операції не завжди можливі.

## Посилання на адреси комірок



Адреси клітинок і діапазонів, використані у формулах, називають **посиланнями**.

Клітинки, на які у формулах зроблено посилання, називатимемо **адресними**.

Розрізняють три типи посилань: *відносні*, *абсолютні*, *мішані*.

Пригадаємо, що адреса клітинки складається із заголовка стовпця (зазвичай — це літера) та номера рядка. Посилання, записані так, як розглянуто раніше, називають **відносними**. Під час копіювання формул до інших клітинок посилання у формулах модифікуються так, що імена стовпців і рядків змінюються на величину зміщення нової клітинки з формулою відносно вихідної.

- 2 Якщо формулу  $=A4+B5$  скопіювати в клітинку, яка знаходиться на два стовпці праворуч і на три рядки вище, то в новій формулі всі значення номерів стовпців збільшаться на два, а номерів рядків зменшаться на три, і отримаємо формулу:  $=C1+D2$  (рис. 15.2).

	A	B	C	D	E
1			17		$=C1+D2$ 37
2				20	
3					
4	10		$=A4+B5$ 26		
5		16			

Diagram illustrating the relative reference adjustment. A blue box labeled "Ctrl+C" is positioned below cell B5. A blue box labeled "Ctrl+V" is positioned below cell E1. Arrows show the movement from B5 to E1: a vertical arrow from row 5 to row 1 (labeled "4-3=1") and a horizontal arrow from column B to column E (labeled "A+2=C").

Рис. 15.2

Якщо формула містить адресу клітинки, у якій вона ж і записана, то виникає помилка «циклічне посилання».

**Копіювання формул у клітинки** можна виконати за допомогою контекстного меню комірки або сполучень клавіш Ctrl+C, Ctrl+V. **Копіювання формул до суміжних клітинок** можна здійснити перетягуванням маркера клітинки з формулою (рис. 15.3). При цьому модифікована формула копіюється до всіх клітинок діапазону.

Якщо у формулах як незмінні коефіцієнти застосовуються посилання на клітинки зі значенням цих коефіцієнтів, то посилання на такі клітинки не повинні змінюватись. У цьому випадку використовують **абсолютні посилання**, які не змінюються під час копіювання формул в інші клітинки. Для **уведення абсолютних посилань** використовують знак \$. Якщо в будь-яку клітинку скопіювати формулу =\$A\$1, вона не змінить свого вигляду і вказуватиме на клітинку з адресою A1 (рис. 15.4).

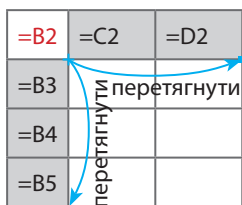


Рис. 15.3

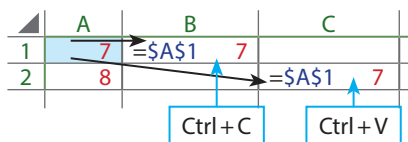


Рис. 15.4

Якщо посилання на клітинку є різнотипними, то такі посилання називають **мішаним**. Під час копіювання формул з мішаними посиланнями модифікуються лише відносні імена рядків і стовпців, а абсолютні — у посиланнях залишаються незмінними.

Автоматична зміна типу посилання, на якому встановлено курсор, відбувається при натисненні клавіші F4.

При **копіюванні адресних клітинок або їх вмісту** до інших клітинок посилання на них у формулах не модифікуються.

При **переміщенні адресних клітинок чи діапазонів** до інших клітинок модифікуються усі посилання на них незалежно від типу посилань та їх місцезнаходження.

3 Так, під час переміщення клітинки A1 на місце клітинки C3 формули

=A1; =\$A\$1; =A\$1; =\$A1

модифікуються на

=C3; =\$C\$3; =C\$3; =\$C3 (рис. 15.5).

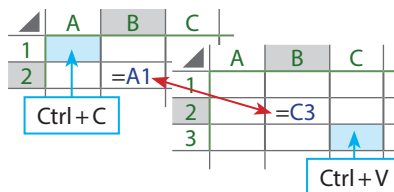


Рис. 15.5



Так само модифікуються усі посилання при переміщенні адресної клітинки в результаті вставки або видалення стовпців, рядків, клітинок чи діапазонів вище або ліворуч від адресної.

Видалення адресних клітинок призводить до помилок у формулах, які посилались на видалені клітинки (рис. 15.6).



Рис. 15.6

### Питання для самоперевірки



1. Як виділити прямокутний діапазон клітинок?
2. Скільки клітинок містить зв'язний діапазон A2:B3?
3. Як виділити незв'язний діапазон (несуміжні клітинки)?
4. Що таке відносне посилання?
5. Як записують абсолютні посилання?
6. Як модифікується формула =B5 при копіюванні в клітинку, що міститься на два стовпці ліворуч і на три рядки нижче від клітинки з вихідною формулою?

### Вправа 15



►► Розрахувати масу продуктів для приготування борщу у шкільній їдальні. Маса продукту розраховується як добуток норми його закладки на кількість порцій.

- 1) Запустіть програму Excel. Заповніть таблицю за зразком (рис. 15.7).

	A	B	C	D	E	F
1		за нормою, г.	85	172	155	57
2	Класи	Порцій	М'ясо	Картопля	Капуста	Буряк
3	8	117				
4	9	109				
5	10	107				
6	11	121				
7	Усього	454				

Рис. 15.7

- 2) Уведіть у клітинку С3 формулу розрахунку маси м'яса для учнів і учениць восьмих класів.  
Установіть у розрахункових клітинках числовий формат з одним десятковим знаком після коми.
- 3) Уведіть у клітинки D3; E3; F3 формули розрахунку маси інших продуктів харчування для учнів і учениць восьмих класів.  
Перетягуванням маркера скопіюйте усі формули для восьмих класів у вертикальні діапазони клітинок для інших класів.
- 4) Використовуючи функцію SUM, створіть формулу підрахунку загальної кількості порцій для учнів усіх класів у клітинці B7.
- 5) Перетягуванням маркера клітинки B7 скопіюйте формулу в клітинки горизонтального діапазону C7:F7.  
У клітинку G2 впишіть Маса продуктів (кг), а в клітинку G3 уведіть формулу для розрахунку загальної маси продуктів (м'яса, картоплі, капусти, буряка) для учнів восьмих класів.
- 6) У клітинках G4:G6 розрахуйте загальну масу продуктів для решти класів.  
Збережіть файл з іменем Вправа 15 у папці за вказівкою вчителя. Заверште роботу за комп'ютером.



### Комп'ютерне тестування



Виконайте тестове завдання 15 з автоматичною перевіркою результату.



## Практична робота 6

### Розв'язування задач на обчислення

**Завдання:** розрахувати заробітну плату працівників за заданий місяць, використовуючи формули і функції, посилання різних типів.

**Обладнання:** комп'ютер зі встановленим табличним процесором Excel.

## Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму Excel. Заповніть початкову таблицю розрахунку зарплат за зразком (окрім клітинок E2; B8; C4:E8).

	A	B	C	D	E
1		місяць			курс €
2		22			25,55
3	Прізвище	Днів	Оклад	Гривні	Євро
4	Петренко	20	5 900,00₴	5 363,64₴	€ 209,93
5	Сидоренко	13	4 880,00₴	2 883,64₴	€ 112,86
6	Іванова	12	6 175,00₴	3 368,18₴	€ 131,83
7	Петров	17	6 266,00₴	4 841,91₴	€ 189,51
8		62	5 805,25₴	16 457,36₴	€ 644,12

2. Знайдіть в інтернеті курси валют і заповніть клітинку E2.
3. У клітинці D4 за формулою розрахуйте зарплату Петренка у гривнях.

*Приклад розрахунку:* «Оклад» — це зарібок працівника за дні, вказані у клітинці B2. Петренко заробляє  $5900/22 \approx 268,18$  грн на день. Оскільки він відпрацював 20 днів, то його зарплата становитиме:  $268,18 \cdot 20 \approx 5363,6$  (грн).

4. Установіть для усіх розрахункових клітинок відповідний грошовий формат із двома десятковими знаками після коми.
5. Скопіюйте формулу зарплати в гривнях для решти працівників.
6. За допомогою формули розрахуйте зарплату Петренка в євро.
7. Скопіюйте формулу в клітинки для розрахунку зарплати усіх працівників у євро.
8. Скориставшись функцією, розрахуйте загальну кількість днів, які відпрацювали робітники, у клітинці B8.
9. Скориставшись функцією, розрахуйте середній оклад у клітинці C8.
10. Копіюванням формул розрахуйте загальний зарібок усіх працівників у гривнях та євро (клітинки D8:E8).
11. Збережіть файл з іменем Практична робота 8 у папці за вказівкою вчителя.
12. Заверште роботу за комп'ютером.

**Зробіть висновок:** від чого залежить правильність обчислень за скопійованими формулами; у яких випадках посилання у формулах повинні бути абсолютними і чому.

## § 16. Логічні функції

У Excel разом із формулами використовують функції.

### Класифікація функцій

Як ви вже знаєте, **функція** — це особливе (словесне) позначення дій над даними. Excel має декілька сотень вбудованих функцій, які за призначенням поділяють на окремі категорії: *математичні, статистичні, логічні, фінансові, інженерні* тощо.

Дані, які опрацьовує функція, називають **аргументами**. Аргументами функції можуть бути інші функції, числа, вирази, посилання, умови тощо.

Будь-яка функція в Excel характеризується *іменем, кількістю аргументів, типом аргументів, результатом дії*.

Розглянемо відому вам із 7 класу функцію (рис. 16.1):

- ім'я — MAX;
- кількість аргументів — не менше одного;
- тип аргументу — числовий;
- результат — найбільше значення зі списку аргументів.

Аргументи функції записують у дужках і за наявності у функції декількох аргументів розділяють символом «;».

Функції зручно вставляти в клітинку кнопкою fx (Встановити функцію), що міститься в рядку формул. Тоді у вікні Вставлення функції можна вибрати функцію за категорією та ім'ям, ознайомитися з описом вибраної функції, кількістю та типами аргументів або знайти ім'я функції за описом її дії (рис. 16.2).

	A	B	C
1	=MAX(B1;C2)	4	1 2
2		3	4

↑ ім'я  
↑ аргументи  
↑ результат

Рис. 16.1

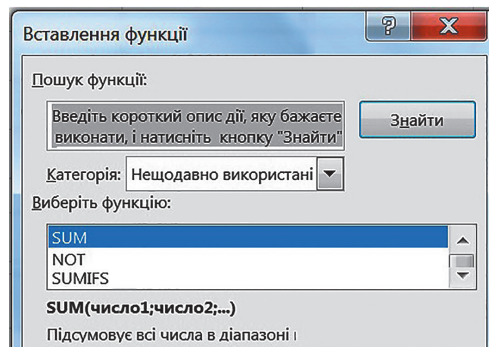


Рис. 16.2

## Логічні функції

У 7 класі ви вже зустрічались із висловлюваннями, що бувають істинними чи хибними.



**Логічний вираз** — це вираз, який набуває одного із двох значень: True — істина або False — хибність.

Прості логічні вирази складаються з двох виразів, пов'язаних операцією відношення:

- більше >
- менше <
- дорівнює =
- не менше (більше або дорівнює) >=
- не більше (менше або дорівнює) <=
- не дорівнює <>

1

Наприклад:

$2 > 1$  — істина,  $2 < 1$  — хибність.



**Логічна функція** — це функція, один або декілька аргументів якої — логічні вирази.

Розглянемо одну з основних логічних функцій IF.

IF(Умова; Вираз1; Вираз2).

Вираз1 — обчислюється, якщо умова істинна.

Вираз2 — обчислюється, якщо умова хибна.

2

Нехай у підсумковій відомості успішності учнів потрібно біля оцінки записати слово «добре», якщо учень має оцінку, вищу від 6 балів, і слово «задовільно», якщо учень має іншу оцінку. Алгоритм розв'язування цієї задачі подано у фрагменті блок-схеми (рис. 16.3).

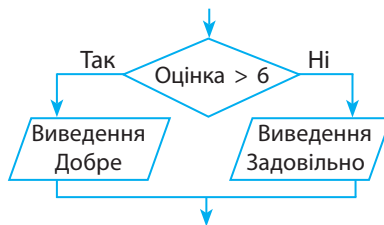


Рис. 16.3

У нашому випадку виконавцем алгоритму є табличний процесор. В Excel цей алгоритм реалізується за допомогою функції IF (рис. 16.4): IF(B1>6; "Добре"; "Задовільно")

	A	B	C	D	E
1	Петренко	5	=IF(B1>6; "Добре"; "Задовільно")		
2	Сидоренко	8	Добре		
3	Петрова	12	Добре		
4	Сидорова	4	Задовільно		

Рис. 16.4

У функціях Excel на місці логічного виразу може міститися арифметичний вираз. Табличний процесор перевіряє, чи не дорівнює значення арифметичного виразу нулю.

- 3 Наприклад, арифметичний вираз:  
 $2+2$  — трактується як істина (тому що  $2+2 \neq 0$ );  
 $2-2$  — трактується як хибність (тому що  $2-2=0$ ).

Іноді для розв'язання задачі потрібно комбінувати кілька умов.



**Складений логічний вираз** — вираз, у якому кілька простих логічних виразів пов'язані логічними операціями.

Основними логічними операціями є NOT (*заперечення*), AND (*кон'юнкція*, або *логічне множення*), OR (*диз'юнкція*, *логічне додавання*). В Excel вони реалізуються за допомогою відповідних функцій.

- 4 Наприклад, у відомості (див. рис. 15.4) потрібно дописати слово Посередньо для тих учнів і учениць, які мають оцінки, вищі від 6 балів, але нижчі за 10. Решті учнів і учениць нічого не писати. Тоді використовуємо логічні функції IF та AND у такому виразі:  
`=IF(AND(B1>6;B1<10); "Посередньо"; "")`

Третій аргумент функції IF — це лапки без символів між ними, тобто порожній рядок.

Іноді буває доцільно використати вкладені логічні функції IF.

- 5 Нехай потрібно підписати успішність учнів і учениць за трьома рівнями: «початковий» (1–4 бали); «середній» (5–8 балів); «високий» (9–12 балів).

В Excel цю задачу можна розв'язати таким чином (рис. 16.5):

`=IF(B1<5;"Початковий"; IF(AND(B1>4; B1<9); "Середній"; "Високий"))`

	A	B	C	D	E	F	G	H
1	Петренко	5	=IF(B1<5;"Початковий"; IF(AND(B1>4; B1<9); "Середній"; "Високий"))					
2	Сидоренко	8	Добре					
3	Петрова	12	Добре					
4	Сидорова	4	Задовільно					

Рис. 16.5

В Excel може бути вкладено одна в одну до семи функцій IF. Наведемо стислий опис ще кількох логічних функцій:

Ім'я	Опис	Кількість аргументів
IFERROR	Якщо перший аргумент не є помилкою, то повертає його значення, інакше повертає значення другого аргумента	Два
FALSE	Повертає значення «хибність»	Без аргументів
TRUE	Повертає значення «істина»	Без аргументів

### Питання для самоперевірки



1. Що таке функція?
2. Що таке аргумент функції?
3. Назвіть декілька категорій функцій в Excel.
4. Чи існують функції без аргументів в Excel?
5. Як позначають операції відношень?
6. Що є першим аргументом логічної функції IF?

### Вправа 16



- ▶▶ Створити в Excel таблицю успішності учнів з інформатики.
- 1) Запустіть програму Excel.  
Створіть таблицю успішності учнів вашої групи з інформатики:
    - у стовпці А — прізвища;
    - у стовпці В — оцінки за семестр.
  - 2) За допомогою функції IF у стовпці С підпишіть оцінки учнів і учениць за двома умовними рівнями: «початковий» (оцінка менша за 5 балів) і «високий» (для решти оцінок).
  - 3) У стовпці D за допомогою логічних функцій підпишіть оцінки за трьома рівнями: «початковий» (1–4 бали), «середній» (5–8 балів), «високий» (9–12 балів).
  - 4) Збережіть файл з іменем Вправа 16 у папці за вказівкою вчителя. Заверште роботу за комп'ютером.



### Комп'ютерне тестування



Виконайте тестове завдання 16 з автоматичною перевіркою результату.



## § 17. Математичні та статистичні функції

Математичних функцій в Excel значно більше, ніж ви вивчали раніше. Із повним переліком таких функцій можна ознайомитись на вкладці Функції → Бібліотека функцій → Математичні.

### Математичні функції



**Математичні функції** в Excel — це функції, призначені для виконання обчислень над даними клітинок: округлення, перетворення чисел тощо.

Коротко розглянемо призначення деяких математичних функцій:

Ім'я	Опис	Кількість аргументів
ROUND	Округлення до вказаної кількості десяткових цифр	Два
MOD	Остача від ділення	Два
PRODUCT	Добуток	Не менше одного
SUM	Сума	Не менше одного
SUMIF	Сума за умови, що...	Три

Аргументами одних функцій можуть бути інші функції.

1

Наприклад, якщо нам потрібно знайти добуток числових даних у діапазоні клітинок A1:B2 й округлити його до десятих, то формула матиме такий вигляд:

```
=ROUND(PRODUCT(A1:B2);1)
```

### Статистичні функції



**Статистичні функції** в Excel — це функції для аналізу значень діапазонів клітинок.

За допомогою статистичних функцій можна знайти найбільше і найменше значення, розрахувати середнє значення тощо.



- 2 Підрахуємо кількість учнів і учениць, які навчаються на 12 балів. Це легко зробити, використовуючи статистичну функцію COUNTIF. Першим аргументом є діапазон клітинок з оцінками, а другим аргументом — критерій вибору:

=COUNTIF(B3:B4;"=12")

Знак рівності в цій логічній умові можна пропускати:

=COUNTIF(B3:B4;12)

Критерієм вибору може бути число, дата, символ, слово тощо. За критерієм будується умова: чи містить поточна клітинка зазначені дані.

- 3 Наприклад, можна підрахувати кількість пропущених учнем або ученицею уроків (див. рисунок).

	A	B	C	D	E	F	G	H
1	Прізвище	02.вер	09.вер	16.вер	23.вер	30.вер	07.жовт	кількість пропусків
2	Петренко	6	7	н	8		н	=COUNTIF(B2:G2;"н")
3	Іванова	9					9	

Стисло розглянемо призначення деяких статистичних функцій:

Ім'я	Опис	Кількість аргументів
MIN	Мінімальне значення	Не менше одного
MAX	Максимальне значення	Не менше одного
AVERAGE	Середнє арифметичне	Не менше одного
COUNT	Кількість клітинок із числовими даними	Не менше одного
COUNTIF	Кількість клітинок, дані в яких задовольняють певній умові	Не менше одного
COUNTIFS	Кількість клітинок, дані в яких задовольняють декільком умовам	Не менше одного
COUNTA	Кількість непорожніх клітинок	Не менше одного

- » Френсіс Елізабет Аллен — американська вчена, піонерка в галузі оптимізації компіляторів, перша жінка, яка отримала премію Тюрінга.

## Питання для самоперевірки



1. Яке призначення математичних функцій?
2. Назвіть кілька математичних функцій.
3. Назвіть кілька статистичних функцій.
4. Назвіть призначення статистичних функцій.
5. Наведіть приклад застосування однієї зі статистичних функцій.
6. Вкажіть імена і призначення статистичних функцій з умовою.

## Вправа 17



►► Проаналізувати в Excel успішність учнів і учениць вашого класу з інформатики.

- 1) Запустіть програму Excel.  
Створіть таблицю успішності учнів з інформатики:
  - у стовпці A — прізвища,
  - у стовпцях B, C, D — поточні оцінки;
  - стовпець E (незаповнений) — підсумкові оцінки.
 Заповніть таблицю прізвищами учнів і учениць та поточними оцінками.
- 2) У стовпці E за допомогою статистичних функцій порахуйте підсумкові оцінки учнів і учениць як середню арифметичну, округлену до цілого значення оцінку.
- 3) У довільній клітинці під прізвищами за допомогою функції COUNTIFS порахуйте, скільки учнів навчаються на «середньому» рівні, тобто мають оцінки 5, 6, 7, 8 балів.
- 4) У довільній клітинці під прізвищами за допомогою функції COUNTA порахуйте загальну кількість поточних оцінок усіх учнів і учениць.
- 5) Збережіть файл з іменем Вправа 17 у папці за вказівкою вчителя.
- 6) Заверште роботу за комп'ютером.



## Комп'ютерне тестування



Виконайте тестове завдання 17 з автоматичною перевіркою результату.



## § 18. Параметри сторінки. Друкування таблиці

Налаштування параметрів сторінки в Excel та параметрів сторінок у текстовому процесорі Word, який ви вже вивчали, є аналогічними.

### Параметри сторінки

Установити параметри сторінки можна на вкладці Розмітка сторінки.

Наприклад, у вікні Параметри сторінки вибираємо потрібний інструмент (скажімо, Поля), а далі — одне із запропонованих значень або вводимо розміри вручну.

### Встановлення розмірів стовпців (рядків)

Для зміни розміру рядка (стовпця) достатньо перетягнути його межу між заголовками за нижній кут справа. При цьому розміри сусідніх рядків (стовпців) не змінюються, а в інформаційному вікні виводиться розмір у пунктах і пікселях (рис. 18.1).

Для встановлення однакової ширини кількох стовпчиків (висоти рядків) потрібно виділити, наприклад, протягнувши вказівник по заголовках рядків (стовпців), а потім або перетягнути одну з меж, або, вибравши в контекстному меню Ширина стовпця (Висота рядка), встановити потрібне значення.

На рис. 18.2 показано: 1 — виділення рядків; 2 — виділених рядків; 3 — всього рядків; 4 — виділення стовпців; 5 — всього стовпців; 6 — виділених стовпців.



Рис. 18.1

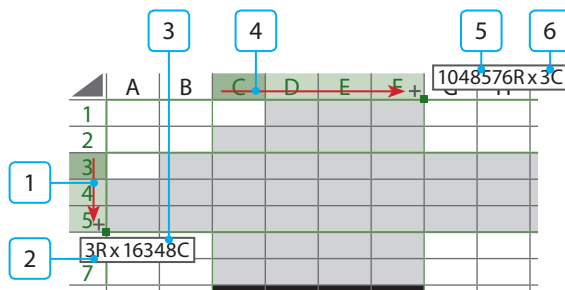


Рис. 18.2

## Оптимізація таблиці

Якщо таблиця не вміщується на сторінку, то вона автоматично розподілиться на кілька сторінок (рис. 18.3). При ручній оптимізації таблиці змінюють орієнтацію сторінки, розмір та орієнтацію тексту, розміри рядків і стовпців тощо (рис. 18.4).

Неоптимізований варіант

A	B	C	D	E	F	G	H
Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Астрономія	Географія
Петренко	12	10	8	9	11	10	12
Іваненко	7	8	9	8	7	9	8
Петров	5	6	5	7	6	6	7
Іванова	10	11	11	12	10	10	9

Рис. 18.3

Після вирівнювання і орієнтації тексту у клітинках верхнього рядка


Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Астрономія	Географія
Петренко	12	10	8	9	11	10	12
Іваненко	7	8	9	8	7	9	8
Петров	5	6	5	7	6	6	7
Іванова	10	11	11	12	10	10	9

Рис. 18.4

Отже, для **оптимізації таблиці** потрібно:

- 1) виділити потрібні клітинки верхнього рядка;
- 2) у вкладці Основне вибрати вкладку Вирівнювання;
- 3) у вікні Формат клітинок задати вирівнювання й орієнтацію тексту;
- 4) змінити висоту потрібного рядка і ширину стовпців з оцінками.

## Об'єднання клітинок

Часто при оформленні таблиць потрібно об'єднати кілька клітинок. Найпростіший спосіб — це виділити їх і вибрати у вкладці Основне в групі Вирівнювання кнопку Об'єднати й помістити у центрі . У такій клітинці залишаються тільки дані з верхньої лівої клітинки. Скасувати об'єднання можна повторним клацанням тієї самої кнопки (дані клітинок не відновлюються).



## Питання для самоперевірки



1. Як змінити розмір одного стовпця або рядка?
2. Як встановити однаковий розмір кількох стовпців (рядків)?
3. Що включає оптимізація таблиці?
4. Як об'єднати декілька клітинок?
5. Як відкрити вікно для налаштування заливки або меж клітинок?
6. Що передбачає підготовка електронної таблиці до друку?

## Вправа 18



▶▶ Створити електронну таблицю за зразком.

- 1) Запустіть програму Excel. Встановіть параметри сторінки:
  - усі поля — по 1 см;
  - розмір — А4 (21 × 29,7 см);
  - орієнтація — книжкова.

Встановіть режим перегляду сторінки: Вигляд → Розмітка сторінки.

- 2) Налаштуйте ширину стовпців: А — 0,5 см; В — 3,5 см. Іншим 27 стовпцям задайте ширину 0,5 см.
- 3) Встановіть висоту першого рядка — 1,5 см. Іншим 48 рядкам (з 2-го по 49-й) установіть висоту 0,5 см, 4-му рядку — 1,5 см.
- 4) Об'єднайте клітинки першого рядка над таблицею. Об'єднайте потрібні клітинки в майбутній таблиці. Тепер верхня частина таблиці виглядатиме, як на рис. 18.6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1																													
2																													
3																													
4																													
5																													

Рис. 18.6



- 5) Виділіть таблицю A2:AC49. Встановіть межі (окрім бокових) мінімальної товщини. У найбільшій клітинці встановіть діагональну межу середньої товщини. Виділіть потрібні діапазони і встановіть для них межі різної товщини (рис. 18.7).

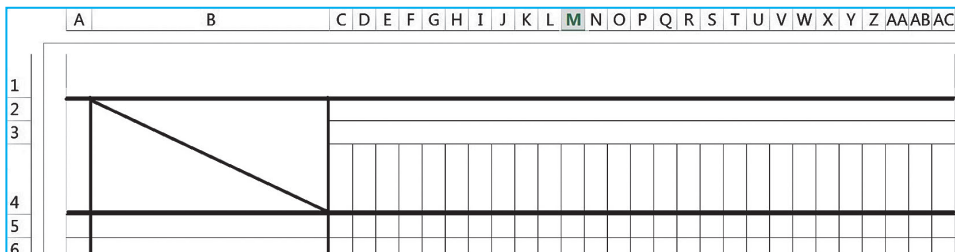


Рис. 18.7

- б) Збережіть файл з іменем Вправа 18 у папці за вказівкою вчителя. Заверште роботу за комп'ютером.



### Комп'ютерне тестування

Виконайте тестове завдання 18 з автоматичною перевіркою результату.



## § 19. Упорядкування даних. Умове форматування

Зазвичай табличні дані, отримані в результаті розрахунків або після ручного введення, потребують систематизації й упорядкування.

### Сортування даних



**Сортування даних** в Excel — це встановлення порядку розташування рядків за даними певного стовпця (стовпців).

Дані можна сортувати за *зростанням* (або в алфавітному порядку) чи *спаданням* (або у зворотному до алфавітного порядку).

Для **сортування даних** потрібно виділити *зв'язний* діапазон клітинок із даними й вибрати тип сортування: Дані → Сортування й фільтрування або Основне → Редагування → Сортування й фільтрування (рис. 19.1).

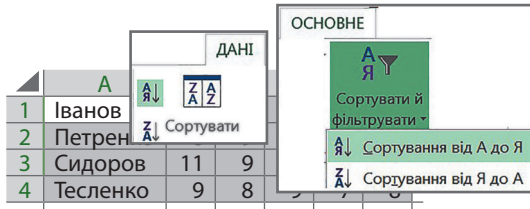


Рис. 19.1

При швидкому сортуванні за зростанням або спаданням, рядки виділеного діапазону переставляються за даними першого стовпця, порожні клітинки якого розташуються знизу незалежно від типу сортування. Якщо виділити один стовпець, то дані можуть бути відсортовані лише в ньому або в усьому діапазоні за даними виділеного стовпця.

Для складного сортування даних слід виконати дії: Дані → Сортування й фільтрування → Сортування, або Основне → Редагування → Сортувати й фільтрувати → Настроюване сортування. Тоді можна вказати, за даними яких стовпців сортувати, за якими параметрами, у якому порядку тощо. При налаштуванні складного сортування за даними кількох стовпців верхній рядок виділеного діапазону сортуванню не підлягає.

Порядок одночасного сортування задається у вікні Сортування. За кожним наступним стовпцем сортуватимуться лише ті рядки, в яких у попередньо відсортованому стовпці дані збігалися (рис. 19.2).

Не відсортовано	...за ім'ям	...за прізвищем	...за балом
Вихідна таблиця	Проміжний етап	Проміжний етап	Кінцева таблиця
Прізви.	Прізви.	Прізви.	Прізви.
Ім'я	Ім'я	Ім'я	Ім'я
Бал	Бал	Бал	Бал
Бабич Лев 11	Бабич Лев 11	Бабич Лев 11	Бабич Лев 11
Бойко Ганна 12	Валеев Олег 10	Валеев Олег 10	Валеев Олег 10
Зленко Ганна 10	Бойко Ганна 12	Бойко Ганна 12	Бойко Ганна 9
Валеев Олег 10	Зленко Ганна 10	Бойко Ганна 9	Бойко Ганна 12
Бойко Ганна 9	Бойко Ганна 9	Зленко Ганна 10	Зленко Ганна 10

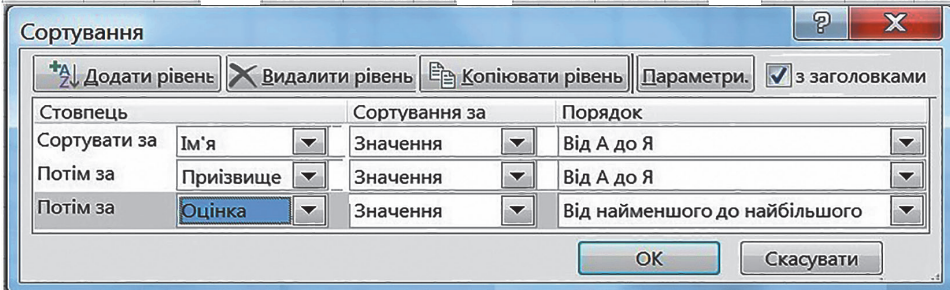


Рис. 19.2



## Фільтрування даних



**Фільтрування даних** — це відображення в таблиці тільки тих рядків, дані яких відповідають певним умовам (умовам фільтрування).

При фільтруванні решта рядків таблиці не знищується, а лише приховується і може бути відображена за інших умов.

Верхній рядок виділеного діапазону не фільтрується і відображається завжди, тому в нього зазвичай вписано заголовки, тобто назви даних, що знаходяться у відповідних стовпцях.

Для увімкнення режиму фільтрування активують кнопку Фільтр (рис. 19.3). При увімкненні фільтру для поточної клітинки у верхньому рядку зв'язаного з нею діапазону з'являться значки списку ▼ (розгортання фільтру).

	A	B	C	D	E	F
1	Іванова	7	10			
2	Тесленко	9	8		7	12
3	Сидоров	11	9		9	12
4	Петренко	8	11			11

Значок розгортання фільтру

Увімкнено: Фільтр

Рис. 19.3

Зазвичай для фільтрування виділяють потрібну кількість стовпців таблиці разом із заголовками. Тоді при увімкненні режиму фільтрування в рядку заголовків з'являться значки списку.

- Фільтр за кольором можна встановити, якщо колір шрифту або заливки відрізняється від чорного і білого.
- Для стовпців із числовими даними активним буде пункт Фільтри чисел.
- Для налаштування діапазону, відсутнього в переліку меню Фільтри чисел, наприклад *більше п'яти та менше дев'яти*, слід вибрати Користувачький фільтр.
- При встановленні нового фільтру в тому самому стовпці старе правило анулюється.

Фільтри можна застосовувати до кількох стовпців одночасно. При цьому кожен фільтр застосовується до рядків, які залишились видимими після застосування попередніх фільтрів.

Якщо стовпець містить дані текстового формату, то для нього буде активним пункт Текстові фільтри (рис. 19.4).

## Приклади фільтрації текстових даних

Вихідна таблиця

Прізвище	Алгебра	Хімія	Фізика	Інформатика	середній бал
Іванов	7	10	11	12	10
Петренко	12	11	8	10	10,3
Ісханов	11	9	8	9	9,3
Тесленко	9	8	9	7	8,3

Текстові фільтри

Пошук (Виділити все)

Іванов

Петренко

Сидоров

Тесленко

Користувачський автофільтр

Показати лише ті рядки, значення яких:

Іванов

дорівнює I\*ОВ

І  ДБО

починаються з букви "І" та закінчуються на "ов"

Відображення рядка з оцінками Іванова

Прізвище	Алгебра	Хімія	Фізика	Інформатика	середній бал
Іванов	7	10	11	12	10

Прізвище	Алгебра	Хімія	Фізика	Інформатика	середній бал
Іванов	7	10	11	12	10
Ісханов	11	9	8	9	9,3

Рис. 19.4

## Умове форматування

Для унаочнення даних вдаються до умовного форматування.



**Умове форматування даних** — це автоматична зміна формату клітинок залежно від значення даних у них.

Для умовного форматування слід виділити діапазон і вибрати потрібне у списку: Основне → Стили → Умове форматування. Умове форматування надає можливість оформити клітинки за стандартними правилами і стилями оформлення або встановити свої. До тих самих клітинок можна застосовувати кілька правил умовного форматування. Якщо ж вони суперечать одне одному, то «сильнішим» буде правило, встановлене пізніше.

При заповненні червоним кольором клітинок зі значеннями «більше 5» та зеленим — клітинок зі значеннями «менше 7» клітинки зі значенням «шість» зафарбуються зеленим кольором (рис. 19.5).

Більше

Форматувати клітинки, які БІЛЬШІ ЗА:

5 з Біло-червона заливка з темно-червоним текстом

OK Скасувати

Менше

Форматувати клітинки, які МЕНШІ ЗА:

7 з Зелена заливка з темно-зеленим текстом

OK Скасувати

Рис. 19.5

## Питання для самоперевірки



1. У чому полягає сортування даних?
2. Як відбувається сортування за даними кількох стовпців?
3. У чому полягає фільтрування даних?
4. Що відбувається під час фільтрування даних таблиці за кількома стовпцями?
5. Що таке умовне форматування?
6. Що відбувається, коли правила умовного форматування даних суперечать одне одному?

## Вправа 19



► Здійснити умовне форматування та впорядкування даних.

- 1) Запустіть програму Excel. Заповніть таблицю за зразком (рис. 19.6), об'єднавши потрібні клітинки та встановивши потовщені червоні зовнішні межі.
- 2) Налаштуйте умовне форматування клітинок для 1-го, 2-го і 3-го місць таким чином:

- 1-ше місце — фон *червоного* відтінку...
- 2-ге місце — фон *зеленого* відтінку...
- 3-тє місце — фон *жовтого* відтінку...

Колір шрифту — за бажанням.

- 3) Налаштуйте умовне форматування непризових місць:
  - шрифт — жирний,
  - колір — синій, без заливки клітинок.
- 4) Упорядкуйте рядки таблиці так, щоб навчальні предмети розташовувались в алфавітному порядку.
- 5) Додайте до таблиці фільтри і відобразіть тільки ті предмети, за якими учні школи у 2020 році посіли перше місце.
- 6) Збережіть файл з іменем Вправа 19 у папці за вказівкою вчителя. Заверште роботу за комп'ютером.

	A	B	C	D	E	
1	Олімпіадні досягнення школи					
2		Рік	2018	2019	2020	2021
3	Математика		2	4	2	3
4	Фізика		4	4	3	3
5	Хімія		3	2	1	2
6	Біологія		4	2	7	5
7	Інформатика		3	1	1	3

Рис. 19.6



## Комп'ютерне тестування



Виконайте тестове завдання 19 з автоматичною перевіркою результату.





## Практична робота 7

### Використання математичних, логічних та статистичних функцій табличного процесора. Умове форматування

**Завдання:** створити таблицю успішності учнів, взявши за зразок шкільний журнал, з автоматичним обчисленням підсумкових оцінок та рівня навчальних досягнень учнів і учениць.

**Обладнання:** комп'ютер зі встановленим табличним процесором Excel.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

- Запустіть програму Excel. Встановіть параметри сторінки:
  - всі поля — по 1 см;
  - колонтитули — 0 см;
  - орієнтація — книжкова;
  - розмір паперу — А4.
- Підготуйте порожню таблицю за зразком (див. рисунок). Об'єднайте потрібні клітинки, змініть розміри відповідних стовпців та рядків. Установіть межі середньої товщини для таблиці з 16 рядків і 13 стовпців.
- У відповідних клітинках встановіть потрібну орієнтацію тексту та формати даних:
  - для оцінок — числовий без десяткових знаків;
  - для написів — текстовий;
  - для дат — короткий формат дати.
- Впишіть прізвища учнів і учениць і їх оцінки, інші написи.
- Відсортуйте в алфавітному порядку прізвища учнів і учениць.
- У режимі автозаповнення введіть нумерацію і дати.
- Обчисліть відповідною функцією підсумкову оцінку кожного учня як округлене середнє арифметичне його оцінок.
- Функцією COUNTIF підрахуйте кількість відсутніх для кожної дати.
- Функцією COUNTIF за підсумковими оцінками учнів і учениць підрахуйте кількість тих, хто має початковий (менше за 4 бали) та високий (більше ніж 9 балів) рівні.

