

ВИДАВНИЦТВО
РАНОК



**НАВЧАННЯ
БЕЗ МЕЖ**

ІНФОРМАТИКА



УДК 004:37.016(075.3)

I-74

Авторський колектив:

Олена Бондаренко, Василь Ластовецький,
Олександр Пилипчук, Євген Шестопапов

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 05.02.2024 № 124)

Видано за рахунок державних коштів.

Продаж заборонено

Підручник створено відповідно до модельної навчальної програми
«Інформатика. 7–9 класи» для закладів загальної середньої освіти
(автори Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопапов Є. А.)

Інформатика : підруч. для 7 кл. закл. загал.
I-74 серед. освіти / [О. О. Бондаренко, В. В. Ластовецький,
О. П. Пилипчук, Є. А. Шестопапов]. — Х. : Вид-во
«Ранок», 2024. — 272 с. : іл.

ISBN 978-617-09-8755-6

УДК 004:37.016(075.3)



ISBN 978-617-09-8755-6

© Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопапов Є. А., 2024
© ТОВ Видавництво «Ранок», 2024

ДОРОГІ СЕМИКЛАСНИКИ ТА СЕМИКЛАСНИЦІ!

У 6 класі ви з'ясували особливості побудови й опрацювання растрових і векторних зображень, навчилися додавати анімації до об'єктів слайдів презентації, ознайомилися з табличним процесором і електронною таблицею. Ви дізналися більше про різні пристрої комп'ютера та цифрові технології, а також продовжили вивчення мови програмування Python.

У цьому навчальному році на вас очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Пригадаймо, як із ним працювати.

Навчальний матеріал підручника налічує шість розділів. Кожен розділ складається з параграфів і практичних робіт. Кожен параграф містить теоретичні відомості за темою уроку, приклади практичного застосування отриманих знань, матеріали за рубриками «Повторюємо», «Питання для самоперевірки», «Вправи», «Комп'ютерне тестування».

«Повторюємо» допоможе вам пригадати відомості, які ви вивчали в попередніх класах і які будуть корисні для засвоєння нового матеріалу. «Питання для самоперевірки» допоможуть з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером. «Вправи» складаються із завдань теоретичного і практичного спрямування. Вони призначені для самостійного та групового виконання за комп'ютером.

У підручнику ви знайдете покроковий опис *практичних робіт*. Щоб виконати кожну з цих робіт, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді ви зможете успішно застосувати свої знання та виконати завдання.

До підручника пропонується електронний інтерактивний додаток, який містить додаткові матеріали до вправ, тестові завдання з автоматичною перевіркою відповідей, 3D-моделі, навчальні відео, інтерактивні вправи, довідкові матеріали й ін.

Бажаємо натхнення та успіхів!

Авторський колектив

У тексті використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Зверніть увагу



Дайте відповідь на питання



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



Знайдіть відповідь в інтернеті



Виконайте тестові завдання за комп'ютером

РОЗДІЛ 1

СЛУЖБИ ІНТЕРНЕТУ



§ 1. Поштові служби інтернету

§ 2. Робота з електронною поштою

Практична робота 1. Створення електронної поштової скриньки

§ 3. Налаштування поштової скриньки

§ 4. Особливості електронного листування

§ 5. Етичне та безпечне користування електронною поштою

§ 6. Використання інтернет-ресурсів. Гіперпосилання в листах

§ 7. Колективна робота з документами

Практична робота 2. Використання хмарних технологій для колективного опрацювання документів

§ 8. Штучний інтелект. Інтернет речей

ПОВТОРЮЄМО



Ви вже знайомі з поняттям *комп'ютерної мережі*, розрізняєте *локальну мережу*, що об'єднує комп'ютери, розташовані на невеликій відстані один від одного, і *глобальну мережу*, яка об'єднує мільйони комп'ютерів і локальних мереж. Найбільшою глобальною мережею є «*Інтернет*».

Ви знаєте, що *сторінки сайтів (вебсторінки)* є гіпертекстовими документами, тобто вони містять *гіперпосилання* — об'єкти (слова, малюнки тощо), що дають змогу перейти до іншого місця в документі, до іншого документа, на інший сайт.

Ви пам'ятаєте про можливі загрози під час роботи в інтернеті, дотримуетесь *правил безпечного користування* ним, критично оцінюєте інформацію.

1. Що таке локальна мережа; глобальна мережа?
2. Які можливості надає нам інтернет?
3. Якою пошуковою системою ви користуєтесь?
4. Що таке гіпертекстовий документ; гіперпосилання?
5. Опишіть відомі вам інтернет-загрози.
6. Яких правил безпечного користування інтернетом ви дотримуетесь?




У цьому розділі ви розширите свої знання про службу електронної пошти, створите власну поштову скриньку. Дізнаєтесь, що таке хмарні технології та хмарний диск, попрацюєте з онлайнним перекладачем, колективно опрацюєте документи. З'ясуєте, що таке штучний інтелект і як його застосовують.

§ 1. ПОШТОВІ СЛУЖБИ ІНТЕРНЕТУ

Із давніх часів люди використовували для обміну повідомленнями малюнки, сигнальні вогні, дими, згодом — телеграф, пошту й ін. Із появою комп'ютера та мереж з'явилися нові можливості для спілкування. Нині існує багато служб (сервісів) інтернету, і цей перелік постійно поповнюється. Із деякими службами інтернету ви вже знайомі.

Електронна пошта є однією з найдавніших комунікаційних служб інтернету; нині вона повсюдно замінює «паперову» пошту.

 **Електронна пошта** (від англ. *electronic* — електронний, *mail* — пошта) — це служба обміну повідомленнями через комп'ютерну мережу з використанням поштової системи адресації.

Принципи функціонування електронної пошти

Як звичайний лист складається з поштового конверта, що містить службову інформацію, та вкладеного листа, так і електронний лист — із заголовків і тіла листа (рис. 1.1). До листа можна прикріпляти (вкладати) файли з текстом, малюнками.

Паперовий лист	Електронний лист
<p>Адреса відправника, індекс Іванчук Іванна Іванівна вул. Київська, 123, кв. 45 м. Київ Київська обл. 12345</p>  <p>Адреса одержувача, індекс Петренко Петро Петрович вул. Львівська, 67 с. Лугове, Бродівський р-н Львівська обл. 56789</p>	 <p>Нове повідомлення</p> <p>Від Іванчук Іванна Іванівна <lvanchuk@ukr.net> Кому Petrenko@gmail.com Тема Вітання</p> <p>Доброго дня, шановний Петре, вітаю тебе з днем народження, бажаю успіхів, міцного здоров'я</p> <p>Без зачіпок</p> <p>Надіслати</p>

Рис. 1.1

Між клієнтськими комп'ютерами (відправника й адресата) існує проміжна ланка — поштовий сервер (рис. 1.2).

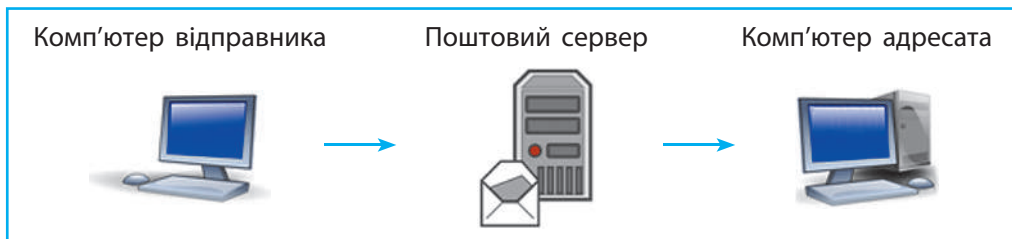


Рис. 1.2

Поштовий сервер — це комп'ютер, який забезпечує роботу електронної пошти завдяки встановленому на ньому спеціальному програмному забезпеченню. Саме на поштовий сервер потрапляє лист, надісланий відправником, і зберігається в електронній скриньці адресата. Сучасні поштові служби надають можливість керувати поштовою скринькою за допомогою вебінтерфейсу або засобами браузерів Google Chrome, Mozilla Firefox, Safari та ін.

Структура електронної адреси

Кожна електронна поштова скринька має унікальну адресу. Щоб електронний лист потрапив до потрібної скриньки, відправник має зазначити адресу поштової скриньки одержувача.

Електронна адреса складається із двох частин: *назви поштової скриньки* та *доменного імені поштового сервера*, на якому розташована ця скринька, розділених символом @.

Символом @ позначають прийменник «на» (англ. *at*). Так, `bober@ukr.net` слід розуміти як «bober на сервері `ukr.net`» (рис. 1.3).



Рис. 1.3



Є класичні правила формування адреси поштової скриньки.

- Допускається використання літер латинського алфавіту, арабських цифр і деяких символів («_», «.»).
- Допускається використання лише одного символу @.
- Не допускається використання пропусків, розділових знаків, символів кирилиці.

Разом із тим в електронній адресі можна знехтувати регістром літер, тобто великі й малі літери в ній рівнозначні.



Розглянемо варіанти запису однієї й тієї самої адреси.

bober@ukr.net
BOBER@UKR.net

Bober@ukr.net
BOBER@UKR.NET

BOBER@ukr.net

- Історія виникнення електронної пошти пов'язана з іменем американського інженера і програміста Рея Томлінсона (рис. 1.4). 1971 року він створив програму для пересилання текстових повідомлень між комп'ютерами. Томлінсон придумав використовувати в адресі символ @ (читається «ет»), який називають по-різному: «равлик», «собачка» тощо.



Рис. 1.4

Добираючи назву поштової скриньки (логін користувача), не варто робити її занадто довгою або складною, адже тоді її важко запам'ятати чи повідомити комусь, скажімо, телефоном.

Правила створення паролів

До створення пароля варто поставитися серйозно. Надійність пароля зростає, якщо дотримуватися кількох правил.

- Не використовуйте як пароль своє (рідних, друзів) ім'я, прізвище, ініціали, дату народження, номери телефонів тощо.
- Створюйте пароль завдовжки понад 8 символів.
- Комбінуйте малі й великі літери, цифри, розділові знаки.

Варто створити надійну основу пароля для різних випадків реєстрування. Це підвищить вірогідність того, що людина не заплутається, запам'ятовуючи власні паролі.

- 2 Надійні основи паролів: lvaN!007, LiNa#2020.
 2 Ненадійні основи паролів, які легко вгадати: Ivan, Lina, ABCD.

Щоб до кожного сайту створити унікальний пароль, можна до основи пароля додавати ознаки поточного сайту.

- 3 Пароль з основою LiNa#2020 для створення скриньки на ukr.net може виглядати так: UkLiNa#2020, для реєстрації на Facebook так: FaLiNa#2020, для Instagram так: InLiNa#2020.

Створення поштової скриньки

Поштову скриньку можна безкоштовно створити на багатьох поштових серверах; на рис. 1.5 наведено логотипи найбільш популярних із них.



Рис. 1.5

Для створення скриньки на поштовому сервері потрібно:

- 1) відкрити сторінку реєстрації на поштовому сервері;
- 2) ввести затребувані дані та підтвердити реєстрацію.

Іноді факт реєстрації потрібно підтвердити, прийнявши телефонний дзвінок або СМС-повідомлення з цифровим кодом підтвердження. Номер телефону або адреса іншої скриньки можуть знадобитися, наприклад, для відновлення пароля. Деякі поштові сервери, такі як bigmir.net, для цього пропонують дати відповідь на контрольне запитання, вибране під час створення скриньки.

Реєструючи поштову скриньку, ви створюєте обліковий запис, який дає змогу використовувати додаткові інтернет-сервіси.

- ! Логін користувача, вибраний під час реєстрації скриньки, змінити неможливо. Особисті дані, що вводяться в ході її створення, можна змінити в налаштуваннях.

Після роботи на чужому комп'ютері необхідно вийти зі свого облікового запису, щоб запобігти втручанню сторонніх осіб.



Питання для самоперевірки

1. Назвіть відомі вам служби інтернету.
2. Які функції виконує поштовий сервер?
3. Яка структура електронної адреси?
4. Як створити надійний пароль?
5. Для чого може знадобитися телефон або адреса додаткової скриньки після реєстрації на поштовому сервері?
6. Що з наведеного можна змінити після створення поштової скриньки: назву скриньки, прізвище та ім'я користувача/ користувачки, пароль?



Вправа 1

- Створити електронну поштову скриньку на поштовому сервері mail.ukr.net.
- 1. Придумайте пароль для поштової скриньки. Запустіть браузер. Відкрийте сторінку реєстрації електронної скриньки на поштовому сервері mail.ukr.net (або іншому за вказівкою вчителя/вчительки). Виберіть команду Створити скриньку.
 - ⚙️ 2. Уведіть реєстраційні дані. Установіть прапорець на згоду щодо опрацювання персональних даних. Прапорець імпорту листів з інших скриньок не встановлюйте.
 3. Клацніть кнопку Отримати код. Отримайте СМС-повідомлення.
 4. Уведіть отриманий код у віконці, що з'явилося, і натисніть кнопку Зареєструвати скриньку.
 5. Запишіть адресу своєї поштової скриньки. Запам'ятайте (або запишіть на окремому аркуші) пароль.
 6. Клацніть значок меню поштової скриньки і вийдіть зі свого облікового запису командою Вихід. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106139).



§ 2. РОБОТА З ЕЛЕКТРОННОЮ ПОШТОЮ

Існує думка, що електронна пошта втрачає позиції. Проте за даними компанії з телекомунікаційних досліджень Radicati (США), кількість її користувачів зростає приблизно на 3% щороку.

Переваги електронної пошти

Факти, що свідчать про переваги використання поштової скриньки порівняно з іншими засобами комунікації, такі:

- електронною поштою можна безперешкодно відправити повідомлення на будь-яку поштову адресу;
- можливість користування інтернет-сервісами залежить від електронної пошти: реєстрація на більшості з них передбачає наявність скриньки для підтвердження реєстрації тощо;
- понад 50 років існування поштового сервісу свідчать про те, що електронна пошта є надійним засобом комунікації.

Вхід до поштової скриньки

Для початку роботи потрібно увійти до поштової скриньки. Щоб увійти до поштової скриньки, необхідно:

- 1) в адресному рядку браузера записати адресу поштового сервера, на якому створено скриньку — з'явиться вікно реєстрації та/або входу до скриньки (рис. 2.1);

Вхід до gmail.com

Вхід до bigmir.net

Вхід до outlook.com

Вхід до ukr.net

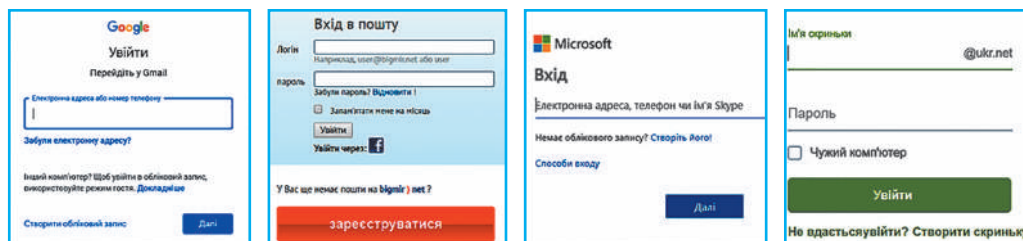


Рис. 2.1

- 2) у вікні зазначити назву скриньки (логін користувача) або адресу скриньки чи телефон залежно від поштового сервера;
- 3) у цьому ж вікні або після клацання кнопки Далі в наступному вікні ввести пароль і натиснути кнопку Увійти.

Під час роботи на чужому комп'ютері задля безпеки слід поставити відповідний прапорець і/або в спливному вікні браузера скасувати запам'ятовування пароля.

Якщо ви забули пароль, слід увести свою поштову адресу (або логін користувача) і натиснути кнопку Забув пароль або Відновити пароль тощо, і далі керуватися запропонованими інструкціями.

Вихід із поштової скриньки

Як ви вже знаєте, після завершення роботи на чужому комп'ютері або обліковому записі необхідно вийти з електронної пошти.

Щоб **вийти з поштової скриньки** (ще говорять «із пошти», «з облікового запису», «з акаунта»), слід знайти команду чи кнопку Вихід. Вона розташовується окремо або відкривається значком меню у вигляді ≡, ⋮ або 👤 та зазвичай міститься в правому верхньому кутку вікна скриньки (рис. 2.2).

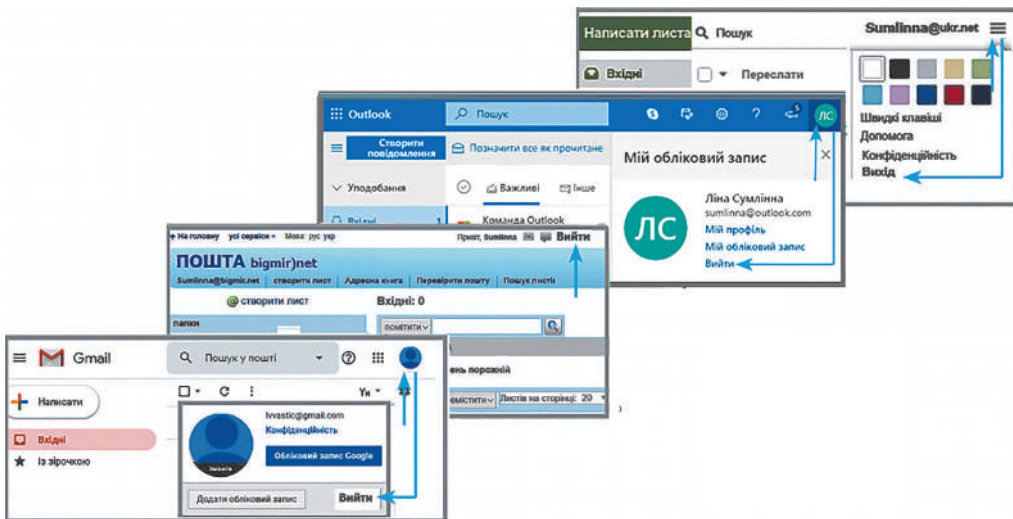


Рис. 2.2

Створення та надсилання електронного листа

Нині без електронного листування майже неможливо обійтися ні в діловій сфері, ні в повсякденному житті.

Щоб **створити електронного листа**, потрібно:

- 1) натиснути кнопку Написати листа, або Написати;
- 2) набрати текст повідомлення в тілі листа;
- 3) зазначити тему листа (поле Тема) й адресу отримувача (поле Кому);
- 4) за потреби додати адреси інших отримувачів у полі Кому, або Копія; натиснути кнопку Надіслати.

Адресу отримувачів листа бажано вказувати в останню чергу.

▶ Пересилання файлів

Разом із повідомленням можна надсилати фотографії, музику тощо. Щоб **надіслати файл** із комп'ютера відправника до скриньки адресата, його слід прикріпити до листа й надіслати.