10. Функцією COUNTIFS за підсумковими оцінками учнів і учениць підрахуйте кількість тих, хто має достатній (від 7 до 9 балів) та середній (від 4 до 6 балів) рівні.
11. Проведіть умовне форматування клітинок з оцінками залежно від рівня досягнень:
  - високий — червоний колір тексту;
  - середній — зелений;
  - достатній — синій;
  - початковий — чорний (усюди фон клітинок — білий).
12. Збережіть файл з іменем Практична робота 9 у папці за вказівкою вчителя. Закрийте програму Excel. Заверште роботу за комп'ютером.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	№ за порядком	Прізвище	Дата										Підсумкова
2			09.02.2021	16.02.2021	23.02.2021	02.03.2021	09.03.2021	16.03.2021	23.03.2021	30.03.2021	06.04.2021	13.04.2021	
3	1	Іванова	6		8	4	4	5	н	н		5	
4	2	Петренко	12	10		12	9		н		н		
5	3	Сидоров	10		9		н	8					
6	4	Тесленко	11		10	9		8		8			
7	5	Айсман	5		7	8		6					
8	6	Бойко					12	12		12		н	
9	7	Грицюк		9	8		9	10	11		н		
10	8	Буренко	н		12				12	9			
11	9	Вівчар		н			8	8	н	8	8		
12		Кількість відсутніх											
13		Початковий рівень											
14		Достатній рівень											
15		Середній рівень											
16		Високий рівень											

Зразок

**Зробіть висновок:** які типи функцій використовують для розрахунків; яка з використовуваних функцій має найбільше аргументів.

## § 20. Розширені фільтри. Проміжні підсумки

Автоматичні фільтри придатні не для всіх випадків.

### Розширені фільтри

Розглянемо приклад.

- 1 За потреби відобразити успішність учнів і учениць, які мають більше від 9 балів з алгебри або інформатики, застосування фільтрів з умовою ( $>9$ ) до стовпців Алгебра, Інформатика приведе до відображення лише рядка балів Петренка, у той час як вказаним критеріям відповідають рядки: Іванова, Петренко, Сидоров (рис. 20.1).

	A	B	C	D	E	F
1	Прізвище	Алгебра	Хімія	Фізика	Інформатика	Середній бал
2	Іванова	7	10	11	12	10
3	Петренко	12	11	8	10	10,3
4	Сидоров	11	9	8	9	9,3
5	Тесленко	9	8	9	7	8,3

Рис. 20.1

У випадку застосування складних логічних виразів (складних умов) використовують розширені фільтри.

Для встановлення розширених фільтрів необхідно:

- 1) скопіювати в окреме місце заголовки стовпців, за якими відбудеться фільтрування (наприклад, Алгебра, Інформатика);
- 2) у клітинки під скопійованими заголовками увести критерії фільтрування (наприклад,  $>9$ );
- 3) вибрати Дані  $\rightarrow$  Сортування й фільтрування  $\rightarrow$  Додатково;
- 4) встановити у вікні Розширений фільтр необхідні прапорці і вказати діапазони (рис. 20.2).

При цьому критерії, об'єднані логічною операцією OR, вказуються в різних рядках, а об'єднані логічною операцією AND — в одному.

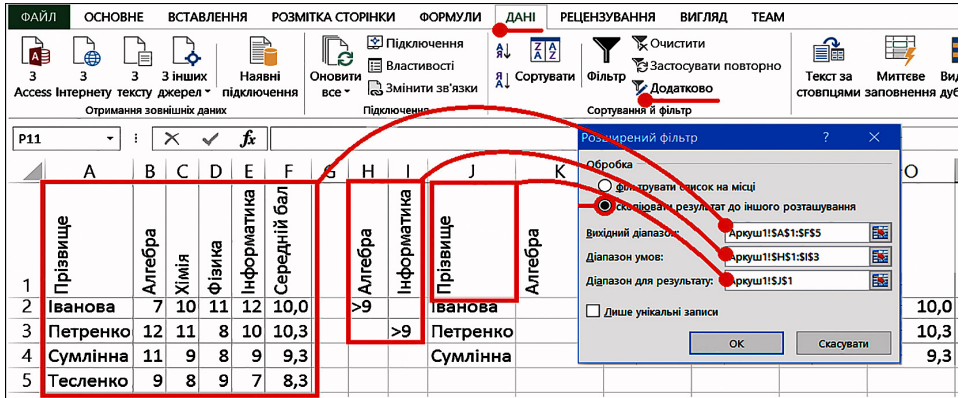


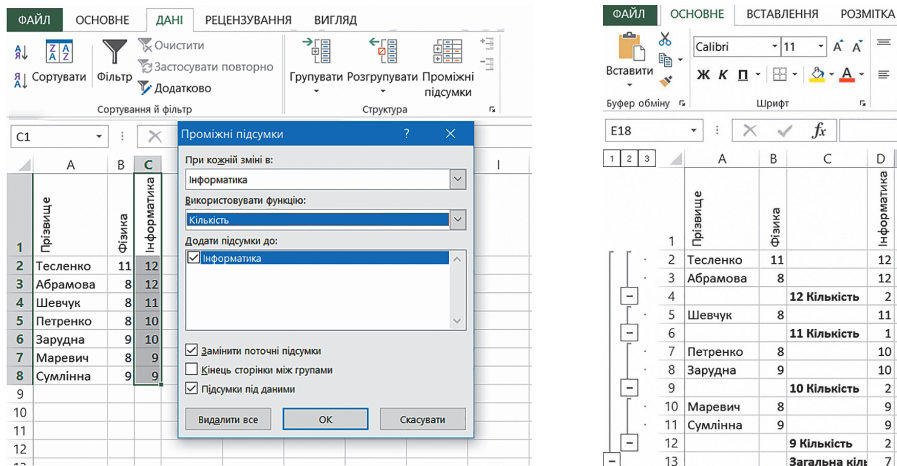
Рис. 20.2

Для дублювання достатньо у відповідному діапазоні вказати адресу верхнього лівого кута майбутньої відфільтрованої таблиці.

## Проміжні підсумки

У таблицях Excel зручно також аналізувати повторювані дані.

- 2 Як підрахувати, скільки учнів і учениць навчаються з інформатики на 10, 11, 12 балів? Одним зі способів є застосування проміжних підсумків, коли підрахунок проводиться за окремими даними стовпця (рис. 20.3).



а

Рис. 20.3

б

Для налаштування проміжних підсумків можна виділити клітинки та встановити критерії у вікні Проміжні підсумки на вкладці: Дані → Структура → Проміжні підсумки. Попередньо таблицю відсортовують за стовпцем, дані якого групуватимуться; кнопками ліворуч дані можна приховати.

### Питання для самоперевірки



1. Коли застосувати автоматичний фільтр не можна?
2. У якому випадку доцільно застосовувати розширені фільтри?
3. Як встановити розширені фільтри?
4. Як у розширених фільтрах записуються умови, об'єднані AND?
5. Коли доцільно використовувати проміжні підсумки?

### Вправа 20



▶▶ Створити таблиці із застосування розширеного фільтра за проміжних підсумків за описом.

- 1) Запустіть Excel. Заповніть таблицю за зразком (рис. 20.4).
- 2) Відформатуйте написи і межі таблиці.



- 3) Застосуванням розширеного фільтра створіть нову таблицю, яка міститиме дані про стаціонарні комп'ютери або такі, обсяг продажів яких перевищив 10 000.

	A	B	C
1	Тип ПК	Виробник	Обсяг продажів
2	Ноутбук	Asus	7000
3	Стаціонарний	Sony	5000
4	Ноутбук	Lenovo	30 000
5	Стаціонарний	Lenovo	100 000
6	Стаціонарний	Acer	7500
7	Планшет	Acer	10 000
8	Ноутбук	Acer	150 000
9	Ноутбук	Apple	15 000

Рис. 20.4

- 4) Для початкової таблиці за допомогою проміжних підсумків обчисліть сумарні обсяги продажів комп'ютерів кожного типу.
- 5) Приховайте дані початкової таблиці, залишивши проміжні підсумки продажів комп'ютерів за типами.
- 6) Збережіть файл з іменем Вправа 20 у відповідній папці. Заверште роботу за комп'ютером.

### Комп'ютерне тестування



Виконайте тестове завдання 20 з автоматичною перевіркою результату.





## § 21. Діаграми і графіки

В Excel включені спеціальні засоби, названі діловою графікою, які дозволяють зобразити табличні дані у графічному вигляді.

### Діаграми

Існує багато типів стандартних діаграм, вибір яких залежить від того, що діаграма відобразить (рис. 21.1).

**Кругові діаграми** (рис. 21.2) наочно показують співвідношення між частинами в цілому. Для побудови кругової діаграми використовують числа, розміщені в *одному рядку* або *стовпчику*.

**Стовпчасті діаграми** (рис. 21.3) краще ілюструють числові дані, розміщені *одночасно в кількох рядках і стовпцях*.

**Лінійний графік** (рис. 21.4) краще використовувати для зображення змін показників *протягом певного часу*.

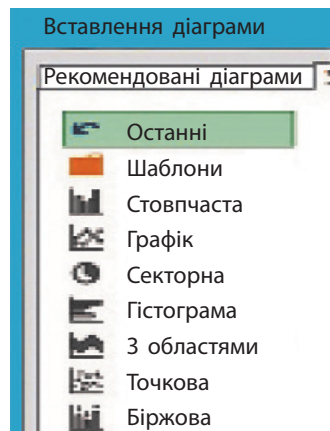


Рис. 21.1

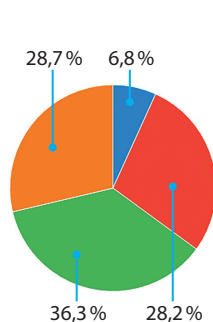


Рис. 21.2

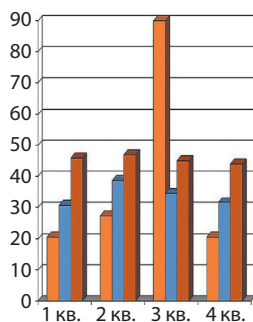


Рис. 21.3

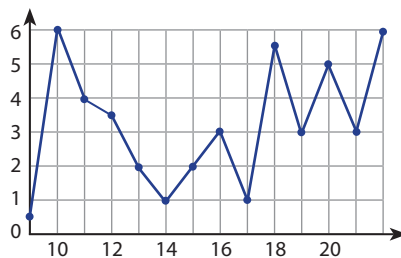


Рис. 21.4

**Точкову діаграму** використовують для побудови графіків математичних функцій. Для її побудови потрібно не менш ніж два ряди даних (один ряд — аргумент, а інший — значення функції).

Після побудови діаграми зміна даних таблиці приводить до зміни відповідних об'єктів діаграми.

## Побудова діаграми

Пригадаємо, що діаграма містить такі об'єкти (рис. 21.5): 1 — назва діаграми; 2 — область діаграми; 3 — фоновий прямокутник; 4 — легенда; 5 — маркер даних; 6 — назва осі X.

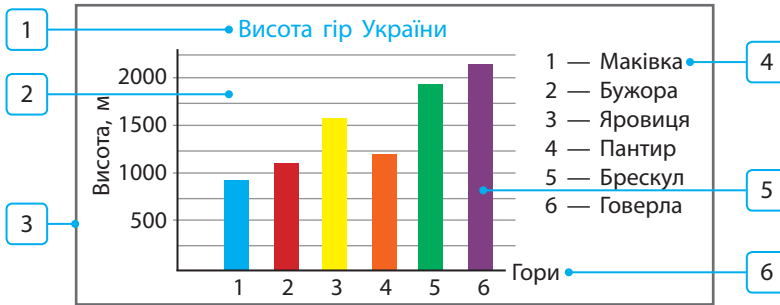





Рис. 21.5

Для створення діаграми потрібно:

- 1) заповнити (створити) таблицю даних;
- 2) виділити табличні дані разом із назвами рядків (стовпців);
- 3) вставити діаграму потрібного типу.

За допомогою стрічки інструментів або кнопок швидкого налаштування можна змінити тип діаграми, зовнішній вигляд тощо.

Зображення	Призначення кнопки швидкого налаштування
 Елементи	Редагування назви, легенди, підписів даних
 Стилi	Налаштування стилю та колірної гами
 Фiльтри	Налаштування відображення даних та назв

Щоб перемістити діаграму на інше місце аркуша, слід перетягнути фоновий прямокутник із натиснутою лівою кнопкою миші.

Щоб розмістити діаграму на окремому робочому аркуші, слід виконати команду, перемістивши діаграму, що є у контекстному меню.

Щоб змінити розміри діаграми або її складових, треба виділити (клацнути) потрібний об'єкт і перетягти відповідні маркери зміни розміру. Сектори кругової діаграми, наприклад, можна роз'єднати, перетягуючи їх мишею. За допомогою контекстного меню можна змінити колір заливки секторів, їхні контур, розмір, рельєф тощо.

## Графіки

На уроках математики ви вивчали функції та їх графіки. Розглянемо хід побудови графіка функції на прикладі.

Побудуємо в Excel на координатній площині графік функції  $y = \frac{2}{x}$

на проміжку [1; 19] з кроком  $h=1,5$ .

1. Запустимо Excel і створимо таблицю. Впишемо в клітинках A1, B1, C1 назви величин  $h$  (крок),  $x$ ,  $y$  (координати). У клітинку A2 введемо крок зміни координати  $x$  (1,5).
2. У клітинку B2 введемо перше значення (із діапазону) координати  $x=1$ .
3. У клітинку B3 введемо формулу, яка додаватиме до поточної координати  $x$  одне і те саме значення кроку  $=B2+\$A\$2$ .
4. Скопіюємо формулу клітинки B3 у решту клітинок для координати  $x$ , доки не отримаємо кінцеве значення діапазону 19.
5. У клітинку C2 введемо формулу розрахунку координати  $y=2/C2$ .
6. Скопіюємо формулу до решти клітинок для координати  $y$ .
7. Установимо для клітинок зі значеннями функції числовий формат із двома десятковими знаками.
8. Маємо таблицю аргументів ( $x$ ) та значень ( $y$ ) функції  $y = \frac{2}{x}$  на проміжку [1; 19] з кроком  $h=1,5$ . Виділимо потрібні дані (разом із назвами  $x$  і  $y$ ), виберемо Вставка → Діаграми та значимо тип графіка (рис. 21.6).

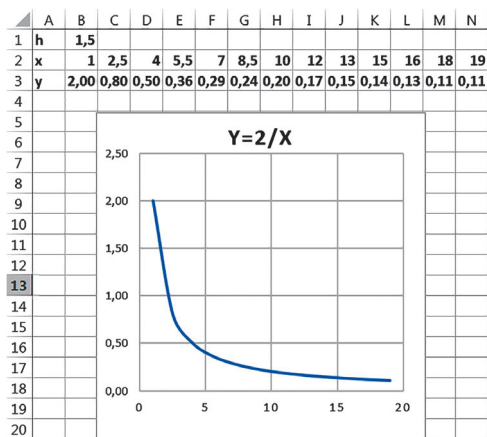


Рис. 21.6

## Питання для самоперевірки



1. Які типи діаграм вам відомі?
2. Коли доцільно застосовувати кругову діаграму?
3. Коли доцільно застосовувати стовпчасту діаграму?
4. Які основні кроки побудови діаграм?
5. Назвіть основні складові зображення діаграми.
6. Як змінити розміри діаграми або її складових?

## Вправа 21



►► Побудувати графік функції  $y = \frac{a}{x}$ .

- 1) Запустіть програму Excel.
- 2) Побудуйте графік заданої функції.



Установіть крок зміни та діапазон координати  $x$ .

Відведіть окрему клітинку для змінної  $a$  та встановіть довільне значення.

- 3) Збережіть файл з іменем Вправа 21 у папці за вказівкою вчителя. Заверште роботу за комп'ютером.

## Комп'ютерне тестування



Виконайте тестове завдання 21 з автоматичною перевіркою результату.



## Практична робота 8

### Упорядкування даних у таблицях.

### Автоматичні та розширені фільтри

**Завдання:** створити таблицю за зразком, упорядкувати дані за різними умовами і правилами.

**Обладнання:** комп'ютер зі встановленим табличним процесором Excel.

## Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму Excel. Створіть таблицю за зразком.

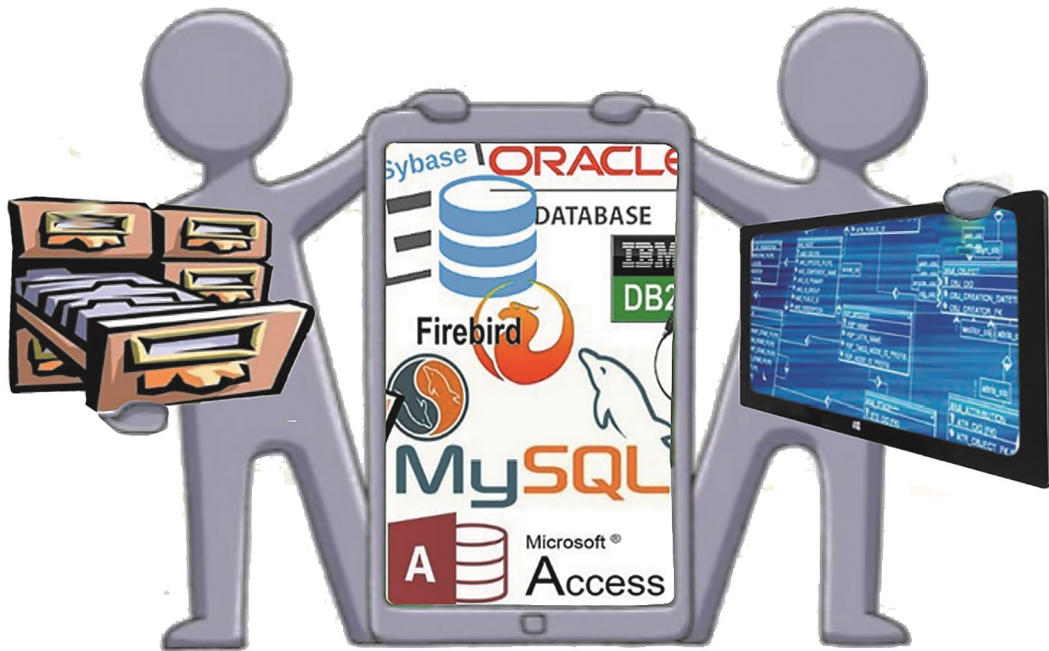
Ціна, \$	Марка автомобіля	Кількість
12 000	МАЗ	100
15 000	Форд	125
11 000	Форд	150
11 000	Мерседес	200
10 000	Хонда	225
12 500	Мерседес	250

Скопіюйте аркуш зі створеною таблицею для завдань 2, 3, 4–5, 6, 7.

2. Відсортуйте дані в таблиці за зростанням ціни.
  3. Відсортуйте рядки таблиці, щоб марки автомобілів були розташовані в алфавітному порядку, а кількість — у порядку спадання.
  4. Увімкніть автоматичні фільтри для всіх стовпців.
  5. Налаштуйте фільтр відповідного стовпця на відображення рядків тільки марок автомобілів «Форд» і «Мерседес».
  6. Налаштуйте розширений фільтр відображення даних для автомобілів, ціна яких менша за 12 300 або кількість яких не перевищує 150.
  7. Встановіть умовне форматування даних так, щоб напис «Мерседес» відображався червоним кольором, а «Форд» — зеленим.
  8. Збережіть файл з іменем Практична робота 10 у папці за вказівкою вчителя.
- Заверште роботу за комп'ютером.

**Зробіть висновок:** як можна впорядкувати дані в таблиці; які фільтри можна застосувати; для чого використовують умовне форматування.

## Розділ 4. СИСТЕМИ КЕРУВАННЯ БАЗАМИ ДАНИХ



§ 22. Бази даних і системи керування ними

§ 23. Реляційна база даних

§ 24. Заповнення та зв'язки таблиць

§ 25. Впорядкування таблиць

§ 26. Запити

Практична робота 9. Робота з таблицями і запитам

§ 27. Форми

§ 28. Звіти

## ПОВТОРЮЄМО

Ви вмієте будувати таблиці в середовищі текстового процесора **Microsoft Office Word**, знаєте, як працювати з таблицями в середовищі табличного процесора **Microsoft Office Excel**: виділяти окремі частини таблиці, змінювати розміри стовпців, заповнювати та редагувати таблицю, форматувати клітинки, виправляти помилки тощо.

Розглядаючи частину таблиці як базу даних, ви мали можливість упорядковувати в таблиці рядки за значеннями одного або декількох стовпців, тимчасово приховувати непотрібні рядки за допомогою *автофільтра*. Ви знаєте, як за допомогою *розширеного фільтра* задати кілька умов і вибрати з бази даних саме ті дані, що відповідають заданим умовам; як виконати обчислення за даними таблиці. Ви також отримали деяке уявлення про базу даних.

1. Як упорядкувати частину електронної таблиці?
2. Опишіть порядок використання автофільтра.
3. Як користуватися розширеним фільтром?
4. Як виправити помилку в клітинці електронної таблиці?
5. Поясніть, як переміщуватися клітинками електронної таблиці.

У цьому розділі ви дізнаєтесь, що таке база даних і система керування базою даних (СКБД), які особливості мають реляційні бази даних, які об'єкти містить база даних у СКБД **Microsoft Access**; навчитесь створювати просту базу даних.

## § 22. БАЗИ ДАНИХ І СИСТЕМИ КЕРУВАННЯ НИМИ

Ще недавно вважалось, що обсяг даних, який опрацьовує людство, подвоюється кожні 10 років. За оцінками сучасних фахівців, 90 % даних, які існували, наприклад, у 2017 році, було створено за попередні 2 роки. У цьому параграфі ви дізнаєтеся як можна упорядковувати великі обсяги інформації.

### База даних

Кілька тонн книжкової макулатури містять багато інформації, якою практично неможливо скористатися. Навіть якщо ви впевнені, що в стосі книжок є потрібна, то знайти і скористатися нею надзвичайно важко. Інша річ, коли книжки упорядковані за жанрами на полицях бібліотеки, тоді таку сукупність книжок можна назвати базою даних.



**База даних** — це упорядкована сукупність взаємопов'язаних даних.

Наприклад, базами даних (БД) також можна назвати шкільні журнали обліку успішності, архіви, описи майна і матеріалів, бухгалтерські документи, особові справи учнів, реєстраційні дані власників авто тощо.

**Електронна БД** являє собою файл або сукупність файлів спеціального формату, які містять належним чином структуровані дані, призначені для зберігання, накопичення, опрацювання та використання за допомогою комп'ютера.

Залежно від структури упорядкування даних розрізняють такі моделі БД (рис. 22.1):

- ієрархічну (а);
- мережеву (б);
- реляційну (в);
- об'єктно-орієнтовану (г);
- гібридну та інші (гіпертекстові, багатовимірні тощо).

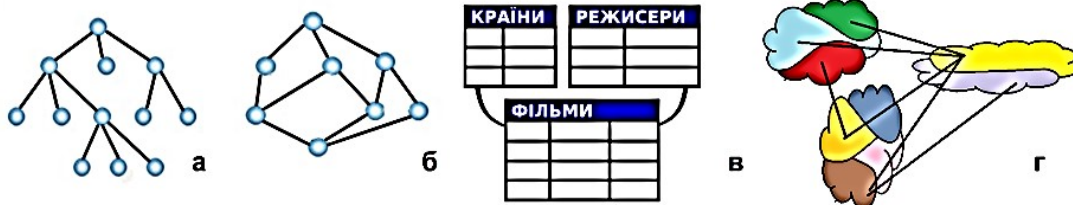


Рис. 22.1

### Ієрархічна база даних

До основних понять ієрархічної структури БД належать вузол, рівень, зв'язок (рис. 22.2). Ознайомимося з ними докладніше.



**Вузол** — це інформаційна модель, що містить певні дані.

**Зв'язок** — це взаємозалежність двох вузлів.

**Рівень** визначає ступінь підпорядкування вузлів БД.

В ієрархічній БД кожен вузол на нижчому рівні має зв'язок тільки з одним вузлом вищого рівня (рис. 22.2).

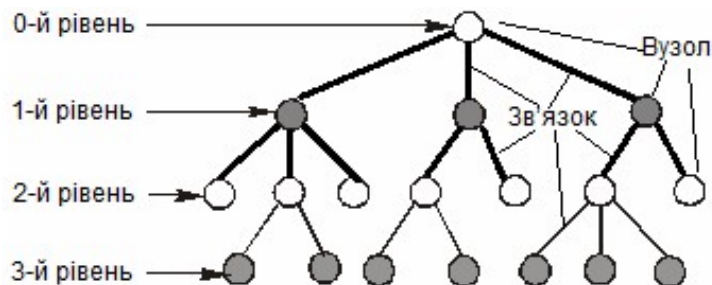


Рис. 22.2

### ПРИКЛАД 1.

За ієрархічною моделлю функціонують файлова система комп'ютера (рис. 22.3), підрозділи збройних сил, побудовано адміністративно-територіальний устрій більшості країн, створюється генеалогічне (родинне) дерево тощо.

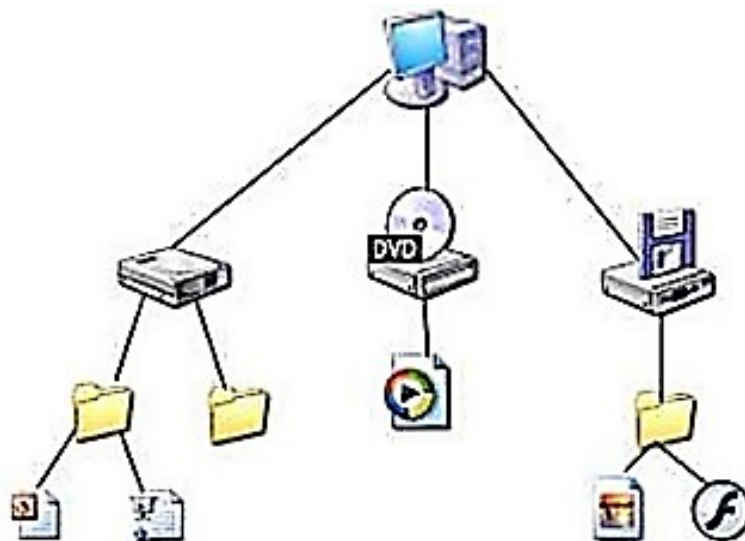


Рис. 22.3

### Мережева база даних

**Мережева модель бази даних** — це сукупність об'єктів різного рівня, де кожен об'єкт може бути зв'язаний з іншими.

Складовими мережевої моделі БД також є рівень, вузол і зв'язок. Проте в ній можливий довільний зв'язок між елементами сусідніх рівнів, тобто елемент нижчого рівня може мати зв'язок з декількома елементами свого або вищого рівня.

## ПРИКЛАД 2.

Дані про клієнтів банків можуть зберігатися у БД різних банків і бути пов'язаними між собою (рис. 22.4).

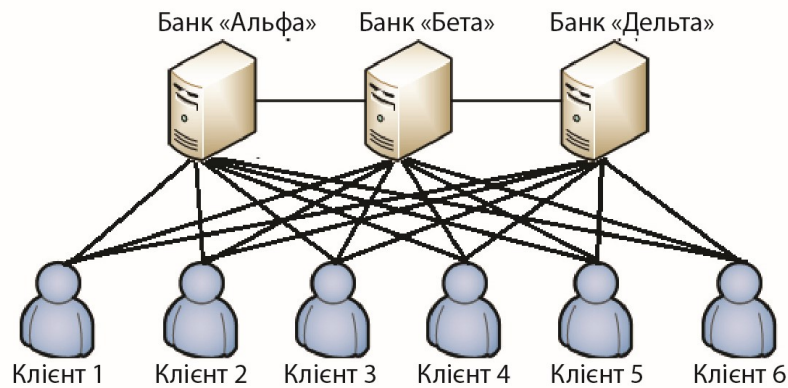


Рис. 22.4

### Реляційна база даних

Реляційна БД являє собою сукупність взаємопов'язаних таблиць.

Пошук даних у реляційній БД зводиться до аналізу зв'язків між таблицями та пошуку перетину рядків і стовпців у відповідних таблицях.

## ПРИКЛАД 3.

Школярам добре відома реляційна БД у вигляді таблиць успішності класного журналу (рис. 22.5).

Інформатика 1 група Облік навчальних

(Назва предмета)

№ п/п	Прізвище та ім'я учня (учениці)	Місяць і число									
		05/09	12/09	19/09	26/09	03/10	10/10	17/10	24/10	31/10	07/11
1	Індошченко Данил			8			10	9			
2	Беласенко Ярослав	н		8			11	10			
3	Талубов Іван			8			11	10			

Рис. 22.5

Ви з 7-го класу знайомі з реляційними БД, створюваними в табличному процесорі Excel, наприклад, у вигляді тих же таблиць успішності (рис. 22.6).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Дата										
2	№ з/п	Прізвище	9.2	16.2	23.2	2.3	9.3	16.3	23.3	30.3	6.4	13.4	Підсумкова
3	1	Іванова	6		8	4	4	5	4	4		5	
4	2	Петренко	12	10		12	9		4		4		
5	3	Сидоров	10		9		4	8					

Рис. 22.6

### Об'єктно-орієнтовані та гібридні бази даних

В **об'єктно-орієнтованих** БД дані зберігаються у вигляді абстрактних об'єктів, наділених певними властивостями і можливостями (методами) взаємодії з іншими об'єктами. У таких БД зручно встановлювати складні взаємозв'язки між об'єктами (рис. 22.7).

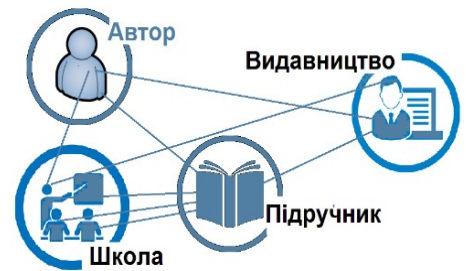


Рис. 22.7

**Гібридні БД** поєднують можливості реляційних і об'єктних моделей, тому їх часто називають *об'єктно-реляційними*.

### Модель «сутність — зв'язок»



**Модель «сутність — зв'язок»** — це схематичне відображення змістового (семантичного) зв'язку об'єктів БД.



**Сутність** — це іменованій інформаційний об'єкт.

### **ПРИКЛАД 3.**

Приклади сутностей: **клас, учень, оцінка, тема, підручник, учитель, співробітник, відділ, підприємство** тощо.

Сутність у моделі зазвичай позначається прямокутником, у якому крім назви сутності можуть бути записані її характеристики — **атрибути** (рис. 22.8). Іноді атрибути поміщають в овали (рис. 22.9).

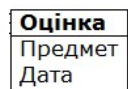


Рис. 22.8



**Зв'язок** — це залежність між двома сутностями, охарактеризована дієсловом.

Зазвичай зв'язки позначаються підписаними лініями або ромбами (рис. 22.9).

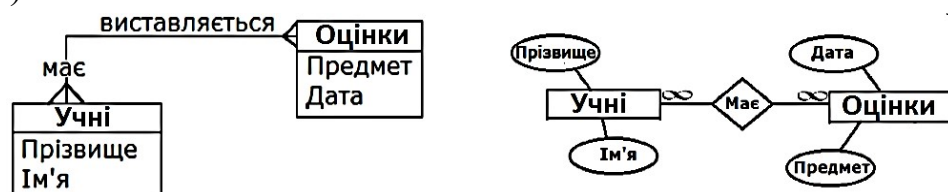


Рис. 22.9

Існує три типи зв'язків між сутностями. Розглянемо таблицю.

Тип зв'язку	Приклад	Позначення
Один-до-одного	<i>Кожен учень має один щоденник</i>	—————
Один-до-багатьох (багато-до-одного)	<i>Один учитель навчає багатьох учнів</i>	—————<
Багато-до-багатьох	<i>Багато учнів вивчають по кілька предметів</i>	>—————<

#### ПРИКЛАД 5.

Якщо сутностями БД класного журналу вважати **Учні**, **Оцінки**, то модель «сутність — зв'язок» такої БД може бути проілюстрована рис. 22.9.

### Системи керування базами даних

База даних є лише сховищем упорядкованої сукупності даних.



**Системи керування базами даних** — це комплекс засобів і

методів для створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до них.

#### Приклад 6.

Книжковою базою даних є бібліотека, а системою керування такої БД можна вважати бібліотечний каталог (рис. 22.10), який дозволяє відстежувати обіг книжок, проводити пошук за назвами чи прізвищами авторів та/або за кодами УДК (універсальний десятковий код) чи ББК (бібліотечно-бібліографічна класифікація), позначеними на перших сторінках книжки.



Рис. 22.10



**Електронна система керування базою даних** — це комплекс

програмних засобів для забезпечення пошуку та опрацювання даних у БД.

Системи керування базами даних (СКБД) дозволяють ефективно працювати з БД, в яких обсяг і/або представлення даних робить неможливим їх ручне опрацювання.

#### **ПРИКЛАД 7.**

Операційні системи (ОС) комп'ютерів виконують функції СКБД, дозволяючи створювати, змінювати, відшукувати та проводити інші операції з об'єктами ОС (файлами, папками, налаштуваннями тощо).

#### **ПРИКЛАД 8.**

Прикладами електронних СКБД є **Microsoft Access, Microsoft SQL Server, MySQL, PostgreSQL, Oracle, Sybase, Interbase, Firebird, IBM DB2** та інші.

**ЦІКАВИНКА.** Історія розвитку СКБД налічує майже 50 років. У 1968 році у США була введена в експлуатацію перша промислова СКБД **IMS** фірми **IBM**.

У розвиток теорії баз даних значний внесок було зроблено американським математиком Едгаром Франком Коддом (рис. 22.11), який є творцем реляційної моделі баз даних. У 1981 році Е. Ф. Кодд одержав за створення реляційної моделі і реляційної алгебри престижну премію Тюрінга американської асоціації з обчислювальної техніки.

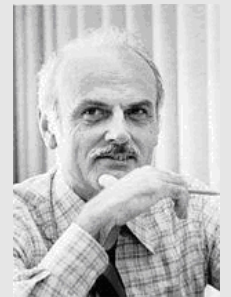


Рис. 22.11

#### **Питання для самоконтролю**

1. Чи містить гора книжкової макулатури інформацію (дані) і чи можна назвати її базою даних? Чому?
2. Наведіть приклади паперових баз даних.
3. Наведіть приклади ієрархічних зв'язків.
4. У чому схожість і відмінність ієрархічних і мережевих баз даних?
5. Що являє собою реляційна база даних?
6. Яке призначення СКБД? Наведіть приклади електронних СКБД.

#### **Вправа 22**

Опрацювати моделі та об'єкти баз даних.

1) Створіть на хмарному диску (наприклад, в обліковому записі **Google**) документ, у якому зберігатимете розв'язки завдань.

Переіменуйте документ своїм прізвищем, додавши до назви номер вправи: (наприклад: **Сумлінна Ліна Вправа 22**).

2) Надайте доступ учителю для редагування створеного вами документа.

3) Запишіть у документ назву та автора (авторів) друкованного видання (книжки), наданого вам вчителем.

Знайдіть у книжці і запишіть у документ коди УДК і /або ББК.

4) Знайдіть в інтернеті таблиці УДК і/або ББК, завантажте та/або відкрийте їх. Визначте за знайденими таблицями і запишіть у документ класифікаційну назву, що відповідає кодам УДК і/або ББК наданої вам книжки.

5) Запишіть (зобразіть) у документі довільний приклад зв'язку сутностей БД **один-до-одного** (окрім наведених у підручнику).

Запишіть (зобразіть) у документі довільний приклад зв'язку сутностей БД **один-до-багатьох** (окрім наведених у підручнику).

Запишіть (зобразіть) у документі довільний приклад зв'язку сутностей БД **багато-до-багатьох** (окрім наведених у підручнику).

6) У довільному графічному редакторі зобразіть ієрархічну модель структури (рис. 22.12) вашої школи до рівня класів.

<p>Школа умовно ділиться на Молодшу, Середню і Старшу; Молодша школа — це паралелі 1–4 класів, Середня — 5–9 класи, Старша — 10–11 класи; Паралель перших класів — це 1-А, 1-Б і т. д. Паралель других класів — 2-А, 2-Б, і т. д. І т. д.</p>
---

Рис. 22.12

Підпишіть рівні та вузли схеми, зображеної у пункті 11. Скопіюйте створений рисунок у документ на хмарному диску.

### **Комп'ютерне тестування**

Виконайте тестове завдання 22 з автоматичною перевіркою результату.

## **§ 23. РЕЛЯЦІЙНА БАЗА ДАНИХ**

Роботу з БД можна поділити на *професійну* (проєктування, конструювання тощо) і *користувацьку* (заповнення, упорядкування, пошук і використання даних тощо).

Далі ви ознайомитеся з особливостями реляційної БД і навчитесь проєктувати і створювати таблиці електронної БД **Access**.