Щоб у скриньці ukr.net **прикріпити і переслати файл**, потрібно:

- 1) підготувати лист, клацнути кнопку Файл у полі Прикріпити;
- 2) у вікні провідника, що відкриється, знайти й виділити файл (файли — утримуючи натисненою клавішу Ctrl);
- 3) натиснути кнопку Відкрити — файл прикріпиться до листа: завантажиться на поштовий сервер, після чого ім'я файлу з'явиться під кнопкою Файл у вікні скриньки (рис. 2.3);

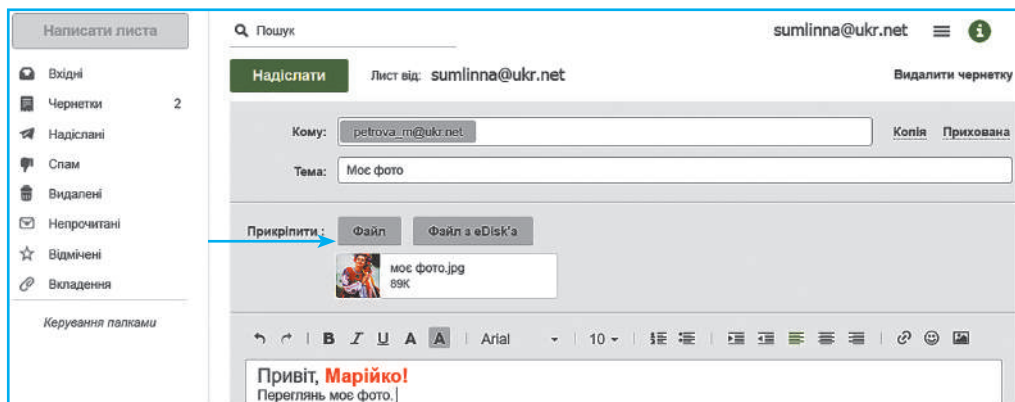


Рис. 2.3

4) після підготовки листа клацнути кнопку Надіслати — лист разом із файлом (файлами) буде надіслано всім адресатам.

Після цього у вікні скриньки з'явиться напис Ваш лист надіслано (рис. 2.4). Лист буде збережено в папці Відправлені в скриньці відправника, а після одержання — у скриньці адресата в папці Вхідні.

► Редагування та форматування повідомлення

Текст електронного листа можна редагувати та форматовувати у вікні вбудованого текстового редактора (рис. 2.5). Такі дії аналогічні опрацюванню документів у текстовому процесорі, із яким ви знайомі.

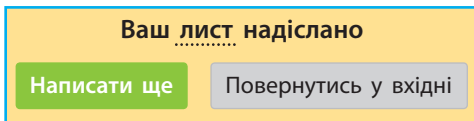


Рис. 2.4

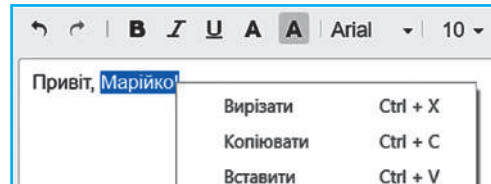


Рис. 2.5

Перегляд та переміщення листів до папок

У папці Вхідні можна переглянути нові електронні листи.

Якщо лист має вкладений файл, який необхідно зберегти на комп'ютері, слід клацнути кнопку Завантажити, а в діалоговому вікні збереження зазначити потрібну папку.

У поштової скриньці є стандартні папки:

- Вхідні — для отриманих листів;
- Відправлені — для надісланих листів;
- Видалені — для видалених листів;
- Спам — для небажаних листів;
- Чернетки — для ненадісланих.



Окрім зазначених стандартних папок, користувачі та користувачки можуть створювати й інші папки (наприклад, Важливо, Різне тощо) на свій розсуд. Це допомагає впорядковувати електронні листи та краще в них орієнтуватися.

У випадках коли виникає потреба показати або приховати папку, слід скористатися командою Керування папками (рис. 2.6).

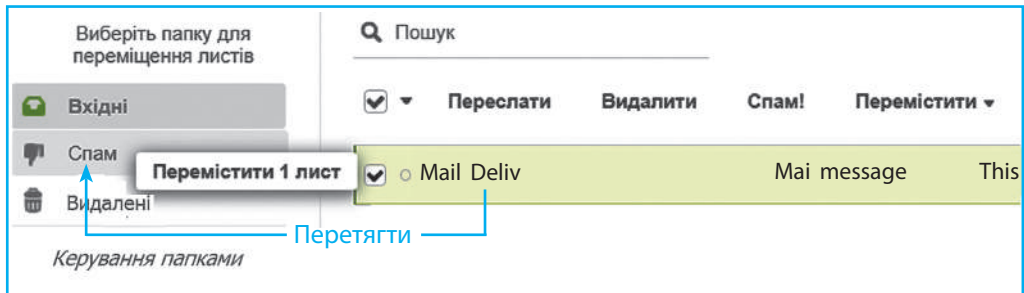


Рис. 2.6

Електронний лист можна перенести до потрібної папки перетягуванням теми на назву папки або за допомогою команди Перемістити, що над списком листів. Перед цим лист потрібно відкрити або позначити в списку прапорцем (див. рис. 2.6).




Питання для самоперевірки

1. Опишіть переваги електронної пошти порівняно з «паперовою» поштою або з іншим засобом комунікації.
2. Як увійти до власної поштової скриньки та як із неї вийти?
3. Який порядок створення та надсилання електронного листа?
4. Поміркуйте, навіщо в електронному листі потрібне поле Тема. Знайдіть відповідь в інтернеті.
5. Які стандартні папки існують у поштовій скриньці?
6. Як створити папку в поштовій скриньці і перемістити до неї листа?



Вправа 2

1. Оформити листа зі вкладеним файлом і надіслати електронною поштою.
1. Запустіть браузер. Знайдіть в інтернеті зображення символу @ і збережіть його на комп'ютері.

2. Зайдіть до своєї поштової скриньки. У вікні скриньки натисніть кнопку Написати листа. Напишіть у темі листа своє прізвище, клас. Уведіть у тіло листа, наприклад, такий текст:
Привіт! Я вмію користуватися електронною поштою.
3. Збільшіть розмір символів тексту. Змініть колір слова Привіт!, наприклад, на червоний.
4. Прикріпіть до листа збережений графічний файл (див. п. 1).
-  5. У рядку Кому введіть дві адреси — вчителя/вчительки та однокласника або однокласниці. Відправте листа (кнопка Надіслати).
Упевніться в тому, що всі листи надіслані адресатам.
6. Якщо лист не надійшов, перевірте написання адреси та повторіть спробу.
Вийдіть із поштової скриньки. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106140).




ПРАКТИЧНА РОБОТА 1 СТВОРЕННЯ ЕЛЕКТРОННОЇ ПОШТОВОЇ СКРИНЬКИ

Завдання: створити електронну поштову скриньку на пошто-
вому сервері gmail.com; оформити й надіслати листа
зі вкладеним графічним файлом кільком особам.




Обладнання: комп'ютер, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Придумайте назву і пароль для поштової скриньки, відкрийте браузер. Зайдіть на сторінку створення облікового запису google.com.
2. У вікні, що з'явилося, виберіть команду Створити обліковий запис.
3. За вказівками введіть потрібні дані та створіть обліковий запис.
4. Увійдіть до облікового запису, розгорніть меню додатків Google значком .

Відкрийте електронну поштову скриньку .

5. У вікні поштової скриньки натисніть кнопку Написати. Уведіть, наприклад, такий текст:
Доброго дня. Це мій обліковий запис Google.
6. Відформатуйте слово Google за зразком: 
Для форматування натисніть кнопку , що розгорне додаткову панель інструментів.
7. Упишіть у рядок Тема свої прізвище, ім'я, клас.
8. Кнопкою  приєднайте до листа графічний файл, збережений на вашому комп'ютері.
9. У рядку Одержувачі (Кому) упишіть адресу вчителя та однокласника або однокласниці.
10. Натисніть кнопку Надіслати і перевірте, чи всі листи надійшли.
11. Якщо лист до адресата не дійшов, перевірте правильність написання адреси і повторіть спробу.
12. Вийдіть із поштової скриньки. Завершіть роботу.

Зробіть висновок як створити електронну поштову скриньку на поштовому сервері; як надіслати листа зі вкладеним графічним файлом кільком особам.

§ 3. НАЛАШТУВАННЯ ПОШТОВОЇ СКРИНЬКИ

Розгляньмо особливості налаштування профілю користувача на прикладі облікового запису Google та поштового сервісу Gmail.

Профіль користувача

Профіль користувача — сукупність персональних даних і налаштувань робочого середовища.

Після створення облікового запису можна відредагувати профіль користувача: змінити прізвище, дату народження, пароль.

► Редагування персональних даних

Перед початком редагування особистих даних потрібно **увійти до свого облікового запису**, відредагувати персональні дані.

Для цього необхідно виконати такі дії (рис. 3.1):

- 1) у правому верхньому кутку вікна клацнути значок із літерою (зазвичай це перша літера в імені користувача) або із зображенням і кнопку **Обліковий запис Google**;
- 2) у меню зліва вибрати команду **Особисті дані**.

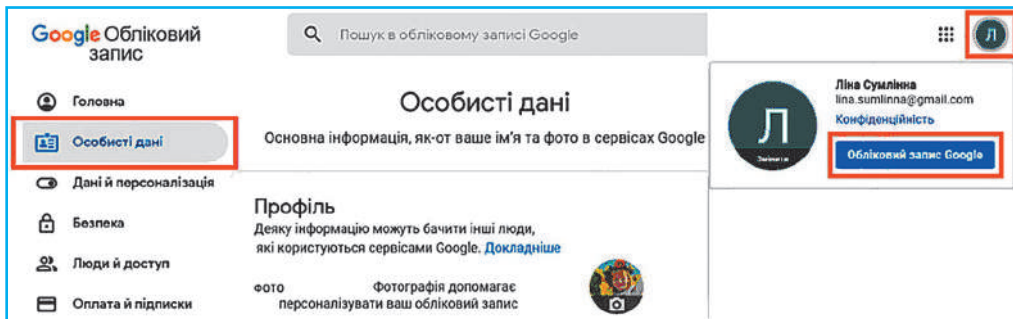



Рис. 3.1

► Додавання персонального зображення



Якщо ви хочете додати до свого профілю персональне зображення (наприклад, власну фотографію) чи змінити його, то слід:

- 1) зберегти на комп'ютері фотографію (інше зображення);
- 2) увійти до свого облікового запису Google, у правому верхньому кутку вікна клацнути значок із літерою чи зображенням;
- 3) у віконці, що з'явилося, клацнути кнопку Змінити;
- 4) керувачись підказками, завантажити фотографію;
- 5) вибрати кадрувальною рамкою потрібний фрагмент;
- 6) клацнути кнопку Зробити фотографією профілю.

Не варто хвилюватися, якщо після додавання зображення воно одразу не з'явилося на значку вашого облікового запису .


Згодом додане зображення з'явиться: .

Налаштування поштової скриньки

Для зручності користування поштовим сервісом і автоматизації деяких процесів під час написання листів у поштових скриньках передбачено низку налаштувань, наприклад, щоб відкрити поштову скриньку, слід клацнути:  → .

▶ Налаштування мови інтерфейсу Gmail

Щоб налаштувати мову інтерфейсу Gmail, потрібно:

- 1) розгорнути значком  меню налаштувань (рис. 3.2);
- 2) вибрати команду Налаштування і відкрити вкладку Загальні;
- 3) вибрати потрібну мову та клацнути кнопку Зберегти зміни в нижній частині вікна налаштувань.

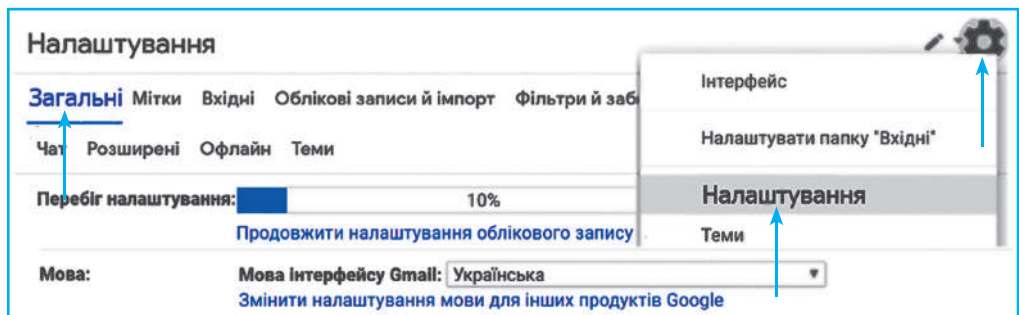


Рис. 3.2

▶ Налаштування автоматичного підпису до листа

Діловий лист варто закінчувати підписом, у якому зазначаються дані відправника: ім'я, прізвище, організація, телефон тощо. Доцільно налаштувати автоматичний підпис.

Для налаштування автоматичного підпису потрібно:

- 1) відкрити вікно налаштувань на вкладці Загальні (див. рис. 3.2);
- 2) прокрутити список і вибрати форму Підпис (рис. 3.3);
- 3) установити перемикач у режим підпису і ввести текст підпису; клацнути кнопку Зберегти зміни.

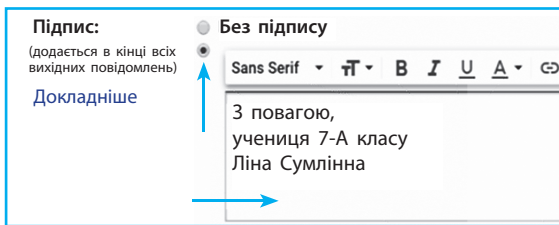



Рис. 3.3

▶ Перенаправлення повідомлень

Як зробити так, щоб листи автоматично перенаправлялися з однієї поштової скриньки на іншу? Щоб задіяти службу перенаправлення повідомлень і перенаправляти на поштовий сервер gmail.com листи з інших скриньок, наприклад, з ukr.net, потрібно:

- 1) зайти до скриньки, з якої ви бажаєте перенаправляти листи до Gmail, і відкрити вікно налаштування пересилання листів:  → Поштові програми → Пересилання листів (рис. 3.4);

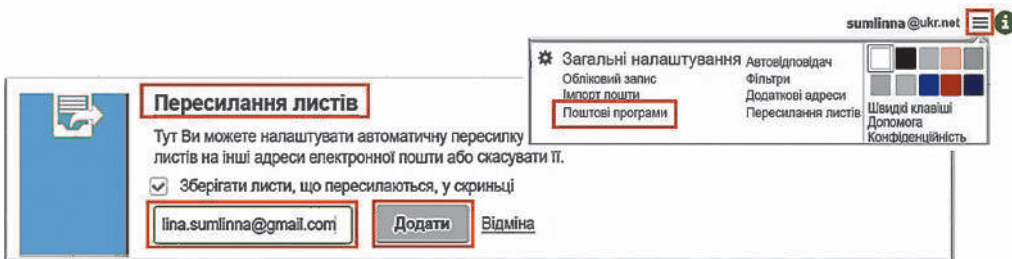


Рис. 3.4

- 2) увести адресу скриньки Gmail, до якої планується пересилати листи, та клацнути кнопку Додати — на пошту Gmail буде надіслано лист із формою підтвердження;
- 3) зайти до скриньки Gmail, відкрити лист, отриманий від ukr.net, і клацнути подане в листі посилання — на адресу скриньки на сервері ukr.net, надійде лист із повідомленням (рис. 3.5).

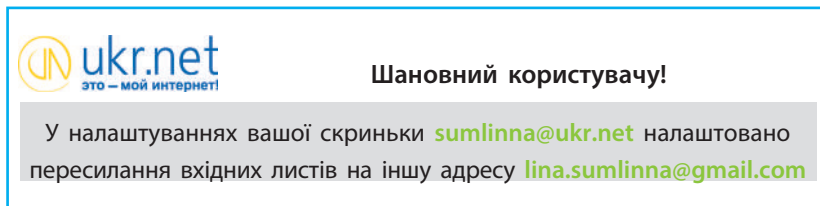


Рис. 3.5

Електронну скриньку можна налаштувати так, що її поштова програма сама збиратиме листи з інших скриньок і навіть перенесе всі наявні в інших скриньках листи до основної скриньки.

► Групування листів

Для комфортної роботи з кореспонденцією листи варто згрупувати за темами, адресами, призначенням тощо. Окрім стандартних папок: Вхідні, Відправлені, Видалені (Кошик), Спам, Чернетки, — можна створити додаткові папки. Так само як для впорядкування файлів на електронних носіях їх розташовують у папках, так і електронні листи в поштовій скриньці можна розподілити в окремі папки за різними ознаками.

У поштовій скриньці Gmail папки називаються **мітками**.

Для створення у поштовій скриньці нової папки (мітки) слід:

- 1) розгорнути головне меню (значок ☰ ліворуч угорі) (рис. 3.6);
- 2) клацнути кнопку Більше (рис. 3.6, а);
- 3) вибрати команду Створити нову мітку (рис. 3.6, б);
- 4) увести у вікні, що з'явилося, назву папки та клацнути кнопку Створити (рис. 3.7).

Прапорець у полі Розмістити мітку під означає, що ви бажаєте розмістити папку в тій, яку виберете зі списку.

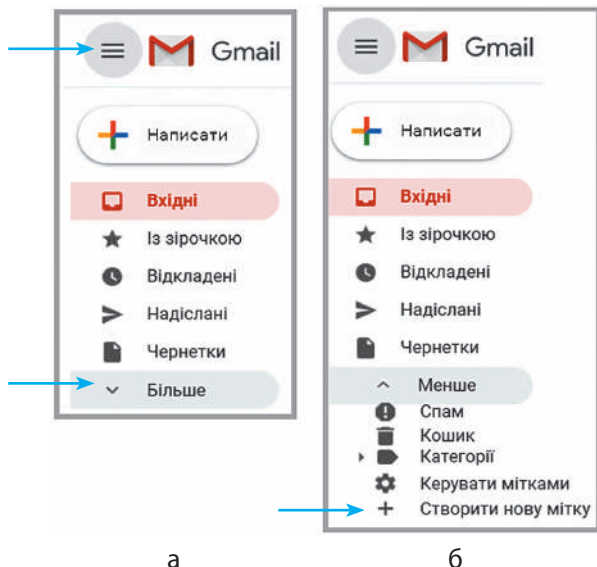


Рис. 3.6

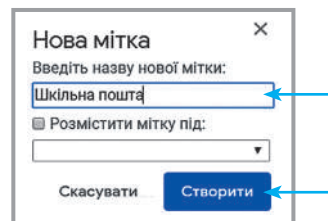




Рис. 3.7

Лист можна **перемістити** з однієї папки до **іншої** мишею або інструментом . Слід розгорнути папку (наприклад, Вхідні); вибрати лист і перетягнути до іншої папки, що в головному меню, або скористатися інструментом  Перемістити у (рис. 3.8).

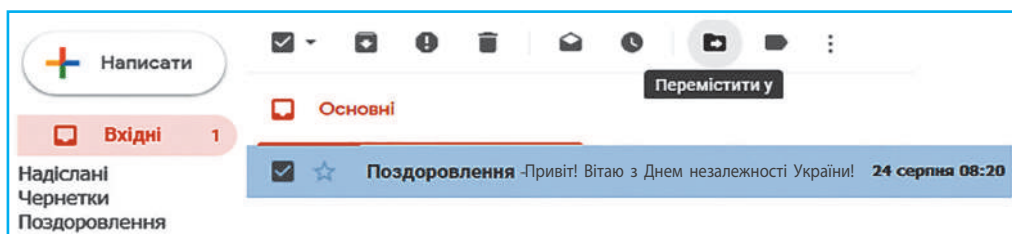



Рис. 3.8

Налаштування фільтрів дає змогу розподіляти листи під час надходження за певними правилами та умовами, пересилати тощо.

Для **створення фільтра** слід: увійти до скриньки й клацнути:  → команда Налаштування → вкладка Фільтри і заблоковані адреси; на вкладці вибрати команду Створити новий фільтр.

Далі у вікні зазначити параметри листів, які треба відфільтрувати, клацнути кнопку Створити фільтр; у новій формі прапорцем позначити необхідні дії та клацнути кнопку Створити фільтр.



Питання для самоперевірки

1. Що називають профілем користувача?
2. Як додати персональне зображення до профілю користувача?
3. Як додати до листів автоматичний підпис?
4. Як перенаправити лист зі скриньки mail.ukr.net до gmail.com?
5. Як створити в поштової скриньці папку?
6. Яка функція фільтра в поштової скриньці?



Вправа 3

- Навчитися редагувати профіль користувача та налаштовувати параметри поштової скриньки Google.
- 1. Запустіть браузер, увійдіть до свого облікового запису Google і установіть персональне зображення.
 2. Відредагуйте профіль, напишіть українською мовою з великої літери своє прізвище та ім'я, виправте помилки, якщо є. Установіть українську мову інтерфейсу поштової скриньки.
 3. Налаштуйте автоматичний підпис листа для ділового листування з учителем. Наприклад, так:

З повагою,
учениця 7-А класу Ліна Сумлінна
 - ⚙️ 4. Відкрийте нову вкладку браузера. Увійдіть до своєї поштової скриньки на mail.ukr.net. Налаштуйте перенаправлення листів із mail.ukr.net на gmail.com (звертайте увагу на листи від поштових служб).
 5. Створіть у поштової скриньці Gmail папку Різне.
 6. Перемістіть до папки Різне листи від поштових служб.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106141).



§ 4. ОСОБЛИВОСТІ ЕЛЕКТРОННОГО ЛИСТУВАННЯ

Розгляньмо особливості роботи з електронною поштою на прикладі одного з найпопулярніших поштових серверів — Gmail від американської корпорації Google (рис. 4.1). Gmail почав працювати з квітня 2004 року.



Рис. 4.1

Вхід до облікового запису Google

Зареєстрованому користувачу/користувачці Google перш за все необхідно **увійти до свого облікового запису**. Для цього слід:


- 1) увести в адресному рядку браузера назву gmail.com і відкрити сторінку входу;
 - якщо з'явиться список із кількох облікових записів, вибрати потрібний;
 - якщо потрібного запису в списку немає, натиснути Вибрати інший обліковий запис;
- 2) за потреби ввести свою поштову адресу або номер телефону, а потім увести пароль.

Увійти до облікового запису можна також зі сторінки google.com.



Створення та використання адресної книги



Електронна пошта широко застосовується в діловому листуванні. Наприклад, деяким фахівцям за день доводиться надсилати десятки, а то й сотні листів різним адресатам. Для зручного пошуку й упорядкування контактів: адрес, прізвищ, імен, телефонів та іншої інформації — використовують адресну книгу.

 **Контакт** — це сукупність даних про адресата в комунікаційних сервісах інтернету.

Адресна книга — це впорядкована сукупність контактів.

Адресна книга призначена для зберігання, редагування та використання контактів електронної поштової скриньки. Час від часу контакти варто переглядати, редагувати або вилучати.


Щоб **переглянути адресну книгу та відредагувати контакти**, необхідно виконати такі дії:

- 1) увійти до свого облікового запису;
- 2) у правому верхньому кутку сторінки натиснути значок меню Додатки Google ; відкрити додаток Контакти .

Створення та використання списків розсилки

Уявіть ситуацію, коли електронного листа треба розіслати кільком адресатам одночасно. Адреси двох або трьох осіб можна ввести вручну (ви вже так робили, виконуючи вправи). А якщо листа потрібно надіслати більшому числу адресатів? Тут варто користатися списком розсилки.

▶ Створення списку розсилки

 **Список розсилки** — це засіб електронної пошти, який надає можливість об'єднати певну кількість контактів у єдину групу для спрощення процесу розсилання листів.


Списки розсилки у Gmail створюють додаванням до контактів міток приналежності до певної групи. Для створення списку розсилки (мітки) слід застосувати команду Створити мітку. Потім за цими мітками можна вибрати потрібних адресатів для одночасного надсилання їм листів. Для більш зручного користування можна створити різні мітки, наприклад Учителі, Друзі, Клас та ін.

▶ Уведення контактів до списку розсилки

Щойно створений список розсилки порожній і не містить інформації, окрім назви списку. Додати нові контакти до списку можна, скориставшись кнопкою Створити контакт (рис. 4.2).

Щоб **увести наявні контакти до списку розсилки** (позначити контакт міткою), потрібно виконати такі дії:

- 1) відкрити додаток Контакти  та вибрати Контакти або Інші контакти;

- 2) установити прапорці (галочки) на вибраних контактах;
- 3) клацнути кнопку Керування мітками  (рис. 4.3);
- 4) вибрати потрібну назву в списку міток, який з'явиться;
- 5) клацнути команду Застосувати (Застос.).

Потрібно пам'ятати, що після цього позначені міткою контакти зі списку Інші контакти перемістяться до списку Контакти.

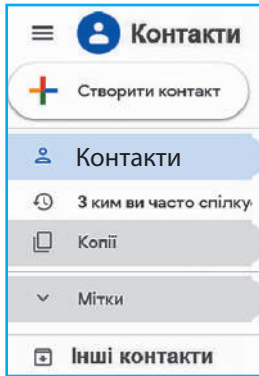


Рис. 4.2

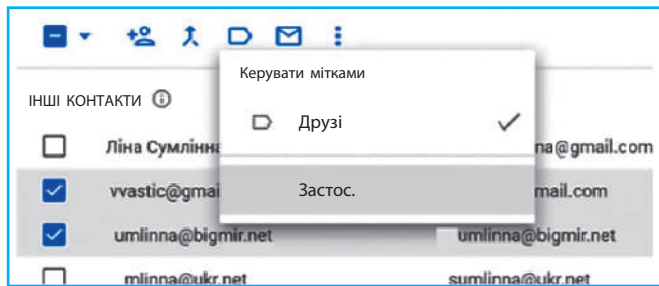


Рис. 4.3

► Надсилання листів за списками розсилки (мітками)

Надіслати листа за списком розсилки (мітками) можна як з адресної книги, так і з поштової скриньки.

Розгляньмо обидва способи.

Для надсилання листа з адресної книги потрібно:

- 1) відкрити адресну книгу  →  і список контактів (або вибраних мітку);
- 2) виділити контакти й клацнути значок Надіслати лист ;
- 3) написати й надіслати листа.

Для надсилання листа з поштової скриньки потрібно:

- 1) зайти до поштової скриньки;
- 2) під час написання листа в рядку Кому ввести назву списку розсилки (мітки), наприклад, Друзі, яка є в адресній книзі, — з'являться всі адреси, позначені міткою.

Тепер залишається написати електронного листа й надіслати його одночасно за всіма адресами зі списку розсилки.


**Питання для самоперевірки**

1. Як увійти до свого облікового запису?
2. Як переглянути адресну книгу; відредагувати контакти?
3. Що таке список розсилки (мітки)?
4. Як створити список розсилки (мітки)?
5. Які ви знаєте способи надсилання листа за списком розсилки (мітками)?
6. Яке призначення адресної книги?

**Вправа 4**

■ Створити список розсилки та надіслати створений електронний лист кільком особам.

➤ 1. Запустіть браузер. Увійдіть до свого облікового запису Google.

2. Відкрийте Адресну книгу (додаток Контакти .

Додайте до списку Контакти три контакти: вчителя/вчительки та двох однокласників або однокласниць (кнопка

).

⚙️ 3. Створіть мітку Інформатика (команда Створити мітку).

Позначте міткою Інформатика три щойно створені контакти.

4. Відкрийте список контактів із міткою Інформатика.

⚙️ 5. Відкрийте поштову скриньку (додаток Gmail .

Напишіть текст листа з поясненням, навіщо потрібні списки розсилки (мітки).

👤 6. Надішліть листа вчителю/вчительці, однокласникам або однокласницям, скориставшись списком розсилки (міткою Інформатика).

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106142).



§ 5. ЕТИЧНЕ ТА БЕЗПЕЧНЕ КОРИСТУВАННЯ ЕЛЕКТРОННОЮ ПОШТОЮ

Під час роботи в інтернеті важливо дотримуватися етичних і безпекових норм. Які є вимоги до оформлення листів?

Правила безпечного користування поштою

Часто правила спілкування людей засобами інтернету називають **нетикетом** (від англ. *net* — мережа та *etiquette* — етикет). Наведімо деякі правила нетикету.

- Заповнюйте всі поля, призначені для службової інформації.
- Ураховуйте призначення листа, чітко окреслюйте тему.
- Пишіть стисло, дотримуйтеся правил правопису.
- Будьте ввічливими, стриманими й толерантними.

Робота з електронною поштою може становити певну загрозу, якщо не дотримуватися правил безпеки. Пригадаймо ці правила.

- Не повідомляйте приватну інформацію незнайомим людям.
- Не розголошуйте паролі облікових записів.
- Не відповідайте на листи із заманливими пропозиціями чи погрозами, повідомляйте про них близьких і вчителів.
- Не переходьте за посиланнями від незнайомих людей.

Запобігання спаму та фішингу

Зі спамом знайомі майже всі, хто відвідує мережі.

 **Спам** — небажані листи переважно рекламного характеру.


Спам може містити посилання на шкідливі сайти, вкладення з вірусами, пропозиції з метою переказу коштів на певні рахунки (рис. 5.1).


Спамери збирають адреси потенційних клієнтів на сайтах за допомогою спеціальних програм-роботів. Найкращий спосіб боротьби з ними — не дати змоги розпізнати електронну адресу.



Рис. 5.1

Для захисту від спаму існує кілька правил. Ось деякі з них:

- Не оприлюднюйте свою електронну адресу та номери телефонів на сайтах, або кодуйте адресу, або подавайте у форматі картинки, яку розпізнає тільки людина.
- Не відповідайте на спам (не переходьте за посиланнями, не відкривайте картинок), адже це збільшить його кількість.
- Позначайте підозрілі листи як спам (у скриньці Gmail це значок ) , і тоді вони переміщуватимуться в папку Спам. Одним зі способів шахрайства за допомогою спаму є фішинг.

 **Фішинг** (від англ. *phishing*, за співзвучністю з *fishing* — риболовля) — шахрайські методи отримання особистих даних (банківських реквізитів рідних і ін.) або коштів через інтернет.

До фішингу належить створення підробних сайтів, розсилка листів, які маскуються під офіційні повідомлення банків, повідомлення про призи, виграші, спадок та ін. (рис. 5.2). Щоб убезпечити себе від фішингу, не слід реагувати на підозрілі повідомлення, відкривати листи від незнайомих людей тощо.



Рис. 5.2

Особливості написання електронного листа

Основними вимогами до написання листа, як електронного, так і паперового, є ввічливість і лаконічність. Разом із тим існують і деякі особливості, притаманні написанню саме електронних листів. Це низка засобів для позначення емоцій у листі, вставлення зображень до тексту тощо.

Так, *смайл* (від англ. *smile* — усмішка) — це схематичне зображення людського обличчя, яке відпочатку повернуто набік і в якому зазвичай не зображають «ніс» (рис. 5.3).

Смайли можуть складатися з різноманітних символів. Окрім суму та радості, вони можуть висловлювати й інші емоції, наприклад, підморгування ;), подив :0, регіт :D та ін.

Обличчя	Смайли
	:-) :)
	:-(:(

Рис. 5.3

Під час написання листа виклик меню смайлів здійснюється інструментом Вставити смайли 😊.

! На відміну від приватного, в діловому листуванні передусім цінуються точність і інформативність. В офіційних документах смайли не використовують.

► Позначення наголосів і знаків оклику

Іноді в запозичених словах, деяких прізвищах або, наприклад, в омографах — словах з однаковим написанням, але різним звучанням і значенням — для уникнення подвійного тлумачення доцільно ставити наголос (наприклад, замОк — зАмок).

Фрагмент тексту, введений великими літерами (рис. 5.4), зазвичай сприймається як вираження негативних емоцій і прояв роздратованості, агресії. Писати великими літерами неетично.

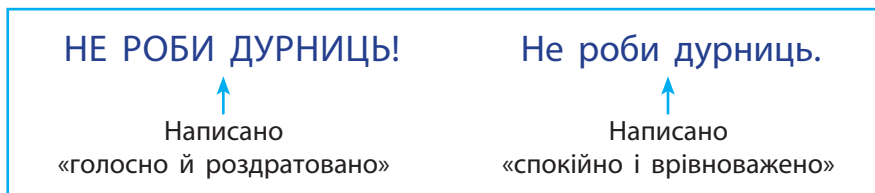


Рис. 5.4

► Додавання зображень до тексту

Для надання тексту привабливого вигляду в тіло листа можна вставити малюнок і оформити як, наприклад, вітальну листівку (рис. 5.5).

Для цього можна скористатися відомими вам сполученнями клавіш: Ctrl+C (скопювати), Ctrl+V (вставити), або інструментом Вставити фотографію 🖼️.



Рис. 5.5

На відміну від вкладеного файлу такі зображення не треба відкривати, отримувач одразу бачить текст листа із малюнком. Окрім того, зображення можуть бути анімованими (формат GIF), що «оживить» лист і надасть йому привабливого вигляду.



Питання для самоперевірки

1. Назвіть правила етикету електронного листування.
2. Назвіть правила безпечного листування. Яких із них ви дотримуетесь?
3. Що таке спам? Назвіть правила захисту від спаму.
4. Що таке фішинг? Як убезпечити пошту від фішингу?
5. Чому не варто писати текст електронного листа великими літерами?
6. Що таке смайлики і коли їх слід застосовувати?



Вправа 5

- Створити й надіслати електронного листа із графічним зображенням.
1. Запустіть браузер. Знайдіть в інтернеті дату державного свята України. Наприклад, День захисників і захисниць України, День Українського козацтва, Всесвітній день пошти, Міжнародний день Чорного моря, День працівників освіти.
 2. Знайдіть в інтернеті анімоване (gif) зображення до вибраного свята та збережіть на комп'ютері.
 3. Увійдіть до свого облікового запису Google. Напишіть текст електронного листа з привітанням із вибраним святом.
 4. Додайте до тексту листа збережене анімоване зображення (не плутайте з прикріпленим файлом зображення). За необхідності змініть у листі розмір зображення, щоб воно не було завеликим або замалим.
 5. Надішліть листа однокласнику чи однокласниці та вчителю/вчительці.
 6. Поцікавтеся, чи отримали адресати листи і чи бачать вони анімоване зображення. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106143).




§ 6. ВИКОРИСТАННЯ ІНТЕРНЕТ-РЕСУРСІВ. ГІПЕРПОСИЛАННЯ В ЛИСТАХ


Уявімо, ви розмістили або знайшли на каналі YouTube начальне відео і хочете поділитися ним із друзями. Які є способи?

Гіперпосилання в листах


Друзям і подругам можна надіслати не сам файл, а гіперпосилання (його ще називають посиланням) на інтернет-ресурс.

 **Гіперпосилання** — це пов'язаний з адресою ресурсу об'єкт сторінки, клацанням якого здійснюється перехід до цього ресурсу.

Щоб надіслати в листі посилання на ресурс, достатньо скопіювати адресу ресурсу в адресному рядку браузера, а потім вставити її до тексту листа. Тепер, клацнувши зазначену адресу, отримувач перейде до відповідного ресурсу в інтернеті.

 Гіперпосилання в електронному листі може мати такий вигляд:
<https://www.slovnnyk.ua/index.php?swrd=%D1%96%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%B0%D1%86%D1%96%D1%8F>

Щоб не захаращувати лист рядками адрес ресурсів, можна зробити гіперпосиланням фрагмент тексту листа. Для цього під час написання листа достатньо виконати такі дії:

- 1) відкрити нову вкладку браузера та знайти потрібний ресурс;
- 2) скопіювати з адресного рядка браузера адресу знайденого ресурсу (сполучення клавіш Ctrl+C);
- 3) перейти на вкладку зі сторінкою поштового сервера, де розпочато написання листа;
- 4) виділити в листі фрагмент тексту та клацнути значок  Вставити посилання (рис. 6.1, с. 34);
- 5) у вікні, що з'явилося, ввести скопійовану адресу (Ctrl+V).

Як це зробити, зрозуміло з рис. 6.2.

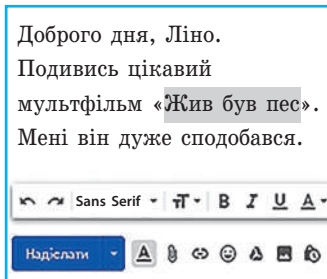


Рис. 6.1

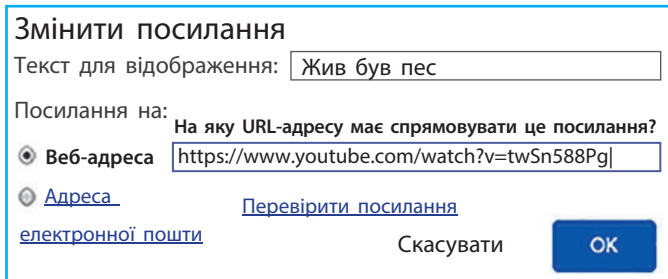


Рис. 6.2




Текст із гіперпосиланням буде виділено кольором і підкресленням, а наведений вказівник миші змінюватиме вигляд (на  на ) , що свідчитиме про наявне гіперпосилання (рис. 6.3).



Рис. 6.3

Хмарні сервіси

Із розвитком інтернету все більшого поширення набуває використання віддалених ресурсів, так званих хмарних сервісів.

 **Хмарні сервіси** — сервіси, пов'язані з наданням постійного доступу до віддалених інтернет-ресурсів (серверів, сховищ).


Хмарні сервіси мають багато переваг, таких як:

- користування послугами поштових серверів;
- зберігання даних у віддалених сховищах: Dropbox, SkyDrive, Google Drive та ін.;
- використання вебпрограм: Office Web Apps (Office 365), сервісів Google-Apps та ін.;
- користування елементами дистанційної освіти;
- телекомунікація засобами Skype, Viber, Zoom та ін.;
- використання ресурсів віддалених серверів для обчислень.