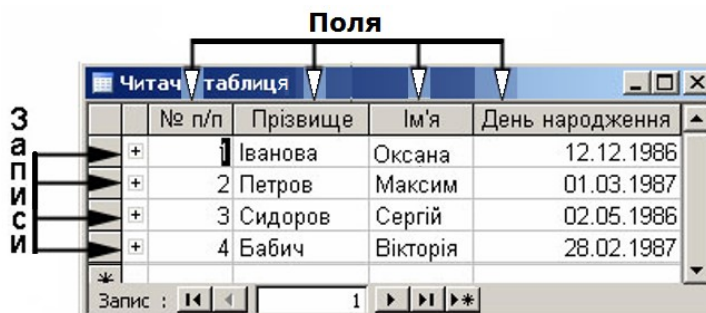


## Структура реляційної бази даних

**Реляційна БД** являє собою сукупність зв'язаних таблиць, що містять інформацію про об'єкти певного виду.

Рядок таблиці містить дані про один об'єкт (наприклад, про людину, автомобіль тощо), а стовпці таблиці містять різні характеристики цих об'єктів — атрибути (наприклад, прізвище, ім'я, дату народження, номер двигуна тощо).

У таблицях реляційної БД рядки називають **записами**, а стовпці — **полями** (рис. 23.1).



	№ п/п	Прізвище	Ім'я	День народження
+	1	Іванова	Оксана	12.12.1986
+	2	Петров	Максим	01.03.1987
+	3	Сидоров	Сергій	02.05.1986
+	4	Бабич	Вікторія	28.02.1987

Рис. 23.1



*Поле* реляційної БД (стовпець) містить дані одного типу.  
*Записи* (стовпці) містять сукупну інформацію про певний об'єкт.

Навіть порожня структурована БД містить певну інформацію про імена полів, опис типів даних тощо.

*Структура* реляційної БД визначає зв'язки між таблицями і можливі типи та властивості даних у них.

## Проектування реляційної бази даних

**Проектування БД** починається з визначення її структури за такими кроками:

- 1) визначається перелік даних, які будуть зберігатися;
- 2) визначається кількість і вміст таблиць для зберігання даних;
- 3) визначаються для кожної з таблиць назви полів, їх тип, властивість тощо.

### **ПРИКЛАД 1.**

Прикладом не зовсім раціональні «бази даних» може бути класний журнал, у якому на кожній сторінці повторюються прізвище та ім'я учня. Уникнути повторювань можна, «зв'язавши» таблиці з оцінками із таблицею з прізвищами. Для цього частину сторінок можна обрізати так, щоби було видно список учнів на першій сторінці (рис. 23.2). Але зазвичай цього не

роблять, оскільки клас може ділитись на групи, учні можуть вибувати або прибувати з інших шкіл тощо.

Петренко І	8	11
Іванова К		10
Петрова М	9	
Сидоров О		12
Богдан П	7	

Рис. 23.2

Повторення даних у таблицях *електронної* реляційної БД знижує її надійність у цілому: помилившись під час уведення, наприклад, прізвища в одну з таблиць, під час пошуку, ми ризикуємо отримати неповну інформацію про людину. Проблем можна уникнути, застосовуючи кілька зв'язаних таблиць замість однієї великої.

### СКБД Microsoft Access

Однією із електронних систем керування реляційними базами даних (СКБД) є **Microsoft Office Access** (далі **Access**). Файл **Access** (розширення **.accdb**) містить таблиці, запити, форми, звіти, зв'язки, фільтри тощо.



**Таблиці бази даних** — це основний елемент СКБД, який

містить розподілені за певними ознаками дані.

На прикладі шкільного журналу спроектуємо спрощену електронну БД з назвою «Інформатика», в якій відслідковуватимемо успішність групи учнів з інформатики (рис 23.3, 23.4).

Таблиця <b>Учень</b>	
Ім'я поля	Тип даних
Номер учня	Число
Прізвище	Текст
Ім'я	Текст
Дата народження	Дата
Вивчає інформатику із 2-го кл	Так/Ні

Рис. 23.3

Таблиця <b>Успішність</b>	
Ім'я поля	Тип даних
Порядковий номер	Число
18 січня	Число
20 січня	Число
25 січня	Число
Тематична	Обчислення

Рис. 23.4




Зрозуміло, що потрібно буде встановити зв'язок таблиць, оскільки в таблиці **Успішність** відсутні прізвища учнів. Це буде зв'язок порядкових номерів відповідних записів (**один-до-одного**).

### Створення порожньої бази даних

Електронну БД Access можна створити на основі шаблону або вручну на основі порожніх таблиць. Розглянемо алгоритм створення порожньої електронної реляційної БД Access.

Для створення порожньої електронної реляційної БД Access потрібно:

- 1) запустити СКБД Access; 
- 2) вибрати **Пуста настільна база даних** (рис. 23.5);
- 3) у вікні, що з'явиться (рис. 23.6), ввести назву створюваної БД (у нас: **Інформатика**);
- 4) за необхідності кнопкою **Знайти розташування для бази даних** змінити розташування файлу БД;
- 5) клацнути значок **Створити**.

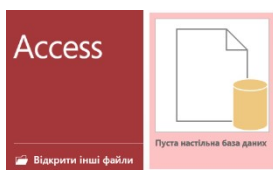


Рис. 23.5

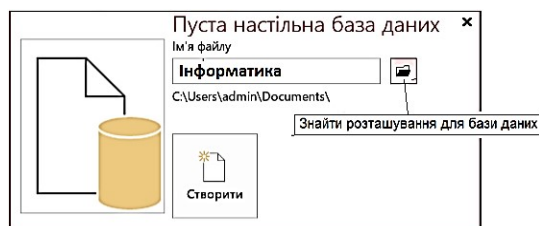


Рис. 23.6

У створеній БД автоматично буде збережено порожню таблицю **Таблиця 1**, яка міститиме єдине поле з ім'ям **Ідентифікатор** (рис. 23.7).



Рис. 23.7

Пригадаємо основні елементи інтерфейсу офісних додатків на прикладі додатку **Access** (див. рис. 23.7). Це допоможе вам у подальшому розумінні термінології підручника.

### Створення таблиць

Для створення нової таблиці на вкладці **СТВОРЕННЯ** у групі **Таблиці** обирають значок **Таблиця** або **Конструктор таблиць**.

Кожна таблиця **Access** може бути відображена у двох режимах: **режим конструктора** (кнопка або команда **Конструктор**) і **режим перегляду чи заповнення** (подвійне клацання на назві таблиці або команда контекстного меню **Відкрити**).

**!** Зміну режиму відображення таблиці зручно робити кнопкою **Режим**, що на вкладці **КОНСТРУКТОР** зліва.

### Збереження змін

За допомогою контекстного меню вкладки таблиці (рис. 23.8) можна зберегти зміни, внесені у структуру таблиці.

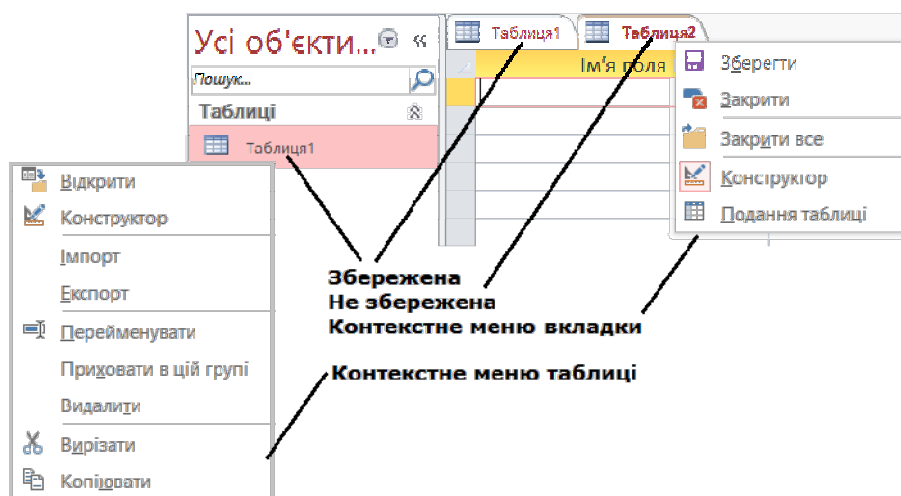


Рис. 23.8

Таблиця створена в режимі **Конструктор таблиць** не містить жодного поля, тому її не можна зберегти, доки не буде створено хоча б одне поле (див. рис. 23.8).

Імена збережених таблиць відображаються зліва в **Області переходів** у полі **Усі об'єкти**.

Таблиці можна перейменувати за допомогою контекстного меню значка таблиці, що в полі **Усі об'єкти**.

**!** Назви таблиць не можуть містити лапки ("").

## Створення полів у таблицях

Наступним кроком конструювання таблиці є створення її полів. У режимі конструктора в рядки вписують імена полів (стовпець **Імя поля**).

! Імена полів **НЕ** можуть:

- бути довгими за 64 символи;
- починатись із пробілу;
- містити крапку (.), знак оклику (!), апостроф ('), квадратні дужки ([]);
- містити керуючі символи (значення ASCII від 0 до 31).

За допомогою контекстного меню поля в режимі конструктора (рис. 23.9) можна додавати поля (рядки), видаляти їх, перейменовувати тощо.

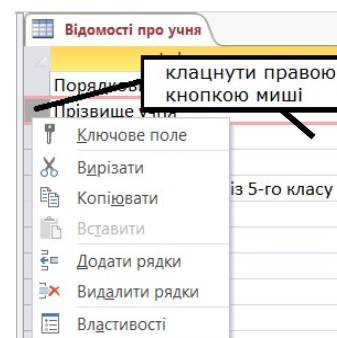


Рис. 23.9

! Контекстне меню поля в режимі конструктора викликається клацанням правої кнопки миші будь-де в рядку поля.

## Типи даних

За замовчуванням у режимі конструктора усім даним надається один і той самий тип: **Короткий текст**. Але це незручно, оскільки може виникати необхідність, наприклад, проведення арифметичних операцій над числовими даними, порівняння віку за датами народження тощо.

У режимі конструктора справа можна вибрати зі списку найоптимальніший тип даних (рис. 23.10). Опис типів даних можна отримати скориставшись довідкою (клавіша **F1** або кнопка ?).

Імя поля	Тип даних
Порядковий номер учня	Число
Прізвище	Короткий текст
Імя	Короткий текст
Дата народження	Дата й час
Вивчає інформатику з 5-го класу	Так/Ні

Рис. 23.10

Ознайомимося з деякими типами даних, які стануть нам у пригоді під час створення БД. Розглянемо таблицю:

Тип	Призначення
Короткий текст	Для зберігання текстів обсягом до 255 символів
Автономерація	Для зберігання натуральних чисел з автоматичним нарощуванням

Число	Для зберігання чисел
Дата й час	Для зберігання календарних дат і часу
Обчислювальний	Для зберігання арифметичних і логічних виразів та результатів обробки даних
Так/Ні	Для зберігання логічних даних у вигляді <b>Так</b> або <b>Ні</b>

Під час заповнення таблиці в полі, яке має тип **Автонумерація**, числа з'являються автоматично, щоразу збільшуючись на 1. Проте в разі вилучення записів їхні номери повторно не використовуються, тому не варто застосовувати поле такого типу, як порядковий номер запису.

Обрання оптимальних типів даних дозволяє заощадити ресурси обчислювальної системи та сприяє більш швидкому і гнучкому пошуку і опрацюванню даних у БД.

### Питання для самоконтролю

1. Як називаються рядки і стовпці таблиць реляційної бази даних?
2. Перелічіть кроки проектування реляційної бази даних.
3. Що називають моделлю **сутність — зв'язок** бази даних? Наведіть приклади зв'язків сутностей.
4. У чому відмінність опрацювання таблиць у режимі **Таблиця** і в режимі **Конструктор таблиць**?
5. Чому порожню таблицю СКБД **Access** можна назвати «порожньою» лише умовно?
6. Опишіть призначення кількох відомих вам типів даних СКБД **Access**.

### Вправа 23

Виконати проектування та створення порожньої бази даних.

- 1) Запустіть СКБД **Access**.
- 2) Виберіть створення **Пустої** настільної бази даних.
- 3) Уведіть ім'я файлу **Інформатика**.
- 4) Оберіть місце збереження БД за вказівкою вчителя (за замовчуванням зберігається у папці **Документи**).
- 5) За допомогою контекстного меню таблиці **Таблиця1** відкрийте її в режимі конструктора, увівши ім'я **Учень**.
- 6) Уведіть імена усіх полів таблиці **Учень** та оберіть для кожного поля оптимальний тип даних (рис. 23.12).

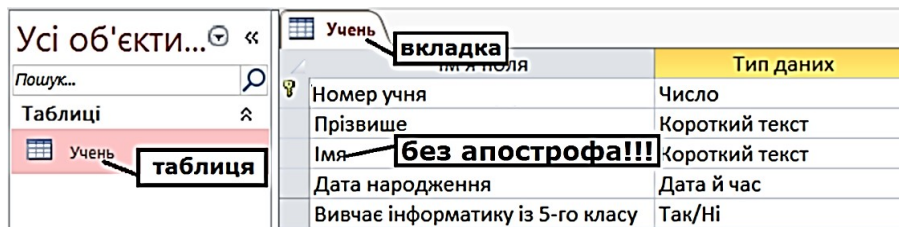


Рис. 23.12

7) За допомогою контекстного меню вкладки **Учень** збережіть зміни та закрийте вкладку.

8) У режимі конструктора створіть іще одну порожню таблицю.

9) Уведіть імена усіх полів нової таблиці та оберіть для кожного поля відповідний тип даних (рис. 23.4).

Для властивості **Обчислюваний** у вікні **Побудовник виразів** (рис. 23.13) поставте **0** і натисніть кнопку **ОК**.

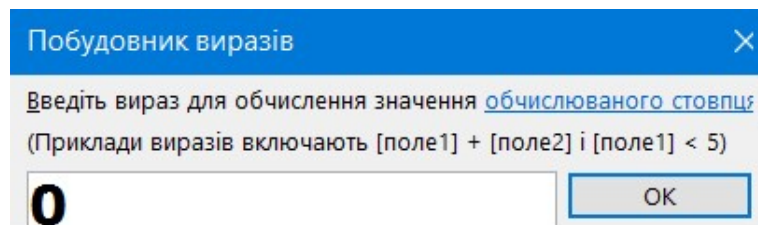


Рис. 23.13

10) Збережіть зміни, давши ім'я таблиці **Успішність**.

У діалоговому вікні щодо створення первинного ключа (рис. 23.14) натисніть кнопку **Ні**.

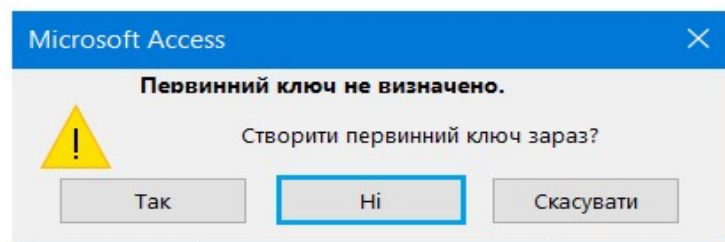


Рис. 23.14

11) Закрийте вкладку **Успішність**.

12) Закрийте вікно СУБД **Access**.

При додаткових запитах на збереження чи закриття таблиць відповідайте **Так**.

### Комп'ютерне тестування

Виконайте тестове завдання 23 з автоматичною перевіркою результату.

## § 24. ЗАПОВНЕННЯ ТА ЗВ'ЯЗКИ ТАБЛИЦЬ

Створені нами таблиці **Access** не можна вважати повною мірою порожніми, оскільки вони містять інформацію про поля даних. Але самих даних, придатних для пошуку й опрацювання, у «порожніх» таблицях немає. На цьому уроці ви навчитеся заповнювати таблиці **Access** необхідними даними та встановлювати зв'язки між ними.

### Властивості полів

Кожне поле залежно від обраного типу даних має певні властивості, які налаштовуються після клацання по назві типу даних (рис. 24.1). Налаштування властивостей поля можна вписати в рядок уведення власноруч або обрати зі списку, що розгортається.

Опис обраної властивості надається у вікні знизу справа (рис. 24.1).

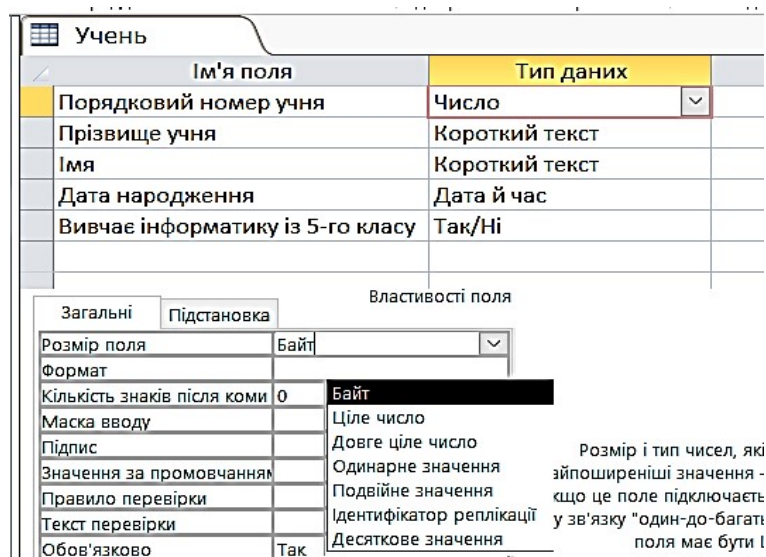



Рис. 24.1

Ознайомимося з деякими властивостями полів. Розглянемо таблицю:

Властивість	Опис
Розмір поля	Максимальне значення типу даних (кількість символів або числове значення)
Формат	Вигляд даних у таблиці
Маска вводу	Шаблон для полегшення уведення даних у таблицю
Значення за промовчанням	Уводиться в таблицю автоматично
Обов'язково	Чи є дане поле обов'язковим для введення в таблицю
Правило перевірки	Обмеження на значення, що вводиться в таблицю

**Маска вводу** містить знаки, які певним чином інтерпретуються СКБД **Access**. Розглянемо деякі з них.

Знак (Engl.)		Інтерпретація
Заповн. обов'язкове	Заповн. необов'язкове	
0	9	Допускає введення лише десяткових цифр
	#	Допускає введення лише десяткових цифр, пропусків, та знаків "+" і "-"
L	?	Допускає введення лише літер
A	a	Допускає введення лише літер і десяткових цифр
&	C	Допускає введення будь-якого символу або пропуску
.: - /		Позначення роздільників (десятьового, тисяч, дат, часу...)
<		Вмикає нижній регістр для літерних клавіш
>		Вмикає верхній регістр для літерних клавіш

**ПРИКЛАД 1.** Якщо у режимі конструктора для короткого формату часу встановити маску вводу **00:00**, то під час введення даних в режимі заповнення таблиці комірка виглядатиме так: . Після введення хоча би однієї цифри потрібно буде обов'язково завершити введення решти цифр (не плутайте з властивістю поля **Обов'язкове**).

### Первинний ключ



**Первинним ключем** таблиці називають одне або декілька

полів, значення яких дозволяють відрізнити будь-які два записи в таблиці (тобто є унікальними в таблиці).

У жодних двох записах значення ключа не можуть збігатися, і за цим слідкує СКБД **Access**.

Ключових полів у таблиці може бути декілька і тоді первинний ключ таблиці називається **комбінованим**.

**ПРИКЛАД 2.** Оскільки прізвища учнів і учениць можуть повторюватися, то для однозначного визначення особистості можна застосувати комбінований ключ до поля **Прізвище** та поля **Імя**. Але і це не гарантує унікальності даних,



адже існує вірогідність того, що прізвище й ім'я учнів можуть збігтися. Абсолютно унікальними, наприклад, є серія та номер паспорта, ідентифікаційний код, номер особової справи учня (учениці), та навіть порядковий номер учня (учениці), який в одному списку не може збігтися для двох осіб.

Для створення первинного ключа (ключового поля) потрібно:

- 1) відобразити таблицю в режимі конструктора;
- 2) виділити потрібне поле (клацнути будь-де в рядку поля);

3) вибрати відповідну команду контекстного меню поля або на вкладці **КОНСТРУКТОР** у групі **Знаряддя** натиснути кнопку **Ключове поле** (рис. 24.2).



Рис. 24.2

Зліва від імені виділеного поля з'явиться символ ключа як підтвердження того, що дане поле є ключовим.

Для створення комбінованого ключа потрібно виділити кілька полів в області вибору, а потім призначити їм властивість **Ключового поля**, утримуючи при цьому клавішу **Ctrl**.

### Уведення даних у режимі таблиці

Використання режиму таблиці є найпростішим способом **уведення даних**. Для цього необхідно:

- 1) подвійним клацанням на назві таблиці (поле **Усі об'єкти**) відкрити потрібну таблицю (рис. 24.3);
- 2) ввести дані, переходячи до комірок клавішами стрілок, **Tab** чи **Enter** або клацаючи мишею по комірці;
- 3) після натискання **Tab** або **Enter** в останньому полі курсор перевести на початок наступного запису.

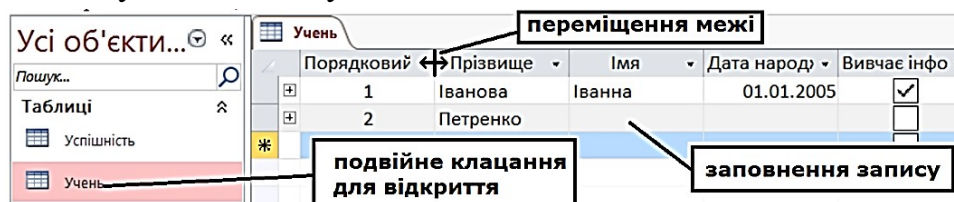


Рис. 24.3

**!** Без заповнення обов'язкових полів (властивість **Обов'язкове**) неможливо перейти до заповнення наступного запису (рядка) і навіть видалити поточний запис.

Збереження кожного запису відбувається автоматично після переходу до наступного.



Зліва від першого поля таблиці знаходиться область вибору запису, в якій з'являються піктограми, що вказують на його стан:

- ▶ — цей запис є поточним (вибраним);
- ✎ — у записі редагуються дані.

### Зв'язки між таблицями

Кожному учневі (учениці) із таблиці **Учень** відповідає єдиний запис із таблиці **Успішність**. Ви вже знаєте, що такий зв'язок називається **один-до-одного**.

Розглянемо **алгоритм встановлення зв'язку** між таблицями **Учень** і **Успішність**. Для цього:

- 1) натисніть кнопку **Зв'язки** на вкладці **ЗНАРЯДДЯ БАЗИ ДАНИХ** у групі **Зв'язки**;
- 2) у контекстному меню робочої області вікна **Зв'язки** оберіть команду **Відобразити таблицю** або натисніть кнопку **Відобразити таблицю** на вкладці **КОНСТРУКТОР** у групі **Зв'язки** (рис. 24.4);
- 3) у вікні **Відображення таблиці** (рис. 24.5) послідовно виділяйте назву таблиці і натискайте кнопку **Додати**;



Рис. 24.4

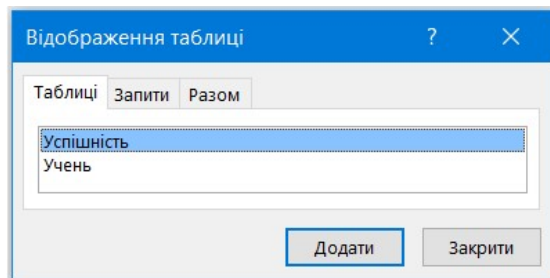


Рис. 24.5

- 4) заверште відображення таблиць кнопкою **Закрити**;
- 5) для встановлення зв'язку перетягніть, наприклад, поле **Номер учня** (із таблиці **Учень**) на поле **Порядковий номер** (у таблицю **Успішність**) (рис. 24.6);

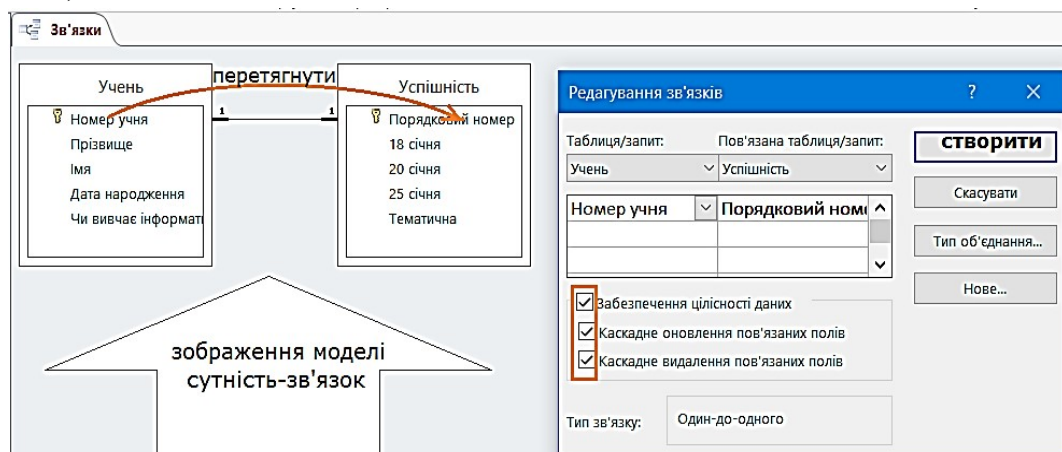


Рис. 24.6

б) у вікні **Редагування зв'язків** установіть прапорці **Забезпечення цілісності даних**, та **Каскадне оновлення пов'язаних полів** (щоб при зміні значення поля зв'язку в головній таблиці **Учень** автоматично змінювалося значення у відповідному полі підлеглої таблиці **Успішність**) і **Каскадне видалення пов'язаних полів** (при видаленні запису із головної таблиці видаляються зв'язані дані в підлеглих);

7) натисніть кнопку **Створити** — у схемі даних з'явиться лінія зв'язку двох полів типу **один-до-одного**.

Для збереження схеми даних потрібно в контекстному меню вкладки **Зв'язок** вибрати команду **Зберегти**.

Для **редагування зв'язків** потрібно двічі клацнути на лінії зв'язку, а для **видалення зв'язку** — закрити всі таблиці, клацнути по лінії зв'язку та натиснути клавішу **Delete**.

### Підстановки

Якщо поля таблиці містять обмежену кількість значень, то ці дані краще не вводити, а обирати зі списку. Так, наприклад, зручно обирати зі списку номер групи, клас, колір тощо (рис. 24.7).

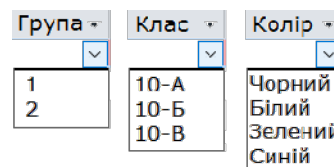


Рис. 24.7

У створеній нами БД **Інформатика** заповнювати оцінками таблицю **Успішність** незручно, оскільки в ній відсутні прізвища учнів. Підстановка дає можливість замінити порядкові номери таблиці **Успішність** прізвищами учнів зі зв'язаної таблиці **Учень**.

Розглянемо алгоритм здійснення такої підстановки:

1) відобразіть таблицю **Успішність** у режимі конструктора та для поля **Порядковий номер учня** розгорніть список **Тип даних** і виберіть команду **Майстер підстановок** (рис. 24.8);

2) у вікні **Майстер підстановок** клацніть на кнопці **Далі** (рис. 24.9);

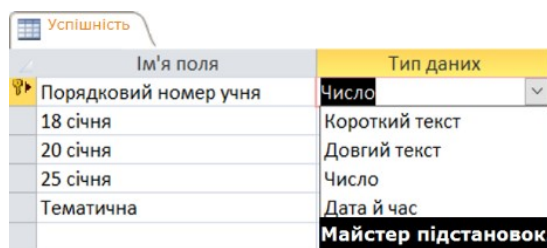


Рис. 24.8

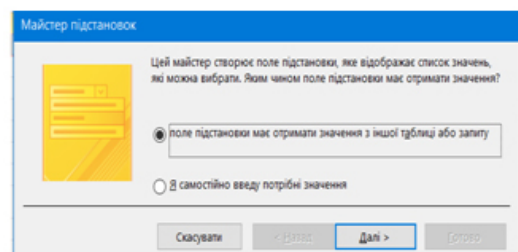


Рис. 24.9

3) у наступному вікні виберіть таблицю **Учень** і клацніть **Далі** (рис. 24.10);

4) у наступному вікні зі списку **Доступні поля** виберіть **Прізвище** і клацніть на кнопці **>** та кнопці **Далі** (рис. 24.11);

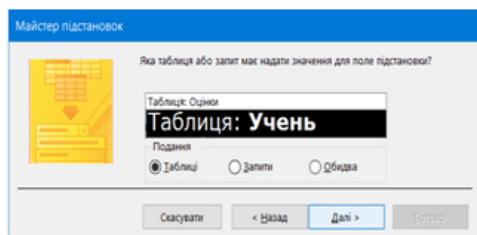


Рис. 24.10

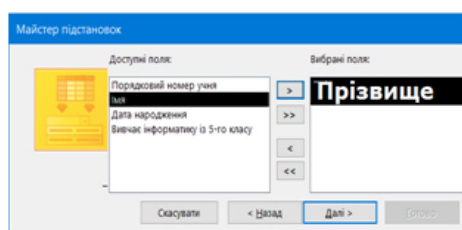


Рис. 24.11

5) у наступних вікнах, що з'являтимуться, погодьтеся з усіма налаштуваннями, натискаючи кнопки **Далі**, **Готово**, **Так**.

Після виконання зазначених дій обидві таблиці стають зв'язаними відповідними полями і в заповненій таблиці **Успішність** у полі **Порядковий номер** замість номерів відобразатимуться прізвища учнів із таблиці **Учень**.

**!** Під час заповнення таблиці **Успішність** у полі **Порядковий номер** учня замість введення номерів потрібно буде обирати прізвища зі списку, що розгортається.

Для редагування наявних підстановок потрібно відкрити таблицю з підстановкою, перейти на відповідне поле і скористатись командою **Змінити підстановку контекстного меню обраного поля** або на вкладці **ПОЛЯ** у групі **Властивості** натиснути кнопку **Змінити підстановку** (рис. 24.12).

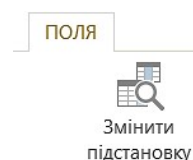


Рис. 24.12

### Питання для самоконтролю

1. Що визначає властивість **Розмір поля**?
2. Яке призначення наступної маски вводу: **00:00** для короткого формату часу?
3. Що таке первинний ключ? Коли виникає необхідність створення комбінованого первинного ключа?
4. Як відкрити таблицю для введення даних?
5. Як розпочати встановлення зв'язків між таблицями?
6. Яке призначення підстановок (наведіть приклади їх застосування)?

### Вправа 24

Виконати налаштування властивостей полів, заповнення та зв'язок таблиць.

- 1) Запустіть СКБД **Access**. Відкрийте створену раніше базу даних **Інформатика**.
- 2) Відкрийте в режимі конструктора таблицю **Учень**.

3) Для кожного типу даних налаштуйте властивості полів на свій розсуд з урахуванням таких міркувань:


- Учні в групі не повинно бути більше 15-ти (**Розмір поля**).

*Підказка.* Пригадайте, що десяткове значення, задане одним байтом, не перевищує 255.

- Навіть подвійні прізвища та імена навряд чи будуть довгими за 30 символів (**Розмір поля**).

- У шкільному журналі не може існувати запис з оцінками без прізвища учня (**Обов'язково**).

- Дату народження уводять залежно від обраного формату (**Формат**) (**Шаблон**).

4) Зробіть ключовими  поле **Номер учня**. Збережіть налаштовану таблицю **Учень**.

5) Відкрийте в режимі конструктора таблицю **Успішність** та налаштуйте для числових типів даних властивості: **Розмір поля**, **Значення за промовчанням**, **Правило перевірки**, **Обов'язково**.

*Підказка.* Правилем перевірки для оцінок можна встановити верхню межу: **< 13**.

6) Для типу **Обчислюваний** клацніть на властивості **Вираз** та значком справа (рис. 24.13) відкрийте **Побудовник виразів**.



Рис. 24.13

7) У вікні **Побудовник** (або одразу в рядку введення виразу) уведіть наступний вираз для розрахунку й округлення тематичної (середньої) оцінки: **Round((18 січня)+[20 січня]+[25 січня])/3;0** — та натисніть **Enter**.

- 8) Зробіть ключовим (якщо не зроблено) поле **Порядковий номер**. Збережіть налаштовану таблицю **Успішність**.

- 9) Заповніть таблицю **Учень** п'ятьма записами та збережіть таблицю.

- 10) Зв'яжіть обидві таблиці їхніми ключовими полями.

11) У поле **Порядковий номер** (таблиця **Учень**) зробіть підстановку поля **Прізвище** (із таблиці **Успішність**).

12) Заповніть таблицю **Успішність** п'ятьма записами та збережіть таблицю.

Закрийте (збережіть усі зміни) БД **Access**.

### Комп'ютерне тестування

Виконайте тестове завдання 24 з автоматичною перевіркою результату.

## § 25. ВПОРЯДКУВАННЯ ТАБЛИЦЬ

Ефективно працювати з великими обсягами даних дозволяють засоби їх впорядкування, пошуку та фільтрування. Ви вже користувались подібними засобами, працюючи з електронними таблицями **Excel**. У цьому параграфі ви дізнаєтеся, як упорядковувати, фільтрувати та шукати дані реляційної бази даних **Access**.

### **Впорядкування полів**

Іноді постає необхідність зміни розташування полів таблиці в порядку, відмінному від початкового.



**Впорядкування полів таблиці** — це зміна порядку взаємного розташування полів.

У режимі конструктора чи перегляду таблиці **Access** НЕ можна налаштувати автоматичне впорядкування полів за певними умовами.

**Переміщення полів** з одного місця таблиці в інше здійснюється лише в ручному режимі за таким алгоритмом:

- 1) виділіть потрібне поле(я), клацнувши по заголовку (таблиця) або по області вибору (конструктор);
- 2) перетягніть мишею виділене поле (стовпець) до іншого місця розташування;
- 3) збережіть зміни.

**!** Зміни в розташуванні полів, зроблені та збережені в режимі конструктора таблиць, *впливають* на розташування полів у режимі відображення таблиці для перегляду і заповнення.

Зміни в розташуванні полів, зроблені та збережені в режимі перегляду таблиці, *НЕ впливають* на розташування полів у конструкторі.

### **Впорядкування (сортування) записів**

Часто при перегляді таблиць буває потрібно розмістити записи за алфавітним порядком прізвищ, віком, показниками успішності тощо.



**Впорядкування записів** — це встановлення порядку записів за зростанням або спаданням значень певного поля.

Впорядкування даних у відкритій для перегляду таблиці відбувається за таким алгоритмом:

- 1) установіть курсор на поле, за даними якого треба провести впорядкування;

2) натисніть кнопку **За зростанням** або **За спаданням** (рис. 25.1), що у контекстному меню, або у списку поряд з іменем поля, чи на вкладці **ОСНОВНЕ** в полі **Сортування і фільтри**.

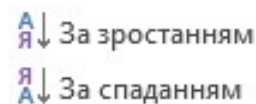


Рис. 25.1

**Відміна усіх впорядкувань записів і повернення до стану, за яким відбувалось початкове заповнення таблиці, виконується командою **Видалити сортування** на вкладці **ОСНОВНЕ** в полі **Сортування і фільтри**.**

Якщо встановити впорядкування (сортування) записів за значеннями кількох полів, то записи впорядковуються за значеннями поля, сортування якого увімкнуте останнім.

Впорядкування записів, на відміну від впорядкування полів таблиці **Access**, не можна здійснювати в ручному режимі перетягуванням рядків з одного місця в інше.

### Пошук даних за зразком

Пошук даних за зразком приводить до послідовного виділення в записах знайдених зразків.

**!** Пошук даних можливий лише в таблиці, відкритій у режимі перегляду (заповнення).

Для **виконання пошуку даних** у заповненій таблиці потрібно:

- 1) клацнути по полю, за даними якого проводиться пошук;
- 2) натиснути сполучення клавіш **Ctrl + F** або на вкладці **ОСНОВНЕ** у групі **Пошук** кнопку **Знайти** (рис. 25.2);
- 3) у вікні **Пошук і заміна** клацнути вкладку **Знайти** (рис. 25.3);



Знайти

Рис. 25.2

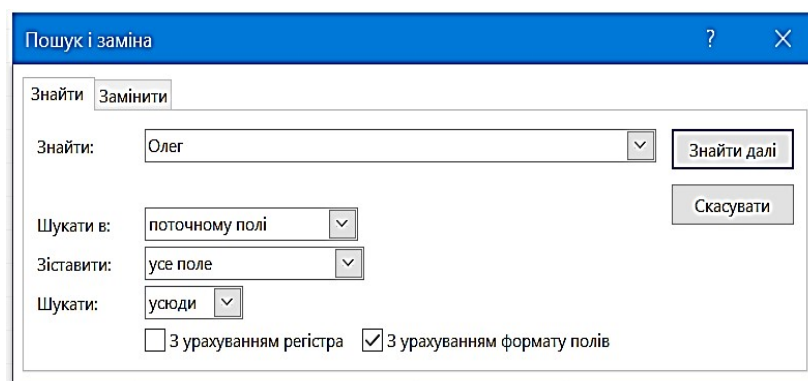


Рис. 25.3

- 4) у поле **Знайти**: ввести зразок розшукуваних даних (тут **Олег**);
- 5) у списку **Шукати** вибрати місце пошуку (у нас в **поточному полі**);
- 6) у списку **Зіставити** вибрати частину поля для пошуку;
- 7) встановити за необхідності прапорець **З урахуванням регістра**;

8) встановити за необхідності прапорець **З** урахуванням формату **полів**, обмеживши пошук певним форматом (без прапорця за зразком 01/01/20011 буде, наприклад, знайдено також дату, представлену у форматі 01.01.2001);

9) натиснути кнопку **Знайти далі** — в таблиці буде виділено текст, який збігається із заданим зразком;

10) для продовження пошуку натиснути кнопку **Знайти далі**.