Нині найактивнішими постачальниками хмарних сервісів є корпорації Google і Microsoft (США).



Хмарний диск

Одним із хмарних сервісів є хмарний диск.

 **Хмарний диск** — сховище даних, розташоване на віддаленому сервері й доступне через інтернет.

На відміну від традиційного жорсткого хмарний диск «не прив'язаний» до фізичного пристрою користувача. Головне мати підключення до мережі. Для використання хмарного диску Google Диска достатньо створити обліковий запис на сайті google.com.

 Власнику скриньки Gmail для збереження кореспонденції та інших файлів безкоштовно надається хмарний диск обсягом 15 Гб. За певних умов або за додаткову плату можна отримати й більший обсяг.

Щоб відкрити Google Диск, слід зайти до облікового запису та кнопкою  відкрити перелік додатків і вибрати  Диск (рис. 6.4).

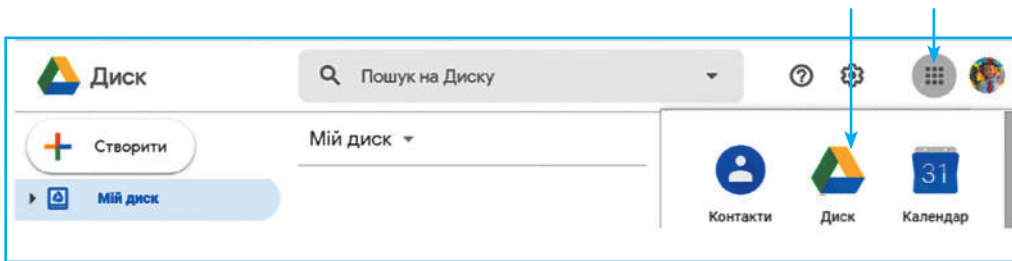


Рис. 6.4

На Google Диск можна завантажити файли й папки з комп'ютера (рис. 6.5).

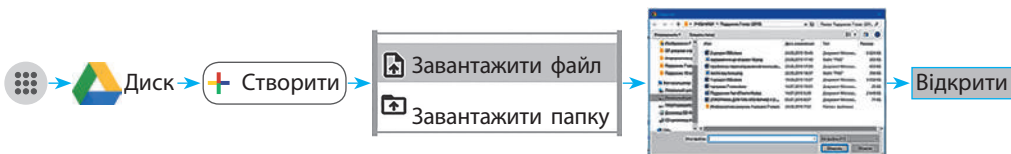


Рис. 6.5

Створити папку та/або необхідні вам документи також можна безпосередньо на Google Диску (рис. 6.6).

Google Диск надає можливість створити текстовий документ, таблицю, зображення, мапу, форму опитування тощо. Крім браузера, жодного додаткового програмного забезпечення для цього не потрібно.

Отже, такий спосіб зберігання даних є зручним і ефективним, він забезпечує можливість співпраці в режимі реального часу.

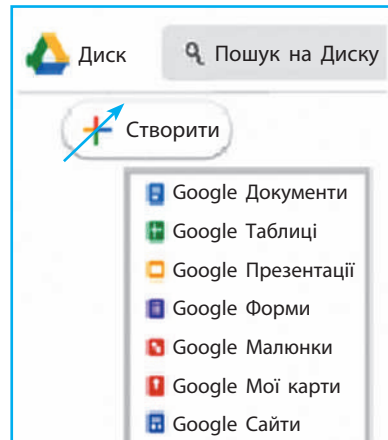


Рис. 6.6



Онлайн-перекладачі

До хмарних сервісів належать і онлайн-перекладачі.

Минулого року ви вже мали досвід роботи з Google Перекладачем (рис. 6.7). За його допомогою можна перекладати з понад 100 різних мов; він надає можливість голосового введення тексту, прослуховування правильної вимови вихідного або перекладеного тексту.



Рис. 6.7

Після здійснення перекладу Google Перекладач пропонує варіанти, розташовані за частотою використання. Клацнувши текст, можна запропонувати свій варіант перекладу, який буде враховано в подальшому. Для голосового введення з мікрофона треба клацнути значок , для прослуховування тексту — значок .

Ті, хто користується браузером Google Chrome, мають можливість перекладати цілі сторінки сайтів мовою, на яку налаштовано браузер. Достатньо викликати контекстне меню в будь-якому місці сторінки і вибрати команду Перекласти такою мовою...

У мобільній версії перекладача, окрім іншого, можна ввести текст фотокамерою. Також варта уваги наявна функція Діалог, за допомогою якої під час з'єднання з інтернетом можна спілкуватися в голосовому режимі.



Питання для самоперевірки

1. Що таке гіперпосилання?
2. Як додати гіперпосилання до фрагмента тексту листа?
3. Наведіть кілька прикладів використання хмарних сервісів.
4. Що таке хмарний диск і як його відкрити?
5. Як створити на хмарному диску презентацію?
6. Як перекласти іншою мовою текст та/або прослухати його за допомогою онлайн-перекладача?



Вправа 6

- Використати онлайн-ресурси для створення презентації.
- 1. Запустіть браузер. Зайдіть до свого облікового запису Google.
 2. Відкрийте хмарний диск і створіть на ньому порожню презентацію з трьох слайдів.
 3. Знайдіть в інтернеті зображення прапора однієї з тих країн, що є дружніми до України.
 4. Додайте на *перший* слайд зображення прапора вибраної країни.
 5. Знайдіть в інтернеті державний гімн вибраної країни мовою оригіналу й додайте на *другий* слайд.
 6. Перекладіть за допомогою онлайн-перекладача гімн вибраної країни українською мовою та додайте текст на *третій* слайд. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106144).



§ 7. КОЛЕКТИВНА РОБОТА З ДОКУМЕНТАМИ

Для підвищення продуктивності роботи іноді доцільно опрацювати документи одночасно кільком особам. Це дають можливість робити хмарні технології, що передбачають віддалене опрацювання та зберігання даних.

Колективна робота з документами

Документи, збережені на хмарному диску, може опрацювати будь-яка людина, що має до них доступ.

► Типи та рівні доступу до файлів

Доступ до файлів надає власник хмарного диска (рис. 7.1).

Існують *два типи доступу* до файлу:

1) усім, хто має посилання на файл:

увімкнено: , вимкнено: .

2) лише тим особам, кому надано доступ за їхніми іменами або електронними адресами.

Існують *три рівні доступу* до файлу:

1) перегляд; 2) коментування; 3) редагування.

! Кожен наступний рівень надає всі можливості попередніх рівнів доступу. Так, найвищий рівень — редагування — передбачає можливість перегляду та коментування даних у файлах.

► Відкриття доступу

Відкрити доступ до даних можна у два способи:

- у контекстному меню файлу вибрати команду Надати доступ (рис. 7.2);
- у режимі онлайн-перегляду файлу клацнути Спільний доступ (рис. 7.3).



Рис. 7.1

У формі, що відкриється, потрібно встановити необхідні тип і рівень доступу та клацнути кнопку Готово (рис. 7.4).

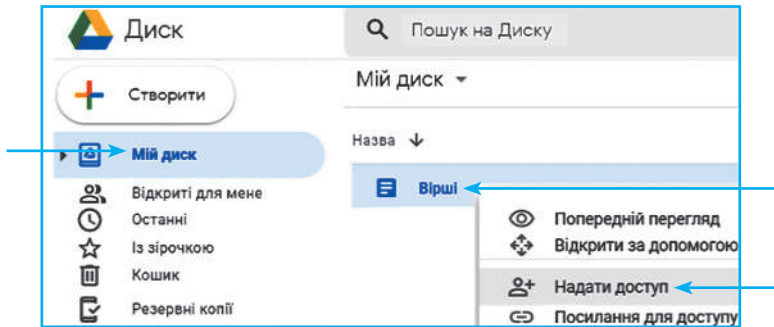


Рис. 7.2

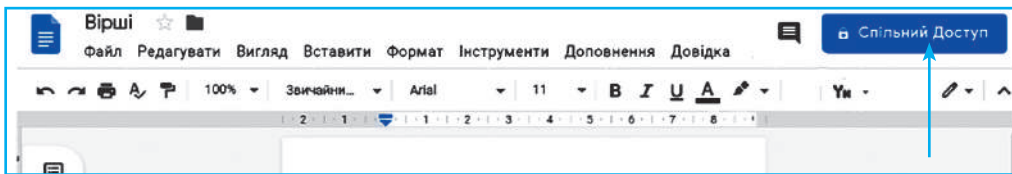


Рис. 7.3

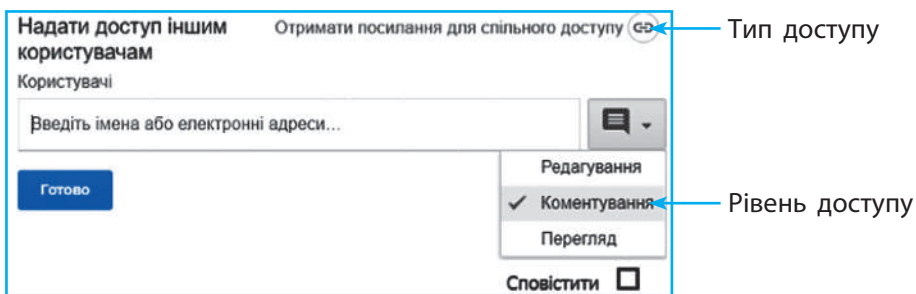


Рис. 7.4

Якщо в адресному рядку форми доступу зазначити електронні адреси, то адресати або отримають автоматичний лист-повідомлення про те, що їм надано доступ до файлу (за умови встановлення прапорця Сповістити), або дізнаються про це, увійшовши до свого Google Диска.



Питання для самоперевірки

1. Які переваги надають хмарні технології?
2. Що таке хмарні сервіси? Назвіть відомі вам хмарні сервіси.
3. Що таке хмарний диск?
4. Як відкрити Google Диск?
5. Як створити документ на Google Диску?
6. Які типи та рівні доступу до файлів вам відомі?
7. Які функції можуть надавати онлайн-перекладачі?



Вправа 7

- Створити на хмарному диску й колективно опрацювати документ за допомогою онлайн-перекладача.
- 1. Запустіть браузер і зайдіть до свого облікового запису Google. Відкрийте Google Диск і створіть на ньому Google-документ. Дайте документу назву Хмарні технології.
 2. Знайдіть в інтернеті означення поняття «хмарні технології». Скопіюйте текст означення до створеного Google-документа.
 3. Відкрийте на новій вкладці Google Перекладач. Перекладіть знайдене означення однією з мов, яку ви вивчаєте. За можливості прослухайте переклад. Скопіюйте переклад до Google-документа.
 - 👥 4. Відкрийте доступ для коментування документа вчителю, однокласнику чи однокласниці (з автоматичним сповіщенням).
 - ⚙️ 5. Зайдіть до своєї поштової скриньки Gmail. Відкрийте лист-сповіщення про доступ і перегляньте документ, доступ до якого вам надано.
 6. Додайте в документі коментар до перекладеного означення, наприклад, чи задовольняє вас якість перекладу. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106145).





ПРАКТИЧНА РОБОТА 2

ВИКОРИСТАННЯ ХМАРНИХ ТЕХНОЛОГІЙ ДЛЯ КОЛЕКТИВНОГО ОПРАЦЮВАННЯ ДОКУМЕНТІВ

- Завдання:** створити Google-презентацію із використанням хмарних сервісів, наприклад, геосервісів, мап тощо.
- Обладнання:** комп'ютер, з'єднаний з інтернетом.

Хід роботи


Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте браузер. Зайдіть до поштової скриньки Gmail.
2. Оформіть листа вчителю/вчительці з поясненням, яке з великих міст України чи іншої європейської держави ви хотіли б відвідати. Листа не надсилайте.
3. У новій вкладці браузера відкрийте Google Диск. Створіть на хмарному диску презентацію. За назву презентації візьміть назву міста, зазначеного в листі.
4. На *першому* слайді помістіть тему, своє прізвище, ім'я і клас.
5. У новій вкладці браузера знайдіть в інтернеті автомобільний маршрут «пункт, в якому мешкаєте, — вибране місто».
6. Скопіюйте карту з маршрутом і відстанню в кілометрах і помістіть зображення на *другий* слайд.
7. У новій вкладці браузера знайдіть сторінку Google Earth (Google Планета Земля). Завантажте з офіційного джерела інсталяційний файл Google Earth і запустіть. Дочекайтеся встановлення програми і запустіть.
Можна скористатись онлайн-сервісом Планета Земля (Google Earth), що відкривається в браузері Google Chrome.
8. У пошуковому рядку вікна Google Earth уведіть назву вибраного міста та клацніть кнопку Пошук.
Прокрутіть колесо й перетягніть із натиснутою лівою кнопкою миші карту так, щоб «опуститися» на мінімально можливу висоту над містом до появи 3D-зображення вулиць.

9. «Помандруйте» вулицями, «зайдіть» до кафе, музею, вокзалу тощо. Зробіть скриншот із зображенням одного з приміщень і помістіть на *третій* слайд.
 10. Відкрийте доступ усім, хто має посилання для перегляду та коментування Google-презентації, із автоматичним поштовим сповіщенням.
 11. Дочекайтеся отримання листа-сповіщення про доступ до документа від однокласників чи однокласниць. Відкрийте лист-сповіщення, перегляньте документ (презентацію) і додайте до назви міста свій коментар.
 12. До листа вчителю/вчительці додайте гіперпосилання на файл Google-презентації, пов'язавши його з назвою вибраного місця в тексті листа. Надішліть листа, завершіть роботу.
- Зробіть висновок:** які хмарні сервіси було використано під час створення презентації.

§ 8. ШТУЧНИЙ ІНТЕЛЕКТ. ІНТЕРНЕТ РЕЧЕЙ

Під час ознайомлення з цифровими технологіями в 6 класі ви вчилися користуватися системою машинного перекладу, робота якої ґрунтується на використанні штучного інтелекту (ШІ). Проте його застосування не обмежується перекладанням текстів.

 **Штучний інтелект** — розділ інформатики, присвячений розробці машин, які моделюють інтелект людини (тобто здатні до самонавчання, розв'язування задач тощо).

Історія штучного інтелекту

Історія розвитку ШІ налічує вже понад 70 років. Невдовзі після винайдення комп'ютерів стало зрозуміло, що з часом їхні можливості зростуть. Так виникла ідея змоделювати мислення

експерта в певній галузі: змусити обчислювальну машину «думати», як людина (наприклад, як лікар, коли ставить діагноз).

● Термін «штучний інтелект» запропонував 1955 року американський інформатик Джон Маккарті (рис. 8.1). Проте лише в 1970-х роках можливості комп'ютерів дозволили створити перші експертні системи, придатні до практичного застосування.



Рис. 8.1

Як відомо, мозок людини складається з нервових клітин — нейронів, які з'єднані між собою й утворюють **нейронну мережу** (нейромережу). У 1990-х роках рівень комп'ютерів дозволив розпочати моделювання нейромереж, що відкрило нові можливості в опрацюванні природної мови та розпізнаванні образів.

Нині в інтернеті з'явилося багато сервісів, де ШІ виконує різноманітні завдання: малює картини, пропонує ідеї для бізнесу тощо. Розвиток ШІ триває, на нас очікує багато цікавих новин.

Експертні системи

Вам доводилося грати в гру «Хто я?» Кожна особа, не дивлячись, чіпляє собі на чоло (спину) картку з назвою певного об'єкта, а потім ставить навідні запитання, щоб відгадати цей об'єкт. Тоді ви легко зрозумієте призначення експертних систем, адже вони діють так само: послідовно ставлять людині питання, враховують усі відповіді і, зрештою, пропонують свою.

● **Експертна система** — програма, що допомагає людині приймати рішення в певній галузі на основі збереженої в базі знань інформації, отриманої від фахівців (експертів).

Пригадаймо з курсу інформатики 6 класу, як ви склали мовою програмування Python програму, що моделює діяльність «експерта»: комп'ютер ставить низку питань і за відповідями визначає тип принтера.



Програма може бути такою:

```
w = input("Чи є в принтері фотобарабан? ")
if w == "так":
    print("Цей принтер — ксерографічний")
else:
    w = input("Чи використовується в принтері чорнило? ")
    if w == "так":
        print("Цей принтер — струминний")
```

У текст програми включено «знання» про особливості принтерів і правила отримання потрібної інформації з цієї «бази знань». Звичайні користувачі можуть просто користуватися програмою, а експерт може доповнити, додавши нові «знання» і правила. Скажімо, для ксерографічного принтера можна уточнити, лазерний він або світлодіодний:

```
w = input("Чи є в принтері фотобарабан? ")
if w == "так":
    print("Цей принтер — ксерографічний")
    w = input("Чи є в принтері лазер? ")
    if w == "так":
        print("Цей принтер — лазерний")
    else: print("Цей принтер — світлодіодний")
```

...

Далі діалог із комп'ютером може виглядати так (рис. 8.2):

```
Чи є в принтері фотобарабан? так
Цей принтер – ксерографічний
Чи є в принтері лазер? ні
Цей принтер – світлодіодний
```

Рис. 8.2

Реальні експертні системи мають значно більший обсяг бази знань і складніші правила отримання відповідей.

Чат-боти на основі штучного інтелекту

Опрацювання текстової інформації потребує менше ресурсів комп'ютера, ніж інші види інформації, тому перші доступні

практики з ШІ стосувалися роботи з текстом. Ви вже працювали з системою машинного перекладу, проте існують і інші напрямки опрацювання тексту засобами ШІ. Так, протягом останніх років набули популярності онлайнві чат-боти на основі ШІ.

Чат-бот — програма, що імітує відповіді людини в діалозі з нею. Може бути вбудованою в операційну систему, доступною онлайн або існувати як окремий застосунок.

Прикладом чат-бота, доступного онлайн, є ChatGPT. Популярні також чат-боти, убудовані у віртуальні помічники, такі як Apple Siri, Microsoft Cortana та Google Assistant. Зайшовши на сайт із чат-ботом, людина може вводити з клавіатури або голосом питання, отримувати відповіді, тобто спілкуватися, як із людиною. Чат-бот має велику базу знань, завдяки якій може виконувати різні завдання. Спілкуючись, він розширює базу знань, тобто навчається.

2 Помітивши в лісі рослину з плодами, схожими на суниці (рис. 8.3), Наталка (Н) вирішила дізнатися про незнайому рослину в онлайнного чат-бота ChatGPT (Б). Відбувся такий діалог (відповіді чат-бота скорочено):



Рис. 8.3

Н Плід якої рослини схожий на суниці, але несмачний?

Б Трохи схожим на суниці, але несмачним, може бути плід рослини «лісова суниця». Її плоди нагадують суничні плоди, але можуть бути менш соковитими та не такими солодкими.