Для забезпечення більш гнучкого пошуку в зразок можна підставляти спеціальні знаки. Розглянемо таблицю:

Знак	Що визначає	Приклад запису	Умова
*	Будь-яка кількість будь-яких символів	*a	Закінчується літерою «а»
		a*	Починається літерою «а»
		*a*	Містить літеру «а»
		a*б	Починається літерою «а» і закінчується літерою «б»
?	Будь-який <i>один</i> символ	????a	Містить п'ять символів, останній з яких — літера «а»
		квіт?	Містить п'ять символів, перші з яких — «квіт»
		10 – ? клас	На місці знака «?» може знаходитись будь-який символ
#	Будь-яка <i>одна</i> десяткова цифра	## - A клас	На місці «##» може знаходитись будь-яка комбінація із двох десяткових цифр
		*##*	Містить будь-яку цифру
		*##	Закінчується двома цифрами
[ ]	Один із символів, що в дужках (діапазон позначається «-»)	квітк[ai]	Слова «квітка» та «квітки»
		[аб]*	Починається літерою «а», або «б»
		*[а-г]*	Містить одну з літер: «а», «б», «в», або «г»
!	Знак заперечення «Не». Окрім символів у дужках, що стоять після «!»	квітк[!y]	Містить шість символів. Перші п'ять — «квітк». Останній не може бути «у»
		*[!a]	Не закінчується літерою «а»
		*[!a]*	Не містить літери «а»



Якщо у вікні **Пошук і заміна** відкрити вкладку **Замінити**, то з'являється поле **Замінити на** для введення даних, якими можна буде замінити знайдені значення, натиснувши кнопки **Замінити** або **Замінити все** (рис. 25.4).

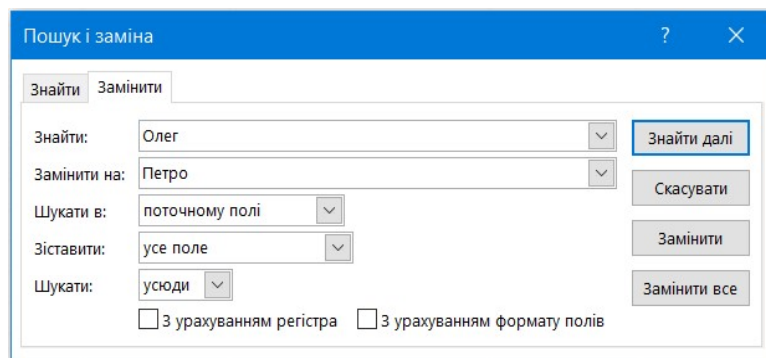


Рис. 25.4

### **Приховування полів**

Поля таблиці в режимі перегляду можна приховувати.

Для **приховування поля** необхідно клацнути по ньому, а потім зі списку **Додатково** вибрати команду **Приховати поля** на вкладці **ОСНОВНЕ** у групі **Записи** (рис. 25.5). Обране поле одразу зникне. Після збереження змін, при повторному відкритті таблиці, прихованих полів не буде видно, доки вони не будуть знову відображені командою **Відобразити поля**.

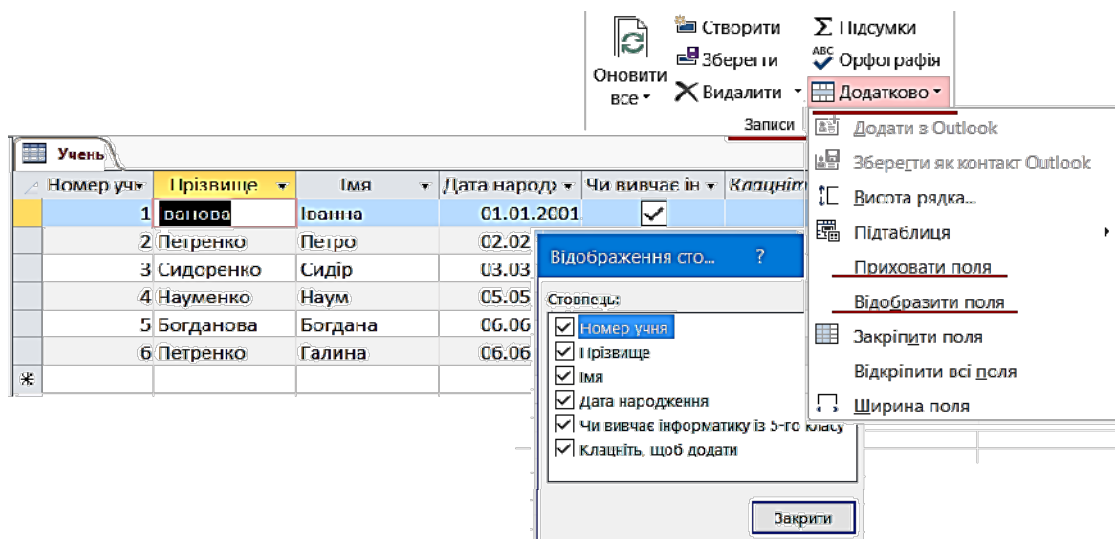


Рис. 25.5

Приховування полів не видаляє їх зі структури таблиці, а лише не відображає в режимі перегляду.

### **Фільтрування записів**

Впорядкування даних не зменшує кількості записів, які доводиться переглядати. Зручним засобом для відбору потрібних записів є **фільтри**.





**Фільтрування даних** — це відображення в таблиці тільки тих

записів, дані в яких відповідають певним умовам (умовам фільтрування).

Є кілька способів встановлення фільтрів:

- **Фільтр за виділеним зразком** — відображає записи, в яких є певні дані

- **Фільтрування за формою** — дозволяє налаштувати фільтрування за кількома зразками (умовами)

- **Розширений фільтр** — відображає записи із даними, відібраними за критеріями користувача у формі **Запиту**

Фільтри не змінюють даних у таблиці, а лише приховують ті записи, що не відповідають умовам фільтрування. При повторному відкритті збереженої БД усі табличні фільтри вимкнені, і для застосування їх вмикають відповідною командою контекстного меню чи кнопкою на вкладці **ОСНОВНЕ** у групі **Сортування й фільтр** (рис. 25.6).



Рис. 25.6

При застосуванні до таблиці послідовно кількох фільтрів кожен наступний фільтр застосовується до записів, які залишилися видимими після попереднього фільтрування.

### Фільтрування за виділеним зразком

Щоб скористатися найпростішим і швидким засобом **відбору даних у таблиці**, відкритій для перегляду, необхідно виконати такі дії:

- 1) розгорнути контекстне меню обраного елемента даних;
- 2) в основному (або додатковому) списку вибрати потрібну команду (рис. 25.7).

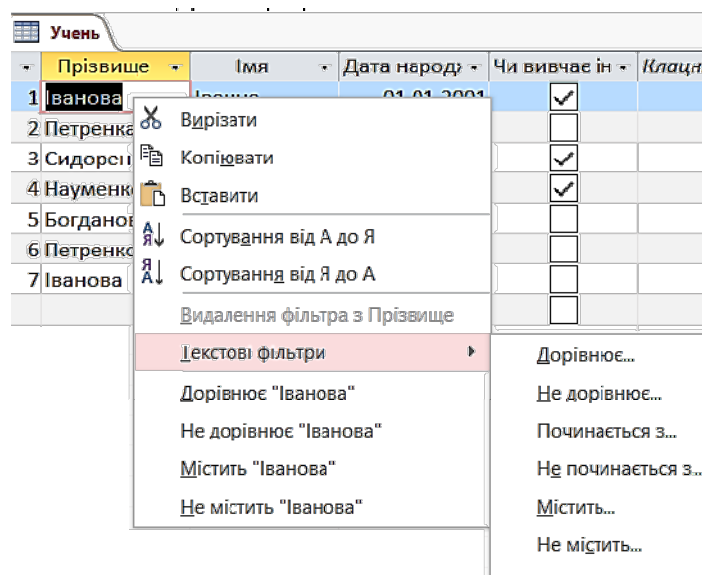


Рис. 25.7

У результаті фільтрування в таблиці залишаються видимими записи, які мають дані, що збігаються зі зразком. Вимикання/вмикання дії фільтрів здійснюється кнопкою (див. рис. 25.6) на вкладці **ОСНОВНЕ** у групі **Сортування й фільтр**.

### Фільтрування за формою

Для створення простого фільтра у відкритій для перегляду таблиці необхідно виконати такі дії:

1) на вкладці **ОСНОВНЕ** у групі **Сортування й фільтр** відкрити список **Додатково** (рис. 25.8);



2) у списку вибрати команду **Змінити фільтр**;

Рис. 25.8

3) у вікні **Фільтрування за формою** (рис. 25.9) вибрати одне або кілька даних зі списку чи вписати умови з використанням підстановочних знаків.

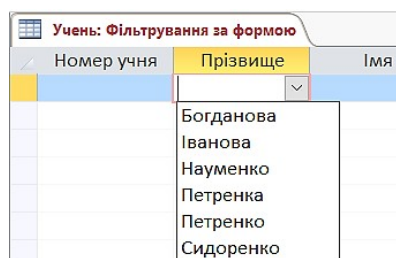


Рис. 25.9

### Розширений фільтр

У режимі **Розширеного фільтру** створюється новий об'єкт **Access**, який називається **Запит**. Роботу із **Запитами** ми розглянемо в наступному параграфі.

#### ▶ Збереження фільтра

Створені фільтри зберігаються під час збереження таблиці.

При повторному відкритті таблиці фільтр викликається командою **Застосувати фільтр**, що на вкладці **ОСНОВНЕ** у групі **Сортування й фільтр**.

Створений фільтр заміняє той, що був збережений з таблицею.

#### ▶ Видалення фільтра

! Перед створенням нового фільтра доцільно видалити попередні, щоби їх дія не впливала на результат фільтрування.

Для видалення фільтра необхідно на вкладці **ОСНОВНЕ** у групі **Сортування й фільтр** розгорнути список **Додатково** та натиснути кнопку **Очистити всі фільтри** (рис. 25.10).



Рис. 25.10

Усі фільтри, раніше збережені з цією таблицею, буде видалено, а кнопка **Очистити всі фільтри** стане неактивною.

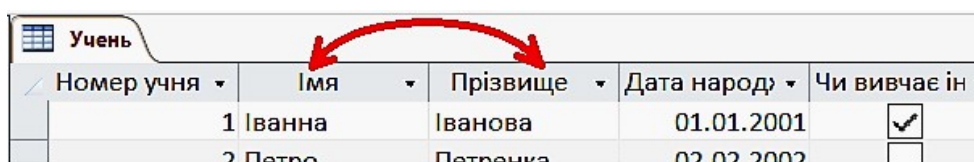
### Питання для самоконтролю

1. Як можна перемістити або приховати поле таблиці в режимі перегляду?
2. У чому полягає впорядкування (сортування) записів таблиці?
3. Як використати підстановочні знаки для пошуку слова, що містить літеру «а»?
4. Що відображає фільтр за виділеним зразком?
5. Як діють кілька фільтрів, застосованих до записів?
6. Як видалити встановлені фільтри до таблиці?

### Вправа 25

Виконати впорядкування полів і записів та пошук і фільтрування даних у таблиці.

- 1) Створіть на хмарному диску (наприклад, **Google**) документ, у якому зберігатимете розв'язки завдань.
- 2) Переіменуйте документ своїм прізвищем, додавши до назви: **Вправа 20** (наприклад: **Петренко Вправа 20**).
- 3) Надайте доступ учителю для редагування створеного вами документа.
- 4) Запустіть СКБД **Access**. Відкрийте створену раніше базу даних **Інформатика**.
- 5) Відкрийте таблицю **Учень** у режимі перегляду (заповнення).
- 6) Перемістіть поле **Прізвище** праворуч від поля **Імя** (рис. 25.11).



Номер учня	Імя	Прізвище	Дата народу	Чи вивчає ін
1	Іванна	Іванова	01.01.2001	<input checked="" type="checkbox"/>
2	Петро	Петренко	02.02.2002	<input type="checkbox"/>

Рис. 25.11

- 7) Приховайте поле **Чи вивчає інформатику із 5-го класу**.
- 8) Впорядкуйте записи за зростанням дати народження.  
Зробіть **скріншот1** із зображенням таблиці та вставте в **Google**-документ.
- 9) Знайдіть у таблиці записи, що містять ім'я **Іванна** або інше ім'я за вашим вибором.  
Не закриваючи вікна **Пошук і заміна**, зробіть **скріншот2** із зображенням таблиці та вставте в **Google**-документ.

10) Встановіть фільтр, для відображення записів лише тих учнів, хто народився узимку (або іншого сезону за вашим вибором).

Зробіть **скріншот3** із зображенням таблиці та вставте в **Google-документ**.

11) Видаліть створений фільтр та відмініть впорядкування таблиці (видаліть сортування).

12) Відобразіть приховане поле **Чи вивчає інформатику із 5-го класу**.

Зробіть **скріншот4** із зображенням таблиці та вставте в **Google-документ**.

Закрийте базу даних **Access** (збережіть усі зміни).

### Комп'ютерне тестування

Виконайте тестове завдання 25 з автоматичною перевіркою результату.

## § 26. ЗАПИТИ

Запити, окрім фільтрування даних, дозволяють автоматично створювати нові таблиці, додавати, видаляти чи змінювати дані в існуючих таблицях тощо.



**Запит** — це засіб відбору та/або опрацювання табличних даних

за заданими правилами і критеріями.

У цьому параграфі ми розглянемо особливості створення деяких типів запитів (рис. 26.1).

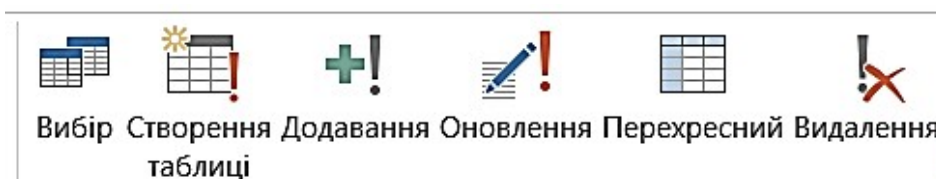


Рис. 26.1

### Особливості запитів

Коли користувач або користувачка відкриває запит, відбувається пошук інформації в таблицях БД згідно із заданими в запиті правилами і критеріями. Результати роботи відкритого запиту виводяться у вигляді таблиці з упорядкованими, якщо це потрібно, полями і записами, та відфільтрованими і, можливо, зміненими згідно з вимогами запиту даними.

Запит є більш потужним засобом відбору даних, ніж фільтр, ще й тому, що він може обробляти одночасно декілька об'єктів (таблиць, запитів).

## Створення запиту вибору

Створення запитів в **Access** можливе за допомогою **Майстра запитів** і **Конструктора запитів**. Режим конструктор запитів є більш універсальним, тож розглянемо насамперед його.

Конструювання запиту починається зі створення схеми даних.

Після натискання кнопки **Конструктор запитів** (рис. 26.2), що на вкладці **СТВОРЕННЯ** у групі **Запити**, відкривається вкладка запиту вибору (див рис. 26.1). Вона складається із поля для **Схеми даних** та із **Бланка запиту**. Якщо в області схеми даних немає об'єктів, то відкривається вікно **Відображення таблиці**, з якого можна здійснити додавання об'єктів (таблиць, запитів) кнопкою **Додати**, або подвійним клацанням імені об'єкта (рис. 26.3).

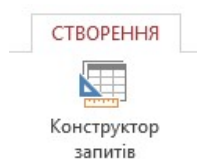


Рис. 26.2

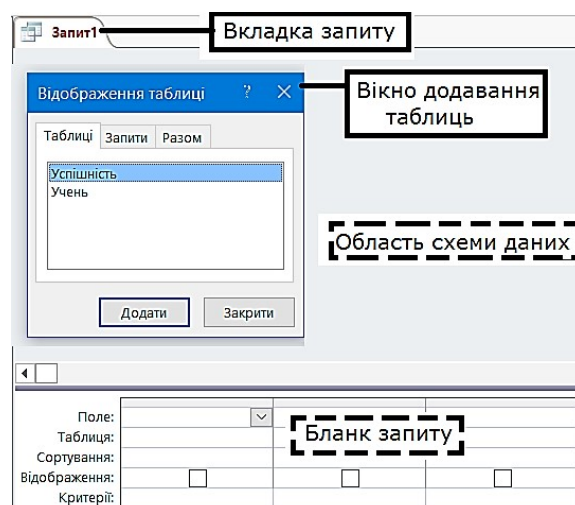


Рис. 26.3

Якщо ви випадково закрили вікно **Відображення таблиці**, то відновити його можна відповідною командою контекстного меню **Області схеми даних**.

## Запит вибору із впорядкуванням (сортуванням) даних

Розглянемо алгоритм створення запиту вибору даних у БД **Інформатика** для відображення упорядкованих за алфавітом прізвищ тих учнів і учениць, чії тематичні оцінки становлять 10, 11, 12.

Очевидно, що нам знадобляться обидві зв'язані таблиці, оскільки в одній є прізвища, а в іншій — оцінки. Отже:

1) додаємо до схеми даних дві таблиці: **Учень** та **Успішність** (див. рис. 26.3). З'являться сутності таблиць (рис. 26.4).

**!** Для видалення зайвої сутності зі **Схеми даних** клацніть по ній та натисніть клавішу **Delete**.

Якщо з якихось причин таблиці не є зв'язаними, то:

2) зв'язуємо ключові поля: **Номер учня** (табл. **Учень**) та **Порядковий номер** (табл. **Успішність**) (рис. 26.4).

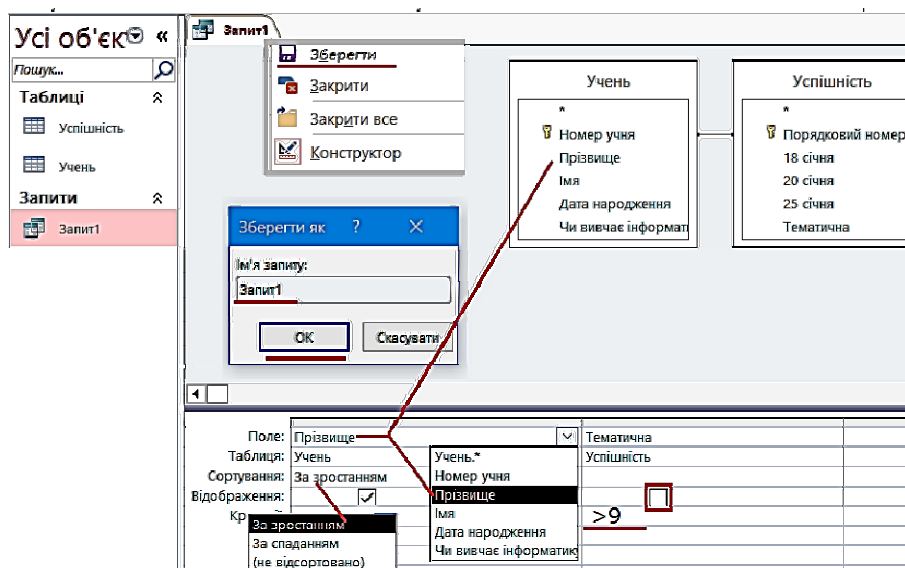



Рис. 26.4

У запиті йдеться про **Прізвища** та **Тематичні оцінки учнів**. Тож:

3) розміщуємо назви відповідних полів у **Бланку запиту**, перетягуючи (або подвійно клацаючи) назви із сутностей **Схеми даних** до рядка **Поле** або обираючи назви зі списку, що розгортається.

! Для видалення зайвого стовпчика (поля) його виділяють, клацнувши над ним при появі стрілки  (стовпчик почорніє), і натискають клавішу **Delete**.

Оскільки в запиті прізвища потрібно впорядковувати за алфавітом, то:

4) у рядку **Сортування** поля **Прізвище** обираємо зі списку команду **За зростанням**.

Далі слід фідфільтрувати ті записи, тематична оцінка в яких більша за 9 балів. Для цього:

5) у рядку **Критерії** поля **Тематична** вписуємо умову відбору даних >9.

При введенні умов використовуються відомі вам із **Excel** оператори: <, >, =, <>, **and**, **or**, **not** тощо.

У запиті не вимагається відображення оцінок, а лише прізвищ. Тож:

6) у рядку **Відображення** поля **Тематична** знімаємо прапорець.

Залишилось зберегти запит, давши за необхідності йому нове ім'я замість стандартного **Запит1**. Для цього:

7) у контекстному меню вкладки запиту обираємо команду **Зберегти**.

В області переходів на панелі **Усі об'єкти** з'явиться значок з ім'ям нового запиту.

### Запуск запиту

Для запуску запиту, тобто відкриття його в режимі таблиці, потрібно двічі клацнути на його значку (панель **Усі об'єкти**), або вибрати команду **Відкрити** в контекстному меню значка, або ж натиснути кнопку **Запуск** (рис. 26.5) вкладки **Конструктор** у групі **Результати**.



Рис. 26.5

Зручним способом зміни представлення запиту в режимі таблиці та режимі конструктора є кнопка **Вигляд**, що на вкладці **Конструктор** у групі **Результати**.

### Запит з параметрами

На відміну від звичайного запиту вибору, запит з параметрами надає користувачу (користувачці) можливість уведення критеріїв відбору в діалоговому вікні під час виконання запиту. Для створення запиту з параметрами в режимі конструктора замість вказування в рядку **Критерії умов відбору** туди разом із підстановочними чи порівняльними знаками у квадратних дужках вписують пояснювальний текст, наприклад **>[Уведіть оцінку, вище якої здійснювати відбір]**, який з'являтиметься у **Вікні введення значень параметрів** щоразу після запуску запиту (рис. 26.6).

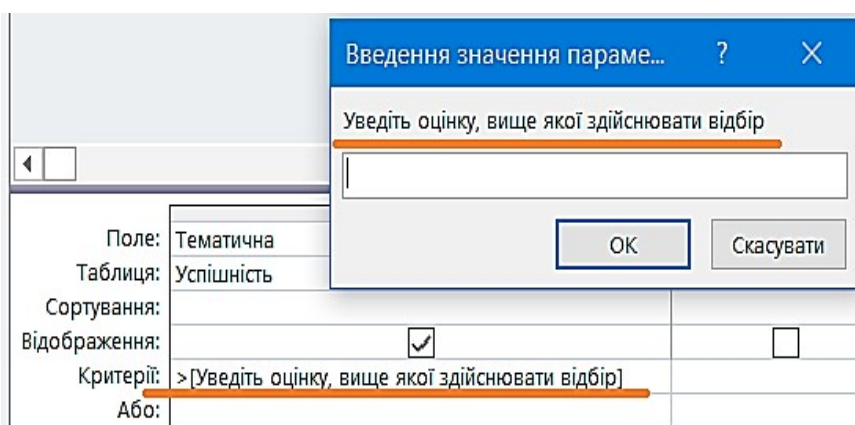


Рис. 26.6

### Обчислення в запитах

Іноді в запитах вибору буває необхідно підрахувати середнє арифметичне, кількість, суму тощо для усіх даних певного поля. Запит, відображений у режимі таблиці, надає таку можливість у вигляді рядка підсумків.



Для додавання рядка підсумків потрібно:

- 1) запустити створений запит (відкрийте в режимі таблиці);
- 2) клацнути кнопку **Підсумки** (рис. 26.7) на вкладці **ОСНОВНЕ** у групі **Записи**;
- 3) розгорнути в потрібному полі список створеного рядка (знизу) і додати потрібну дію над даними поля (рис. 26.8).



Рис. 26.7

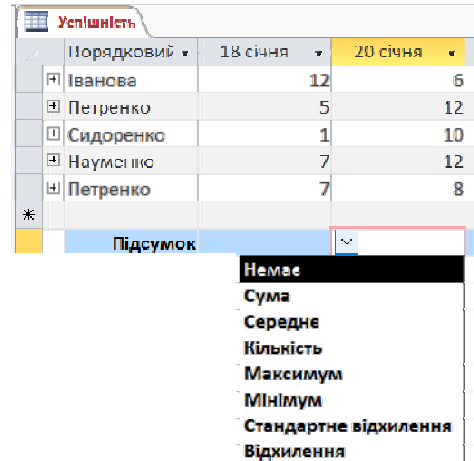


Рис. 26.8

Для приховання рядка підсумків використовується та сама кнопка **Підсумки** (рис. 25.7), що на вкладці **ОСНОВНЕ** у групі **Записи**.

Обчислення в запиті вибору можна додати і в режимі конструктора. Для цього в додаткове поле потрібно ввести необхідний вираз із зазначенням імен полів і таблиці у квадратних дужках: [ **Ім'я Таблиці** ] ! [ **Ім'я поля** ].

Наприклад, нехай нам поряд із прізвищем і тематичною оцінкою потрібно відобразити значення середньої оцінки з точністю до десятих.

Алгоритм **створення запиту** буде таким:

- 1) створюємо запит із відповідною схемою даних;
- 2) у режимі конструктора додаємо потрібні за умовою поля;
- 3) у вільному полі додаємо ім'я нового поля і через двокрапку - обчислювальний вираз (рис. 26.9).

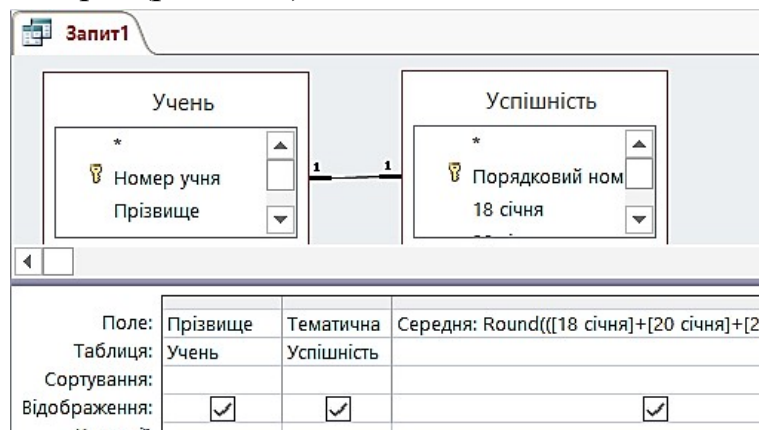


Рис. 26.9



Якщо вираз у формі запити вводиться незручно (дрібний шрифт), можна скористатись сполученням клавіш **Shift + F2**, яке дає змогу вводити дані в зручному масштабі.

Слід враховувати, що на відміну від даних табличного поля **Тематична** розрахунки в полі запити **Середня** не зберігаються в БД і щоразу під час виконання запити перераховуються.

Для розрахункових полів у режимі конструктора можна викликати із контекстного меню **побудовник виразів**, а в режимі таблиці застосовувати **рядки підсумків** тощо.

### Пошук повторюваних значень

Швидке створення запити для пошуку повторюваних даних можна зробити за таким алгоритмом:

1) натисніть кнопку **Майстер запитів**, що на вкладці **Створення** у групі **Запити**;

2) у вікні **Новий запит** оберіть команду **Майстер пошуку повторюваних записів** і натисніть кнопку **ОК**;

3) у вікнах **Майстер пошуку повторюваних записів** оберіть потрібні таблиці і поля.

Розглянемо приклади формування критеріїв (умов) відбору текстових даних у запитах вибору:

Критерій	Результат відбору
Петренко	Лише Петренко
Not Петренко	Усі окрім Петренко
Іванова and Петренко	Лише Іванова і Петренко
Іванова or Петренко	Лише Іванова або Петренко
< К	Починається з літер А...Л
>= К	Починається з літер К...Я

Критерій	Результат відбору
К*	Починається з літери К
*К	Закінчується літерою К
*К*	Містить літеру К
??К	Містить три символи, останній з яких К
Not Like «*К*»	Не містить літеру К
Null	Пусті значення
Is Not Null	Не пусті значення

### Питання для самоконтролю

1. Що таке запити і для чого вони використовуються?
2. Чим запити відрізняються від фільтрів?
3. Як створити запит вибору даних за певною умовою?
4. Як записати умову відбору даних: містить літеру «а»?
5. Як у запиті створити обчислювальні рядки та/або поля?
6. Як створити запит для пошуку повторюваних даних?

### Вправа 26

Створити запити вибору.

- 1) Запустіть СКБД **Access**.
- 2) Відкрийте створену раніше базу даних **Інформатика**.
- 3) Створіть, збережіть і запустіть **Запит1** на відображення прізвищ тих учнів, які мають тематичну оцінку 12.
- 4) Створіть, збережіть і запустіть **Запит2** на відображення за алфавітом імен тих учнів, які народились узимку.
- 5) Створіть, збережіть і запустіть **Запит3** на відображення прізвищ тих учнів, в іменах яких присутня літера «а».

**!** В україномовному інтерфейсі деяких версій СКБД **Access** (наприклад, **Access 2013**) можуть не спрацьовувати запити з критеріями типу: **\*К, К???** тощо. Тоді, для виконання подібних запитів, слід налаштувати іншу мову інтерфейсу. (**Файл** → **Параметри** → **Мова** → **Вибір мов інтерфейсу та довідки...**).

- 6) Додайте до **Запиту3** рядок підсумків із підрахунком кількості учнів.
- 7) Створіть, збережіть і запустіть **Запит4** на відображення кількості кожної із тематичних оцінок.
- 8) Створіть, збережіть і запустіть **Запит5** на відображення кількості повторюваних прізвищ у списку.

### Комп'ютерне тестування

Виконайте тестове завдання 26 з автоматичною перевіркою результату.

## **Практична робота 9. РОБОТА З ТАБЛИЦЯМИ І ЗАПИТАМИ**

Завдання: створити таблиці бази даних **Access** за зразком і опрацювати табличні дані за допомогою запитів.

Обладнання: комп'ютер зі встановленою СУБД **MS Access**.

### **Хід роботи**

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

1. Запустіть **MS Access** і створіть **Пусту настільну базу даних** з іменем файлу **Практична робота9.accdb**.

Відкрийте в режимі конструктора таблицю **Таблиця 1**, перейменувавши її на **Товари**, і додайте імена трьох полів: **Код**, **Група**, **Назва**.

2. Установіть необхідні типи даних для кожного з трьох полів відповідно до заповненого зразка (рис. 1). Зробіть поле **Код** ключовим.

Код ▾	Група ▾	Назва ▾
1	молочна продукція	молоко
2	молочна продукція	йогурт з наповнювачем
3	макаронні вироби	лапша
4	крупя	гречка
5	макаронні вироби	спагеті
6	чай	зелений
7	чай	чорний
8	крупя	рис
9	молочна продукція	морозиво
10	макаронні вироби	макарони

Рис. 1

3. У вікні **Властивості поля** → **Загальні** виконайте такі дії:

1) оптимізуйте (мінімізуйте) **Розмір поля** для кожного з трьох полів відповідно до зразка, якщо відомо, що кількість записів у таблиці **Товари** не перевищуватиме 250, а найдовші рядки в полях **Група** і **Назва** не перевищуватимуть відповідні рядки для товару з кодом 2 (див. рис. 1);

2) у разі потреби решту властивостей полів змініть на свій розсуд;

3) збережіть таблицю.

4. Заповніть таблицю **Товари** за зразком і збережіть.

5. Створіть у конструкторі таблиць нову таблицю. Для цього:

1) дайте назву полям таблиці відповідно до зразка (рис. 2);

Код угоди ▾	Код товару ▾	Кількість товару ▾	Ціна товару ▾	Дата ▾	Вид розрахунку ▾	Оплата
1	1	1230	8	22.12.2020	безготівковий	<input checked="" type="checkbox"/>
2	6	2301	15,23	05.10.2021	безготівковий	<input checked="" type="checkbox"/>
3	4	2365	7	14.10.2021	готівкою	<input type="checkbox"/>
4	7	235	10,24	26.10.2021	безготівковий	<input checked="" type="checkbox"/>
5	7	256	12,35	30.10.2021	готівкою	<input type="checkbox"/>

Рис. 2

- 2) установіть типи даних для кожного з полів відповідно до зразка, встановивши, де це можливо, тип **Число**;
  - 3) установіть ключове поле на свій розсуд;
  - 4) оптимізуйте (мінімізуйте) розмір полів на власний розсуд, керуючись зразком на рис. 2;
  - 5) у разі потреби решту властивостей полів змініть на свій розсуд;
  - 6) збережіть порожню таблицю з назвою **Угоди**;
  - 7) заповніть таблицю **Угоди** за зразком (див. рис. 2);
  - 8) збережіть таблицю.
6. Установіть зв'язок *один-до-багатьох* для кодів товару в таблицях **Товари** й **Угоди**. Налаштуйте **Параметри об'єднання** (рис. 3).

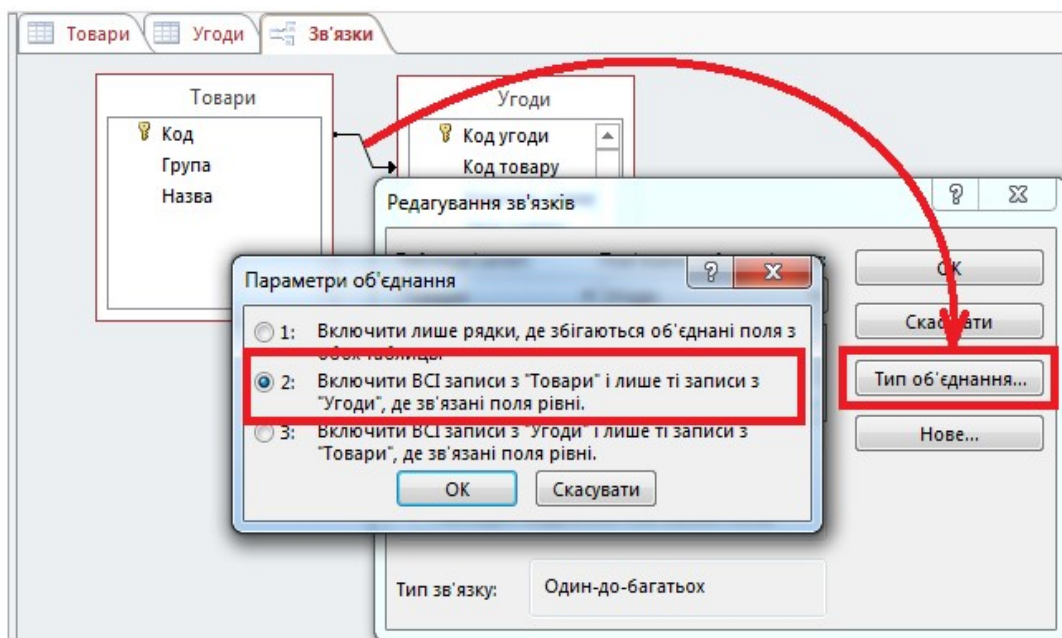


Рис. 3

7. Використовуючи **Конструктор запитів**, створіть запит на відображення в алфавітному порядку всіх назв макаронних виробів (рис. 4).

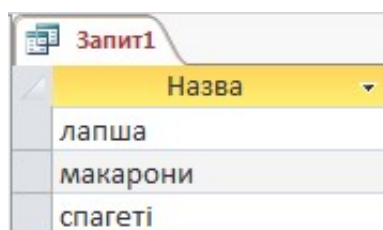
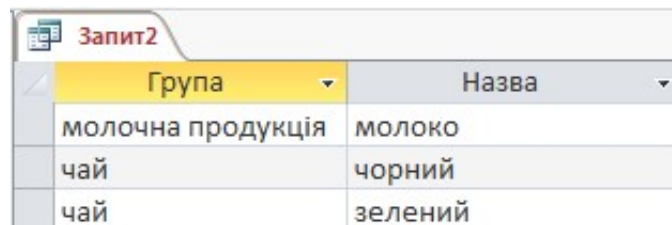


Рис. 4

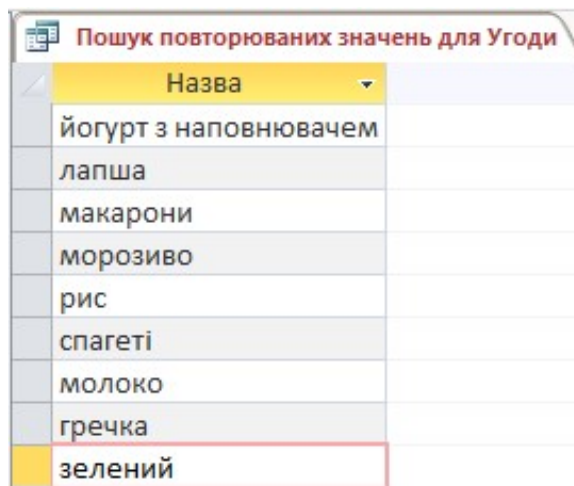
8. Використовуючи **Конструктор запитів**, створіть запит на відображення в порядку збільшення ціни групи і назви всіх товарів із безготівковим видом розрахунку за них (рис. 5).



Група	Назва
молочна продукція	молоко
чай	чорний
чай	зелений

Рис. 5

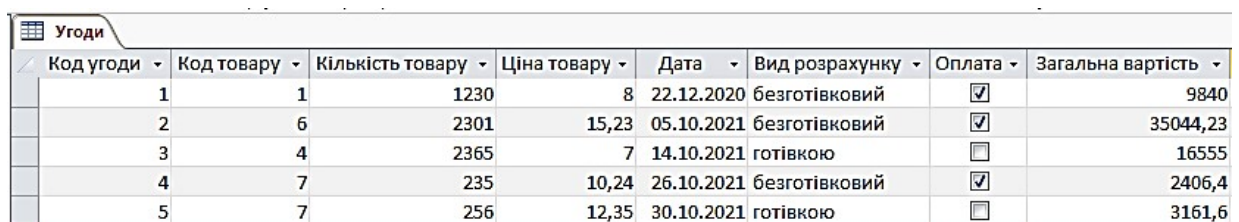
9. Використовуючи **Майстер запитів** і **Конструктор запитів**, створіть запит на відображення групи і назви всіх товарів, на закупівлю яких укладено не більше однієї угоди (рис. 6).



Назва
йогурт з наповнювачем
лапша
макарони
морозиво
рис
спагеті
молоко
гречка
зелений

Рис. 6

10. Доповніть таблицю **Угоди** полем **Загальна вартість**, в якому б автоматично підраховувалася вартість товару, придбаного за цією угодою (див. рис. 7).



Код угоди	Код товару	Кількість товару	Ціна товару	Дата	Вид розрахунку	Оплата	Загальна вартість
1	1	1230	8	22.12.2020	безготівковий	<input checked="" type="checkbox"/>	9840
2	6	2301	15,23	05.10.2021	безготівковий	<input checked="" type="checkbox"/>	35044,23
3	4	2365	7	14.10.2021	готівкою	<input type="checkbox"/>	16555
4	7	235	10,24	26.10.2021	безготівковий	<input checked="" type="checkbox"/>	2406,4
5	7	256	12,35	30.10.2021	готівкою	<input type="checkbox"/>	3161,6

Рис. 7

11. Збережіть базу даних.

Зробіть висновки стосовно особливостей створення таблиць і запитів у базі даних Access.

## § 27. ФОРМИ

Уведення даних у табличному поданні електронної БД зручне для спеціалістів і може бути незрозумілим для недосвідченого користувача. Щоб робота з даними була схожою на звичні багатьом користувачам паперові аналоги, застосовують електронні форми. Тепер ви навчитеся створювати форми роботи з даними та встановлювати і налаштовувати кнопки поширених дій над даними.

### Створення форми



**Форма** — це інструмент, який являє собою зручний бланк для

перегляду, заповнення і редагування вмісту таблиць та перегляду результатів запитів.

Після завантаження БД для створення форми можна скористатися одним із засобів, доступних у групі інструментів **Форми** на вкладці **СТВОРЕННЯ**.

Створення форми в режимі майстра форм дає хороший результат, і саме цей режим буде далі розглянуто. Цей режим ставить детальні питання про записи, поля, макет, формат і створює форму на основі відповідей. Доопрацювати і відредагувати форму можна в режимі конструктора.

Для створення форми в режимі майстра форм потрібно:

- 1) натиснути кнопку **Майстер форм** (рис. 27.1);

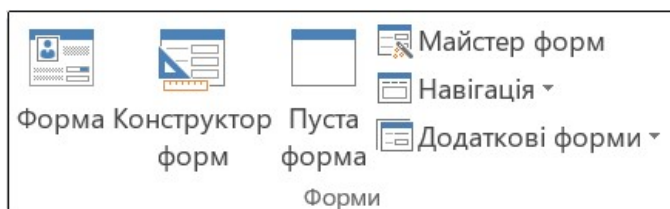


Рис. 27.1

- 2) у вікні **Майстер форм** вибрати потрібну таблицю або запит та кнопкою «<>» вибрати поля, які міститиме форма;

- 3) у наступному вікні вибрати варіант розміщення об'єктів на формі;

- 4) змінити, якщо потрібно, ім'я форми та вибрати дії (відкрити форму для роботи чи продовжити розробку в режимі конструктора), після чого натиснути кнопку **Готово**.

У результаті виконаних дій буде збережена форма і під заголовком **Форми**, що в **Області переходів** у вікні **Усі об'єкти**, з'явиться ім'я нової форми (рис. 27.2).

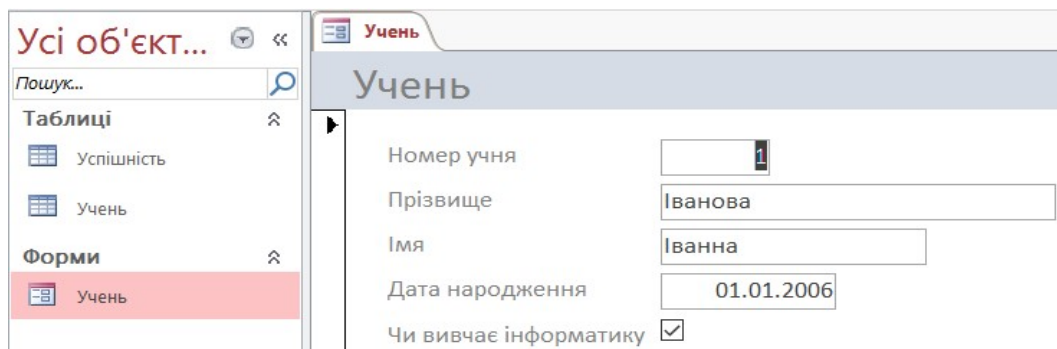


Рис. 27.2

### Редагування форми

Розміри вікна форми, розташування, розміри і налаштування її об'єктів змінюють у **Режимі розмітки** (макета) або **Конструктора**, які вмикаються відповідними командами контекстного меню форми або кнопкою **Вигляд**, що в лівому верхньому куті вкладки **Основне** (рис. 27.3).

Форма може містити **Текстові поля**, **Підписи**, **Кнопки**, **Прапорці** тощо, які можна додатково встановити на формі із вкладки **КОНСТРУКТОР** (рис 27.4).

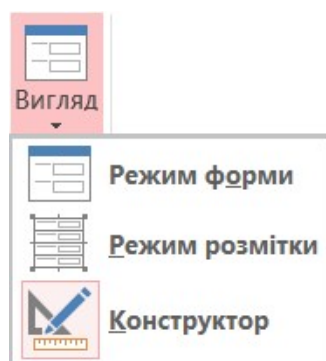


Рис. 27.3

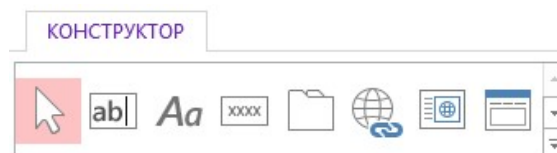


Рис. 27.4

### Редагування підписів

Підписи не виводять даних БД, а лише пояснювальний текст (написи, назви, пояснення тощо).

Зміст підписів можна змінити на інший, а також встановити потрібний формат (колір, написання, вирівнювання, заливку тощо) інструментами групи **Форматування тексту** на вкладці **ОСНОВНЕ**.

Можна додати нові підписи кнопкою **Aa** (див. рис. 27.4) або видалити наявні. Видалення **Підписів** (як і інших об'єктів форми) здійснюється клавішею **Delete** і відновлюється кнопкою ↶. Редагування чи видалення **Підпису** не впливає на працездатність форми, на існуючі імена табличних полів і зміст наявних даних.



## Редагування текстових полів даних

Текстові поля в режимі форми і режимі розмітки виводять дані, а в режимі конструктора — назву поля таблиці чи запиту, звідки ці дані отримані (рис. 27.5).

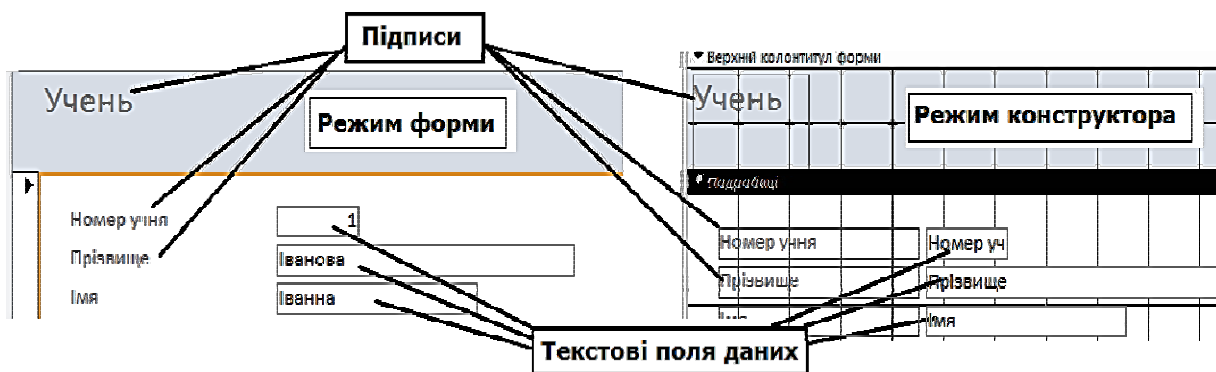


Рис. 27.5

Зміст даних у режимі розмітки чи конструктора змінити неможливо. Але можна, як і для написів, здійснити форматування текстів у них, а також змінити розташування і розміри самих Текстових полів.

Видалення текстового поля з даними призводить до видалення пов'язаного з ним підпису та втрати доступу до відповідної інформації із вікна форми.

## Оформлення форми

Крім області даних форма може мати заголовок та примітку, на яких розміщують як корисну інформацію, так і елементи дизайну. Ці частини форми залишаються видимими, навіть якщо для перегляду області даних необхідна смуга прокрутки. Наприклад, для стрічкової форми в області заголовка зручно розмістити назви стовпців.

Щоб змінити зовнішній вигляд форми, можна виконати такі дії:



- 1) відкрити форму в режимі конструктора;
- 2) змінити за необхідності розміри вікна форми, для чого встановити вказівник миші на межу і при появі двонаправленої стрілки перетягнути її до отримання потрібного розміру вікна;
- 3) розширити за потреби область заголовка (примітки), перетягнувши її нижню межу;
- 4) додати, якщо вважаєте за потрібне, зображення, для чого скористатися кнопкою **Вставити зображення**, що на вкладці **КОНСТРУКТОР** у групі **Елементи керування**.



## Додавання кнопок

Часто, для розширення можливостей користування формою, на ній створюють додаткові елементи керування: кнопки, списки, гіперпосилання тощо.

На прикладі створення кнопки видалення існуючого запису розглянемо алгоритм додавання до форми такого елемента:

- 1) відкрийте форму в режимі конструктора;
- 2) встановіть (якщо не встановлено) прапорець  **Застосувати майстри елементів керування** (рис. 27.6);
- 3) виберіть елемент керування **Кнопка** , що на вкладці **Конструктор** у групі **Елементи керування** (рис. 27.6);

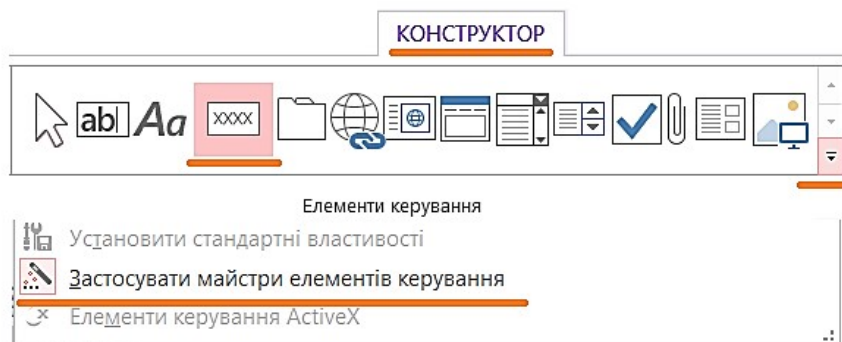


Рис. 27.6

- 4) в обраному місці форми намалюйте мишкою майбутню кнопку (прямокутник);

- 5) у вікні **Майстер кнопок** оберіть категорію: **Операції з записами** та дію: **Видалити запис** (рис. 27.7);

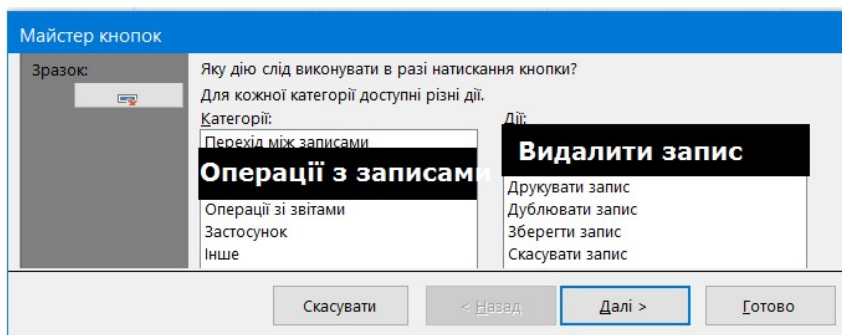


Рис. 27.7

- 6) у наступному вікні оберіть, що відобразатиметься на кнопці (**Текст** чи **Зображення**) і за необхідності змініть текст чи зображення, на майбутній кнопці;

7) в останньому вікні натисніть кнопку **Готово**, залишивши програмне ім'я кнопки без змін.

Створена і налаштована кнопка автоматично збережеться і працюватиме на формі лише в режимі форми. Для запобігання випадковому видаленню запису дію кнопки потрібно буде підтвердити в діалоговому вікні.

### Стандартні елементи користування формою

Для перегляду і пошуку записів унизу форми знаходиться рядок з елементами керування (рис. 27.8).

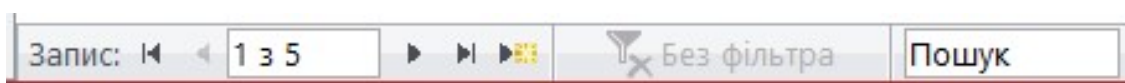


Рис. 27.8

Розглянемо дію його елементів:

Елемент	Дія
	Перехід до першого запису
	Перехід до останнього запису
	Перехід до наступного запису
	Перехід до попереднього запису
	Створення нового запису
<input type="text" value="1 з 5"/>	Параметри поточного запису (тут: перший із п'яти)
<input type="text" value="Пошук"/>	Рядок пошуку (автоматично відшукує уведені значення в даних)
	Інформує про наявність увімкненого фільтра

### Питання для самоконтролю

1. Для чого використовуються форми?
2. Як створити форму за допомогою майстра форм?
3. Як викликати режим конструктора для редагування форми?
4. Як змінити параметри шрифту окремого поля?
5. Як перейти від редагування до користування формою?
6. Як додати кнопку або інший елемент керування на форму?

## Вправа 27

Виконати створення форм.

1) Запустіть СКБД **Access** та відкрийте створену раніше базу даних **Інформатика**.

2) Додайте до таблиці **Успішність** текстове поле з ім'ям **Рівень** та збережіть таблицю.

3) Створіть форму із відображенням полів: **Прізвище, Імя, Тематична, Рівень**.

4) Встановіть однакові розміри текстових полів форми.

5) Змініть заголовок форми на «Відомості про успішність учнів з інформатики» та відцентруйте його.

6) Вирівняйте усі написи в **Текстових полях** і **Підписах** по лівому краю та зробіть їх синіми.

7) Додайте до форми довільний малюнок.

8) Текстове поле **Рівень** перетворіть на **Поле зі списком**. Для цього в контекстному меню текстового поля **Рівень** виберіть команди: **Перетворити елемент на** → **Поле зі списком**.

9) Додайте список даних до зміненого поля. Для цього:

а) у контекстному меню текстового поля **Рівень** виберіть команду **Властивості**;

б) у вікні **Властивості**, що справа, на вкладці **Дані** в рядку **Тип джерела даних** замість **Таблиця чи запит** встановіть **Список значень**;

в) у вікні **Властивості** на вкладці **Дані** в рядок **Джерело рядків** впишіть через крапку з комою рівні успішності: **Початковий; Середній; Достатній; Високий**;

г) збережіть форму.

10) Відкрийте форму в режимі **Форми** та заповніть текстові поля **Рівень** для усіх п'яти записів БД.

11) У режимі **Форми** додайте ще один (шостий) запис до бази даних та навчіться користуватись стандартними елементами керування форми (див. рис. 27.8).

12) Створіть на формі кнопку для збереження записів.

## Комп'ютерне тестування

Виконайте тестове завдання 27 з автоматичною перевіркою результату.

## § 28. ЗВІТИ

Відкриті таблиці, запити та форми бази даних **Access** можна роздрукувати за допомогою команд: **Файл** → **Друк**. Але часто виникає потреба в особливому розміщенні написів та текстових полів даних, наданні додаткових пояснень, додаванні заголовків, діаграм, малюнків, номерів сторінок тощо. Тут у пригоді стануть звіти, з якими ви познайомитеся у цьому параграфі.

### Структура звітів



**Звіт** — це підготовлений до публікації чи друку електронний документ.

У режимі конструктора звіту, який відкривається кнопкою **Конструктор звітів**, що на вкладці **СТВОРЕННЯ** в групі **Звіти**, відображається його стандартна (порожня) структура, яка складається із колонтитулів сторінок і області **Подробиці для розміщення даних**. У **Режимі розмітки** (рис. 28.1) області **Колонтитулів** і **Подробиць** виділяються помаранчевим кольором при клацанні по них.

**Колонтитули сторінок**, як і в інших офісних документах (**Word**, **Excel** тощо), можуть містити номер сторінки та інші службові дані.

До звіту можна включити заголовок, що розташовується зверху усього звіту (над верхнім колонтитулом сторінки) і може містити автоматично відформатовані емблему, назву, поточну дату і час, які додаються до заголовку відповідними кнопками на вкладці **КОНСТРУКТОР** у групі **Колонтитули** (рис. 28.2).

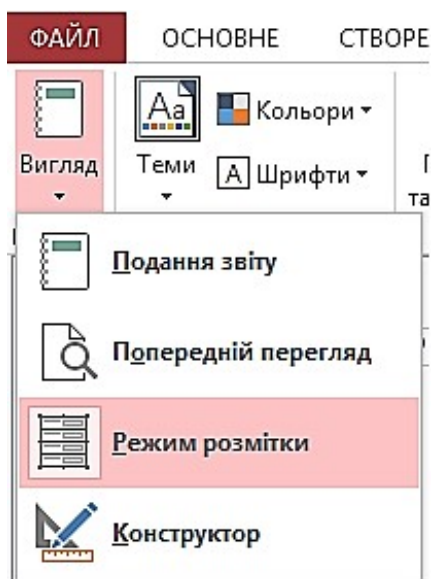


Рис. 28.1

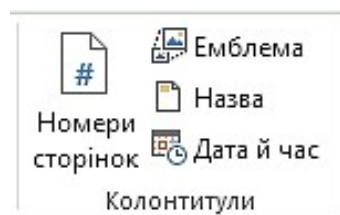


Рис. 28.2

Звіт може містити *верхній* і *нижній колонтитули* групи записів, де розміщують елементи керування, або узагальнену інформацію групи даних (суму, кількість тощо).

Верхній колонтитул сторінки може містити заголовки стовпців, які є на сторінках під час перегляду чи друку.

### Створення звіту об'єкта

Якщо клацнути по імені таблиці, запиту чи форми в **Області переходів** на панелі **Усі об'єкти**, а потім кнопку **Звіт** (рис. 28.3), що на вкладці **ОСНОВНЕ** у групі **Звіти**, то з'явиться зображення звіту вибраного об'єкта у **Режимі розмітки**. Цей звіт міститиме усі поля вибраного об'єкта, але його можна в подальшому змінювати, редагувати і формувати, надаючи бажаного вигляду.

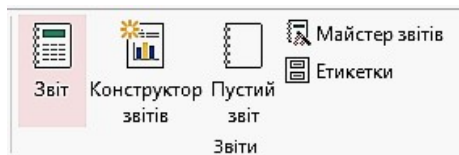


Рис. 28.3

### Створення порожнього звіту

Звіти зручно використовувати для друку додатків, атестатів, свідоцтв тощо. Як приклад розглянемо створення звіту для друку грамот (рис. 28.4).



Рис. 28.4

Конструювання такого звіту полягає у створенні фонового малюнка, додаванні та розташуванні в потрібних місцях підписів «*Нагороджується*» та «*за відмінні успіхи у вивченні інформатики*», а також у додаванні

текстових полів із даними про прізвища та ім'я учнів, що мають тематичні оцінки 12 балів.

Розглянемо алгоритм створення звіту.

#### 1. Налаштування сторінки звіту

1) Натисніть кнопку **Пустий звіт**, що на вкладці **СТВОРЕННЯ** у групі **Звіти**.

2) На вкладці **Параметри сторінки** налаштуйте розмір сторінки (наприклад, **A4**) і її поля.

3) У режимі конструктора перетягуванням меж встановіть нульові колонтитули.

#### 2. Встановлення зображення

4) Відскануйте (або завантажте з інтернету) зображення грамоти.

5) Кнопкою **Вставити зображення**, що на вкладці **КОНСТРУКТОР** у полі **Елементи керування**, вставте зображення грамоти, обравши потрібне зображення та клацнувши у верхньому лівому кутку звіту.

6) У контекстному меню зображення командою **Розташування** встановіть зображення **На задній план**.

7) У контекстному меню зображення виберіть команду **Властивості** та на **Аркуші властивостей** (список **Формат**) встановіть **Вирівнювання зображення** (**Вгорі ліворуч**) та **Ширину** і **Висоту** зображення згідно з розмірами сторінки (наприклад, для формату **A4** — Ширина: 21 см, Висота: 29,7 см).

8) У режимі конструктора, за необхідності, підтягуванням меж звіту встановіть його розміри згідно з розмірами сторінки (або зображення).

#### 3. Додавання підписів

9) Додайте два елементи **Підпис**, розмістіть їх (стрілками на клавіатурі або перетягуванням мишею) у потрібному місці, зробіть написи, відредагуйте і відформатуйте їх.

#### 4. Додавання текстових полів із даними

10) Натисніть кнопку **Додавання наявних полів**, що на вкладці **КОНСТРУКТОР** у групі **Знаряддя** (рис. 28.5).

11) На панелі **Список полів** розгорніть список **Учень** та подвійним клацанням додайте до звіту поля **Прізвище** та **Імя**.

12) Видаліть клавішею **Delete** **Підписи**, які знаходяться ліворуч від доданих **Текстових полів з даними** (рис. 28.6), та відформатуйте тексти в **Текстових полях** з прізвищем та ім'ям.

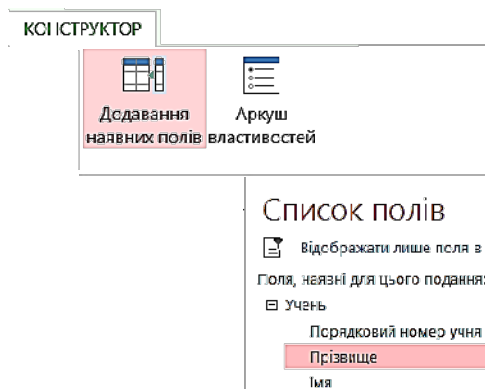


Рис. 28.5



Рис. 28.6

## 5. Відбір відмінників

13) Натисніть кнопку **Додавання наявних полів** (див. рис. 28.5) та в контекстному меню наявного поля (наприклад, **Прізвище**) оберіть команду **Редагувати джерело записів** (рис. 28.7).

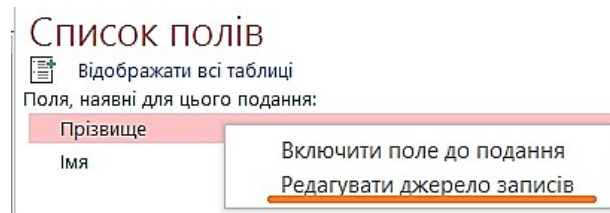


Рис. 28.7

14) У **Побудовнику запитів** додайте до схеми таблицю **Успішність**, із якої на бланк запиту додайте поле **Тематична**, і встановіть **Критерій 12** та зніміть прапорець **Відображення** (рис. 28.8).

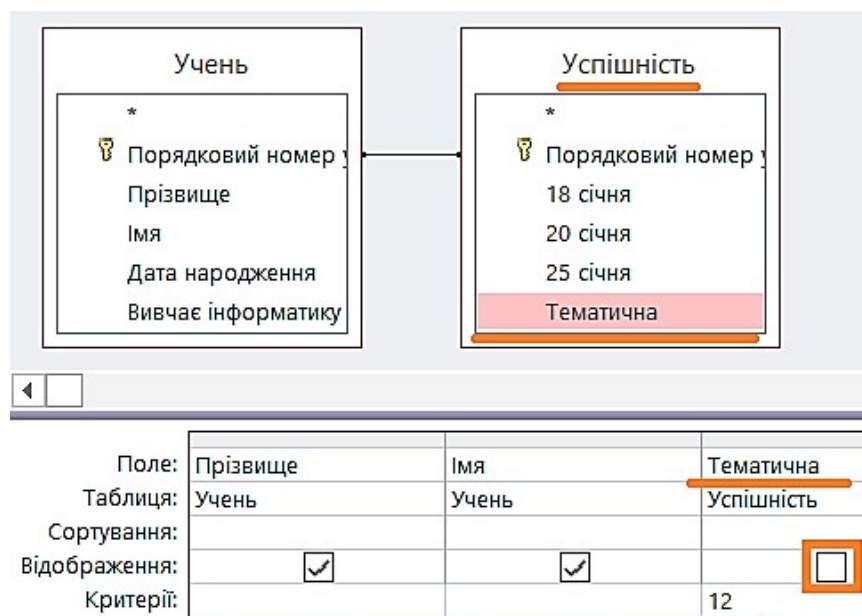


Рис. 28.8

15) Збережіть **Побудовник запитів** та створений **Звіт**.

Звіт готовий. Кількість його сторінок дорівнює кількості відмінників у таблиці **Учень**. Тепер можна видалити зображення грамоти і друкувати звіт на пустих бланках грамот або, за необхідності, друкувати грамоти разом із зображеннями на порожніх аркушах.

### Створення звіту за допомогою майстра звіту

Майстер звітів, на відміну від режиму конструктора чи пустого звіту, автоматично розташовує обрані елементи і раціонально форматує їх.

Для створення звіту за допомогою майстра звітів потрібно:

1) вибрати команду **Майстер звітів** (рис. 27.3) на вкладці **СТВОРЕННЯ** у групі **Звіти** — відкриється вікно **Майстер звітів**;

2) у полі **Таблиці та запити** вибрати таблицю або запит — джерело даних для звіту (наприклад, **Таблиця: Учень**);

3) за допомогою кнопки «>» вибрати потрібні поля або скористатися кнопкою «>>>» для вибору всіх полів і натиснути кнопку **Далі** — з'явиться вікно, призначене для задання рівнів групування;

4) для прийняття рівня, заданого за мовчазною згодою, натиснути кнопку **Далі**;

5) вибрати тип впорядкування та натиснути кнопку **Далі**;

6) вибрати вид макета (наприклад, **Табличний**) та орієнтацію аркуша (наприклад, **Альбомна**) і натиснути кнопку **Далі**;

7) у вікні з фінішним прапорцем ввести назву звіту **Учні** та натиснути кнопку **Готово**.

У результаті виконаних дій з'явиться звіт **Учні**.

### Редагування й оформлення звіту

Редагування й оформлення звіту виконується в режимі конструктора подібно до того, як це робилось вище.

Наприклад, для редагування заголовка **Учні** необхідно виділити текст і на панелі інструментів виконати відповідні дії.

Звіт може містити таке:

- заголовок — друкується на початку першої сторінки;
- верхній і нижній колонтитули — друкуються вгорі та внизу кожної сторінки;
- примітку — друкується в кінці останньої сторінки.

Для збереження звіту слід натиснути кнопку **Зберегти** на панелі інструментів; закрити звіт кнопкою **✕**.



### Друкування звіту

Перед друком необхідно ретельно перевірити параметри сторінки, зокрема поля й орієнтацію сторінки. **Access** зберігає параметри сторінки разом зі звітом, тому їх потрібно налаштувати лише раз. Ви зможете змінити їх згодом відповідно до своїх потреб.

Щоб змінити параметри сторінки, потрібно виконати такі дії:

1) відкрити звіт у режимі **Попередній перегляд** (див. рис. 28.2) — внесені зміни негайно відбиваються на зображенні на вкладці **Попередній перегляд**, у групах **Макет сторінки** та **Розмір сторінки** вибрати орієнтацію сторінки (**Книжкова** або **Альбомна**), натиснути кнопку **Розмір**, щоб установити розмір паперу, кнопку **Поля**, щоб налаштувати поля, тощо;

2) внісши потрібні зміни, за допомогою кнопок переходу переглянути кілька сторінок, щоб упевнитись у відсутності проблем із форматуванням на наступних сторінках.

Для передавання звіту на принтер потрібно виконати такі дії:

1) відкрити звіт у будь-якому поданні або вибрати його в області переходів;

2) вибрати **Файл** → **Друк** → **Друк** — відкриється діалогове вікно **Друк**;

3) налаштувати запропоновані параметри, зокрема принтер, діапазон друку та кількість копій;

4) клацнути кнопку **ОК** — принтер надрукує звіт.

### Експорт даних до Word

До **Microsoft Word** можна експортувати об'єкти бази даних: таблиці, запити, форми або звіти. **Access** створює файл **Microsoft Word** із копією даних об'єкта у форматі **RTF** у вигляді таблиці. Тому коли екпортуєте звіт, документ **Word** максимально нагадує звіт.

Щоб підготувати дані до екпортування до **Word**, потрібно:

1) відкрити вихідну базу даних **Access**;

2) в області переходів вихідної бази даних вибрати об'єкт із даними, які потрібно експортувати;

3) переконатися, що в об'єкті немає помилок.

Для екпортування даних потрібно:

1) на вкладці **ЗОВНІШНІ ДАНІ** в групі **Експорт** натиснути кнопку **Додатково**, щоб відкрити розкритий список параметрів (рис. 28.9);

2) вибрати **Word** — відкриється майстер **Експорт** → **RTF**-файл;

3) у вікні майстра **Експорт** → **RTF**-файл указати ім'я кінцевого файлу;

4) встановити прапорець **Відкрити кінцевий файл після завершення операції експорту**;

5) якщо не вибрано жодного запису, встановити прапорець **Експортувати лише вибрані записи недоступний (сірий)**;

6) клацнути кнопку **ОК**.

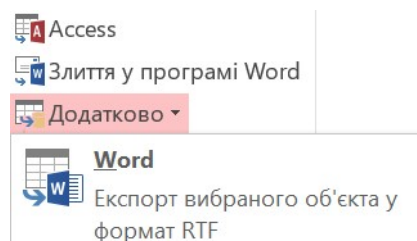


Рис. 28.9

Якщо кінцевий документ створено, вам буде запропоновано натиснути кнопку **Так**, щоб перезаписати файл. Клацніть **Ні**, щоб змінити назву кінцевого файлу, а потім знову натисніть кнопку **ОК**.

**Access** експортує дані та відкриває кінцевий документ у **Word** залежно від параметрів експорту, вказаних у майстрі.

### Питання для самоконтролю

1. Яке призначення звітів?
2. Яку структуру має стандартний звіт?
3. Як швидко створити звіт обраної таблиці, запиту чи форми?
4. У чому полягає конструювання звіту для заповнення бланків, грамот, атестатів тощо?
5. Як створити звіт за допомогою майстра?
6. Яким чином експортувати звіт до **Word**?

### Вправа 28

Виконати створення звітів.

1) Запустіть СКБД **Access** і відкрийте створену раніше базу даних **Інформатика**.

2) Створіть і збережіть звіт, який складається з усіх полів таблиці **Учень**.

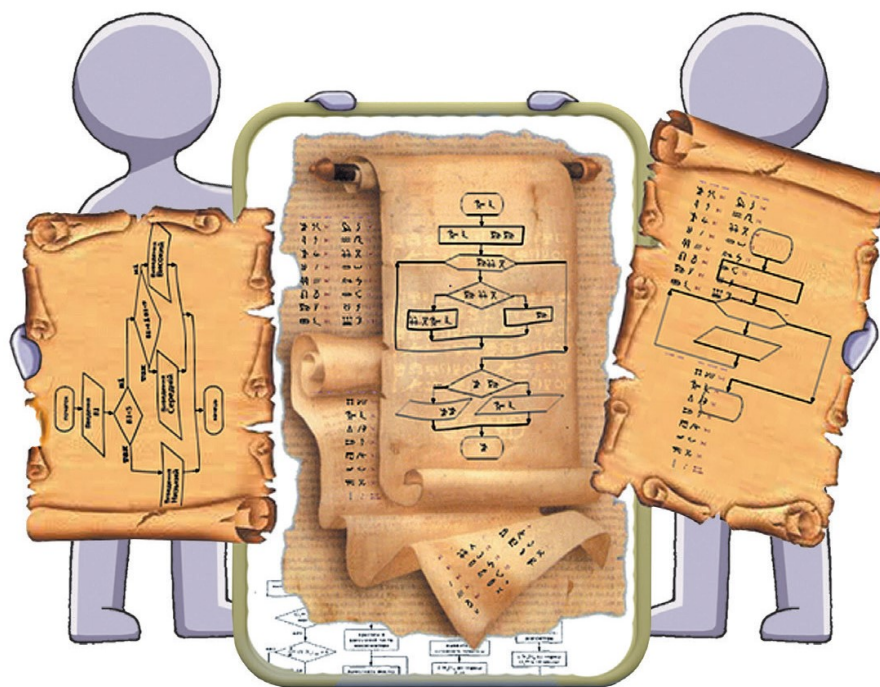
3) Завантажте з інтернету та збережіть файл із зображенням незаповненої грамоти.

4) За алгоритмом, наведеним у підручнику, створіть та збережіть разом з малюнком звіт для заповнення (друку) грамот, якими нагороджуватимуться відмінники.

### Комп'ютерне тестування

Виконайте тестове завдання 28 з автоматичною перевіркою результату.

## РОЗДІЛ 5 АЛГОРИТМИ ТА ПРОГРАМИ



§ 29. Поняття одновимірного масиву. Списки

§ 30. Функції та методи списків

§ 31. Опрацювання елементів списку

§ 32. Алгоритми впорядкування масиву

§ 33. Об'єкти класу **Listbox**

§ 34. Візуалізація елементів табличної величини

Практична робота 8. Складання і виконання алгоритмів знаходження сум і кількостей значень елементів табличних величин за заданими умовами

§ 35. Поняття двовимірного масиву. Вкладені списки

Практична робота 9. Робота з двовимірними масивами у програмі з графічним інтерфейсом

§ 36. Опрацювання двовимірних масивів даних

§ 37. Поняття складності алгоритмів

## ПОВТОРЮЄМО

У 5–8 класах ви опанували *основи мови програмування Python*, знаєте, як реалізувати базові алгоритмічні структури, і вмієте створювати програми з використанням розгалужень і повторень.

Ви розширили уявлення про *типи величин* і *алгоритми опрацювання даних різних типів* та розробляли програми з графічним інтерфейсом користувача, створюючи об'єкти класів графічних компонентів. Ви також навчилися задавати властивості найуживаніших віджетів та програмувати обробку подій для цих об'єктів.

Ви оволоділи прийомами роботи в інтегрованому середовищі розробки **PyCharm**, знайомі з можливостями інтелектуального введення та автодоповнення коду, засобами для аналізу коду тощо.

1. Назвіть основні характеристики величини.
2. Щовизначає тип величини?
3. З якими типами даних ви працювали?
4. Наведіть приклади величин простих типів.
5. Як можна змінити значення величини?
6. Які базові алгоритмічні структури використовуються для конструювання алгоритмів?

У цьому розділі ви продовжите вивчення основ програмування мовою **Python**, ознайомитесь із поняттям масиву даних і алгоритмами опрацювання табличних величин.

## § 29. ПОНЯТТЯ ОДНОВИМІРНОГО МАСИВУ. СПИСКИ

Ви вже стикались із необхідністю зберігання наборів значень, об'єднаних за певною ознакою. Наприклад, у комп'ютерній грі можуть зберігатися координати персонажів на екрані, таблиці рекордів, ведення рахунку тощо. В обчислювальних програмах виникає потреба зберігати набори чисел: значення середньодобових температур за тиждень, відстаней між об'єктами, кількісних показників певного процесу тощо. Як вам вже відомо, для збереження сукупностей даних у програмуванні існують структуровані типи, або контейнери. Контейнер можна порівняти з картотекою — він зберігає набір значень у певному порядку.

**Складені або структуровані типи** — це типи даних, які мають внутрішню структуру та можуть бути сконструйовані з простих типів даних. Змінна або константа структурованого типу має декілька компонент, об'єднаних спільним ім'ям.