Н А така, що її плоди не обвисають, а піднімаються вгору?

Б Якщо плід піднятий вгору, це може вказувати на «підлісну суницю» (лат. *Potentilla erecta*), також відому як «тризубець піднятий». Ця рослина насправді не належить до родини суниць (*Fragaria*), але її плоди подібні до суниць і також ростуть на низькому кущі.

Н Але «*Potentilla erecta*» — це латинська назва перстачу ви-
прямленого.

Б Ви праві, вибачте за плутанину. Піднятий вгору плід, що нагадує суниця, але несмачний, має «перстач індійський» (лат. *Potentilla indica*). Це рослина, що належить до родини трояндових (*Rosaceae*). Її плоди нагадують плоди суниці, але зазвичай мають кислий або невиразний смак.

! Відповідь, отриману від чат-бота, обов'язково перевіряйте за іншими джерелами. А також ніколи не куштуйте знайдені в лісі плоди незнайомих рослин!

Штучний інтелект працює із зображеннями

Засоби ШІ знайшли застосування і в опрацюванні зображень, наприклад, у створенні зображень за описом і ін.

Оскільки ви ознайомилися з технологією розпізнавання тексту, слід зазначити, що завдяки ШІ якість розпізнавання значно зростає, навіть якщо початкове зображення має певні недоліки. Розробляються також програми для розпізнавання довільних образів (окремих об'єктів, облич) на світлинах, малюнках тощо.

В інтернеті доступні сервіси, що створюють зображення за словесним описом. При цьому можна «замовити» деякі його параметри: фотографічну якість, художню техніку, стиль тощо.

3 На сторінці сервісу для створення зображень за запитом «З дерев падає жовте листя, а ялинка — зелена» було створено чотири різні зображення, які нагадують світлини (рис. 8.4, *а, б*). Коли ж до запиту додали слова «Малюнок крейдою», отримали зображення, що імітують малюнок (рис. 8.4, *в*).



а



б



в

Рис. 8.4

- 2023 року в Україні вийшла перша дитяча книга, написана та проілюстрована за допомогою штучного інтелекту. Спочатку творчий колектив редакції сформулював запитання стосовно теми та описав, які потрібні ілюстрації, — решту роботи виконав штучний інтелект.

Штучний інтелект в інтернеті речей

Список застосування ШІ в різних галузях довгий. Зокрема ШІ підсилює ефективність інтернету речей. Нині кількість фізичних пристроїв, під'єднаних до інтернету, вже перевищила число користувачів-людей. За різними даними, у світі до інтернету під'єднано понад 16 млрд пристроїв (рис. 8.5).



Рис. 8.5

- **Інтернет речей** (*Internet of Things*, скорочено *IoT*) — це сукупність під'єднаних до інтернету фізичних пристроїв — «речей», оснащених засобами приймання й передавання даних.

Інтернет речей дає змогу багатьом процесам відбуватися без участі людини. Ознайомімося зі складовими реалізації інтернету речей. Розгляньмо таблицю:

Складова	Опис
Кінцеві пристрої (речі)	Завдяки датчикам і сенсорам приймають і/або передають інформацію, а завдяки виконавчим механізмам виконують потрібні дії
Мережа передавання даних	Може бути бездротовою та/або дротовою. Для обміну даними IoT існує кілька стандартів: на основі мобільних мереж, на основі власних незалежних мереж або звичайних електромереж
Система керування мережею і пристроями	Базується на використанні комп'ютерів, контролерів і відповідного програмного забезпечення

Як бачимо, розвиток ШІ пов'язаний із практичними потребами суспільства в різних галузях, таких як промисловість, фінанси й ін. Уже зараз ШІ змінює ринок праці, впливає на зайнятість людей. Разом із тим фахівці у сфері ШІ вважають, що ця технологія потребує особливо відповідального керування з боку людей.





Питання для самоперевірки

1. Що таке нейронна мережа?
2. Що таке експертна система?
3. Які «знання» про принтери включає програма з прикладу 1?
4. Як записано правила отримання інформації в програмі, наведеній у прикладі 1 (с. 44)?
5. Що таке чат-бот?
6. Прокоментуйте діалог людини і чат-бота із прикладу 2 (с. 45).



Вправа 8

1. Отримайте від онлайн-чат-бота (наприклад, ChatGPT) поради, як установити на свій комп'ютер IDLE Python.
2. Випробуйте онлайн-чат-бот (приклад 2, с. 45), наприклад, у середовищі IDLE.
- 
 3. Доповніть програму розпізнаванням іще двох видів принтерів: матричного (друкує за допомогою стрічки, просоченої фарбою) і термопринтера (друкує на спеціальному термочутливому папері).
 Перевірте роботу програми в різних випадках.
- 
 4. Виготовте з однокласником/однокласницею 10 карток із назвами об'єктів (наприклад: ялинка, шафа, літак, штангіст, помідор), перемішайте їх і пограйте в гру «Хто я?».



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106146).



РОЗДІЛ 2

КОМП'ЮТЕРНА АНІМАЦІЯ



§ 9. Принцип анімації. Редактор двовимірної анімації TurbTube

§ 10. Тло анімації. Керування кольором

§ 11. Автоматичне створення анімації

§ 12. Створення анімаційних ефектів

§ 13. Бібліотека об'єктів. Імпортування зображень.
Експортування проекту

Практична робота 3. Створення анімаційного проекту
«Грибний дощ»

§ 14. Покадрова анімація

Практична робота 4. Створення анімації з декількох сцен

ПОВТОРЮЄМО



На уроках інформатики ви вже створювали рухомі зображення — *анімації*. У середовищі Scratch за допомогою команд руху ви змушували окремі об'єкти рухатися відповідно до задуму. Ви знаєте, що процес просувається швидше, якщо перед цим продумати сценарій анімації, докладно описавши поведінку всіх об'єктів.

Працюючи в редакторі презентацій, ви додавали анімаційні ефекти до об'єктів, розташованих на слайді. Один об'єкт може мати кілька ефектів: *на вхід*, *виділення*, *вихід*. Також об'єкт може рухатися заданою траєкторією. Анімації в презентаціях запускаються клацанням або виконуються автоматично (почергово чи одночасно).

Вам стануть у пригоді вміння користуватися інструментами графічного редактора Inkscape.

1. Поясніть, що таке анімація.
2. Які команди дають змогу рухати об'єкти в Scratch?
3. Для чого складають сценарій анімаційного фільму?
4. Як у презентаціях використовують анімаційні ефекти?
5. Назвіть способи запуску анімації в презентації.
6. Які інструменти графічного редактора Inkscape ви використовували?



У цьому розділі ви ознайомитеся з редактором двовимірної анімації та навчитесь розробляти за його допомогою прості мальовані анімації.

§ 9. ПРИНЦИП АНІМАЦІЇ. РЕДАКТОР ДВОВИМІРНОЇ АНІМАЦІЇ TURITUBE

Переглядаючи мультфільми, ви, напевно, іноді замислювалися, як можна змусити зображення рухатися. Нині анімація стала дуже розвиненим напрямом застосування комп'ютерів.

Принцип отримання анімаційного ефекту

Загалом, принцип отримання рухомого зображення простий: слід один за одним показувати нерухомі малюнки, які трохи відрізняються між собою. Якщо їх змінювати достатньо часто, око людини не встигатиме сприймати кожне зображення окремо.

Величину, яка дорівнює кількості кадрів, що демонструються за секунду, називають **частотою кадрів**. Вимірюють її в кадрах на секунду (кадр/с; англ. *frames per second, fps*).



У перших фільмах застосовували частоту від 16 до 24 кадрів/с (остання стандартна частота для кінотеатрів із 1930-х років). Нині в різних системах телебачення вона становить 24, 25 і 30 кадрів/с. Комп'ютерні програми дають змогу *створювати*, а обладнання — *відтворювати* відео з різною частотою кадрів.

Для побудови кадрів використовують малюнки, плоскі перекладні маріонетки тощо. Далі ми розглядатимемо лише мальовану анімацію, що являє собою серію мальованих зображень, на кожному з яких об'єкт подано в іншій фазі руху.

● «Казка про солом'яного бичка» — перший український анімаційний фільм. Його створив режисер, сценарист, художник-аніматор В'ячеслав Левандовський 1927 року в Одесі. Замість малюнків у фільмі використано плоскі маріонетки.

Для створення анімацій існує низка різних програм, таких як Adobe Animate, Blender, Animation Desk, Animaker, TurpTube та ін.

Редактор двовимірної анімації TurīTube

Із розробкою простих анімацій ви ознайомитеся в середовищі TurīTube (рис. 9.1), вільного редактора двовимірної анімації з відкритим кодом.

Розгляньмо головне вікно програми (рис. 9.2). Його вигляд залежить від використовуваного модуля (на рисунку — модуль Анімація).



Рис. 9.1

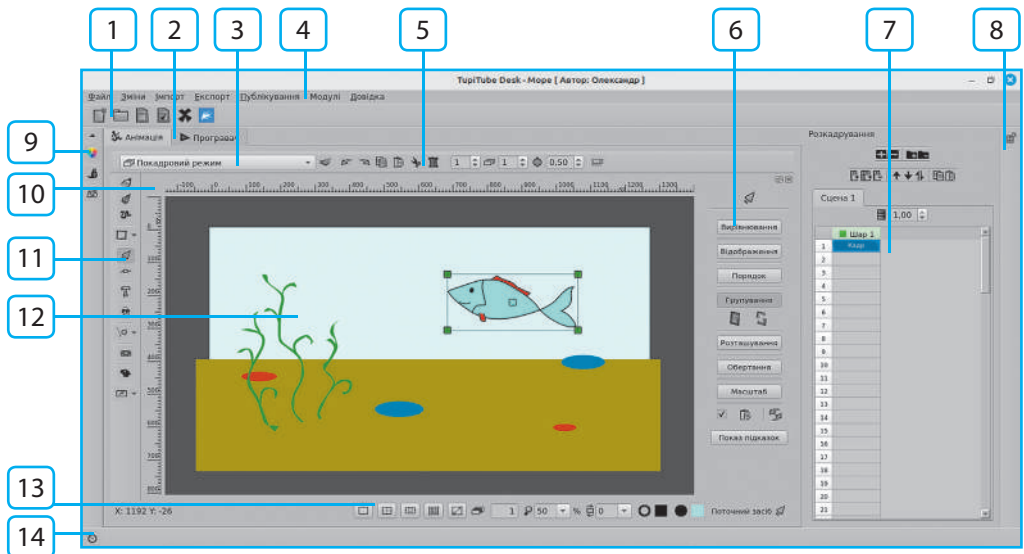









Рис. 9.2

Тут 1 — верхня панель; 2 — вкладки модулів; 3 — список режимів модуля; 4 — рядок меню; 5 — панель команд модуля; 6 — панель параметрів інструментів; 7 — розкривна панель; 8 — права панель; 9 — ліва панель; 10 — лінійки; 11 — панель інструментів; 12 — робоче поле; 13 — рядок стану; 14 — нижня панель.

Верхня панель містить кнопки команд:  — створити новий проєкт;  — відкрити наявний проєкт;  — зберегти проєкт;  — зберегти копію проєкту під іншою назвою;  — закрити проєкт;  — перейти на сайт програми.

Засоби розроблення анімації, які буде докладно розглянуто далі, зібрано в модулі Анімація (див. рис. 9.2). Модуль Програваач дає змогу переглядати створену анімацію в різних режимах (докладніше — далі).

Створення проєкту

Щоб розпочати розроблення проєкту, потрібно запусити застосунок TurbTube і натиснути кнопку Новий проєкт , а потім у діалоговому вікні ввести його параметри (рис. 9.3).

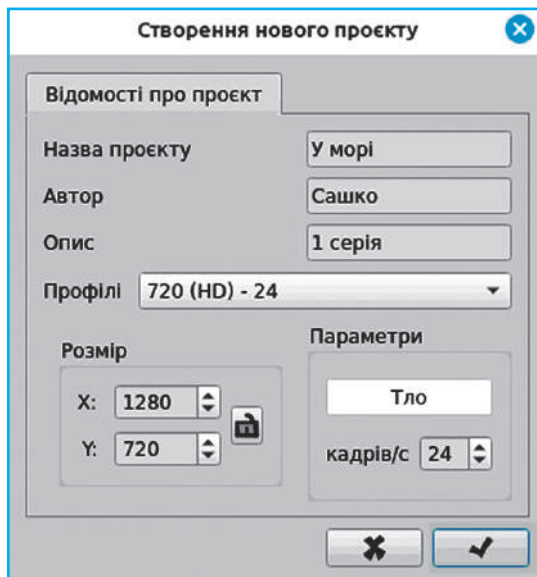





Рис. 9.3

Якщо зі списку Профілі вибрати один із варіантів, то буде автоматично заповнено поля для розмірів (X — ширина, Y — висота) і частоти кадрів (кадрів/с).

Якщо клацнути прямокутник із написом Тло, то відкриється діалогове вікно, де можна вибрати колір тла для кадрів анімації. Після підтвердження налаштувань кнопкою  відкриється вікно програми з модулем Анімація.

Додавання геометричних фігур

Прямокутник кольору тла в робочому полі — це перший кадр нашого майбутнього «фільму». Тут ми малюватимемо об'єкти, а також додаватимемо об'єкти та малюнки, створені в інших застосунках.

На рис. 9.4 показано інструменти групи Фігури, список яких розкривається кнопкою  на панелі інструментів поруч із кнопкою групи  (на кнопці групи показано вибраний інструмент). Щоб побудувати фігуру, слід виконати перетягування в робочому полі.

Побудована фігура має центральний маркер для переміщення та зелені кутові маркери для змінення розмірів (рис. 9.5). Якщо двічі клацнути на одному з маркерів, вони змінять колір на помаранчевий і слугуватимуть для обертання фігури.

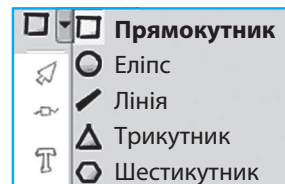


Рис. 9.4

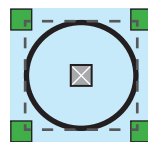





Рис. 9.5

 Створимо анімацію кидання м'яча. Для цього:

- 1) запустимо редактор анімацій і створимо проєкт із профілем 640×480 — 24;
- 2) додамо декілька фігур, у нашому випадку — прямокутник, трикутник і еліпс;
- 3) виберемо інструмент Анімація переміщення , на панелі параметрів інструментів уведемо назву анімаційного ефекту (наприклад, стрибок) і натиснемо  — з'являться назви режимів подальшої роботи (рис. 9.6);

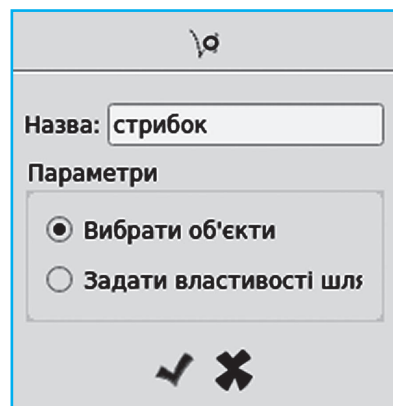


Рис. 9.6

- 4) клацнемо (виберемо) еліпс і перейдемо в режим Задати властивості шляху (рис. 9.7, а);
- 5) побудуємо шлях, яким рухатиметься еліпс, клацнувши над трикутником, а потім праворуч від нього (рис. 9.7, б);
- 6) перетягнемо білі маркери, спочатку суміщені з кінцями відрізків, щоб отримати криву лінію (рис. 9.7, в);
- 7) збережемо анімацію, клацнувши кнопку ✓;
- 8) перейдемо до модуля Програваач, щоб побачити результат роботи.

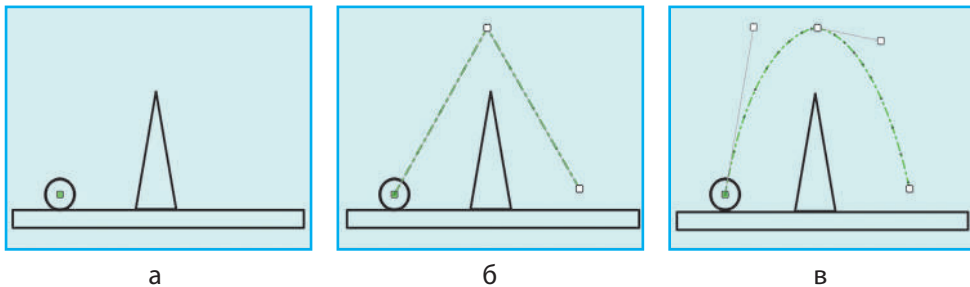


Рис. 9.7

Після додання до еліпса анімаційного ефекту можна помітити, що на розкривній панелі на вкладці Сцена 1 у стовпці Шар 1 з'явилися позначки кадрів (рис. 9.8).

! Якщо проект не зберегти на початку роботи, то час від часу з'являтиметься вікно з нагадуванням. Зміни, зроблені після збереження проекту, зберігатимуться автоматично.

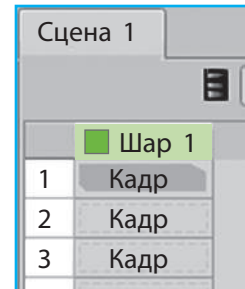


Рис. 9.8

Хай наша перша анімація недосконала, але ж це лише перша спроба! З TurbTube дійсно можна легко створити анімацію навіть новачкам, скориставшись комп'ютером, планшетом або навіть мобільним телефоном. Далі ви дізнаєтеся, як покращити свою роботу, а також навчитесь застосовувати інші способи створення анімацій, опануєте різноманітні корисні засоби.



Питання для самоперевірки

1. Поясніть принцип отримання анімаційного ефекту.
2. Що таке частота кадрів?
3. Яке призначення застосунку TurTube?
4. Значення яких параметрів задають при створенні проєкту?
5. Яке призначення маркерів у об'єктів, побудованих інструментами групи Фігури?
6. Що потрібно зробити, щоб під час роботи не з'являлося вікно з нагадуванням про збереження проєкту?



Вправа 9

■ Ознайомитися з різними способами створення анімацій.

1. У ХІХ ст. була популярною іграшка з назвою «тауматроп». Дізнайтеся, як вона влаштована і стисло опишіть її роботу. Спробуйте виготовити тауматроп самостійно.
2. Створіть анімацію з готових кадрів (див. папку Коник*) за допомогою редактора презентацій. Для цього:
 - 1) запустіть редактор презентацій і створіть 20 порожніх слайдів;
 - 2) на *перший* слайд додайте малюнок із файлу 01.png; розташуйте його в центрі слайда;
 - 3) на *наступні* слайди по чергово додайте малюнки з файлів 02.png, 03.png, 04.png і т. д.; перегляньте презентацію;
 - 4) налаштуйте автоматичну зміну слайдів через 0,1 с; перегляньте презентацію;
 - 5) поекспериментуйте з часом змінення слайдів; збережіть презентацію у файлі Вправа 9-1.
3. Створіть анімацію за описом, поданим у прикладі 2. Збережіть проєкт у файлі Вправа 9-2.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106147).