### Створення списку



**Список (*list*)** — це упорядкований набір об'єктів різних типів

(чисел, рядків, списків тощо), який можна змінювати.

**Значення списку** вкладаються у квадратні дужки [ ] та відокремлюються одне від одного за допомогою коми. У мові **Python** списки використовуються для зберігання **масивів даних**.

**Одновимірний масив** — це пронумерована послідовність значень одного типу, що мають спільне ім'я.

Хоча **Python** дозволяє зберігати у списку елементи різних типів, наша мета — навчитися опрацьовувати масиви даних, тобто сукупності однотипних значень, об'єднаних за певною ознакою.

**Створити порожній список** можна двома способами:

- за допомогою функції **list()**: `a = list()`
- використовуючи квадратні дужки: `a = [ ]`

**ПРИКЛАД 1.** Створити об'єкт **animal** типу список і надати йому значення:

```
animal = ["Кіт", "Собака", "Миша", "Хом'як"]
```

0                      1                      2                      3

Список **animal** →

"Кіт"	"Собака"	"Миша"	"Хом'як"
-------	----------	--------	----------

Перевірити, чи є в списку певний елемент, допоможе ключове слово **in**.

**ПРИКЛАД 2.** Визначити, чи є в списку `letters` елемент `'a'`:

```
if 'a' in letters: print('так')
else: print("символу 'a' в списку немає")
```

Додати елемент до списку можна так:

за допомогою методу `append()`

Додати до списку `a = [2, 6, 4]` елемент 10:

```
>>> a.append(10)
>>> print(a)
[2, 6, 4, 10]
```

за допомогою операції об'єднання

Заповнити список числами від 0 до 4:

```
>>> a = []
>>> for i in range(5):
        a = a+[i]
>>> a
[0, 1, 2, 3, 4]
```

**ПРИКЛАД 3.** Створення списку з 5 цілих чисел, що вводяться з клавіатури.

```
a = []
for i in range(5):
    n = int(input(str(i)+'? '))
a.append(n)
print(a)
```

```
0? 3
1? 5
2? 2
3? 15
4? 90
[3, 5, 2, 15, 90]
>>>
```

**ПРИКЛАД 4.** Заповнення списку 5 випадковими числами.

```
from random import*
a = []
for i in range(5):
    a = a+[randint(1, 10)]
print(a)
```

Результат роботи програми може бути таким: `[9, 1, 6, 3, 8]`

### Генератори списків



**Генератор списків** — це синтаксична конструкція, що

призначена для створення списків і заповнення їх значеннями за певними правилами.

**Синтаксис генератора списків:**

```
list_variable = [<вираз> for x in iterable]
```

де `x` — значення, які вибираються з послідовності `iterable`, `<вираз>` — вираз, що обчислюється для кожного значення `x` і використовується як елемент

списку `list_variable`. Конструкція береться у квадратні дужки, що означає, що результатом роботи генератора є список.

**ПРИКЛАД 5.** Створити список з перших 10 натуральних чисел.

```
>>> a = [i for i in range(1, 11)]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

**ПРИКЛАД 6.** Створити список `a` з квадратів елементів списку `b`.

```
>>> b = [2, -2, 4, -4, 7, 5]
>>> a = [i**2 for i in b]
>>> b
[4, 4, 16, 16, 49, 25]
```

**ПРИКЛАД 7.** Створити список `a` з перших літер елементів списку `animal`.

```
>>> animal = ["Кіт", "Собака", "Миша", "Хом'як"]
>>> a = [x[0] for x in animal]
>>> a
['К', 'С', 'М', 'Х'].
```

### Доступ до елементів списку

Доступ до елемента списку можна отримати за його індексом (номером у списку). Нумерація елементів починається з нуля.

Завдяки нумерації можна визначати довжину списку (кількість елементів у списку), робити зрізи.

**ПРИКЛАД 8.** Дано список `a = [2, 15, 3, -5, 10]`. Проаналізуємо результати виконання в консолі дій зі списком.

	Дія	Оператор	Результат виконання
1.	Визначити довжину списку	<code>len(a)</code>	5
2.	Вивести елемент списку з індексом 3	<code>a[3]</code>	-5
3.	Вивести перші 3 елементи списку	<code>a[0: 3]</code>	[2, 15, 3]
4.	Вивести елементи списку, починаючи з елемента з номером 3 і до кінця	<code>a[3:]</code>	[-5, 10]

5.	Вивести зріз списку довжиною 1 елемент	<code>a[3: 4]</code>	<code>[-5]</code>	
----	--	----------------------	-------------------	--

**ПРИКЛАД 9.** Замінити нулями перші 3 елементи списку `a`:

```
a = [2, 15, 3, -5, 10]
```

```
a[0: 3] = [0]*3
```

```
print(a) # Буде виведено [0, 0, 0, -5, 10]
```

Отже, списки є гнучким і потужним інструментом для збереження сукупностей даних у певному порядку.

### Питання для самоперевірки

- Обговоріть спільні ознаки та відмінності між рядками та списками.
- У змінній `letters` міститься список `['a', 'b', 'c', 'd']`. Обчисліть значення виразів:
  - `letters[int('2'*2)//11]`
  - `letters[3]`
  - `letters[-1]`
  - `letters[::2]`
  - `letters[1: 3]`
  - `letters[:2]`
- Створено список `a = [2, 15, 3, -5, 10]`. Визначте вміст списку `a` після виконання операторів:
  - `a = a+[11]`
  - `a[3] = 0`
  - `a[0], a[4] = a[4], a[0]`
- Збережіть імена кількох своїх друзів у списку з ім'ям `names`. Виведіть ім'я кожного друга, звернувшись до кожного елемента списку.
- За допомогою генератора списків заповніть список квадратами перших 10 натуральних чисел.
- За допомогою генератора списків заповніть список `a` непарними числами в діапазоні від 1 до 20.

### Вправа 29

Створити програму опрацювання списку рядкових даних.

- Завантажте середовище `PyCharm`. Створіть `Python file` з назвою `Список назв`. Поміркуйте над тим, яку інформацію можна було б зберігати у



списку. Створіть список **nazva** з 6 рядкових величин: назв професій, видів спорту, назв країн, видів рослин тощо.

2) Змінній **a** надайте значення третього елемента списку **nazva**. Виведіть значення **a** на екран.

3) Виведіть на екран перші 3 елементи списку **nazva**.

4) Програмно додайте до списку **nazva** ще два значення. Виведіть список на екран.

5) Вставте на 3-тє місце в списку **nazva** елемент **Нове значення**.

6) За допомогою генератора списків створіть список **letters** із перших літер елементів списку **nazva**. Виведіть список **letters** на екран.

### Комп'ютерне тестування

Виконайте тестове завдання 29 з автоматичною перевіркою результату.

## § 30. ФУНКЦІЇ ТА МЕТОДИ СПИСКІВ

Для розв'язування багатьох задач опрацювання сукупностей однотипних даних мова **Python** має вбудовані функції та методи списків. Із методом **append()** ви вже знайомі. Застосування функцій та методів списків допомагає легко і швидко розв'язувати складні прикладні завдання.

### Методи списків

Виклик методів має загальний синтаксис:

**Назва\_списку. Назва\_методу(<параметри>)**

Зауважимо, що методи списків змінюють сам список, тому результат виконання не потрібно зберігати в іншу змінну.

Розглянемо на прикладі списку **a = [2, 15, 3, -5, 3]** деякі методи списків:

Метод	Призначення	Приклад	Результат
<b>list.insert(i, x)</b>	Вставляє на <b>i</b> -те місце елемент зі значенням <b>x</b>	<b>a.insert(2, 78)</b>	<b>a = [2, 15, 78, 3, -5, 3]</b>
<b>list.remove(x)</b>	Видаляє зі списку перший елемент зі значенням <b>x</b>	<b>a.remove(15)</b>	<b>a = [2, 3, -5, 3]</b>
<b>list.index(x)</b>	Повертає індекс першого елемента зі значенням <b>x</b>	<b>k = a.index(0)</b>	<b>k = 2</b>

Метод	Призначення	Приклад	Результат
<b>list.count(x)</b>	Повертає кількість елементів зі значенням x	<b>k = a.count(3)</b>	k = 2
<b>list.sort()</b>	Упорядковує список за зростанням	<b>a.sort()</b>	a = [-5, 2, 3, 3, 15]
<b>list.reverse()</b>	Перебудовує елементи списку у зворотному порядку	<b>a.reverse()</b>	a = [15, 3, 3, 2, -5]

**ПРИКЛАД 1.** Список **results** містить суми балів, набрані спортсменами в ході змагань. Визначити результати трьох переможців (див. рисунок):

```
results.sort()
results.reverse()
print(results[:3])
```

```
results = [2 3 7 15 1 9]
results.sort() → [1 2 3 7 9 15]
results.reverse() → [15 9 7 3 2 1]
                ↑
                зріз списку → results[:3]
```

Методи списків можуть допомогти опрацювати дані інших типів, наприклад рядки. Розглянемо таблицю:

Метод	Призначення	Приклад	Результат
<b>s.split(символ)</b>	Розбиття рядка s на список по роздільнику символ	s = "2 56 12 5" a = s.split() # За замовчуванням роздільником є пробіл	a = ['2', '56', '12', '5']
<b>list(рядок)</b>	Розбиття рядка на список символів	s = "2 56 12 5" a = list(s)	a = ['2', ' ', '5', '6', ' ', '1', '2', ' ', '5']
<b>c.join(список)</b>	Об'єднання списку в рядок із роздільником c між символами	a = ['2', '56', '12', '5'] s = '+'.join(a)	s = "2+56+12+5"

**ПРИКЛАД 2.** Упорядкувати за неспаданням числа, які утворюють рядок **s**.

```
s = "2 56 12 5"  
a = s.split()  
a.sort()  
s = ' '.join(a)  
print(s)                                # Надруковано '2 5 12 56'
```

### Функції для роботи зі списками

Функції для роботи зі списками використовують список як аргумент:

**Назва\_функції(назва\_списку)**

Більшість функцій, на відміну від методів, не змінюють сам список, а повертають певне значення. Розглянемо на прикладі списку **a = [1, 5, 7, 5, 31, -5]** деякі корисні функції.

Функція	Призначення	Приклад	Результат
<b>len(list)</b>	Повертає довжину списку	<b>k = len(a)</b>	k = 6
<b>max(list)</b>	Повертає значення найбільшого елемента	<b>m = max(a)</b>	m = 31
<b>min(list)</b>	Повертає значення найменшого елемента	<b>m = min(a)</b>	m = -5
<b>sum(list)</b>	Повертає значення суми елементів	<b>s = sum(a)</b>	s = 44
<b>del(list[n])</b>	Видаляє елемент з індексом <b>n</b>	<b>del(a[2])</b>	a = [1, 5, 5, 31, -5]

**ПРИКЛАД 3.** Видалити зі списку **a** найбільший елемент.

```
>>> a = [1, 5, 7, 31, -5]  
>>> m = max(a)           # m = 31  
>>> n = a.index(m)      # n = 3  
>>> del(a[n])           # a = [1, 5, 7, -5]
```

Розглянуті вбудовані функції опрацювання списків допомагають реалізувати такі типи задач, як пошук у списку елемента з конкретним значенням, сортування списку, знаходження суми елементів списку. Але для розв'язування задач, що потребують виконання дій над елементами, значення

яких відповідає заданій умові, слід застосовувати типові алгоритми опрацювання масивів, які ми розглянемо в наступному параграфі.

### Питання для самоперевірки

1. У змінній **a** міститься список [2, 8, 1, 4, 3, -5, 3]. Обчисліть значення виразів:

- а) **len(a)**
- б) **a.index(4)**
- в) **a.count(3)**
- г) **max(a)**
- г) **a.index(min(a))**
- д) **sum(a)**

2. У змінній **a** міститься список [2, 8, 1, 4, 3, -5, 3]. Якого значення набуває змінна **a** після виклику методів:

- а) **a.insert(3, 10)**
- б) **a.remove(4)**
- в) **a.sort()**
- г) **a.reverse()**
- г) **del(a[0])**

3. Створіть список із чисел, записаних у рядок "25 3 12 44 12 21".

4. Збережіть у списку назви декількох країн. Простежте за тим, щоб елементи в списку не зберігались в алфавітному порядку. Застосуйте до списку функції **sort()** і **reverse()**. Виведіть результат у консоль.

5. Заповніть список десятьма випадковими цілими числами в діапазоні [0, 10]. Визначте кількість чисел 5 у створеному списку.

6. У базі даних зберігаються номери співробітників і співробітниць, яким була виписана премія за місяць. Напишіть програму для перевірки, чи не потрапив дехто зі співробітників або співробітниць у цю базу двічі.

### Вправа 30

Створити програму опрацювання списку рядкових даних.

1) Відкрийте **Python file** з назвою **Список назв**.

2) Змінній **a** надайте значення найбільшого елемента списку **nazva**. Виведіть значення **a** на екран.

3) Замініть найбільший елемент списку рядком **'\*\*\*'**. Виведіть список на екран.

4) Упорядкуйте список **nazva** за зростанням. Виведіть список на екран.

5) Поміняйте місцями найменший та найбільший елементи списку **nazva**. Виведіть список на екран.

б) Видаліть зі списку **nazva** останній елемент. Виведіть список на екран.

### Комп'ютерне тестування

Виконайте тестове завдання 30 з автоматичною перевіркою результату.

## § 31. ОПРАЦЮВАННЯ ЕЛЕМЕНТІВ СПИСКУ

Розв'язування багатьох прикладних задач, що потребують виконання дій над елементами, значення або індекс яких відповідає заданій умові, базується на типових алгоритмах опрацювання списків, які передбачають послідовний перегляд елементів списку і виконання певних дій із кожним елементом.

### Опрацювання елементів списку

Якщо потрібно виконати дії тільки над елементами списку, що мають задану ознаку, слід послідовно перебрати елементи списку і перевірити кожний елемент на відповідність певній умові. Опрацювання елементів списку зручно виконувати в циклі, де параметр циклу послідовно набуває значень елементів списку або їх індексів.

Розглянемо це на прикладі виведення списку на екран:

Послідовно звертатися до елементів за їх індексами:	<b>a = [2, 5, 14, 62, -5]</b> <b>for i in range(len(a)):</b> <b>print(a[i])</b>
Перебрати всі наявні в списку елементи. Застосовується, коли не потрібно змінювати значення елементів списку	<b>for item in a:</b> <b>print(item)</b>

### Типові алгоритми опрацювання елементів списку

#### ▶ Заміна значень усіх елементів списку

*Алгоритм розв'язування задачі:* послідовно перебрати всі елементи списку і значення чергового елемента змінити за певним правилом (наприклад, змінити знак елементів на протилежний, надати всім елементам нульові значення, збільшити значення всіх елементів на певну величину).

**ПРИКЛАД 1.** Подвоїти значення елементів списку **a = [2, 5, 14, 62, -5]**.

**a = [2, 5, 14, 62, -5]**

```
for i in range(5):
```

```
    a[i] = a[i]*2      # Список набуває значення a = [4, 10, 28, 124, -10]
```

► Заміна значень із перевіркою умови

*Алгоритм розв'язування задачі:* послідовно перебрати всі елементи списку; якщо черговий елемент відповідає деякій умові, його значення змінити за певним правилом. Як умова може перевірятися значення елемента списку на кратність деякому числу, додатність, рівність нулю тощо. Може перевірятися також і значення індексу елемента списку (наприклад, змінюються елементи, що стоять на парних місцях).

**ПРИКЛАД 2.** Поділити на 2 парні елементи списку **a = [2, 5, -14, 62, -5]**.

```
if a[i]%2 == 0:
```

```
    a[i] = a[i]//2    # Список набуває значення a = [1, 5, 7, 31, -5]
```

**ПРИКЛАД 3.** Елементи списку **a = [2, 5, 14, 62, -5]**,

що мають парні індекси, замінити нулем.

```
a = [2, 5, 14, 62, -5]
```

```
for i in range(5):
```

```
    if i%2 == 0:
```

```
        a[i] = 0
```

```
print(a) # Буде надруковано [0, 5, 0, 62, 0]
```

```
=== RESTART: C:/U  
[0, 5, 0, 62, 0]
```

► Визначення кількості елементів із заданою властивістю

**ПРИКЛАД 4.** Визначити кількість додатних елементів списку **a = [2, 5, 14, 62, -5]**.

```
k = 0
```

```
# Лічильник елементів із заданою властивістю
```

```
for item in a:
```

```
    if item > 0:
```

```
        k = k+1
```

```
print('k =', k) # Буде надруковано k = 4
```

► Визначення найбільшого (найменшого) елемента списку

Хоча **Python** має вбудовані функції **max(list)**, **min(list)**, потрібно розуміти сутність алгоритму пошуку найбільшого (найменшого) елемента списку.

*Алгоритм розв'язування задачі:* припустити, що найбільшим є нульовий елемент списку, після чого послідовно перебрати елементи масиву,

починаючи з другого; якщо черговий елемент більший за максимальний, його значення запам'ятати як максимальне.

```
mx = a[0]      # Змінна для збереження найбільшого значення елемента
k = 0        # Змінна для збереження індексу найбільшого елемента
for i in range(1, len(a)): # Перебираємо елементи, починаючи з
першого
    if a[i]>mx:   # Якщо і-й елемент більший за mx,
        mx = a[i] # значення mx замінюється на значення a[i];
    k = i       # у змінній k запам'ятовується значення індексу i
```

► Обчислення суми (добутку) елементів із заданою властивістю

**ПРИКЛАД 5.** Знайти суму додатних елементів списку **a = [2, -3, 1, 6, -5]**.

```
suma = 0                # Змінна для збереження значення суми
for item in a:
    if item>0: suma = suma + item
print('suma = ', suma) # Буде надруковано suma = 9
```

**ПРИКЛАД 6.** Знайти добуток ненульових елементів списку **a = [2, 0, 2, 6, 0]**.

```
dobutok = 1            # Змінна для збереження значення суми
for item in a:
    if item != 0: dobutok = dobutok*item
print('dobutok = ', dobutok) # Буде надруковано dobutok = 24
```

Користуючись типовими алгоритмами опрацювання значень елементів списків, ви зможете створювати ефективні алгоритми розв'язування більш складних прикладних задач.

### Питання для самоперевірки

1. Дано список **a = [3, 8, 5, 7, 6]**. Виконайте дії з елементами списку.

а) **for i in range(5): a[i] = -a[i]**

г) **s = 0**

**for i in range(5):**

**s = s+a[i]**

б) **for i in range(5):**

г) **p = 1**

**for item in a:**

**if a[i]%3 == 0:**

**a[i] = a[i]//3**

**if a[i] <= 5: p = p\*a[i]**

в) **for i in range(5):**

д) **k = 0**

```
if i%2 == 0: a[i] = 0
```

```
for i in range(5):  
    if a[i]>5: k = k+1
```

2. Дано список **a = []**. Запишіть оператори для реалізації таких завдань:
- а) ввести з клавіатури значення елементів списку;
  - б) знайти добуток елементів списку;
  - в) знайти мінімальний елемент списку;
  - г) знайти кількість додатних елементів списку.

3. Створіть список **nazva**, який містить назви 5 країн світу. Сформууйте з елементів списку **nazva** і виведіть на екран речення за зразком: **Україна, Польща, Німеччина — це назви країн.**

4. Складіть програму для виконання завдання: список **numbers** заповнено 50 випадковими числами в межах 100. Визначте, скільки елементів списку відрізняються від останнього елемента.

### Вправа 31

Створити програму **Календар нагляду за погодою**. Проаналізувати список значень середньодобових температур за березень.

1) Завантажте модуль **random**. Створіть порожній список **t** і додайте до списку 31 випадкове число в діапазоні можливих березневих температур. Виведіть список на екран.

```
from random import*  
t = []  
for i in range(31):  
    t.append(randint(-10, 10))  
print(t)
```

2) Запишіть код для підрахунку кількості днів, коли температура була вище нуля. Виведіть знайдене значення з відповідним поясненням:

```
print(k, 'днів температура була вище 0')
```

3) Визначте число стрибків температур (сусідства двох чисел різних знаків):

```
k = 0  
for i in range(30):  
    if t[i]*t[i+1]<0: k += 1  
print('Стрибків температури було: ', k)
```

4) Встановіть дату найнижчої температури, виведіть знайдене значення.

5) Встановіть різницю між найменшим і найбільшим значеннями температур.



б) Знайдіть середнє значення температури за місяць. Виведіть дати, коли температура відрізнялася від середньомісячної не більше ніж на 10° С.

### Комп'ютерне тестування

Виконайте тестове завдання 31 з автоматичною перевіркою результату.

## § 32. АЛГОРИТМИ ВПОРЯДКУВАННЯ МАСИВУ

### Методи сортування масивів

Під час опрацювання таблиць часто виникає потреба впорядкувати дані в таблиці за деякою ознакою. Числові дані можна відсортувати за величиною (наприклад, розташування в масиві значень маси деталей за зростанням), рядкові дані — в алфавітному порядку (упорядкування списку учнів). **Сортування елементів масиву** — це впорядкування їх за деякою ознакою.

Клас **List** у **Python** має метод **sort()**, який упорядковує список за зростанням. Якщо потрібно упорядкувати елементи масиву за спаданням, після сортування застосовують метод **reverse()**.

**ПРИКЛАД 1.** Упорядкувати елементи списку **arr** за незростанням.

```
arr = [23, 12, 3, 45, 6, 7, 8, 4, 21, 81]
arr.sort()
arr.reverse()           # a = [81, 45, 23, 21, 12, 8, 7, 6, 4, 3]
```

Щоб зрозуміти сутність алгоритмів сортування, розглянемо два найпростіші методи сортування масиву. Нехай потрібно впорядкувати елементи списку **arr** із 10 елементів за неспаданням:

$$\mathbf{arr[1] \leq arr[2] \leq \dots \leq arr[10]}$$

#### ► Сортування вибором максимального елемента

Алгоритм сортування вибором полягає в пошуку на неопрацьованому зрізі списку максимального значення і подальшому обміні цього значення з останнім елементом неопрацьованого зрізу. На наступному кроці неопрацьований зріз зменшується на один елемент.

*Алгоритм сортування* (див. рисунок):

- Відшукати максимальний елемент у зрізі **arr[0: 9]**.
- Максимальний елемент цього зрізу поміняти місцями з **arr[9]**.
- Відшукати максимальний елемент у зрізі **arr[0: 8]**.
- Максимальний елемент цього зрізу поміняти місцями з **arr[8]**.

<...>

- Максимальний елемент зрізу `arr[0: 1]`. Поміняти місцями з `arr[1]`.

1	2	3	4	5	6	7	8	9	10
21	27	15	20	1	29	12	18	26	2
21	15	20	1	27	12	18	26	2	29
15	20	1	21	12	18	26	2	27	29
15	1	20	12	18	21	2	26	27	29
1	15	12	18	20	2	21	26	27	29
1	12	15	18	2	20	21	26	27	29
1	12	15	2	18	20	21	26	27	29
1	12	2	15	18	20	21	26	27	29
1	2	12	15	18	20	21	26	27	29
1	2	12	15	18	20	21	26	27	29

Реалізуємо алгоритм у вигляді функції. Список `arr` заповнимо 10 випадковими числами.

```
from random import randint
def sort_select():
    for k in range(len(arr)-1, 0, -1):
        # При кожній ітерації циклу переглядається зріз [0: k]
        mk = max(arr[0: k])
        # Знайшли максимальний елемент у зрізі arr[0: k]
        m = arr[0: k].index(mk)
        # Визначили індекс максимального елемента
        arr[k], arr[m] = arr[m], arr[k]
        # Максимальний елемент поміняли місцями з останнім у зрізі
arr = []
for i in range(10):      # Заповнення списку випадковими числами
    arr.append(randint(1, 30))
print(arr)
sort_select()           # Виклик функції сортування
print(arr)
```

Приклад виконання:

[21, 27, 15, 20, 1, 29, 12, 18, 26, 2]

[1, 2, 12, 15, 18, 20, 21, 26, 27, 29]

### ▶ Сортування обміном (метод бульбашки)



**Метод бульбашки** — це метод упорядкування списку шляхом

послідовного порівняння й обміну сусідніх елементів, якщо попередній елемент виявляється більшим за наступний.

У процесі виконання цього алгоритму елементи з великими значеннями опиняються в кінці списку, а елементи з меншими значеннями поступово переміщуються у напрямку до початку списку. Образно кажучи, важкі елементи падають на дно, а легкі повільно спливають подібно до бульбашок повітря.

Змінна **prap** типу **bool** виконує роль прапорця. Вона отримує значення **True** в тому випадку, якщо відбулася хоча б одна перестановка сусідніх елементів. Якщо значення **prap** не змінилось, це означає, що елементи списку вже впорядковані й подальший перегляд списку не потрібний.

Реалізуємо алгоритм у вигляді функції.

```
def sort_bulb():
    prap = True
    k = len(arr)-1
    while prap:
        prap = False
        for i in range(k):
            if arr[i]>arr[i+1]:
                arr[i], arr[i+1] = arr[i+1], arr[i]
                prap = True
        k = k-1
```

**ПРИКЛАД 2.** Проаналізуємо виконання алгоритму на прикладі списку **arr = [5, 10, 1, 4, 6]**.

Перша ітерація зовнішнього циклу, **k = 4**

- **5 > 10?** Ні.
- **10 > 1?** Так. 10 і 1 міняємо місцями, **prap = True**.
- **10 > 4?** Так. 10 і 4 міняємо місцями, **prap = True**.
- **10 > 6?** Так. 10 і 6 міняємо місцями, **prap = True**.

**arr = [5, 1, 4, 6, 10]**

Друга ітерація зовнішнього циклу, **k = 3**

- **5 > 1?** Так. 5 і 1 міняємо місцями, **prap = True**.
- **5 > 4?** Так. 5 і 4 міняємо місцями, **prap = True**.
- **5 > 6?** Ні.

**arr = [1, 4, 5, 6, 10]**

Третя ітерація зовнішнього циклу, **k = 2**

- **1 > 4?** Ні. **prap = False**.
- **4 > 5?** Ні. **prap = False**

**arr = [1, 4, 5, 6, 10]**

Уже на третій ітерації елементи виявились упорядкованими за зростанням, змінна **prap = False**, тому зовнішній цикл припиняє роботу. Для різних наборів значень списку може знадобитися різна кількість кроків сортування.

### Питання для самоперевірки

1. У чому полягає сутність сортування методом вибору максимального елемента?

2. У чому полягає сутність сортування методом бульбашки?

3. Де може знаходитися найбільший елемент списку, якщо список не впорядковано?

4. Де може знаходитися найменший елемент списку, якщо список упорядковано за зростанням; за спаданням?

5. **int(True) = 1; int(False) = 0**. Для кожної пари сусідніх елементів списку **a** виконується операція

**s = s+int(a[i] >= a[i+1])**

Початкове значення **s** дорівнює 0. Визначте, чому дорівнює кінцеве значення **s**, якщо список:

а) було впорядковано за зростанням;

б) було впорядковано за спаданням;

в) не було впорядковано.

6. Дано список результатів забігу на 100 м восьми спортсменів. Складіть програму для визначення трьох кращих результатів.

### Вправа 32

Скласти програму для розв'язування задачі.

Задача. У фігурному катанні загальна оцінка якості виконання елемента обчислюється як усічене середнє оцінок, даних 9 судьями. Для цього відкидаються найвища і найнижча оцінки, а з решти обчислюється середнє арифметичне. Складіть програму для визначення оцінки за цими правилами.

1) Створіть **Python file** із назвою **Оцінки**. Завантажте модуль **random**:

**import random**

2) Створіть порожній список **grades**.

3) Додайте до списку 9 випадкових чисел з інтервалу (4, 6). Щоб згенерувати випадкове дійсне число, викличте метод **random.uniform(4, 6)** і округліть згенероване число до 1 десяткового знаку:

**for i in range(9):**

**x = random.uniform(4, 6)**

**grades = grades+[round(x, 1)]**

- Виведіть список на екран.
- 4) Упорядкуйте список. Виведіть упорядкований список на екран.
  - 5) Знайдіть середнє арифметичне елементів, що утворюють зріз списку згідно з умовою задачі.
  - 6) Виведіть результат з одним десятковим знаком.

### Комп'ютерне тестування

Виконайте тестове завдання 32 з автоматичною перевіркою результату.

## § 33. ОБ'ЄКТИ КЛАСУ LISTBOX

Вибирати для виконання певних дій елементи списку із заданою ознакою зручніше, якщо ці елементи візуально представлені й доступні у графічному вікні. Таку можливість надають віджети класу **Listbox**.

Об'єкт класу **Listbox** дозволяє відображати список елементів, з якого користувач може обрати один або декілька пунктів.

### Створення об'єктів класу Listbox

Для налаштування віджета **Listbox** можна задати в його конструкторі такі спільні для багатьох віджетів параметри, як **bg, fg** (колір фону; колір шрифту); **font**: настройки шрифту; **width, height**: висота та ширина елемента (за замовчуванням висота встановлюється в 10 рядків, ширина — 20 символів) у рядках. Важливо задати значення параметра **selectmode**, яке визначає, скільки елементів може бути виділено. За замовчуванням у **Listbox** натисканням миші можна вибрати тільки один елемент. Якщо установити **selectmode = EXTENDED**, можна вибрати декілька елементів, утримуючи натиснутою клавішу **Ctrl**.

При створенні віджета класу **Listbox** слід дотримуватися такої послідовності дій:

- 1) створити об'єкт класу **Listbox**;
- 2) заповнити список **Listbox** елементами;
- 3) розмістити віджет у вікні.

**ПРИКЛАД 1.** Створити віджет класу **Listbox**.

```
lbox_mova_progr = Listbox(width = 40, selectmode = EXTENDED)  
lbox_mova_progr.pack()       # Віджет розміщується у вікні
```

Для наповнення списку **Listbox** елементами слід у циклі **for** пройти по всіх елементах списку **mova\_progr** і додати кожний елемент до списку **Listbox** за допомогою методу **insert()**:

```
<назва об'єкта>.insert(index, element)
```

Як перший аргумент методу **insert** передається індекс вставлення елемента. Якщо ми хочемо послідовно додавати елементи, замість індекса можна використовувати значення **END**.

**ПРИКЛАД 2.** Відобразити у віджеті класу **Listbox** список мов програмування (рис. 33.1).

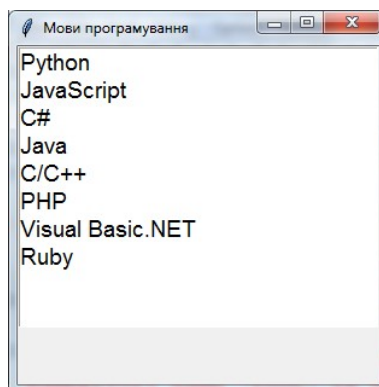


Рис. 33.1

```
mova_progr = ["Python", "JavaScript", "C#", "Java", "C/C++",  
"PHP", "Visual Basic.NET", "Ruby"]  
# Заповнюється за допомогою методу insert()  
for mova in mova_progr:  
    lbox_mova_progr.insert(END, mova)
```

### Основні методи Listbox

Крім методу **insert()**, клас **Listbox** має інші корисні методи для керування поведінкою об'єктів та їхнім вмістом. Розглянемо деякі з цих методів.

- Метод **<назва об'єкта>.curselection()** повертає набір індексів елементів, що виділені.

**ПРИКЛАД 3.** Застосувати метод **curselection()** до списку віджета **lbox\_mova\_progr**. Метод викликається в обробнику події для кнопки **Застосувати метод curselection()** (рис. 33.2).

```
def btn_click():  
    print (lbox_mova_progr.get(3))    # Буде виведено (1, 3, 6)
```

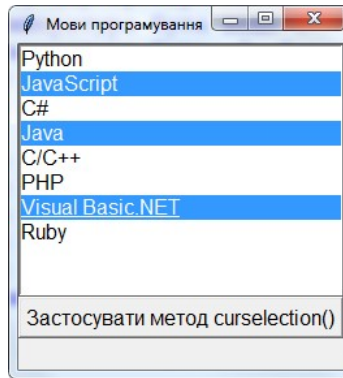


Рис. 33.2

- Метод `<назва об'єкта>.delete(n)` видаляє зі списку **Listbox** елемент з індексом **n**.

**ПРИКЛАД 4.** Видалити елемент з індексом 6.

`lbox_mova_progr.delete(6)`

Вигляд списку після виконання методу подано на рис. 33.3.

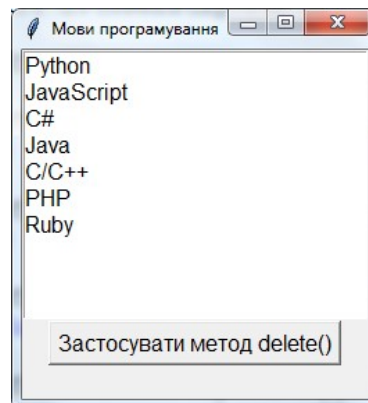


Рис. 33.3

- Метод `<назва об'єкта>.get(n)` повертає текст елемента з індексом **n**.

**ПРИКЛАД 5.** Вивести в консоль елемент списку **Listbox** з індексом 3.

`print(lbox_mova_progr.get(3))`     # Буде виведено Java

**ПРИКЛАД 6.** Створити список **elem** з елементів, що додані до списку **Listbox**.

`elem = list(lbox_mova_progr.get(0, END))`

- Метод `<назва об'єкта>.size()` повертає число елементів у списку **Listbox**.

**ПРИКЛАД 7.** Визначити кількість елементів у списку `lbox_mova_progr`.

```
k = lbox_mova_progr.size()
print(k)           # Буде виведено 8
```

### Застосування методів класу `Listbox`

Додавати рядки до списку `Listbox` можна під час виконання програми, вводячи значення з клавіатури.

**ПРИКЛАД 8.** Запрограмувати додавання рядків до списку віджета `lbox` класу `Listbox`, уводячи по черзі значення до поля віджета `entry_add` класу `Entry` і натискаючи кнопку `badd` (рис. 33.4).

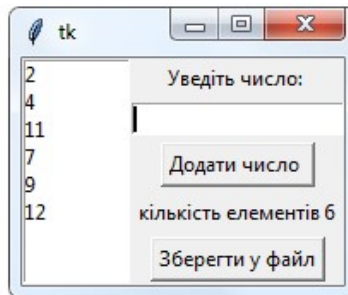


Рис. 33.4

```
from tkinter import*
def badd_cl():
    # Додавання рядка з поля entry_add до списку lbox
    lbox.insert(END, entry_add.get())
    entry_add.delete(0, END) # Очищення поля entry_add
    k = lbox.size()         # Визначення кількості рядків у списку lbox
    lab2.config(text = 'кількість елементів '+str(k))
    # Виведення значення k
root = Tk()
lbox = Listbox(root, selectmode = EXTENDED, width = 10, height = 8)
lbox.grid(row = 0, column = 0, rowspan = 5)
# Віджет lbox розміщується у вікні
lab1 = Label(root, text = 'Уведіть число: ').grid(row = 0, column = 1)
entry_add = Entry(root)      # Створення текстового поля
entry_add.grid(row = 1, column = 1)
badd = Button(text = "Додати число ", command = badd_cl)
badd.grid(row = 2, column = 1)
lab2 = Label(root, text = '****')
lab2.grid(row = 3, column = 1)
root.mainloop()
```



**ПРИКЛАД 9.** Додати до коду з прикладу 8 можливість зберігати список віджета `lbox` у текстовому файлі (рис. 33.5). Для цього слід створити кнопку **Зберегти у файл** і описати обробник події для кнопки.

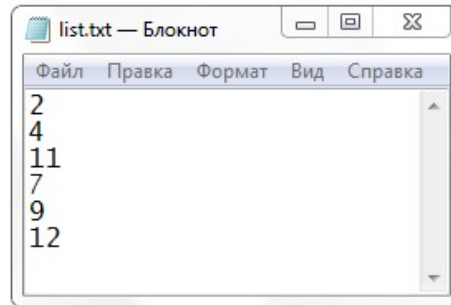


Рис. 33.5

```
def saveList():
    f = open('list.txt', 'w')
    f.writelines("\n".join(lbox.get(0, END)))
    f.close()
bsave = Button(root, text = "Зберегти у файл", command = saveList)
bsave.grid(row = 4, column = 1)
```

Метод `join()` перетворює кортеж рядків, який повернув метод `get()`, в один рядок через роздільник `\n`

**ПРИКЛАД 10.** Обчислити суму чисел, які знаходяться в списку `lbox`, що створений у прикладі 8 (див. рис. 34.4).

До коду основної програми слід додати кнопку **Знайти суму** й описати обробник події для кнопки:

```
def bsum_cl():
    elem = list(map(int, lbox.get(0, END)))
    s = sum(elem)
    lab2.config(text = "Сума елементів "+str(s))
    <...>
bsum = Button(text = "Знайти суму", command = bsum_cl)
bsum.grid(row = 4, column = 1)
```

**ПРИКЛАД 11.** Обчислити суму чисел, які знаходяться у виділених рядках списку `lbox`, що створений у прикладі 8 (рис. 33.6).

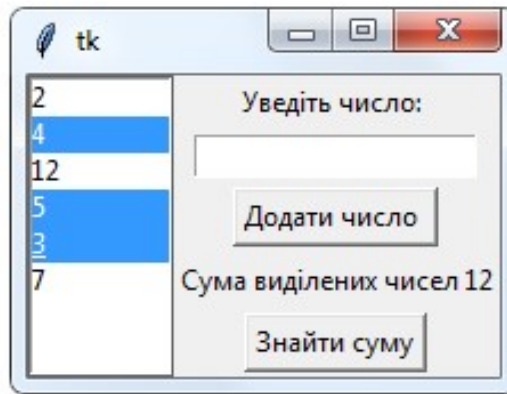


Рис. 33.6

Функцію `bsum_cl()` слід змінити таким чином:

```
def bsum_cl():
```

```
    select = list(lbox.curselection())
```

```
    # Список індексів виділених елементів
```

```
    s = 0
```

```
    for i in select:
```

```
        s = s+int(lbox.get(i))
```

```
    # Додавання елементів зі списку lbox, індекси яких знаходяться в
    # списку select
```

```
    lab2.config(text = 'Сума виділених '+str(s))
```

Отже, віджет **Listbox** дозволяє унаочнити роботу зі списками, полегшити опрацювання значень у списку, вибір елементів тощо.

### Питання для самоперевірки

1. Поясніть алгоритм створення об'єкта класу **Listbox**.
2. Створіть об'єкт **lab** класу **Listbox**. Заповніть список 5 випадковими цілими числами.
3. Виведіть у консоль число елементів у списку **lab**.
4. Виведіть у консоль індекс виділеного елемента списку **lab**.
5. Виведіть у консоль текст 3-го елемента в списку **lab**.
6. Видаліть 4-й елемент списку **lab**.

### Вправа 33

Створити програму **Покупки**.

Завдання. Напишіть програму, що складається з двох списків **Listbox**. Перший список містить перелік товарів, другий — перелік покупок. При натисканні на одну кнопку товар має переходити з одного списку в інший.

При натисканні на іншу кнопку — повертатися (людина передумала купувати). Передбачити можливість множинного вибору елементів списку і їх переміщення.

1) Створіть **Python file** із назвою **Покупки**. Завантажте модуль **tkinter**. Створіть вікно програми із заголовком **Покупки**. Заверште програму оператором **root.mainloop()**.

2) Додайте у вікно програми віджет **box1** класу **Listbox**, занесіть до списку перелік товарів.

```
box1 = Listbox(selectmode = EXTENDED)
box1.grid(row = 0, column = 0, rowspan = 6)
products = ["ЯБЛУКА", "БАНАНИ", "МОРКВА", "ХЛІБ",
"МОЛОКО", "КАВА", "ЧАЙ", "ЦУКОР", "ЧІПСИ", "ЦУКЕРКИ"]
for p in products:
    box1.insert(END, p)
```

Додайте у вікно віджети **btn1**, **btn2** класу **Button** (рис. 33.7).

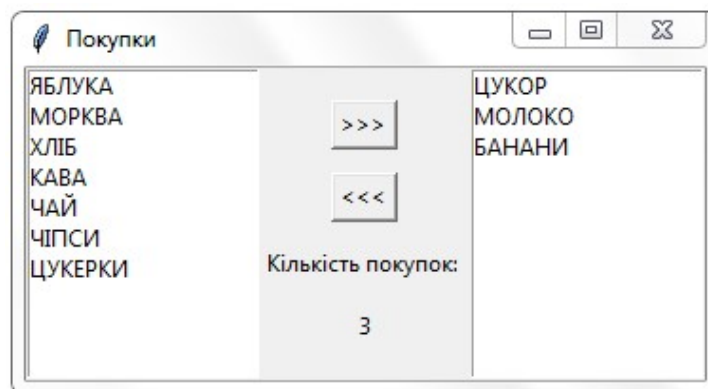


Рис. 33.7

3) Створіть об'єкт **box2** класу **Listbox**, залиште його порожнім.

4) Опишіть обробник події для кнопки **btn1** (перенесення елементів зі списку **box1** до списку **box2**).

```
def to_box2():
    select = list(box1.curselection())
    select.reverse()
    for i in select:
        box2.insert(END, box1.get(i))
        box1.delete(i)
```

*Підказка.* У функції **to\_box2()** кортеж обраних елементів перетворюється в список, після чого виконується його запис у зворотному порядку. Це робиться для того, щоб видалення елементів відбувалося з кінця списку.

Опишіть обробник події для кнопки **btn2** (перенесення елементів зі списку **box2** до списку **box1**).

5) Додайте віджет **lab1** класу **Label** із заголовком **Кількість покупок** і віджет **lab2** класу **Label** для виведення кількості елементів у списку **box2**. Додайте до описів функцій оператори для зміни значення атрибуту **text** віджета **lab2** на значення **box\_2.size()**.

```
k = box2.size()
lab2.config(text = str(k))
```

### Комп'ютерне тестування

Виконайте тестове завдання 33 з автоматичною перевіркою результату.

## § 34. ВІЗУАЛІЗАЦІЯ ЕЛЕМЕНТІВ ТАБЛИЧНОЇ ВЕЛИЧИНИ

У різних сферах діяльності існує багато завдань, у яких вихідні дані й результати повинні бути подані у графічній формі. Уявіть собі, що ви готуетесь до виступу з доповіддю. Ваша доповідь виглядатиме більш наочною, якщо сухі цифри подати у вигляді кольорової діаграми. Діаграми будуються на основі числових даних, що містяться в таблицях, і допомагають аналізувати та порівнювати дані.

Розглянемо два способи побудови діаграм: із використанням графічних методів модуля **tkinter** і за допомогою методів бібліотеки **matplotlib**.

### Побудова діаграм із використанням графічних методів модуля **tkinter**

**Діаграма** — це графічне зображення, у якому числові дані подано геометричними фігурами. Діаграми можна створювати, програмуючи креслення графічних примітивів (лінія, прямокутник, сектор) графічними методами полотна **Canvas**, із якими ви знайомі з курсу 7 класу.

**ПРИКЛАД 1.** Гістограма (стовпчикова діаграма) використовується для порівняння елементів табличної величини. Побудуємо гістограму показника успішності учнів 9А класу з чотирьох предметів, значення якого занесені до списку **ndu** (рис. 34.1).

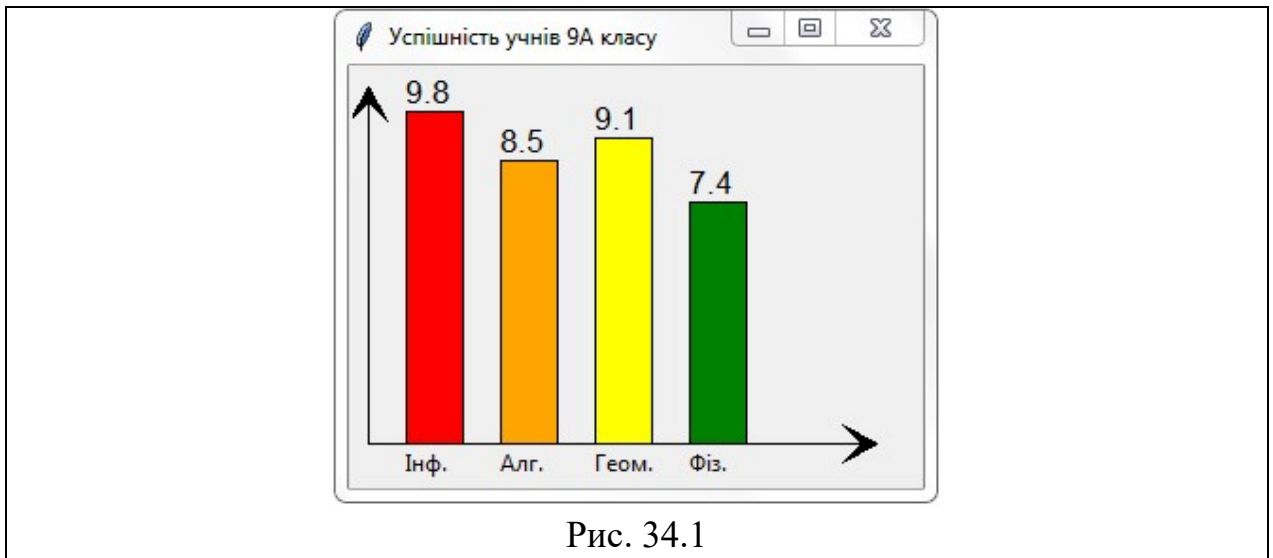


Рис. 34.1

Областю побудови діаграми є полотно об'єкта **Canvas** розмірами  $300 \times 220$  пікселів. Координатні осі намальовані за допомогою методу **canvas.create\_line()**. Для малювання стрілок до переліку параметрів методу додано параметри **arrow** та **arrowshape**. Атрибуту **arrow** надано значення **LAST**, що задає розміщення стрілки в кінці лінії. Значення атрибуту **arrowshape** задає розміри стрілки.

Ряди даних побудовані методом **create\_rectangle()**.

До діаграми слід додати підписи осей і рядів даних. Розмістити текст на полотні можна за допомогою методу **create\_text(x, y, text)**. За замовчуванням у точці  $(x, y)$  розташовується центр напису. Щоб розмістити за вказаними координатами ліву границю тексту, атрибуту **anchor** надається значення **W** (від англ. *west* — захід).

```

from tkinter import*
root = Tk()
root.title("Успішність учнів 9А класу")
canvas = Canvas(root, width = 300, height = 220)
canvas.pack()
ndu = [9.8, 8.5, 9.1, 7.4]
# Побудова координатних осей
canvas.create_line(10, 200, 10, 10, arrow = LAST, arrowshape = "10 20
10")
canvas.create_line(10, 200, 280, 200, arrow = LAST, arrowshape = "10
20 10")
colors = ['red', 'orange', 'yellow', 'green']
subject = ['Інф. ', 'Алг. ', 'Геом. ', 'Фіз. ']
x = 30
for i in range(4):

```

```

y = 220-ndu[i]*20
canvas.create_rectangle(x, y, x+30, 200, fill = colors[i])
canvas.create_text(x, 210, text = subject[i], anchor = W)
canvas.create_text(x, y-10, text = str(ndu[i]), anchor = W, font =
"Arial 13")
x = x+50

```

**ПРИКЛАД 2.** Графіки використовуються для зображення зміни значень певної величини в певний період часу. Побудуємо графік зміни значень середньодобової температури повітря за тиждень, які зберігаються в списку **a** (рис. 34.2).



Рис. 34.2

```

from tkinter import*
root = Tk()
root.title("Середньодобова температура за тиждень")
canvas = Canvas(root, width = 300, height = 220)
canvas.pack()
a = [14, 12, 15, 12, 9, 10, 7]
# Побудова координатних осей
canvas.create_line(10, 200, 10, 10, arrow = LAST, arrowshape = "10 20
10")
canvas.create_line(10, 200, 280, 200, arrow = LAST, arrowshape = "10
20 10")
day = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд']
x1 = 30

```

```

y1 = 220-a[0]*10
canvas.create_text(x1, 210, text = day[0], anchor = W)
canvas.create_text(x1, y1-10, text = str(a[0]), anchor = W, font = "Arial
12")
for i in range(1, 7):
    x = x1+30
    y = 220-a[i]*10
    canvas.create_line(x1, y1, x, y, width = 2)
    canvas.create_text(x, 210, text = day[i], anchor = W)
    canvas.create_text(x, y-10, text = str(a[i]), anchor = W, font = "Arial
12")
x1 = x
y1 = y

```

**ЦІКАВИНКА.** Карен Спарк Джон (рис. 34.3) — британська дослідниця, яка розробила технології пошуку, що дозволили користувачам і користувачкам працювати з комп'ютерами, використовуючи слова замість рівнянь та кодів.

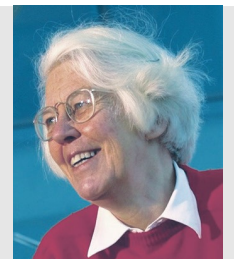


Рис. 34.3

### Побудова діаграм за допомогою методів бібліотеки matplotlib

**Matplotlib** — це кросплатформна бібліотека візуалізації даних і побудови діаграм для **Python**. Щоб установити бібліотеку для роботи в **IDLE** **PyCharm**, відкрийте вкладку **Terminal** і в командному рядку наберіть сполучення клавіш **Ctrl + Z**, щоб вийти з віртуального оточення; у наступному рядку наберіть команду **pip install matplotlib**:

```
(venv) C:\Users\...\PycharmProjects\pythonProject4>^Z
```

```
(venv) C:\Users\...\PycharmProjects\pythonProject4>pip install
matplotlib
```

Після отримання повідомлення про успішне інсталювання **matplotlib** можна у **Python**-файлі підключати модуль **pyplot** даної бібліотеки, який містить колекцію функцій для створення діаграм і налаштування їхнього вигляду. Імпортуємо пакет **pyplot** і коротко позначимо його як **plt**.

```
import matplotlib.pyplot as plt
```

**ПРИКЛАД 3.** Побудуємо стовпчикову діаграму для відображення елементів списку **data** (рис. 34.4).

```
import matplotlib.pyplot as plt
data = [5., 25., 50., 20.]
plt.bar(range(len(data)), data)
plt.show()
```

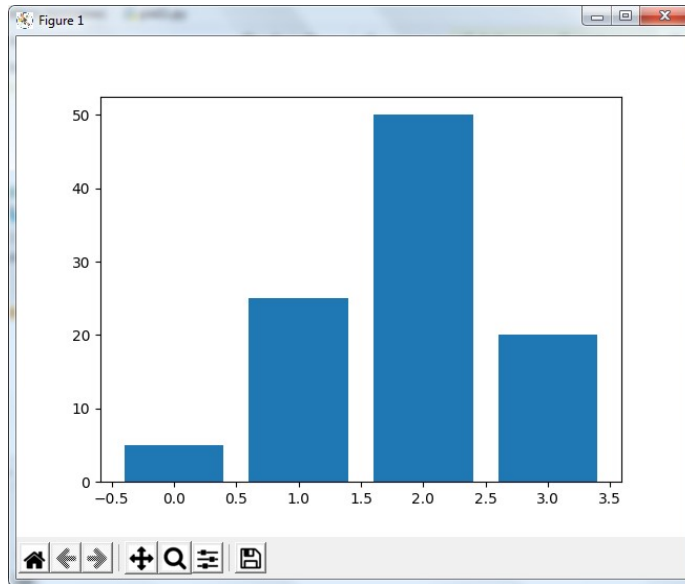








Рис. 34.4

Для виведення зображення на екран використовується функція **plt.show()**. Діаграму побудовано в окремому графічному вікні **plotting window**, яке має заголовок **Figure#**. Кнопки панелі інструментів вікна мають таке призначення:

Кнопка	Призначення
	Повертає графік до первісного стану (вид при першому запуску)
	Переміщує назад/вперед по історії зміни графіка. Наприклад, після збільшення показника певної галузі можна повернутися до попереднього, використовуючи кнопку 2
	Перетягує (ліва кнопка миші) або встановлює масштаб (права кнопка миші) графіка
	Збільшує зображення
	Відкриває додаткові налаштування вікна (наприклад, відступи від країв)
	Викликає діалог збереження графіка у файл



Для підвищення наочності діаграми можна налаштувати такі властивості, як **title** (заголовок діаграми), **xlabel**, **ylabel** (підписи осей), **grid** (координатна сітка).

У прикладі 3 функції **plt.bar()** передано тільки один список чисел, і **matplotlib** пов'язує цей список із послідовністю чисел **x**: 0, 1, 2, 3, ... Зазвичай графік являє собою пару значень (**x**, **y**), тому слід задавати два масиви: у першому будуть значення для осі **x**, у другому — для **y**.

**ПРИКЛАД 4.** Побудуємо діаграму з прикладу 1 (рис. 34.5).

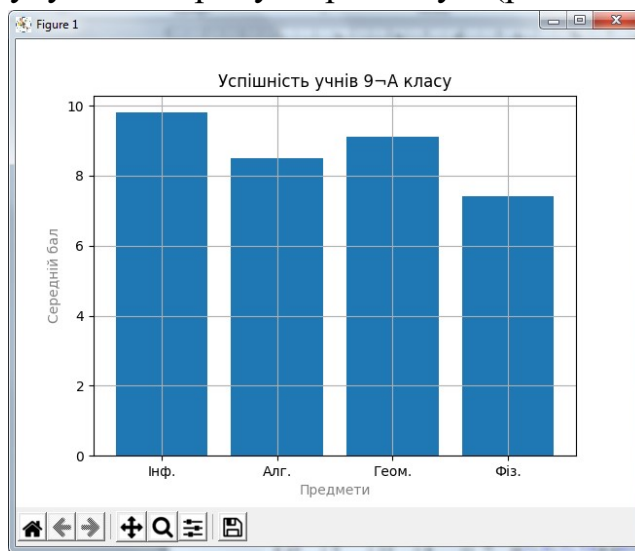


Рис. 34.5

```
import matplotlib.pyplot as plt
plt.title('Успішність учнів 9А класу')
ndu = [9.8, 8.5, 9.1, 7.4]
subject = ['Інф.', 'Алг.', 'Геом.', 'Фіз.']
plt.xlabel('Предмети', color = 'gray')
plt.ylabel('Середній бал', color = 'gray')
plt.grid(True)
plt.bar(subject, ndu)
plt.show()
```

**ПРИКЛАД 5.** Для побудови графіків призначена функція **plot()**. Побудуємо графік із прикладу 2, задавши червоний колір лінії (параметр **'r'**) (рис. 34.6).

```
import matplotlib.pyplot as plt
plt.title('Середньодобова температура за тиждень')
a = [14, 12, 15, 12, 9, 10, 7]
day = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб', 'Нд']
```

```
plt.xlabel('Дні тижня', color = 'gray')
plt.ylabel('Середньодобова температура', color = 'gray')
plt.grid(True)
plt.plot(day, a, 'r')
plt.show()
```



Рис. 34.6

**ПРИКЛАД 6.** Побудувати кругову діаграму прибутку підприємства за 4 квартали року (рис. 34.7):

	I квартал	II квартал	III квартал	IV квартал
Прибуток підприємства, тис. грн	100	40	80	70

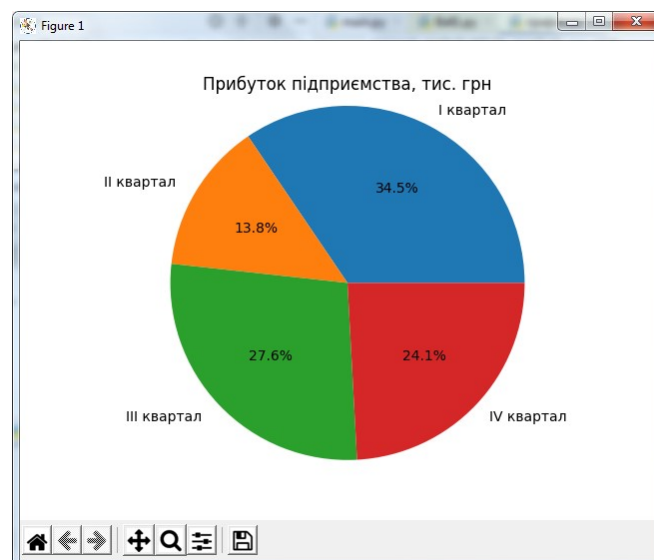


Рис. 34.7

```

import matplotlib.pyplot as plt
plt.title('Прибуток підприємства, тис. грн')
a = [100, 40, 80, 70]
kv = ['I квартал', 'II квартал', 'III квартал', 'IV квартал']
plt.pie(a, labels = kv, autopct = "%.1f%%")
plt.show()

```

Атрибут **autopct** дозволяє відображати частку, використовуючи форматування рядка **Python**.

### Питання для самоперевірки

1. Обговоріть переваги і недоліки двох способів побудови діаграм.
2. Поясніть послідовність операцій при побудові одного ряду даних за допомогою графічного методу модуля **tkinter**.
3. Опишіть алгоритм побудови графіка змін деякої величини з використанням графічних методів модуля **tkinter**.
4. Назвіть методи класу **matplotlib.pyplot**, призначені для візуалізації табличних величин.
5. Поясніть призначення інструментів інтерактивної навігації графічного вікна
6. Додайте до проєкта **Календар нагляду за погодою** (практична робота 8) процедуру побудови графіка опадів за рік.

### Вправа 34

Створити програму для виконання завдання.

Завдання. У таблиці наведено дані про обсяг продажів книжкового магазину за місяцями за період із січня до червня. Визначити сумарний обсяг продажів за півріччя, побудувати гістограму і кругову діаграму обсягів продажу за місяцями.

	Січень	Лютий	Березень	Квітень	Травень	Червень
Обсяг продажів, тис. грн	11,0	17,0	18,0	32,5	22,5	15,5

- 1) Створіть **Python file** з назвою **Продажі**. Завантажте модулі **tkinter**, **matplotlib.pyplot**. Створіть вікно програми із заголовком **Обсяг продажів**. Заверште програму оператором **root.mainloop()**.
- 2) Збережіть масиви назв місяців і значень обсягу продажів у списках **month**, **sales**.

```
month = ['Січень', 'Лютий', 'Березень', 'Квітень', 'Травень',  
'Червень']
```

```
sales = [11.0, 17.0, 18.0, 32.5, 22.5, 15.5]
```

Задайте заголовок вікна діаграми:

```
plt.title('Обсяг продажів')
```

3) Додайте у вікно програми віджет **box1** класу **Listbox**, занесіть до списку значення списку **sales**:

```
box1 = Listbox(selectmode = EXTENDED)
```

```
box1.grid(row = 0, column = 0, rowspan = 6)
```

```
for p in sales:
```

```
    box1.insert(END, p)
```

4) Додайте у вікно віджет **btn1** класу **Button**:

```
btn1 = Button(text = "Стовпчикова діаграма ", command = btn1_cl)
```

```
btn1.grid(row = 2, column = 1)
```

Опишіть функцію **btn1\_cl()**, запишіть оператори для побудови стовпчикової діаграми за значеннями елементів масиву **sales**:

```
plt.xlabel('Місяці', color = 'gray')
```

```
plt.ylabel('тис. грн', color = 'gray')
```

```
plt.bar(month, sales)
```

```
plt.show()
```

5) Додайте у вікно віджет **btn2** класу **Button**:

```
btn2 = Button(text = "Графік ", command = btn2_cl)
```

```
btn2.grid(row = 4, column = 1)
```

Опишіть функцію **btn2\_cl()**, призначену для побудови графіка за значеннями елементів масиву **sales**.

6) Додайте у вікно віджет **btn3** класу **Button**:

```
btn3 = Button(text = "Кругова діаграма ", command = btn3_cl)
```

```
btn3.grid(row = 6, column = 1)
```

Опишіть функцію **btn3\_cl()**, призначену для побудови кругової діаграми за значеннями елементів масиву **sales**.

```
def btn3_cl():
```

```
    plt.pie(sales, labels = month, autopct = "%.1f%%")
```

```
    plt.show()
```

Випробуйте роботу програми.

### Комп'ютерне тестування

Виконайте тестове завдання 34 з автоматичною перевіркою результату.

## Практична робота 10. СКЛАДАННЯ І ВИКОНАННЯ АЛГОРИТМІВ ЗНАХОДЖЕННЯ СУМ І КІЛЬКОСТЕЙ ЗНАЧЕНЬ ЕЛЕМЕНТІВ ТАБЛИЧНИХ ВЕЛИЧИН ЗА ЗАДАНИМИ УМОВАМИ

Завдання: створити програму для розв'язування задачі.

Задача. Дано таблицю кількості опадів протягом року за місяцями:

Номер місяця	1	2	3	4	5	6	7	8	9	10	11	12
К-сть опадів, мм	54	12	30	15	40	32	10	30	46	60	56	38

Визначити:

- кількість опадів за рік;
- у якому місяці кількість опадів була найменшою;
- у якому місяці кількість опадів була найбільшою;
- у які місяці кількість опадів була менша від 40, кількість таких місяців.

Обладнання: комп'ютер із середовищем програмування **PyCharm**.

### Хід роботи

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

- Створіть **Python file** з назвою **Опади**. Завантажте модулі **tkinter**, **matplotlib.pyplot**.
- Створіть вікно програми із заголовком **Таблиця опадів**. Заверште програму оператором **root.mainloop()**.
- Створіть списки для збереження масивів назв місяців і кількостей опадів:  

```
month= ['січень', 'лютий', 'березень', 'квітень', 'травень', 'червень',  
'липень', 'серпень', 'вересень', 'жовтень', 'листопад', 'грудень']  
mas = [54, 12, 30, 15, 40, 32, 10, 30, 46, 60, 56, 38]
```
- Створіть масив об'єктів класу **Label**, який буде відображати назви місяців:

```
list_1=[]
```

```
for i in range (12):
```

```
list_1.append(Label(text = month[i], bg = 'lightblue', width = 4))
```

```
list_1[i].grid(row = 0, column = i, padx = 2, pady = 2)
```

5. Створіть масив **list\_2** об'єктів класу **Label**, який буде відображати значення елементів списку **mas**.

6. Створіть командні кнопки **btn1** **Опрацювати дані**, **btn2** **Побудувати графік** (див. рисунок).

7. Створіть віджет **box1** класу **Listbox**.

```
box1=Listbox()
```

```
box1.grid(row = 3, column = 4, rowspan = 5, columnspan = 5)
```

8. Опишіть функцію **btn1\_cl()** — обробник події для кнопки **btn1**. Обчисліть суму елементів списку **mas** і додайте результат до списку **box1**:

```
def btn1_cl():
```

```
    s = sum(mas)
```

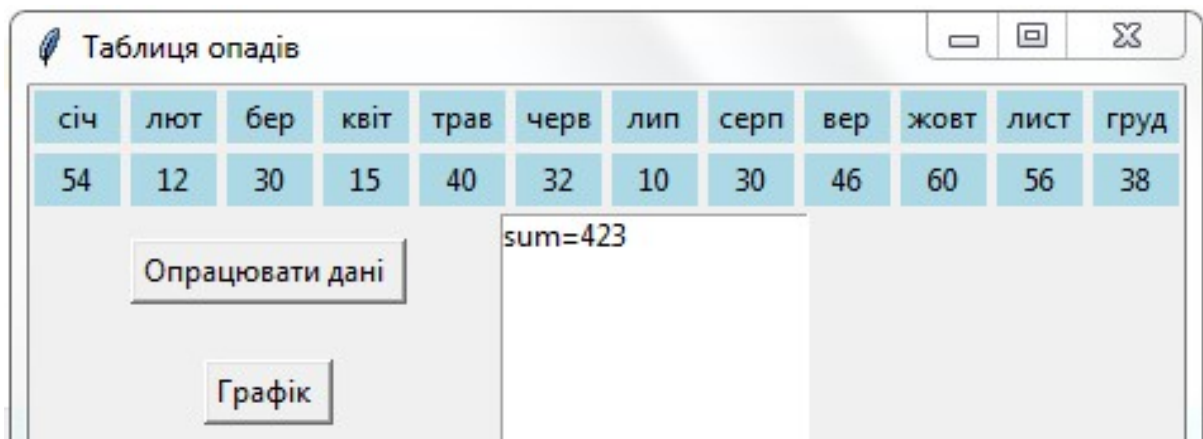
```
    box1.insert(END, "Кількість за рік "+str(s))
```

9. Додайте до списку **box1** назву місяця, в якому кількість опадів була найменшою.

10. Додайте до списку **box1** назву місяця, в якому кількість опадів була найбільшою.

11. Користуючись алгоритмом пошуку елементів із заданою властивістю, знайдіть кількість місяців, у яких кількість опадів менша за 40 мм, і виведіть до списку **box1** назви цих місяців.

12. Опишіть функцію **btn2\_cl()** — обробник події для кнопки **btn2**, призначену для побудови графіка за значеннями списку **mas**.



Зробіть висновок: як застосовувати методи списків і типові алгоритми опрацювання табличних величин.

## § 35. ПОНЯТТЯ ДВОВИМІРНОГО МАСИВУ. ВКЛАДЕНІ СПИСКИ

Списки корисні для опрацювання таблиць, які містять один рядок даних, наприклад таблиця опадів по місяцях за один рік. А якщо потрібно проаналізувати кількість опадів по місяцях за кілька років? Під час розв'язування багатьох задач доводиться працювати з таблицями, у яких кілька рядків і кілька стовпців. Як правило, дані в комірках таблиці належать до одного типу. Такі таблиці прийнято називати двовимірними масивами. У мові **Python** для опрацювання таких таблиць використовуються вкладені списки, тобто списки, кожен елемент якого у свою чергу є списком значень.

Масив називається **двовимірним**, якщо для задання місцеположення елемента в масиві необхідно вказати значення двох індексів.

**ПРИКЛАД 1.** Збережемо дані таблиці, поданої на рис. 35.1, у змінній **a**.

1	2	3	12
23	5	13	4
8	6	7	11

Рис. 35.1

```
a = [[1, 2, 3, 12], [23, 45, 13, 4], [8, 6, 7, 11]]
```

Кожний елемент списку **a** також є списком, що містить дані з одного рядка таблиці. Довжина списку **len(a) = 3**. Звернутися до вкладеного списку (рядка таблиці) можна за його індексом:

```
a[0] = [1, 2, 3, 12].
```

Для перебору рядків списку використовується цикл **for**.

Нехай **b = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]**

Можна перебрати індекси вкладених списків — елементів списку	Можна перебрати всі наявні в списку елементи — рядки таблиці.
<pre>for i in range (len(b)):     print(b[i])</pre>	<pre>for row in b:     print(row)</pre>

При використанні обох варіантів циклу в консоль буде виведено:

```
↑ C:\Users\NMR\P  
[1, 2, 3, 4]  
↓ [5, 6]  
↻ [7, 8, 9]
```

Положення окремого елемента в таблиці визначається 2 індексами — номером рядка і номером стовпця, отже, щоб звернутися до елемента у вкладеному списку, слід указати два індекси.

Елемент, розташований на перетині *i*-го рядка і *j*-го стовпця масиву **A**, позначають **A[i][j]**.

**!** Запам'ятайте, що у двовимірних масивах перший індекс завжди вказує на номер рядка, а другий — на номер стовпця.

Елементи списку **a** з прикладу 1 мають такі значення:

**a[0][0] == 1, a[0][1] == 2, a[1][0] == 23, a[1][1] == 5, a[1][2] == 13.**

Головну діагональ масиву утворюють елементи, для яких **i == j**.

Послідовно отримати доступ до всіх елементів двовимірного масиву можна за допомогою вкладених циклів: зовнішній цикл перебирає вкладені списки (рядки таблиці), внутрішній цикл перебирає елементи рядка.

**ПРИКЛАД 2.** Виведемо значення списку **a** на екран порядково, додаючи пробіл між елементами. Зовнішній цикл **for i** перебирає індекси елементів списку **a** — рядків двовимірного масиву.

Внутрішній цикл **for j** перебирає індекси елементів списку **a[i]** (*i*-го рядка масиву). Згадаємо, що параметр **end = ' '** методу **print()** потрібен, щоб після виведення курсор залишився в тому самому рядку.

Оператор **print()** із порожнім списком виведення переводить курсор на наступний рядок.

```
a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]  
for i in range(len(a)):  
    for j in range(len(a[i])):  
        print(a[i][j], end = ' '  
    print()
```

**ПРИКЛАД 3.** Щоб вивести в консоль значення двовимірного масиву **b** порядково, можна перебирати всі наявні у зовнішньому списку рядки та у внутрішньому циклі — всі наявні у вкладених списках елементи:

```
b = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]  
for row in b:  
    for elem in row:  
        print(elem, end = ' '  
    print()
```

```
=====
1 2 3 4
5 6
7 8 9
>>>
```



### Створення вкладених списків

Нехай потрібно створити двовимірний масив для збереження даних прямокутної таблиці, у якій **n** рядків і **m** стовпців, і заповнити його нулями. Це можна зробити в такий спосіб: створити порожній список, потім додати до нього **n** елементів — рядків масиву.

```
n = 3
m = 4
a = []
for i in range(n):
    a.append([0]*m)
```

Іншим способом створення двовимірного масиву є використання генератора. Список **[0]\*m** заново генерується для заповнення чергового елемента списку **a**.

```
n = 3
m = 4
a = [[0]*m for i in range(n)]
```

### Введення елементів двовимірного масиву

Щоб створити двовимірний масив із **n** рядків та **m** стовпців, уводячи значення елементів масиву з клавіатури, слід **n** разів повторити дії:

1) увести рядок **row**, що містить **m** чисел, відокремлених пробілами, і розбити рядок **row** функцією **split()** на список;

2) кожний елемент **row[i]** рядка **row** перетворити на числовий тип;

3) рядок **row** додати до списку **a**.

```
n = int(input())          # Кількість рядків масиву
a = []                   # Створюється порожній список
for i in range(n):
    row = input().split()
    for j in range(len(row)):
        row[j] = int(row[j])
    a.append(row)
```

**ПРИКЛАД 4.** Заповнення двовимірного масиву з 3 рядків та 4 стовпців випадковими числами:

```
from random import*
a = [], [], []
for i in range(len(a)):
    for j in range(4):
        a[i] = a[i]+[randint(1,10)] # Додавання чергового елемента до i-го
рядка
```

**ПРИКЛАД 5.** Задання значень елементів двовимірного масиву за певною формулою наявне у випадках, коли значення елемента залежить від його індексів. Нехай кожний елемент масиву  $3 \times 4$  дорівнює більшому з його індексів:

```

a = [], [], []
for i in range(len(a)):
    for j in range(4):
        a[i] = a[i]+[max(i, j)]
print()

```

0	1	2	3
1	1	2	3
2	2	2	3

**ПРИКЛАД 6.** Заповнити двовимірний масив  $5 \times 5$  у такий спосіб: елементам головної діагоналі присвоїти значення 1, елементам, що розташовані вище головної діагоналі, — значення 2, елементам, що розташовані нижче головної діагоналі, — значення 0.

```

a = [], [], [], [], []
for i in range(5):
    for j in range(5):
        if i == j: a[i].append(1)
        elif i < j: a[i].append(2)
        else: a[i].append(0)

```

1	2	2	2	2
0	1	2	2	2
0	0	1	2	2
0	0	0	1	2
0	0	0	0	1

### Питання для самоперевірки

1. Назвіть кількість рядків і кількість стовпців у двовимірному масиві, для збереження якого створено такий вкладений список:

а) `a = [[1, 2, 3, 4], [5, 6,1,1], [7, 8, 9,1]]`

б) `b = [], [], []`

в) `c = [[] for i in range(10)]`

2. Збережіть дані таблиці, поданої на рисунку, у змінній `mas`.

1	2	3	4
8	7	6	5
9	10	11	12

3. Запишіть фрагмент коду, призначений для введення значень двовимірного масиву з 5 рядків та 5 стовпців із клавіатури і збереження значень елементів масиву у вкладеному списку.

4. Створіть вкладений список **A** для збереження значень двовимірного масиву з 4 рядків та 6 стовпців, елементами якого є випадкові числа.
5. Запишіть оператори виведення значень масиву **A** на екран.
6. Дані з таблиці занесені до вкладеного списку **a**:

7	2	3	34
4	5	0	-3
1	3	8	1.2
15	2.5	-6	12

Знайдіть значення змінної **x** після виконання фрагмента коду:

```
x = a[3][2]+a[2][2]*a[2][3]
```

```
i = 1; j = 2; x = a[i][j]+a[i][j+1]
```

### **Вправа 35**

Створити програму для виконання завдання.

Завдання. Територіальна громада ухвалила рішення про здавання в оренду частини території, що має форму прямокутника розміром  $5 \times 5$  км. Вартість оренди кожної квадратної ділянки  $1 \times 1$  км була визначена з урахуванням локальних умов і занесена в таблицю **a**. З метою організації відкритого тендеру на оренду на вебсайті громади оприлюднено карту території. Напишіть програму, що дозволяє розрахувати вартість оренди для довільної прямокутної групи сусідніх ділянок, якщо протилежні кути групи ділянок розташовані в елементах таблиці з індексами **(i1, j1)**, **(i2, j2)**.

1) Створіть **Python file** із назвою **Оренда**. Завантажте модулі **tkinter**, **random**. Створіть вікно програми **tk** із заголовком **Вартість оренди**. Заверште програму оператором **tk.mainloop()**.

2) Створіть двовимірний масив об'єктів класу **Label**, який буде відображати карту території.

```
list_b=[[[],[],[],[],[]]]
```

```
for i in range (5):
```

```
    for j in range (5):
```

```
        list_b[i].append(Label(tk, bg = 'lightblue', width = 4))
```

```
        list_b[i][j].grid(row = i, column = j, padx = 2, pady = 2)
```

3) Опишіть функцію **output()**, призначену для виведення значень елементів масиву **a** до елементів масиву **list\_b**.

```
def output():
```

```
    for i in range (5):
```

```
        for j in range (5):
```

```
list_b[i][j].config(text = str(a[i][j]))
```

4) Заповніть масив **a** розміром  $5 \times 5$  випадковими числами в діапазоні (1, 10) і запишіть оператор виклику функції **output()**.

5) Додайте до вікна напис **Координати верхнього лівого кута** і створіть 2 об'єкти класу **Entry** для введення індексів ділянки, що знаходиться у лівому верхньому куті групи ділянок.