§ 10. ТЛО АНІМАЦІЇ. КЕРУВАННЯ КОЛЬОРОМ

В анімації з прикладу 2 (§ 9) нерухомі трикутник і прямокутник з'являються лише на початку руху «м'яча», а потім зникають. Щоб виправити це, їх слід перенести на особливий шар — *тло*. Підтримувані види тла зібрано в списку режимів модуля Анімація (рис. 10.1), їх можна використовувати одночасно.

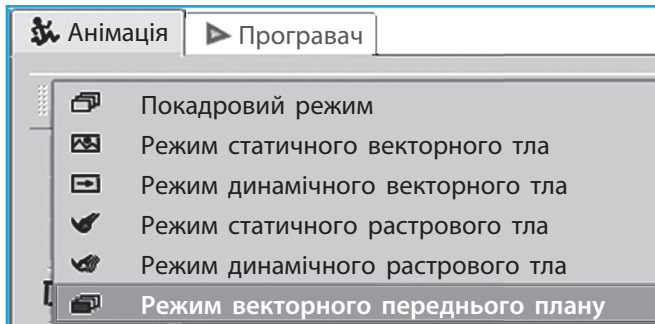


Рис. 10.1

Керування кольорами об'єктів

Змінити кольори штриха та заповнення векторних об'єктів, доданих до сцени, можна на панелі Колірна палітра (рис. 10.2).

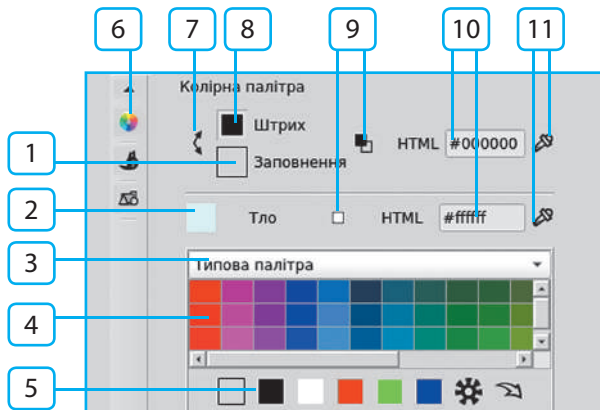




Рис. 10.2

Тут 1 — зразок кольору заповнення; 2 — зразок кольору тла; 3 — список палітр; 4 — поточна палітра; 5 — основні кольори; 6 — кнопка відкриття панелі Колірна палітра; 7 — кнопка для обміну кольорів; 8 — зразок кольору штриха; 9 — кнопки для скидання кольорів до початкових значень; 10 — коди кольорів; 11 — «піпетки» для взяття кольорів із робочого поля.


Рівень прозорості (0 — повна прозорість, 255 — повна непрозорість) задають регулятором Щільність.

Зорієнтуватися, які саме кольори штриха й заповнення вибрано, можна за їх зразками, наведеними в рядку стану модуля Анімація — . Новостворені об'єкти мають кольори штриха й заповнення, задані на панелі Колірна палітра.


Змінити кольори об'єктів, які додано раніше, допоможе інструмент Заповнення . Якщо на панелі Колірна палітра клацнути зразок *кольору штриха*, а потім об'єкти, їхні штрихи набуватимуть вибраного кольору. Якщо клацнути зразок *кольору заповнення*, то інструмент впливатиме на колір заповнення об'єктів.

Векторне тло


Векторне тло дає змогу малювати векторні об'єкти інструментами TurbTube. Ці інструменти подібні до інструментів відомого вам векторного графічного редактора Inkscape.

 **Статичне тло** — зображення на тлі, яке упродовж перегляду анімації залишається нерухомим.


На статичному тлі розміщують зображення пейзажу чи інтер'єру, де відбувається дія анімаційного фільму, а також окремі об'єкти, які мають залишатися нерухомими.

 Перенесемо нерухомі об'єкти (див. приклад 2 з § 9) на статичне векторне тло:

- 1) переконаємося, що активним є кадр 1 (на розкривній панелі на вкладці Сцена 1 у стовпці Шар 1 клацнемо його позначку);

- 2) інструментом Вибір об'єкта  виберемо трикутник і прямокутник (два об'єкти разом можна вибрати їх клацанням з утримуванням клавіші Ctrl або перетягуванням);
- 3) виріжемо вибрані об'єкти, викликавши на них контекстне меню і вибравши команду Вирізати (або сполучення клавіш Ctrl+X);
- 4) перейдемо в режим статичного векторного тла і вставимо об'єкти командою контекстного меню Вставити (або сполучення клавіш Ctrl+V);
- 5) натисканням клавіш зі стрілками пересунемо об'єкти, якщо це потрібно.

Перейшовши в модуль Програвач, побачимо, що тепер трикутник і прямокутник розташовані на нерухомому (статичному) тлі і не зникають протягом перегляду всієї анімації.

 **Динамічне тло** — зображення на тлі, яке під час перегляду анімації прокручується в певному напрямку.

Щоб додати до анімації рухоме тло з векторними об'єктами, слід увімкнути режим динамічного векторного тла (рис. 10.3) і побудувати потрібний малюнок.

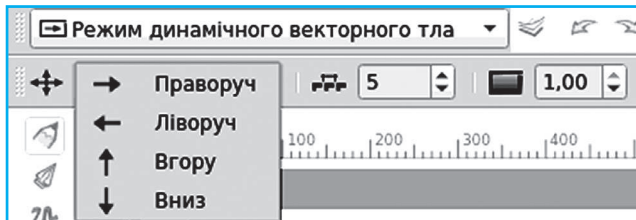





Рис. 10.3

Елементами керування, що з'явилися над робочим полем, можна задати:

- напрям руху (список ; на рис. 10.3 — розкритий);
- величину зміщення тла на 1 кадр (поле );
- щільність (поле ; 0 — повністю прозоре, 1 — повністю непрозоре).



Створимо тло для зимової анімації з зображенням нерухомого снігового покриву та рухомих хмар (рис. 10.4). Для цього:



- 1) розпочнемо новий проєкт зі світло-синім кольором тла; виберемо світло-сірий колір для штриха й заповнення;
- 2) на статичному векторному тлі інструментом Олівець  намалюємо замкнену криву, що зображає сніговий покрив;
- 3) перейдемо в режим динамічного векторного тла і намалюємо у верхній частині кадру інструментом Еліпс  декілька фігур — хмар;
- 4) задамо значення параметрів: напрям руху — праворуч; зміщення — 5; щільність — 1.



Рис. 10.4

Наша анімація має один кадр, тому в модулі Перегляд побачимо нерухому картинку. Додаймо до анімації кадри (зважаючи, що частота кадрів 24 кадри/с): прокрутімо донизу список кадрів на панелі Розкадрування і клацнімо прямокутник навпроти номера 100 — тепер проєкт має 100 кадрів. Список номерів при цьому подовжився, тому кількість кадрів можна ще збільшити.

Тепер у модулі Перегляд видно анімацію тривалістю приблизно 4 с (оскільки $100:24 \approx 4$).

! Рухоме тло дозволяє створити ілюзію руху інших об'єктів. Наприклад, внаслідок руху тла із зображенням дороги здаватиметься, що рухається (насправді, нерухоме) зображення автомобіля.

Растрове тло

У режимах статичного й динамічного растрового тла відкривається вікно, в лівій частині якого розташована панель Властивості пензля (рис. 10.5). Кнопки з назвами наборів пензлів угорі панелі відкривають відповідні набори, з яких можна вибрати потрібний пензель.

Як бачимо, пензлі дуже різноманітні, тому ознайомлюватися з ними найкраще, експериментуючи.

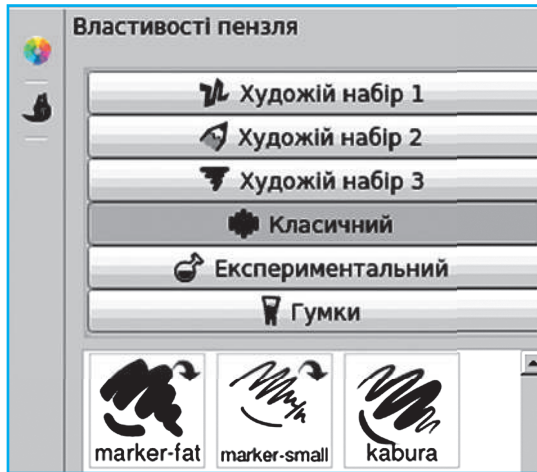


Рис. 10.5

3

Додамо до проекту з прикладу 2 зображення сонця. Для цього:
1) перейдемо в режим статичного растрового тла;
2) виберемо з набору Класичний пензель pointy-ink (рис. 10.6);
3) кнопкою на лівій панелі відкриємо панель Колірна палітра і встановимо для штриха жовтий колір — тепер на статичному растровому тлі можна намалювати сонце (рис. 10.7).



Рис. 10.6

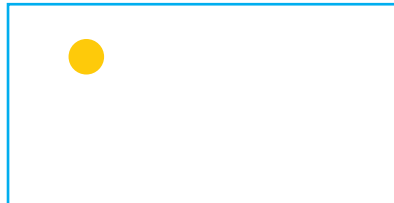



Рис. 10.7

Щоб унесені зміни додати до проекту та повернутися в головне вікно, необхідно вибрати команду меню Файл → Вийти з растрового режиму (або натиснути сполучення клавіш Ctrl+Q).

► Динамічне растрове тло


Робота в режимі динамічного растрового тла не відрізняється від роботи зі статичним растровим тлом. Над робочим полем у цьому режимі (як і в режимі динамічного векторного тла) з'являється поле  для введення величини зміщення тла на 1 кадр.

4 Додамо до проекту (див. приклади 2 і 3) рухоме зображення клаптів снігу. Для цього:



- 1) перейдемо в режим динамічного растрового тла;
- 2) встановимо на панелі Колірна палітра світло-сірий колір штриха;
- 3) виберемо на панелі Властивості пензля з набору Класичний пензель dry-brush;
- 4) намалюємо в різних місцях полотна короткі штрихи;
- 5) задамо для зміщення значення 1;
- 6) виберемо команду Файл → Вийти з растрового режиму і переглянемо результат: небом пливуть хмари і падає сніг.

Упорядкування шарів тла

Одночасно використовуючи різні види тла, можна досягти цікавих ефектів. Тому розглянемо керування шарами тла докладніше.

Кнопкою  поряд зі списком режимів модуля Анімація відкривається вікно для задання параметрів режимів (рис. 10.8).

Тут можна змінити розташування різних шарів тла відносно глядача/глядачки. Перший у списку шар розміщується найдалі від глядача, а кожен наступний — ближче.

Кнопкою  вибраний шар переміщують у списку вгору (тобто від глядача), а кнопкою  — униз (тобто до глядача).

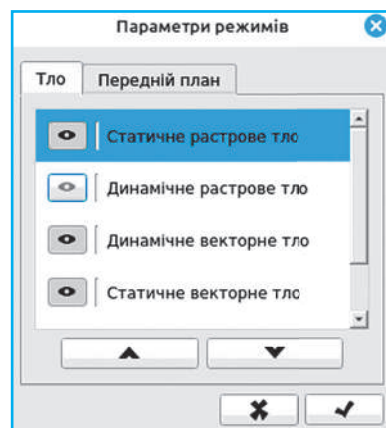




Рис. 10.8

Кнопка  перемикає видимість відповідного шару.

- ! Щоб підтвердити зміни в налаштуваннях, необхідно натиснути кнопку Застосувати . Кнопкою Скасувати  вікно закривається без внесення змін до проекту.

Отже, вправне використання різних видів тла дозволяє створити анімоване середовище, де розгортатиметься сюжет фільму.




Питання для самоперевірки

1. Чому в анімації з попереднього параграфу нерухомі об'єкти зникають?
2. Які види тла підтримує TurbTube?
3. У чому різниця між статичним і динамічним тлом?
4. У чому різниця між векторним і растровим тлом?
5. Як збільшити швидкість руху динамічного тла?
6. Як приховати один із шарів тла?



Вправа 10

- Створити анімацію з різними видами тла.
- 1. Відкрийте створений раніше файл Вправа 9-2.
- 2. Доопрацюйте проєкт відповідно до прикладу 1 (с. 58).
- 3. Розпочніть новий проєкт Вправа 10 із профілем 480 (PAL DV/DVD)-25. Додайте зображення снігу та рухомих хмар, керуючись прикладом 2 (с. 60).
- 4. Намалуйте сонце відповідно до прикладу 3 (с. 61).
-  5. Додайте рухоме зображення снігу, керуючись прикладом 4 (с. 62).
- 6. Збережіть проєкт у файлі Вправа 10.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106148).














§ 11. АВТОМАТИЧНЕ СТВОРЕННЯ АНІМАЦІЇ

Раніше під час створення анімаційного фільму художники-аніматори малювали кожен кадр вручну. Порахуймо, скільки потрібно малюнків для 10-хвилинного фільму за частоти кадрів 25 кадрів/с? 15 000! Комп'ютер значно пришвидшив роботу завдяки методу ключових кадрів, за якого більшість зображень об'єкта будується автоматично.

Метод ключових кадрів (англ. *keyframing* від *key* — ключ, *frame* — кадр) — спосіб автоматичного створення анімації, за якого розробник задає положення (форму, колір або інші властивості) об'єкта на деяких кадрах, а комп'ютер автоматично обчислює ці характеристики для проміжних кадрів.


Можливості інструмента Вибір об'єкта

Для роботи з різними об'єктами ознайомімося з інструментом Вибір об'єкта . Панель параметрів інструмента містить назви операцій, кожна з яких розкриває відповідні засоби керування. Розгляньмо таблицю:

Операція	Що можна зробити
Вирівнювання	Вирівняти об'єкт у робочому полі:  — на горизонтальній осі;  — на вертикальній осі;  — у центрі
Відображення	Віддзеркалити об'єкт:  — горизонтально;  — вертикально;  — в обох напрямках
Порядок	Перемістити об'єкт відносно глядача:  — позаду всіх;  — на один рівень далі;  — попереду всіх;  — на один рівень ближче
Групування	Згрупувати  або розгрупувати  вибрані об'єкти

Закінчення таблиці

Операція	Що можна зробити
Розташування	Задати координати X і Y об'єкта. Початок координат у лівому верхньому куті робочого поля. Вісь X напрямлена праворуч, а вісь Y — донизу
Обертання	Задати кут повороту за годинниковою стрілкою в градусах
Масштаб	Змінити розміри об'єкта: X — ширину; Y — висоту. Якщо встановити прапорець Пропорц. , обидва розміри змінюватимуться разом
Показ підказок	Показати короткі інструкції щодо користування інструментом. Повторне натискання їх приховує

Як і в інших відомих вам застосунках, у TurbTube можна змінювати **масштаб перегляду** робочого поля. Для цього під ним є поле зі списком Масштаб , де можна вибрати значення зі списку або ввести з клавіатури. Також масштаб зміниться, якщо навести вказівник у робоче поле і прокрутити коліщатко миші.

Додавання анімаційних ефектів до об'єктів

У вправі 9 (п. 3) ви створили анімацію кидання м'яча, задавши положення на деяких ключових кадрах і лінію, вздовж якої він мав рухатися. У TurbTube за допомогою ключових кадрів можна створювати й інші анімаційні ефекти (рис. 11.1).

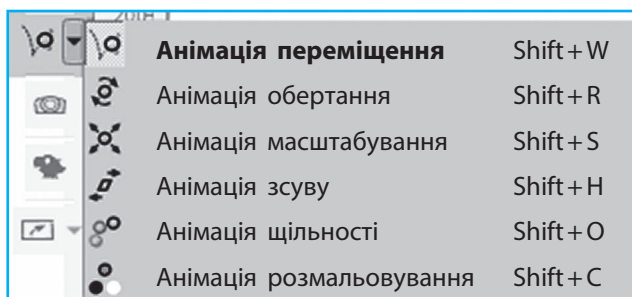




Рис. 11.1

Щоб додати до об'єкта анімаційний ефект, потрібно:


- 1) вибрати інструмент, у поле на панелі параметрів інструменту ввести назву нового ефекту і натиснути ;
- 2) вибрати один або декілька об'єктів, до яких буде додано ефект;
- 3) на панелі параметрів інструмента задати властивості ефекту;
- 4) натиснути кнопку , щоб зберегти створений ефект.

Після цього можна переглянути результат у модулі Програвач.

- !** У поточній версії TurbTube об'єкт, до якого додано автоматичну анімацію, неможливо редагувати як звичайний об'єкт.

Розгляньмо параметри ефектів і приклади їх використання.




Анімація переміщення

Для анімації переміщення ()¹, знайомої вам із прикладу в §9, задають траєкторію (шлях), уздовж якої має рухатися об'єкт. Траєкторію будують подібно до кривої у векторному редакторі.

Основними параметрами анімації переміщення є номер початкового кадру та кількості кадрів, які відводяться для кожної з ділянок траєкторії.



Задамо ключові кадри та форму траєкторії. Для цього:

- 1) розпочнемо проект Казка із профілем 480 (PAL DV/DVD)-25 із тлом світлого кольору. На статичному векторному тлі намалюємо краєвид (рис. 11.2, а);
- 2) повернемося в покадровий режим. Збільшимо масштаб і біля лівого краю кадра інструментом Олівець намалюємо Колобка (рис. 11.2, б);
- 3) інструментом Вибір об'єкта виберемо всі частини Колобка й згрупуємо їх (Групування → );
- 4) виберемо інструмент Анімація переміщення ()², введемо назву анімаційного ефекту рух, натиснемо  і виберемо зображення Колобка;

- 5) перейдемо в режим Задати шлях. Клацанням у робочому полі задамо положення Колобка на ключових кадрах (на рис. 11.2, в — білі маркери). За потреби, перетягуючи маркери, змінимо форму траєкторії;
- 6) збережемо анімацію, натиснувши кнопку ✓.

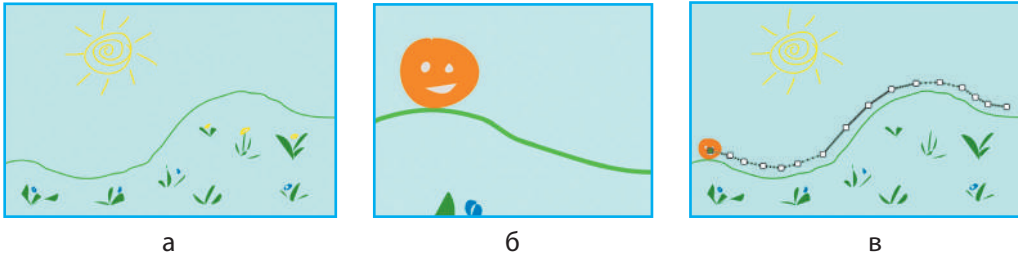


Рис. 11.2

Переглянемо результат у модулі Програвач. Бачимо, що швидкість руху Колобка на деяких ділянках слід змінити. Для цього на панелі параметрів інструмента є список ділянок (кроків). Для кожної зазначено кількість кадрів, відведених на проходження (рис. 11.3).