```
Label(tk, text = 'Координати верхнього лівого кута:').grid(row = 5,  
column = 0, columnspan = 5)
```

```
enx1 = Entry(tk, width = 3)
```

```
enx1.grid(row = 6, column = 1)
```

```
eny1=Entry(tk, width = 3)
```

```
eny1.grid(row = 6, column = 3)
```

Додайте до вікна напис **Координати нижнього правого кута** і створіть 2 об'єкти класу **Entry** для введення індексів ділянки, що знаходиться у правому нижньому куті групи ділянок.

Додайте об'єкт **envart** класу **Entry** для виведення загальної вартості оренди визначеної групи ділянок.

```
envart = Entry(tk, width = 6)
```

```
envart.grid(row = 10, column = 2)
```

6) Створіть командну кнопку — об'єкт **but** класу **Button**:

```
but = Button(text = 'Обчислити', command = but_click, width =  
14).grid(row = 9, column = 0, columnspan = 5)
```

Опишіть функцію **but\_click()** — обробник події **Натискання на кнопку**.

```
def but_click():
```

```
    x1 = int(enx1.get())
```

```
    y1 = int(eny1.get())
```

```
    x2 = int(enx2.get())
```

```
    y2 = int(eny2.get())
```

```
    vart = 0
```

```
    for i in range(x1, x2+1):
```

```
        for j in range(y1, y2+1):
```

```
            vart = vart+a[i][j]
```

```
    envart.delete(0, END)
```

```
    envart.insert(0, str(vart))
```

Випробуйте роботу програми.



### Комп'ютерне тестування

Виконайте тестове завдання 35 з автоматичною перевіркою результату.

## Практична робота 11. РОБОТА З ДВОВИМІРНИМИ МАСИВАМИ У ПРОГРАМІ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ

Завдання: запрограмувати введення з клавіатури і виведення значень двовимірного масиву за допомогою графічних компонентів.

Організувати заповнення двовимірного масиву **a** з 4 рядків та 4 стовпців за такими правилами:

- 1) заповнити масив за формулою  $a[i][j] = (i+1)*(j+1)$ ;
- 2) заповнити масив випадковими числами;
- 3) заповнити масив за формулою  $a[i][j] = \begin{cases} 0, & \text{якщо } i \leq j, \\ 1, & \text{якщо } i > j. \end{cases}$

Обладнання: комп'ютер із середовищем програмування **PyCharm**.

### Хід роботи

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

1. Створіть **Python file** із назвою **Масив**. Створіть вікно програми **tk** із заголовком **Масив** (рис. 1). Заверште програму оператором **tk.mainloop()**.

2. За допомогою генератора списків створіть вкладений список **a** розміром  $4 \times 4$  і заповніть масив нулями.

```
a = [[0]*4 for i in range(4)]
```

3. Створіть двовимірний масив об'єктів класу **Entry**:

```
list_en=[[], [], [], []]
```

```
for i in range (len(a)):
```

```
    for j in range (len(a[i])):
```

```
        en = Entry(tk, bg = 'lightblue', width = 6) # Об'єкт класу Entry
```

```
        en.grid(row = i, column = j, padx = 2, pady = 2)
```

```
        list_en[i].append(en) # Додаємо об'єкт en до i-го рядка масиву
```

```
list_en
```

4. Додайте до вікна програми віджет класу **Label** для виведення тексту **Після змін натисни Enter**.

5. Запишіть оператор виклику для об'єкта **tk** методу **bind()** для обробки натискання клавіші **Enter**.

```
tk.bind('<Return>', read)
```

6. Опишіть функцію **read()**, призначену для зчитування значень із текстових полів масиву **list\_en** і занесення їх до масиву **a**.

```
def read(event):
```

```
    for i in range (len(a)):
```

```
        for j in range (len(a[i])):
```

```
            x = int(list_en[i][j].get())
```

```
            a[i][j] = x
```

7. Опишіть функцію **output()**, призначену для виведення значень

елементів масиву **a** до полів елементів масиву **list\_en**.

```
def output():
```

```
    for i in range (len(a)):
```

```
        for j in range (len(a[i])):
```

```
            list_en[i][j].delete(0, END)
```

```
            list_en[i][j].insert(0, str(a[i][j]))
```

8. Створіть командну кнопку **b1** із заголовком **Завдання 1**.

```
b1 = Button(text = 'Завдання 1', command = b1_click, width = 14).grid(row = 5, column = 0, columnspan = 5)
```

9. Опишіть функцію-обробник події **b1\_click()**, призначену для виконання завдання № 1.

```
def b1_click():
```

```
    for i in range(len(a)):
```

```
        for j in range(len(a[i])):
```

```
            a[i][j] = (i+1)*(j+1)
```

```
    output() # Виклик функції, призначеної для виведення значень елементів масиву a до полів елементів масиву list_en
```

10. Створіть командну кнопку **b2** із заголовком **Завдання 2**, атрибуту **command** надайте значення **b2\_click**.

11. Опишіть функцію — обробник події **b2\_click()**, призначену для виконання завдання № 2.

12. Створіть командну кнопку **b3** із заголовком **Завдання 3**, опишіть функцію — обробник події **b3\_click()**, призначену для виконання завдання № 3.

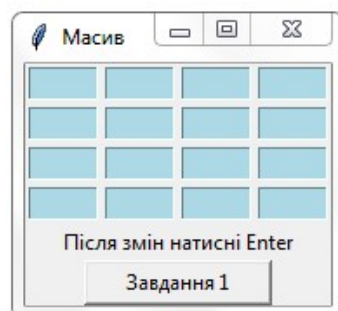


Рис. 1

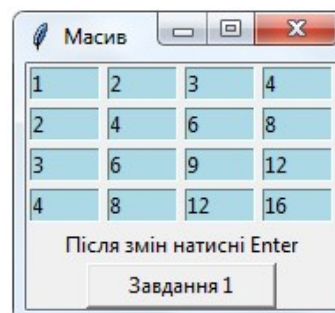


Рис. 2

Зробіть висновок: як організувати введення і виведення елементів двовимірного масиву.

## § 36. ОПРАЦЮВАННЯ ДВОВИМІРНИХ МАСИВІВ ДАНИХ

До типових задач опрацювання двовимірних масивів належать завдання на обчислення суми, добутку, кількості, середнього арифметичного, максимуму, мінімуму елементів кожного рядка або кожного стовпця, заданого рядка або заданого стовпця.

Уявіть, що ви керівник фірми, і вам потрібно проаналізувати динаміку продажів телефонів.

Продаж телефонів, шт.				
№	Марка	Січень	Лютий	Березень
1	Nokia	80	110	120
2	Samsung	80	90	80
3	Alcatel	60	70	130

Якщо вас цікавить сумарний обсяг продажів за січень, то вам потрібно виконати додавання елементів першого стовпця таблиці — числових даних про кількість проданих телефонів. Щоб знайти загальну кількість проданих телефонів марки Samsung, треба знайти суму елементів другого рядка. Якщо ж потрібно знайти загальну кількість проданих телефонів усіх марок за рік, то слід обчислити суму всіх елементів масиву.

### Додавання значень елементів двовимірного масиву

**ПРИКЛАД 1.** Знайти суму елементів масиву `a[4, 6]`.

```
s = 0
for i in range(len(a)):
    for j in range(len(a[i])):
        s += a[i][j]
print(s)
```

```
s = 0
for i in range(len(a)):
    s = s+sum(a[i])
print(s)
```

Рядки двовимірного масиву зберігаються як списки, тому при опрацюванні масиву по рядках зручно застосовувати методи роботи зі списками.

**ПРИКЛАД 2.** Знайти суму елементів 3-го рядка масиву `a[3, 4]` можна двома способами:

Звертаючись до кожного елемента рядка за його індексами

Викликавши метод `sum()` для списку, в якому зберігаються елементи 3-го рядка

<pre>s = 0 for j in range(len(a[2])):     s = s+a[2][j] print(s)</pre>	<pre>s = sum(a[2])</pre>
--	--------------------------

**ПРИКЛАД 3.** Застосуємо обидва способи, щоб знайти суми елементів у кожному рядку масиву:

```
for i in range(len(a)): # Зовнішній цикл по рядках
    s = 0 # Перед початком додавання сума обнуляється
    for j in range(len(a[i])):
        s = s+a[i][j] # Додавання елементів i-го рядка
    print(s) # Друк суми елементів i-го рядка
```

або:

```
for i in range(len(a)):
    print(sum(a[i]))
```

Якщо потрібно здійснити опрацювання двовимірного масиву по стовпцях, то за параметр зовнішнього циклу слід взяти індекс стовпця, а за параметр внутрішнього — індекс рядка.

**ПРИКЛАД 4.** Знайти суму 2-го стовпця масиву  
a = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]:

```
s = 0
for i in range(len(a)):
    s = s+a[i][2]
print(s) # Буде надруковано 13
```

		j=2	
1	2	3	4
5	6	1	1
7	8	9	1

**ПРИКЛАД 5.** Знайти суми по стовпцях масиву a:

```
for j in range(4): # Зовнішній цикл по стовпцях
    s = 0
    for i in range(len(a)):
        s = s+a[i][j] # Додавання елементів j-го
# стовпця
print(s) # Друк суми елементів j-го стовпця
```

		j=2	
1	2	3	4
5	6	1	1
7	8	9	1

**Пошук елементів двовимірного масиву, які задовольняють певній умові**



**ПРИКЛАД 6.** Знайти кількість парних елементів у кожному стовпці масиву **tabl**  $3 \times 4$ .

```

tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
for j in range(4):           # Зовнішній цикл по стовпцях
    k = 0                       # Перед початком перегляду кожного стовпця
    k обнуляється
    for i in range(3):         # Внутрішній цикл по рядках
        if tabl[i][j]%2 == 0:
            k = k+1
    print('Y ', j, '-му стовпці k = ', k)

```

**ПРИКЛАД 7.** Знайти в кожному рядку масиву **tabl**  $3 \times 4$  кількість елементів, які перевищують середнє арифметичне значення масиву:

```

tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
s = 0
for i in range(len(tabl)):
    s = s+sum(tabl[i])        # До значення суми додається сума елементів
    i-го рядка
    sr = s/12                 # Обчислення середнього арифметичного елементів
    масиву
    print('sr = ', sr)
    for i in range(len(tabl)): # Зовнішній цикл по рядках
        k = 0
        for j in range(len(tabl[i])):
            if tabl[i][j]>sr: k = k+1
    print('B ', i, '-тому рядку таких елементів ', k)

```

**ПРИКЛАД 8.** Заповнити список **b** значеннями максимальних елементів стовпців масиву **tabl**  $3 \times 4$ .

```
tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1]]
b = []
for j in range(4): # Зовнішній цикл по стовпцях
    m = tabl[0][j] # Перед початком перегляду кожного стовпця
    # m набуває значення елемента, що знаходиться
    # у цьому стовпці в нульовому рядку
    for i in range(1, 3): # Внутрішній цикл по рядках
        if tabl[i][j]>m: m = tabl[i][j]
    b.append(m) # Максимальний елемент j-го стовпця
# додається до списку b
print('b = ', b)
```



```
C:\Users\Lyceum\PycharmProjects\p
b= [7, 8, 9, 4]
```

Таким чином, опрацювання даних здійснюється шляхом застосування типових алгоритмів опрацювання одновимірних масивів.

**ПРИКЛАД 9.** Визначити номер рядка двовимірного масиву **tabl**, елементи якого утворюють найбільшу суму.

```
tabl = [[1, 2, 3, 4], [5, 6, 1, 1], [7, 8, 9, 1],[7, 2, 3, 4]]
ms = []
for i in range(len(tabl)):
    ms = ms+[sum(tabl[i])] # Сума елементів i-го рядка додається до
# списку ms
    m = max(ms) # Визначається максимальний елемент списку ms
    k = ms.index(m) # Визначається індекс елемента списку ms, який
# дорівнює m
print('m = ', m) # m = 25
print('k = ', k) # k = 2
```

Таким чином, опрацювання даних здійснюється шляхом застосування типових алгоритмів опрацювання одновимірних масивів до рядків або стовпців двовимірного масиву.

### Питання для самоперевірки

1. Запишіть оператор для обчислення суми елементів двовимірного масиву **tabl** ( $4 \times 4$ ).
2. Знайдіть суму елементів головної діагоналі масиву **tabl** ( $4 \times 4$ ).
3. Знайдіть максимальний елемент другого стовпця масиву **tabl** ( $5 \times 4$ ).
4. Знайдіть середнє арифметичне елементів 1-го рядка масиву **tabl** ( $6 \times 6$ ).
5. Заповніть список **a** середніми арифметичними значеннями елементів кожного рядка масиву **tabl** ( $6 \times 6$ ).
6. У 5 міст відправили вагони з десятьма різними видами товарів. Складіть програму для обчислення загальної маси продукції, відправленої в кожне місто, і загальної маси продукції кожного виду.

### Вправа 36

Створити програму для виконання завдання.

Завдання. Торговельна мережа має 4 магазини. Інформація про прибуток кожного магазину по місяцях за квартал (тис. грн) зберігається в рядках двовимірного масиву. Скласти консольну програму для опрацювання масиву даних.

- 1) Створіть **Python file** із назвою **Магазини**.
- 2) Заповніть список **tabl** даними таблиці.

Виконайте завдання опрацювання даних таблиці:

Номер магазину	Січень	Лютий	Березень
1	300	150	200
2	250	230	170
3	120	180	190
4	220	135	210

- 3) За введеним номером магазину вивести його середній прибуток за 3 місяці.
- 4) Визначити сумарний дохід мережі магазинів.
- 5) Визначити, який магазин за квартал мав найбільший прибуток.
- 6) Визначити, у якому місяці торговельна мережа отримала найбільший прибуток.

### Комп'ютерне тестування

Виконайте тестове завдання 36 з автоматичною перевіркою результату.

## § 37. ПОНЯТТЯ СКЛАДНОСТІ АЛГОРИТМІВ

Який час потрібний для виконання програми, що реалізує певний алгоритм? Чи вистачить ресурсів комп'ютера, щоб отримати результати обчислення за даним алгоритмом? На подібні питання відповідає теорія алгоритмів — розділ інформатики, що займається дослідженням складності алгоритмів для розв'язання задач на основі формально визначених моделей обчислювальних пристроїв.

**Складність алгоритму** — це кількісна характеристика, яка визначає час, що необхідний для виконання алгоритму (часова складність), і об'єм пам'яті, необхідний для його розміщення (ємнісна складність).

Складність алгоритмів зазвичай оцінюють за часом виконання або за використовуваною пам'яттю:

- часова складність — час виконання алгоритму;
- ємнісна складність — кількість умовних одиниць пам'яті, необхідних для роботи алгоритму.

Складність алгоритму дозволяє визначитися з вибором ефективного алгоритму серед тих, що побудовані для розв'язання конкретної проблеми.

### Оцінка складності

Часова та ємнісна складність тісно пов'язані між собою. Обидві є функціями від розміру вхідних даних. В обох випадках складність залежить від розмірів вхідних даних: масив зі 100 елементів буде оброблений швидше, ніж аналогічний із 1000. Кількість вхідних даних прийнято позначати літерою  $n$ . Варто розуміти, що нам треба оцінити не те, скільки операцій знадобиться алгоритму при конкретній кількості вхідних даних, а те, як він себе поводитиме при збільшенні їх кількості. При цьому мова йде не про точний час обчислень, який залежить від процесора, типу даних, мови програмування тощо. Оцінюється складність при прагненні розміру вхідних даних до нескінченності.

Терміном **елементарна операція** позначимо присвоювання та операції над значеннями простих типів (порівняння, додавання, множення тощо). При цьому вважається, що кожна елементарна операція виконується за однаковий час. За такого припущення час виконання програми прямо пропорційний кількості елементарних операцій у процесі виконання.



**Часова складність алгоритму** — характеристика продуктивності алгоритму, що визначається кількістю елементарних операцій, які потрібно виконати для реалізації алгоритму.

Часову складність оцінюють для найгіршого випадку і визначають як максимальний час, необхідний для обробки алгоритмом будь-якої множини з  $n$  елементів.

Для позначення оцінки складності алгоритмів використовують так звану  $O$ -нотацію — вираз  $O(f(n))$ , який означає, що час виконання алгоритму зростає з тією ж швидкістю, що і функція  $f(n)$ . Тобто ми хочемо отримати функцію зміни кількості операцій, які виконає алгоритм, залежно від кількості вхідних даних  $n$ .

$O$ -нотація показує, **як буде змінюватися обчислювальна складність алгоритму** зі зміною кількості вхідних даних у найгіршому для алгоритму випадку.

- Якщо час роботи алгоритму не залежить від обсягу вхідних даних, то його часову складність позначають як  $O(1)$ ; наприклад, для визначення значення третього елемента масиву не потрібно ні запам'ятовувати елементи, ні проходити по них декілька разів, тобто на обчислення результату для будь-якої кількості даних потрібен той самий час.

- *Лінійну складність  $O(n)$*  мають алгоритми, час виконання яких лінійно залежить від кількості вхідних даних, наприклад алгоритм пошуку найбільшого елемента в невідсортованому масиві, для чого потрібно переглянути всі  $n$  елементів масиву; алгоритм додавання/віднімання чисел із  $n$  цифр.

- *Квадратична складність  $O(n^2)$*  визначається, якщо час роботи алгоритму зростає пропорційно квадрату кількості оброблюваних елементів, тобто подвоєння розміру задачі вчетверо збільшує необхідний час. Квадратичну складність має алгоритм сортування бульбашкою, що виконує два вкладені цикли перебору масиву.

- *Кубічна складність  $O(n^3)$*  визначається, якщо час роботи алгоритму зростає пропорційно кубу кількості оброблюваних елементів, тобто подвоєння розміру задачі збільшує необхідний час у вісім разів. Припустимо, певним алгоритмом потрібно виконати  $2n^3 + 5n$  умовних операцій, щоб обробити  $n$  елементів вхідних даних. При збільшенні  $n$  на час роботи буде значно більше впливати зведення  $n$  у куб, ніж множення його на 2 або ж додавання  $5n$ .

### Приклади оцінювання складності алгоритмів

**ПРИКЛАД 1.** Нехай дано послідовність із нулів та одиниць і потрібно з'ясувати, чи є там хоч одна одиниця. Яку складність матиме алгоритм розв'язання цієї задачі?

Розв'язання. Нехай  $n$  — кількість символів у послідовності. Алгоритм буде послідовно перевіряти, чи немає одиниці в поточному місці заданої послідовності, а потім рухатися далі, поки вхід не скінчиться. Оскільки одиниця дійсно може бути тільки одна, для отримання точної відповіді на це питання в гіршому випадку доведеться перевірити всі  $n$  символів входу. Таким чином, алгоритм має складність  $O(n)$ , іншими словами, він лінійний.

**ПРИКЛАД 2.** Проаналізуємо програму для визначення кількості додатних елементів у кожному рядку масиву `tabl[n, n]`.

```
for i in range(n):          # Зовнішній цикл по рядках
    k = 0
    for j in range(n):
        if tabl[i][j]>0: k = k+1
    print(' В ', i, '-тому рядку таких елементів ', k)
```

У цьому алгоритмі змінна  $i$  змінюється від 1 до  $n$ . При кожній зміні  $i$  змінна  $j$  теж змінюється від 1 до  $n$ . Під час кожної з  $n$  ітерацій зовнішнього циклу внутрішній цикл теж виконується  $n$  разів. Загальна кількість ітерацій внутрішнього циклу дорівнює  $n \cdot n$ . Це визначає складність алгоритму  $O(n^2)$ .

Для однієї й тієї самої задачі можуть існувати алгоритми різної складності. Часто буває і так, що більш повільний алгоритм працює завжди, а більш швидкий — лише за певних умов. Під складністю задачі розуміють найменшу зі складностей алгоритмів її розв'язання.

Розглянемо різні методи пошуку елемента в деякому масиві. Залежно від організації інформації (впорядковані, невпорядковані дані) використовують різні методи пошуку.

**ПРИКЛАД 3.** Алгоритм послідовного пошуку елемента в масиві.

Сутність методу послідовного пошуку полягає в тому, що елементи масиву послідовно порівнюються із певним значенням (ключем) пошуку. Якщо наявна збіжність, то пошук закінчується, інакше — здійснюється перехід до наступного елемента. Отже, пошук припиняється або в разі досягнення кінця масиву, або в разі знаходження заданого елемента. Розглянемо випадок, коли елементи масиву не повторюються. Оскільки наявність та місцезнаходження елемента наперед невідомі, то пошук елемента масиву проводиться у циклі з передумовою, поки не знайдено відповідний елемент або поки не дійдемо до кінця масиву.

```
a = [23, 12, 3, 45, 6, 17, 8, 4, 21, 81]
```

```
k = 8
```

```
i = 0
```

```
n = 10
```

```
while a[i] != k and i < n-1:
```

```
    i = i+1
```

```
if i < n-1 or a[n-1] == k: print(i)
```

```
else: print('No')
```

Очевидно, що кількість елементарних операцій прямо пропорційна кількості порівнянь  $a[i] \neq k$ . У найгіршому випадку з ключем порівнюються всі елементи масиву, тоді кількість перевірок дорівнює  $n$ . Звідси найбільший час пошуку  $t$  є лінійною функцією від кількості елементів масиву, алгоритм має лінійну складність  $O(n)$ .

Перевагою методу простого пошуку є його простота та наочність. Недоліком методу є те, що в заголовку циклу доводиться здійснювати дві перевірки: на допустимість індексу й на рівність значення. Можна спростити алгоритм, позбувшись перевірки номера ( $i \leq n$ ) за рахунок збільшення масиву на один елемент у кінці, значення якого буде рівним  $k$  (так званий бар'єр). Додаткові операції з установки й зняття бар'єра окупаються спрощенням циклу, в якому витрачається основний час при пошуку. Особливо це позначиться при великих розмірах масиву. У загальному випадку час пошуку буде меншим, ніж у попередньому випадку.

```
a = [23, 12, 3, 45, 6, 17, 8, 4, 21, 81]
```

```
k = 8
```

```
a = a+[k]
```

```
i = 0
```

```
n = 10
```

```
while a[i] != k:
```

```
    i = i+1
```

```
if i == n: print('No')
```

```
else: print(i)
```

```
del(a[n])
```

```
print(a)
```

Шукаємо  $k=8$

i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=n
23	12	3	45	6	17	28	4	21	81	8

a=[23,12,3,45,6,17,8,4,21,81]

a=a+[k]

Знайдено  $a[i]=8$

$i==n?$  Так, отже, елемента  $k$  у масиві немає.

**ПРИКЛАД 4.** Алгоритм двійкового пошуку елемента в масиві.

В упорядкованому масиві процес пошуку елемента можна значно прискорити, застосувавши метод бінарного пошуку. Сутність цього методу

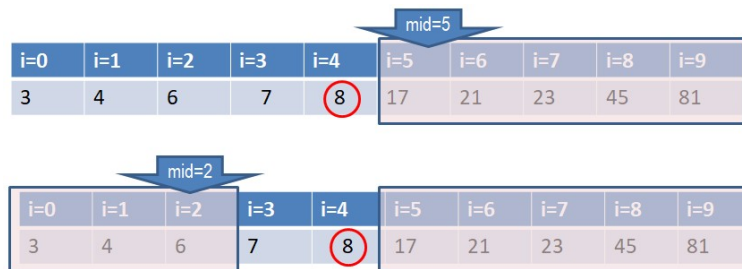
полягає в тому, що на кожному кроці частина масиву, в якій здійснюється пошук, зменшується вдвічі. Ділення здійснюється доти, поки не буде знайдено елемент або від масиву не залишиться жодного неопрацьованого елемента.

Припустимо, що масив впорядкований за зростанням. Поділимо його навпіл і порівняємо центральний елемент із ключем пошуку. Якщо елемент дорівнює ключу, то елемент знайдений. Якщо ключ більший за центральний елемент, то процес поділу й аналізу продовжується у правій частині масиву, інакше — у лівій частині. Таким чином, на кожному кроці потрібно зберігати індекси (номери) верхньої та нижньої меж частини масиву, що аналізується.

```

a = [23, 12, 3, 45, 6, 7, 8, 4, 21, 81]
a.sort()
k = 8
mid = len(a) // 2
low = 0
high = len(a)-1
while a[mid] != k and low <= high:
    if k > a[mid]:
        low = mid+1
    else:
        high = mid-1
    mid = (low+high) // 2
if low > high: print("No value")
else: print("ID = ", mid)

```



Цей метод дуже ефективний, оскільки, наприклад, для масиву з 1000 елементів результат визначається навіть у гіршому випадку після 10 кроків циклу, тоді як для методу послідовного пошуку в середньому потрібно буде 500 кроків. Однак впорядкування вихідного масиву є досить трудомісткою операцією і вимагає часу значно більшого, ніж проведення пошуку першими двома методами. Складність цього алгоритму дорівнює  $O(\log_2 n)$ .



**ПРИКЛАД 5.** Оцінимо складність алгоритму сортування бульбашкою. При виконанні алгоритму при кожному проході по масиву елементи порівнюються парами  $i$ , за необхідності, міняються місцями у заданому порядку (за зростанням або за спаданням).

$O(1)$  😊  
 $O(\log_2 n)$  😊  
 $O(n)$  😐  
 $O(n^2)$  😞

Розглянемо найгірший випадок для цього алгоритму — масив відсортовано у зворотному порядку. Для сортування доведеться виконати  $n$  проходів масивом і  $n$  перестановок на кожному проході, тобто зробити  $n \cdot n$  операцій порівняння й перестановок. Отже, складність цього алгоритму може оцінюватись як  $O(n^2)$ .

Знаючи це, ми можемо приблизно зрозуміти, які обсяги даних можна обробляти за цим алгоритмом, а які — ні. Наприклад, для сортування 1000 елементів алгоритму в гіршому випадку знадобиться 1 секунда, але сортування 100 000 елементів чекати доведеться майже 3 години.

Отже, однією з найбільш значущих та використовуваних властивостей є складність, тому при застосуванні різноманітних алгоритмів дуже важливо вміти правильно її оцінювати.

**ЦІКАВИНКА.** Адель Голдберг — американська вчена в галузі інформатики, відома своєю роботою у галузі об'єктно-орієнтованого програмування і графічних інтерфейсів та розробкою мови програмування **Smalltalk**.



### Питання для самоперевірки

1. Поясніть поняття складності алгоритму.
2. Дано послідовність з  $n$  символів і потрібно з'ясувати, чи є там хоч один символ **A**. Яку складність матиме алгоритм розв'язання цієї задачі?
3. При яких розмірах вхідних даних краще використати для пошуку елемента алгоритм послідовного пошуку? Алгоритм двійкового пошуку?
4. Яку складність має алгоритм сортування масиву вибором найбільшого елемента?
5. Наведіть приклади алгоритмів, які мають складність  $O(1)$ .
6. Наведіть приклади алгоритмів, які мають складність  $O(n)$ .

### Вправа 37

Скласти програму для виконання завдання.

Завдання. Визначити час роботи алгоритму залежно від розміру вхідних даних, якщо для розміру  $n$  вхідних даних час роботи алгоритму дорівнює  $f(n)$  мікросекунд.

1. Створіть **Python file** із назвою **Час роботи**. Завантажте модулі **tkinter, matplotlib.pyplot**.

Створіть вікно програми із заголовком **Час роботи**. Завершіть програму оператором **root.mainloop()**.

2. Створіть список **n** для збереження значень кількості вхідних даних.

```
n = [10, 100, 200, 300, 400, 500]
```

Заповніть список **f** значеннями часу роботи алгоритму при кількості вхідних даних **n[i]** (в секундах), якщо 1 мікросекунда =  $10^{-6}$  секунди.

```
f = []
```

```
for i in n:
```

```
    f = f+[i**2*1e-6]
```

3. Додайте у вікно програми віджет **lab1** класу **Label**.

```
lab1 = Label(text = "Кількість даних").grid(row = 0, column = 0)
```

Додайте віджет **box1** класу **Listbox**, занесіть до списку **box1** значення списку **n**:

```
box1 = Listbox()
```

```
box1.grid(row = 1, column = 0, rowspan = 6)
```

```
for p in n:
```

```
    box1.insert(END, p)
```

4. Додайте у вікно програми віджет **lab2** класу **Label** для виведення напису **Час роботи** і віджет **box2** класу **Listbox**, до списку якого занесіть значення списку **f**.

5. Додайте у вікно віджет **btn1** класу **Button**.

```
btn1 = Button(text = "Графік ", command = btn1_cl)
```

```
btn1.grid(row = 2, column = 3)
```

6. Опишіть функцію **btn1\_cl()**, призначену для побудови графіка за значеннями елементів масивів **n, f**.

```
plt.title('Час роботи, с')
```

```
plt.xlabel('n', color = 'gray')
```

```
plt.ylabel('f', color = 'gray')
```

```
plt.plot(n, f)
```

```
plt.show()
```

Випробуйте роботу програми.

### Комп'ютерне тестування

Виконайте тестове завдання 37 з автоматичною перевіркою результату.

## КОМП'ЮТЕРНИЙ СЛОВНИК

**Аргументи** — дані, які опрацьовує функція.

**Архів** — файл, який містить у стисненому вигляді один або більше файлів.

**Архіватори** — службові програми, за допомогою яких здійснюють операції архівування та розархівування.

**Архів із саморозархівуванням** — виконуваний файл, який після запуску автоматично розархівується.

**Архівування даних** — процес стиснення даних, що зберігаються у файлах.

**База даних** — упорядкована сукупність взаємопов'язаних даних.

**Бот** — програма, яка дає можливість зловмиснику таємно керувати вашим комп'ютером.

**Брандмауер** — це технічний пристрій або програмний засіб для контролю даних, що надходять до комп'ютера через мережу.

**Генератор списків** — синтаксична конструкція, що призначена для створення списків і заповнення їх значеннями за певними правилами.

**Деінсталяція програми** — процес видалення програми з комп'ютера.

**Діаграма** — графічне зображення, у якому числові дані подано геометричними фігурами.

**Діапазон** — сукупність комірок, які можна опрацьовувати як єдине ціле.

**Драйвери** — спеціальні програми, які забезпечують взаємодію комп'ютера з зовнішніми пристроями.

**Дропери** — виконувані файли, які самі не є вірусами, але призначені для встановлення шкідливих програм.

**Електронна система керування базою даних** — комплекс програмних засобів для забезпечення пошуку та опрацьовання даних у базі даних.

**Електронний підпис** — цифровий аналог звичайного підпису, яким можна скріпити всі електронні документи, що гарантуватиме їх цілісність і секретність.

**Запит** — засіб відбору та/або опрацьовання табличних даних за заданими правилами і критеріями.

**Захищений сайт** — сайт, який використовує для обміну даними протоколи захищеного зв'язку.

**Звіт** — підготовлений до публікації чи друку електронний документ.

**Зв'язок** — залежність між двома сутностями, охарактеризована дієсловом.

**Інсталяція програми** — процес встановлення програми на комп'ютер.

**Інформаційна безпека** — захищеність даних та інформаційної системи від випадкових або навмисних пошкоджень та несанкціонованих посягань.

**Комп'ютерний вірус** — програма, яка копіює та приєднує себе до інших програм і може негативно впливати на їх виконання та роботу комп'ютера в цілому.

**Ліцензія на програмне забезпечення** — це угода, яка надає право на його використання.

**Логічний вираз** — вираз, який набуває одного із двох значень: **True** — істина або **False** — хибність.

**Логічна функція** — функція, один або декілька аргументів якої — логічні вирази.

**Математичні функції в Excel** — функції, призначені для виконання обчислень над даними комірок: округлення, перетворення чисел тощо.

**Метод бульбашки** — метод упорядкування списку шляхом послідовного порівняння і обміну сусідніх елементів, якщо попередній елемент виявляється більше наступного.

**Одновимірний масив** — пронумерована послідовність значень одного типу, що мають спільне ім'я.

**Операційна система** — це комплект програм, призначених для керування ресурсами, обчислювальними процесами комп'ютера, а також для забезпечення взаємодії з користувачем.

**Переносна (або портативна) програма** — програма, для запуску якої на комп'ютері не потрібне встановлення.

**Програмне забезпечення** — сукупність програм, призначених для забезпечення роботи інформаційної системи.

**Рендеринг (вимальовування)** — процес побудови зображення з урахуванням усіх налаштувань.

**Розархівування** — процес повернення даних, записаних в архіві, до початкового стану.

**Системи керування базами даних** — комплекс засобів і методів для створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до них.

**Складений логічний вираз** — вираз, у якому кілька простих логічних виразів пов'язані логічними операціями.

**Складність алгоритму** — кількісна характеристика, яка визначає час, що необхідний для виконання алгоритму (часова складність), і об'єм пам'яті, необхідний для його розміщення (ємнісна складність).

**Скрипт-віруси** — програми, що потрапляють на комп'ютер через електронну пошту, маскуючись під документи.

**Сортування даних в Excel** — встановлення порядку розташування рядків за даними певного стовпця (стовпців).

**Спам** — небажана пошта переважно рекламного характеру.

**Список** — упорядкований набір об'єктів різних типів, який можна змінювати.

**Статистичні функції в Excel** — функції для аналізу значень діапазонів комірок.

**Сутність** — іменованій інформаційний об'єкт.

**Таблиці бази даних** — основний елемент СКБД, який містить розподілені за певними ознаками дані.

**Точка відновлення Windows** (контрольна точка відновлення) — збережена інформація про стан операційної системи і встановлених програм у певний час.

**Тривимірна графіка (3D-графіка)** — розділ комп'ютерної графіки, який присвячено методам створення зображень або відео шляхом моделювання об'ємних об'єктів у тривимірному просторі.

**Тривимірна модель** — візуальний об'ємний прообраз певного об'єкта.

**Тривимірне моделювання** — процес створення 3D-моделі об'єкта, що є його візуальним об'ємним образом.

**Трояни** — програми, що поширюються під виглядом нешкідливих програм та виконують несанкціоновані дії, передають її злочинцям через інтернет, самостійно відкривають сайти для зміни рейтингів, хакерських атак тощо.

**Умове форматування даних** — автоматична зміна формату комірок залежно від значення даних у них.

**Фільтрування даних в Excel** — відображення в таблиці тільки тих рядків, дані яких відповідають певним умовам (умовам фільтрування).

**Форма** — інструмент, який являє собою зручний бланк для перегляду, заповнення і редагування вмісту таблиць та перегляду результатів запитів, що дає змогу доручити цю роботу персоналу невисокої кваліфікації.

**Форматування носія даних** — процес розмітки області зберігання даних.

**Хробаки (черви)** — програми, які самостійно поширюються мережею, не «інфікуючи» інші файли.

**Часова складність алгоритму** — характеристика продуктивності алгоритму, що визначається кількістю елементарних операцій, які потрібно виконати для реалізації алгоритму.

## АЛФАВІТНИЙ ПОКАЖЧИК

### А

Архів 20

Архіватор 22

Архів із саморозархівуванням 22

Архівування даних 20

### Б

База даних 123

Боти 34

Брандмауер 34

### В

Впорядкування записів 144

Впорядкування полів таблиці 144

### Г

Генератор списків 177

### Д

Двовимірний масив 210

Деінсталяція програми 18

Діаграма 200

Діапазон 89

Драйвери 11

Дропери 34

### Е

Електронний підпис 35

### З

Запит 151

Захищений сайт 35

Звіт 167

Зв'язок 126

Зв'язний діапазон 87

### І

Інсталяція програми 15

Інформаційна безпека 32

## **К**

Комп'ютерний вірус 28

## **Л**

Ліцензія на програмне забезпечення 12

Логічний вираз 96

Логічна функція 96

## **М**

Математичні функції в Excel 99

Метод бульбашки 189

## **Н**

Незв'язний діапазон 88

## **О**

Одновимірний масив

## **П**

Переносна (портативна) програма 17

Посилання 90

Програмне забезпечення 9

## **Р**

Рендеринг 73

Розархівування 21

## **С**

Системи керування базами даних 127

Складність алгоритму 223

Скрипт-віруси 34

Сортування даних в Excel 106

Спам 36

Список 176

Статистична функція в Excel 99

Ступінь стиснення даних 21

Сутність 126

## **Т**

Точка відновлення Windows 40

Тривимірна графіка 45

Тривимірна модель 46

Тривимірне моделювання 46

Троянські програми 34

## **У**

Умовне форматування даних 109

**Ф**

Фільтрування даних 108, 148

Форма 161

Форматування носія даних 12

**Х**

Хробаки (черви) 34

**Ч**

Часова складність алгоритму 223



# ІНФОРМАТИКА

## 9 клас

### Особливості підручника:

- ▶ Мотивація навчальної діяльності
- ▶ Приклади практичного застосування набутих знань
- ▶ Вправи для формування та закріплення навичок роботи за комп'ютером
- ▶ Питання для самоперевірки за кожною темою
- ▶ Практичні роботи з покроковим описом виконання

### Інтернет-підтримка дозволить:

- ▶ здійснити онлайн-тестування за кожною темою
- ▶ ознайомитися з додатковими матеріалами до уроків

ВИДАВНИЦТВО  
**РАНОК**



**i** Інтернет-підтримка

за QR-кодом  
або посиланням  
[rnk.com.ua/100857](http://rnk.com.ua/100857)