Крок	Кадри	+	-
1	5	+	-
2	4	+	-
3	4	+	-

Всього кадрів: 25




Рис. 11.3

Якщо для певної ділянки натиснути кнопку **+**, кількість кадрів збільшиться і Колобок рухатиметься нею повільніше. Натомість кнопка **-** зменшує кількість кадрів на проходження ділянки, а отже, прискорює рух Колобка.


2 Змінимо швидкість руху на різних ділянках. Нехай Колобок швидко скочується донизу, а на горб викочується повільно. У стовпець Кадри внесемо для 7, 8, 9 і 10 кроків значення 15, 19, 18, 11. Кількість кадрів для решти ділянок також змінимо, щоб загальна кількість становила 120 (це знадобиться далі). Підтвердимо зміни кнопкою ✓.

! Значення параметрів автоматичної анімації, доданої до об'єкта, можна змінити пізніше.

Для цього виконаємо такі дії:

- 1) виберемо інструмент Анімація переміщення ;
- 2) виберемо на панелі параметрів інструментів у списку назву анімаційного ефекту рух (наразі вона там одна);
- 3) натиснемо кнопку Змінити анімацію , розміщену під списком, — відкриється панель параметрів;
- 4) після внесення змін натиснемо кнопку .

Анімація обертання

Обертіві об'єкти зустрічаються в анімаціях часто. Тож розгляньмо, як додати до об'єкта **анімацію обертання** .

Для *неперервного обертання* необхідно задати напрям (за стрілкою годинника, проти стрілки годинника) і швидкість обертання (у градусах на кадр).

3 *Неперервне обертання.* Зробимо так, щоб сонце в проєкті Казка повільно оберталося. Оскільки сонце намальовано на тлі, його спочатку доведеться перемістити зі статичного векторного тла на шар покадрового режиму. Для цього:






- 1) перейдемо в режим статичного векторного тла, виділимо всі складові частини сонця (рис. 11.4), згрупуємо їх і виріжемо .
- 2) перейдемо в покадровий режим і вставимо об'єкт із буфера обміну . За потреби перетягнемо його в потрібне місце;
- 3) виберемо інструмент Анімація обертання , на панелі параметрів інструмента введемо назву «рух сонця», натиснемо .
- 4) виберемо об'єкт сонце, перейдемо в режим Задати властивості, налаштуємо тип — неперервне, швидкість — 3 градуси/кадр — після підтвердження змін  побачимо, що сонце протягом 120 кадрів робить повний оберт ($120 \cdot 3^\circ = 360^\circ$).



Рис. 11.4

Для *часткового обертання*, окрім напрямку та швидкості обертання, необхідно задати (рис. 11.5) величини початкового та кінцевого кутів повороту і спосіб повторення, якщо воно потрібне (**зациклити** — просте повторення; **зациклити з реверсом** — повторення в прямому та зворотному напрямках).



Тип:	Частковий
Напрямок:	За стрілкою годинника
Діапазон градусів:	
Починається з:	0
Закінчується на:	44
<input type="checkbox"/> Зациклити	
<input checked="" type="checkbox"/> Зациклити з реверсом	
Швидкість (градусів/кадр):	5,00

Рис. 11.5

4

Часткове обертання. Створимо анімацію гойдалки. Для цього:

- 1) на статичному векторному тлі намалюємо підлогу й опору гойдалки;
- 2) зобразимо в покадровому режимі на першому кадрі дошку і двох дітей; згрупуємо їх (рис. 11.6);
- 3) додамо до групи об'єктів анімацію обертання (↻) з назвою «гойдалка»;
- 4) перетягнемо зелений маркер центра обертання в точку опори гойдалки;
- 5) задамо такі параметри анімації: початковий кадр — 1; кінцевий кадр — 200; тип — частковий; діапазон градусів — від 340 до 20, зациклити з реверсом; швидкість обертання — 1 градус/кадр (рис. 11.7);

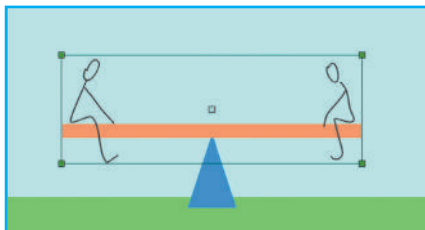


Рис. 11.6

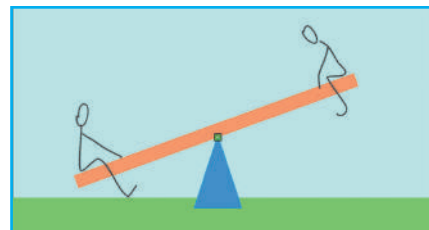


Рис. 11.7

б) підтвердимо зміни (✓), переглянемо анімацію: гойдалка періодично нахиляється то в один, то в інший бік.

Як бачимо, автоматичний розрахунок положення об'єктів під час переміщування чи обертання дає змогу швидко створювати анімаційні ефекти. Далі ви дізнаєтеся про інші анімаційні ефекти, а також опануєте деякі прийоми їх використання.



Питання для самоперевірки

1. Чому створення анімації без комп'ютерів потребувало багато часу або одночасної роботи багатьох людей?
2. Поясніть суть методу ключових кадрів.
3. Яка послідовність додавання до об'єкта автоматичного анімаційного ефекту?
4. Як додати до об'єкта анімацію переміщення вздовж певної траєкторії?
5. Як додати до об'єкта анімацію обертання?
6. Поясніть різницю між неперервним і частковим обертанням.



Вправа 11

■ Створити анімацію з ефектами переміщення та обертання.

1. Створіть проєкт Казка, як описано в прикладі 1 (с. 66).

Для цього:

- 1) збережіть його у файлі з назвою Казка;
- 2) змініть швидкість руху Колобка на окремих ділянках так, щоб рух мав природніший вигляд (див. приклад 2);
- 3) зробіть так, щоб зображення сонця неперервно оберталося (див. приклад 3, с. 68).



2. Створіть проєкт Гойдалка (див. приклад 4, с. 69). Збережіть його у файлі з назвою Гойдалка.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106149).



§ 12. СТВОРЕННЯ АНІМАЦІЙНИХ ЕФЕКТІВ



Окрім уже розглянутих раніше, існують й інші анімаційні ефекти. Ознайомімося з особливостями їх використання.

Додавання декількох ефектів до одного об'єкта

Урізноманітнити анімацію можна, додавши до одного об'єкта декілька анімаційних ефектів.



У прикладі 1 із § 11 Колобок рухався вздовж траєкторії. Щоб рух мав природніший вигляд, додамо до нього ще й ефект обертання. Обчислимо, на який кут слід повертати Колобка на кожному кадрі, щоб протягом 120 кадрів він зробив три повних оберти. Повному оберту відповідає кут 360° , а трьом обертam — $3 \cdot 360^\circ = 1080^\circ$. Оскільки $1080^\circ : 120 = 9^\circ$, то на кожному кадрі слід повернути Колобка на 9° . Для цього:


- 1) у проєкті Казка перейдемо на 1-й кадр і виберемо інструмент Анімація обертання ;
- 2) на панелі параметрів цього інструмента введемо назву «обертання» й натиснемо ;
- 3) виберемо об'єкт Колобок, перейдемо в режим Задати властивості і налаштуємо: тип — неперервний, швидкість — 9 градусів/кадр.

Після підтвердження змін () під час перегляду побачимо, що Колобок тепер справді котиться.

TurnTube дозволяє автоматично розраховувати не лише розташування та кут повороту рухомого об'єкта, а й інші його властивості: розміри, колір тощо. Для цього є низка інструментів у групі Анімації (див. рис. 11.1, с. 65).

Усі розглянуті далі інструменти створюють можливість задати спосіб повторення: зациклити — просте повторення; зациклити з реверсом — повторення в прямому та зворотному напрямках.

Анімація масштабування

Якщо потрібно, щоб змінювалися розміри певного об'єкта, до нього слід додати **анімацію масштабування** (.

Для цього виду анімації слід зазначити такі параметри:

- як саме *масштабувати*: За шириною й висотою, Лише за шириною, Лише за висотою;
- *коефіцієнт масштабування* — множник, на який змінюються відповідні розміри на наступному кадрі порівняно з поточним кадром;
- *число ітерацій* — кількість кадрів, протягом яких змінюватимуться розміри об'єкта.



Створимо анімацію мерехтливої зірки. Для цього:





- 1) намалюємо інструментом Олівець зірку (рис. 12.1) і згрупуємо частини малюнка в один об'єкт;
- 2) виберемо інструмент Анімація масштабування , на панелі параметрів інструмента введемо назву «пульсація» і натиснемо .
- 3) виберемо зірку, перейдемо в режим Задати властивості і налаштуємо таке: кінцевий кадр — 200; масштабувати — за шириною й висотою; коефіцієнт масштабування — 1,050; кількість ітерацій — 20; зациклити з реверсом.




Рис. 12.1

Після підтвердження змін () перейдемо в модуль Програваач і побачимо зірку, розміри якої періодично змінюються.

Анімація зсуву

Ще одним ефектом, який можна додати до об'єкта в ТупіТубе, є **анімація зсуву** (.

Інструмент Анімація зсуву  має такі параметри:

- *напрям зсуву*, який вибирають зі списку Зсунути: За шириною й висотою, Лише за шириною, Лише за висотою;

- *коефіцієнт масштабування* — величину, на яку змінюється зсув на наступному кадрі порівняно з поточним кадром;
- *число ітерацій* — кількість кадрів, протягом яких змінюватимуться розміри об'єкта.

Анімація щільності

Завдяки зміненню щільності в **анімації щільності** (☉☉) об'єкти можуть поступово з'являтися або зникати.

Інструмент Анімація щільності ☉☉ має такі параметри:

- *початкова щільність* і *кінцева щільність* — значення 1 означає, що об'єкт повністю видимий, а 0 — що він повністю зник (отже, щоб об'єкт поступово з'явився, початкова щільність має дорівнювати 0, а кінцева — 1);
- *число ітерацій* — кількість кадрів, протягом яких змінюватиметься щільність об'єкта (менше значення відповідає швидшій появі або зникненню, а більше значення — повільнішій).

Анімація розмальовування

Колір об'єкта також може змінюватися від кадру до кадру, якщо додати до нього **анімацію розмальовування** (●●).

Інструмент Анімація розмальовування ●● має такі параметри (рис. 12.2):

- *колір якої частини об'єкта має змінюватися*: штриха, заповнення чи штриха й заповнення разом;
- *початковий* і *кінцевий кольори*;
- *число ітерацій* — кількість кадрів, протягом яких початковий колір переходитиме в кінцевий.

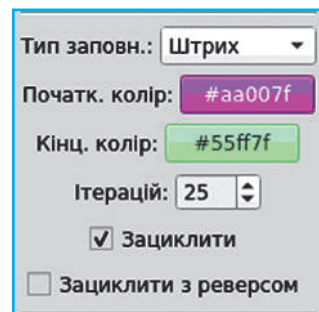


Рис. 12.2

Якщо клацнути прямокутник зі зразком початкового або кінцевого кольору, відкриється діалогове вікно, у якому можна вибрати інший колір.



Створимо анімацію вогнища (рис. 12.3). Для цього:




- 1) на статичному векторному тлі Олівцем намалюємо дрова для вогнища (рис. 12.3) і згрупуємо частини малюнка в один об'єкт;
- 2) перейдемо в *покадровий* режим і червоним кольором намалюємо контур полум'я;
- 3) виберемо інструмент Анімація розмальовування , виберемо полум'я і на панелі параметрів інструмента введемо назву ефекту, колір полум'я та значення відповідно до рис. 12.4. Якщо підтвердити зміни (✓), то в модулі Програвач побачимо, що колір полум'я змінюється;
- 4) виберемо інструмент Анімація масштабування , полум'я і задамо такі значення: назва ефекту — розмір полум'я; масштабувати — За шириною й висотою; коефіцієнт масштабування — 1,02; число ітерацій — 5; зациклити з реверсом — тепер змінюється не лише колір полум'я, а й розмір;
- 5) виберемо інструмент Анімація щільності  і задамо такі значення: назва ефекту — щільність полум'я; початкова щільність — 0,8; кінцева щільність — 1; число ітерацій — 10; зациклити з реверсом.



Рис. 12.3

Початковий кадр:	1
Кінцевий кадр:	200
Всього кадрів: 200	
Тип заповн.:	Заповненн-
Початк. колір:	#ff714e
Кінц. колір:	#ffd322
Ітерацій:	8
<input type="checkbox"/>	Зациклити
<input checked="" type="checkbox"/>	Зациклити з реверсом

Рис. 12.4

Режим векторного переднього плану

Ви вже вмієте використовувати в анімаціях тло різних видів (див. § 10). Тло розташоване позаду решти об'єктів, проте

можна додати ще й шар, розташований *попереду всіх об'єктів* анімації. Для цього застосовують **режим векторного переднього плану** (див. рис. 10.1, с. 57).

! У режимі векторного переднього плану (як і в режимах роботи із тлом) інших шарів не видно.

4 Як видно на рис. 12.3, нижній край полум'я прямий, тому має неприродний вигляд. Щоб його приховати, перенесемо зображення дров на передній план. Для цього:



- 1) перейдемо в режим статичного векторного тла;
- 2) виріжемо інструментом  зображення дров;
- 3) перейдемо в режим векторного переднього плану й уставимо  зображення дров із буфера обміну;
- 4) пересунемо зображення дров так, щоб воно закривало нижній край полум'я (рис. 12.5).



Рис. 12.5

Ви ознайомилися з інструментами, які дозволяють швидко створювати проміжні кадри анімації на основі ключових кадрів та деяких параметрів. Ці прості в опануванні інструменти допомагають створювати цікаві анімації.



Питання для самоперевірки

1. Як додати до об'єкта декілька анімаційних ефектів?
2. На який кут має повертатися об'єкт на кожному кадрі, щоб в анімації із 720 кадрів він здійснив 5 обертів?
3. Які параметри задають для анімації масштабування?
4. Які параметри задають для анімації зсуву?
5. Які параметри задають для анімації щільності?
6. Які параметри задають для анімації розмальовування?
7. Поясніть, чим відрізняється векторний передній план від векторного тла.

**Вправа 12**

■ Створити анімацію з використанням декількох ефектів.

1. Запустіть редактор анімації.

Доповніть проєкт Казка, як описано в прикладі 1 (див. відеофайл Казка.mp4*).

2. За описом у прикладі 2 створіть проєкт Зорі з анімацією двох зірок, які мерехтять із різними періодами (одна — зациклена з реверсом, друга — без реверсу) (див. відеофайл Зорі.mp4*).

3. За описом у прикладах 3 і 4 створіть проєкт Вогнище з анімацією вогнища (див. відеофайл Вогнище.mp4*).



4. Додайте до проєкту Вогнище ще два анімованих язики полум'я.

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106150).




§ 13. БІБЛІОТЕКА ОБ'ЄКТІВ. ІМПОРТУВАННЯ ЗОБРАЖЕНЬ. ЕКСПОРТУВАННЯ ПРОЄКТУ

TurTube має прості засоби для власноручного малювання об'єктів. Проте якщо ви звикли малювати в інших застосунках, то можете використати свої малюнки в анімаційному проєкті.


Розгляньмо засоби застосунку TurTube, призначені для роботи з об'єктами, та дізнаймося, як із власної анімації створити відеофайл і поділитися ним із друзями та подругами.


Бібліотека об'єктів


У найпростішій анімації всі об'єкти можна намалювати вручну. Але якщо об'єкт потрібно використати декілька разів, краще скористатися *бібліотекою* — сховищем об'єктів, доступних для використання в анімаціях.

Панель Бібліотека відкривається натисканням кнопки  на лівій панелі. Спочатку бібліотека порожня.

Щоб додати власний об'єкт, наявний на кадрі, слід:

- 1) вибрати об'єкт інструментом Вибір об'єкта ;
- 2) викликати на ньому контекстне меню;
- 3) вибрати команду Додати до бібліотеки;
- 4) у діалоговому вікні, що відкриється (рис. 13.1), увести назву, яку об'єкт матиме в бібліотеці, і клацнути кнопку ОК — назва об'єкта з'явиться в списку на панелі Бібліотека.

Щоб додати на поточний кадр об'єкт із бібліотеки, потрібно клацнути його назву в списку, а потім — кнопку . Надалі доданий об'єкт використовують, як і інші об'єкти.

До бібліотеки можна додавати об'єкти різних типів: окремі растрові та векторні зображення та їх послідовності, а також звукові файли, кадри з відеофайлів. Тип об'єкта потрібно вибрати зі списку на панелі Бібліотека (рис. 13.2). Кнопка  відкриває діалогове вікно Імпорт..., у якому слід відшукати файл і клацнути Open (Відкрити). Після цього в списку бібліотечних об'єктів з'явиться ім'я доданого файлу.

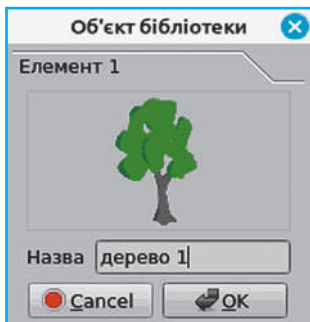


Рис. 13.1

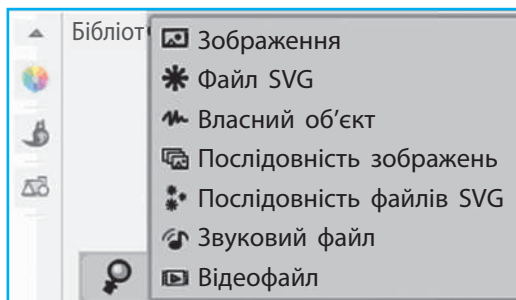


Рис. 13.2

Щоб додати послідовність зображень (растрових) або файлів SVG, потрібно:

- 1) у діалоговому вікні Імпорт... вибрати папку, в якій лежать відповідні файли;
- 2) підтвердити свій вибір — у списку з'являться назви папки і вкладених у неї файлів.

Одночасно з цим зображення з файлів буде додано на поточний кадр і на наступні кадри.

Експортування анімації

Після завершення розроблення анімації її можна експортувати у відеофайл. Для цього потрібно:

- 1) вибрати команду меню Експорт → Експортувати проєкт;
- 2) у лівій частині вікна Експорт у відео вибрати Формати відео, а в правій — один із форматів (наприклад, Відео MP4) і натиснути Далі;
- 3) зі списку в лівій частині вікна вибрати сцени проєкту, що ввійдуть у відеофайл (у найпростішому випадку в переліку є лише одна сцена).

У список сцен для відео вибрану сцену переміщують кнопкою ➔. Кожну сцену можна вибрати декілька разів.

Щоб вилучити сцену зі списку вибраних, слід натиснути кнопку ←. Коли список сцен готовий — натиснути Далі;

- 4) задати ім'я файлу для експортування та папку, куди його буде записано, і натиснути кнопку Експортувати.

По закінченні експортування в зазначеній папці з'явиться відеофайл, який можна опублікувати в інтернеті, переслати друзям і подругам тощо.

! Якщо на кроці 2 нашого алгоритму в лівій частині вікна вибрати варіант Послідовність зображень, а в правій — потрібний графічний формат, то під час експортування кожен із кадрів буде записано в окремий файл зазначеного формату.

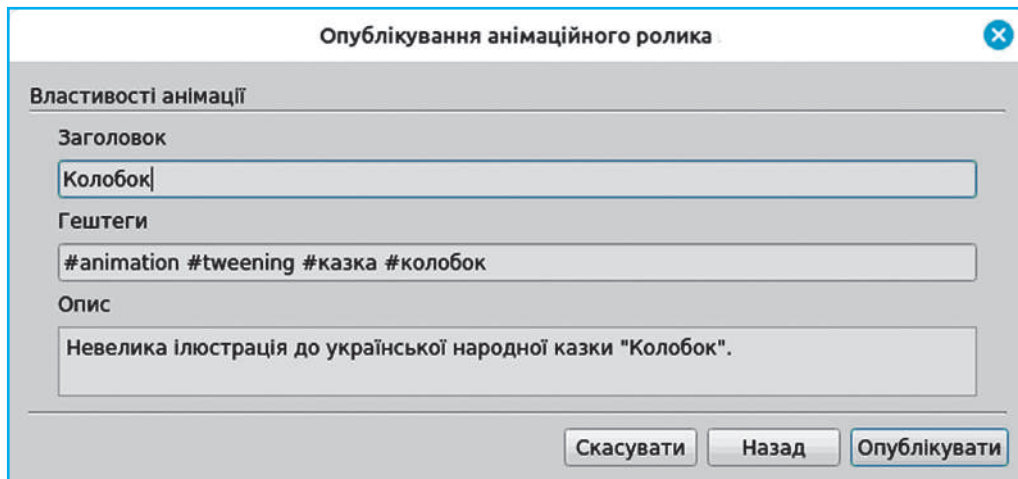


Публікування анімації в спільноті Turitube

Зареєстровані користувачі та користувачки сайту turitube.com можуть швидко опублікувати на ньому анімацію або окремий кадр. Після цього результат можна обговорити в спільноті тих, хто цікавиться Turitube.

Щоб **опублікувати свою роботу в спільноті Turitube**, потрібно:

- 1) у меню Публікування вибрати, що саме публікуєте: анімаційний ролик чи окремий кадр;
- 2) у вікні Авторизація ввести ім'я користувача та пароль із сайту turitube.com, натиснути Експортування анімації (якщо поставити позначку Зберегти пароль, то наступного разу облікові дані буде введено автоматично) і кнопку Прийняти;
- 3) вибрати сцени, які буде опубліковано (так само, як це описано для експортування у відеофайл) і натиснути кнопку Далі;
- 4) у вікні Оpubлікування анімаційного ролика ввести заголовок публікації, гештеги (ключові слова з позначкою # на початку) і короткий опис (рис. 13.3).



Опублікування анімаційного ролика

Властивості анімації

Заголовок
Колобок

Гештеги
#animation #tweening #казка #колобок

Опис
Невелика ілюстрація до української народної казки "Колобок".

Скасувати Назад Опублікувати

Рис. 13.3

Після розгляду модераторами ваш анімаційний ролик або кадр з'явиться в стрічці новин на сайті turitube.com.



Питання для самоперевірки

1. Які переваги надає використання бібліотеки об'єктів?
2. Як у TurbTube відкрити панель Бібліотека?
3. Яка послідовність додавання до бібліотеки власного об'єкта?
4. Які об'єкти можна додавати до бібліотеки?
5. Опишіть послідовність експортування анімації у відеофайл.
6. Як опублікувати власну анімацію в спільноті TurbTube?



Вправа 13

■ Створити бібліотеку об'єктів для проекту Грибний дощ.

1. Запустіть TurbTube та розпочніть проєкт із профілем 480 (PAL DV/DVD)-25 із тлом білого кольору.
2. Намалюйте гриб і згрупуйте його частини (рис. 13.4, а).
Додайте малюнок до бібліотеки з назвою гриб 1.
3. Намалюйте два кущики трави (рис. 13.4, б) і додайте їх до бібліотеки з назвами трава 1 і трава 2.
4. Намалюйте дерево і додайте до бібліотеки з назвою дерево (рис. 13.4, в).
5. Намалюйте краплю води та додайте до бібліотеки з назвою крапля (рис. 13.4, г).
6. Збережіть проєкт у файлі з назвою Грибний дощ.

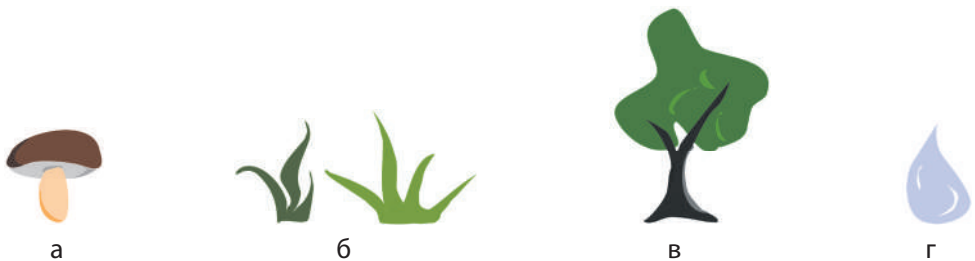


Рис. 13.4



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106151).





ПРАКТИЧНА РОБОТА 3

СТВОРЕННЯ АНІМАЦІЙНОГО ПРОЄКТУ «ГРИБНИЙ ДОЩ»

Завдання: створити анімаційний проєкт Грибний дощ тривалістю 10 с (250 кадрів) із використанням вивчених засобів редактора анімацій TurbTube.

Сценарій. Галявина лісу. Йде дощ. Подекуди з-поміж трави з'являються гриби.

Обладнання: комп'ютер із редактором анімацій TurbTube.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Запустіть TurbTube і відкрийте проєкт Грибний дощ (див. вправу 13). Установіть блакитний колір тла.
2. У режимі статичного векторного тла за допомогою інструмента Олівець та бібліотечних об'єктів сформуєте зображення (рис. 1).
3. Перейдіть у покадровий режим. Перейдіть на 25-й кадр і додайте з бібліотеки зображення гриба. Зробіть його зовсім малим за розміром (він потім виросте).
4. Додайте до гриба анімацію масштабування з параметрами:
 - кінцевий кадр — 250;
 - змінювати — ширину й висоту;
 - коефіцієнт масштабування — 1,05, ітерацій — 30.
5. Гриб збільшується від центра, тому нижній край ніжки зміщується вниз. Щоб він був нерухомим, треба, щоб одночасно зі збільшенням гриба рухався вгору. Додайте до гриба анімацію переміщення на невелику відстань вгору протягом

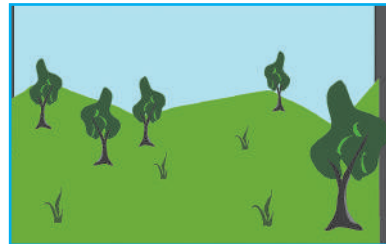


Рис. 1




- 30 кадрів (за кількістю ітерацій масштабування). Можливо, після перегляду відстань доведеться змінити.
6. Наведіть вказівник на гриб і запам'ятайте його координати (їх зазначено ліворуч внизу від робочого поля).
Перейдіть у *режим векторного переднього плану*, додайте з бібліотеки зображення кущика трави і помістіть приблизно в те місце, де розташовано гриб. Тепер в анімації гриб з'являтиметься з-за кущика трави.
 7. Перейдіть на 80-й кадр і додайте ще один гриб. Зменште гриб і зробіть так, щоб він падав з неба (перетягніть за межі кадру, додайте анімацію переміщення протягом 25 кадрів).
 8. Щоб другий гриб після падіння не зникав, на 81-му кадрі додайте нерухомий гриб такого самого розміру. Це легко зробити завдяки режиму калькування (під час роботи на 81-му кадрі видно напівпрозору копію 80-го кадру).
 9. Вставлений гриб скопіюйте , викличте контекстне меню і виберіть команду Вставити на... → наступні 100 кадрів. Повторіть операцію, щоб гриб з'явився на решті кадрів, де він відсутній. Якщо з'являться зайві кадри, то видаліть їх .
 10. У *режимі динамічного векторного тла* додайте з бібліотеки зображення краплі.
Змініть розміри зображення і зробіть копії краплі, щоб зобразити дощ (рис. 2). Задайте параметри: напрям — вниз; зміщення — 4.
 11. Відкрийте вікно параметрів режимів  і розташуйте *динамічне векторне тло* попереду статичного векторного тла.
 12. Додайте на різних кадрах ще два гриби, що виростають, і два — що падають. Перегляньте результат.
Продемонструйте анімацію однокласникам і однокласницям.




Рис. 2

Зробіть висновок: які засоби використано для створення проєкту, які кроки викликали утруднення.

§ 14. ПОКАДРОВА АНІМАЦІЯ

Анімаційні фільми протягом десятиліть створювали шляхом малювання кадрів вручну. Нині в цій непростій справі допомагає комп'ютер.

 **Покадрова анімація** — спосіб створення анімації, за якого художник малює кожен кадр майбутнього фільму.

Калькування


За автоматичного створення анімації людина задає лише вигляд ключових кадрів, а проміжні кадри будує комп'ютер. У покадровому ж режимі художник має будувати новий кадр узгодженим із попереднім.

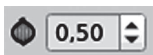
Раніше для цього використовували тоненький папір: на намальований кадр накладали новий аркуш, підсвічували знизу і так малювали новий кадр. Цю техніку називають **калькуванням** (від назви напівпрозорого паперу калька; англійською — *onion skinning*, від назви сорту паперу *onionskin*).


Під час малювання кадру TurbTube надає можливість показувати один (або більше) попередній кадр і один (або більше) наступний кадр. Їх виводять напівпрозорими, щоб не заважати огляду кадру, який малюється.

Розгляньмо засоби TurbTube, які моделюють техніку калькування. Всі вони розміщені на панелі команд модуля Анімація:



— поля для введення кількості попередніх (ліве) та наступних (праве) кадрів; кнопка  вимикає режим калькування та вмикає знову;



— поле для введення рівня прозорості попередніх і наступних шарів; кнопка  швидко встановлює значення прозорості 0,5.



Створимо анімацію стрілки, яка рухається вздовж кривої лінії. Для цього:

- 1) на *статичному векторному тлі* намалюємо допоміжну криву — траєкторію стрілки (рис. 14.1, а). Товщину та форму підберемо так, щоб зручно було малювати стрілку на кожному кадрі (на малюнку лінія має 30 штрихів);
- 2) перейдемо в *покадровий режим* і встановимо для калькування в обох полях значення 2;
- 3) на *першому* кадрі намалюємо початкове положення стрілки, завдовжки 4 штрихи допоміжної лінії (рис. 14.1, б);
- 4) перейдемо на *другий* кадр — завдяки калькуванню щойно намальовану на першому кадрі стрілку буде видно напівпрозорою. Намалюємо наступне положення стрілки, зміщеної на 2 штрихи вперед (рис. 14.1, в);
- 5) продовжимо малювати положення стрілки на *наступних* кадрах (рис. 14.1, г);

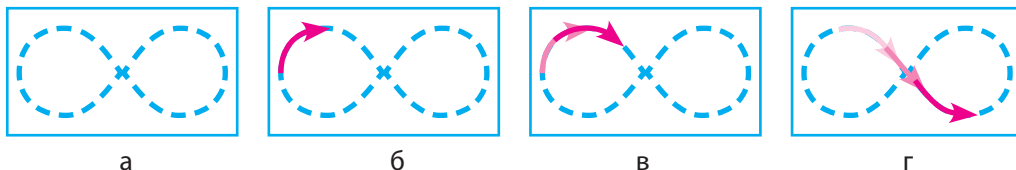




Рис. 14.1

- 6) натиснемо кнопку Параметри режимів  і вимкнемо відображення статичного векторного тла — тепер у модулі Програвач можна побачити, що стрілка рухається криволінійною траєкторією.

Перегляньмо отриману анімацію (файл Покадрова.mp4) за QR-кодом або посиланням rnk.com.ua/106243.











Під вікном перегляду є засоби керування: для швидкості — введення значення частоти кадрів у поле Кадр/с; для перегляду анімації в нескінченному циклі — прапорець Зациклити .


Робота з кадрами

Ми вже зверталися до панелі Розкадрування, коли до анімації потрібно було додати порожні кадри для збільшення тривалості анімації або додати об'єкт на певному кадрі для подальшого створення автоматичного анімаційного ефекту.


Зараз ознайомімося з роботою з кадрами докладніше.

Угорі панелі Розкадрування розміщені кнопки для виконання операцій над кадрами. Розгляньмо таблицю:

Кнопка	Опис
 Додати кадр	Додає порожній кадр після поточного; наступні кадри зсуваються донизу
 Дублювати кадр	Додає після поточного кадру його копію; наступні кадри зсуваються донизу
 Вилучити кадр	Вилучає виділені кадри; наступні кадри зсуваються вгору
 Перемістити кадр назад і  вперед	Поточний кадр зміщується на одну позицію у вибраному напрямі
 Розвернути вибрані кадри	Вибрані кадри розвертаються у зворотному порядку
 Скопіювати кадр	Діє як і в інших застосунках: вміщує в буфер обміну копії виділених кадрів;
 Вставити кадр	Поточний кадр зсувається донизу, над ним вставляються кадри з буфера обміну

 Створимо анімацію появи рукописного напису в такий спосіб. Спочатку побудуємо кадр, на якому напис видно цілком. А потім почергово дублюватимемо кадри та вилучатимемо на кадрі з меншим номером частину напису.

Для цього виконаємо такі дії:

- 1) зробимо в *покадровому режимі* напис (рис. 14.2, а);
- 2) продублюємо кадр , перейдемо на його примірник із меншим номером і вилучимо останній вузол в останній літері (рис. 14.2, б);
- 3) повторимо дублювання кадру і вилучення останнього вузла до досягнення початку напису — тепер при перегляді анімації напис з'являтиметься від початку до кінця.

Щоб після побудови напис протягом деякого часу залишався на екрані, потрібно:

- 1) перейти на останній кадр;
- 2) виділити напис і скопіювати його;
- 3) викликати контекстне меню і вибрати команду Вставити на... → наступні 50 кадрів (рис. 14.2, в).

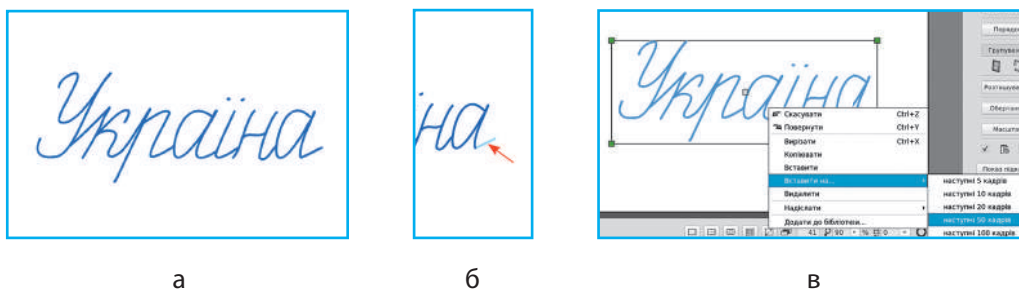



Рис. 14.2

! Щоб на наступних кадрах напис був розташований на тому самому місці, перед його копіюванням слід на панелі параметрів інструмента Вибір об'єкта *зняти* прапорець Вставляти об'єкти на місце вказівника  .

Звуковий супровід анімації

Як ви вже знаєте, до проекту TurTube можна імпортувати звукові файли (меню Імпорт → Звуковий файл). Після цього звук із файлу почне відтворюватися з кадру, на якому його додано.

Імпортовані звукові файли додаються в бібліотеку. Якщо в списку бібліотеки вибрати звуковий файл, то над списком з'явиться панель для налаштування параметрів цього звуку.

Розгляньмо рис. 14.3. Ми побачимо, що:

- у поле Відтворити на кадрі вводять номер кадру, на якому починається відтворення;
- кнопка Відтворити ► дозволяє прослухати вибраний файл; під час прослуховування вона змінюється на кнопку Пауза |||;
- кнопкою із зображенням гучномовця 🔊 можна вимкнути відтворення вибраного звуку в анімації: вмикають звук повторним натисканням цієї кнопки;
- якщо встановити прапорець Зациклити ↺, то звук під час прослуховування неперервно повторюватиметься.

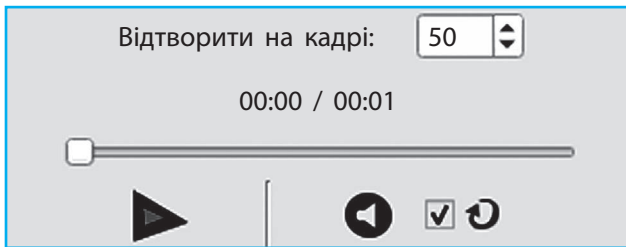


Рис. 14.3

! Якщо певний звук додано до анімації декілька разів, то й у списку бібліотеки він з'явиться стільки ж разів. Починаючи з другого примірника, в кінці назви буде додано порядковий номер.

3 Додамо звуки до проєкту «Грибний дощ» (див. Практичну роботу 3). Для цього:

- 1) відкриємо проєкт, перейдемо на перший кадр;
- 2) імпортуємо до проєкту звуковий файл gain.ogg — тепер анімація супроводжуватиметься шумом дощу;
- 3) перейдемо на 105-й кадр, де закінчується падіння гриба, й імпортуємо звуковий файл drop.ogg — падіння гриба отримає свій звуковий ефект.

Перегляньмо отримане відео (файл Грибний дощ.mp4) за QR-кодом або посиланням rnk.com.ua/106243.



Як ми бачимо, звуки можуть накладатися один на одного: звуковий ефект для падіння гриба не перериває шуму дощу, а звучить разом із ним.

Розробка складних анімацій

Дізнаймося про особливості створення складніших анімацій, ніж ми робили раніше.

► Шари

Подібно до того, як це робиться в графічних редакторах, об'єкти анімаційної сцени можуть розташовуватися на окремих шарах. Це, разом із відомими шарами растрового й векторного тла та растрового переднього плану, додає зручності в разі розроблення складних анімацій.

Розгляньмо панель Розкадрування (рис. 14.4). Угорі на ній розташовані кнопки керування шарами: Вставити шар **+**, Вилучити шар **-**.

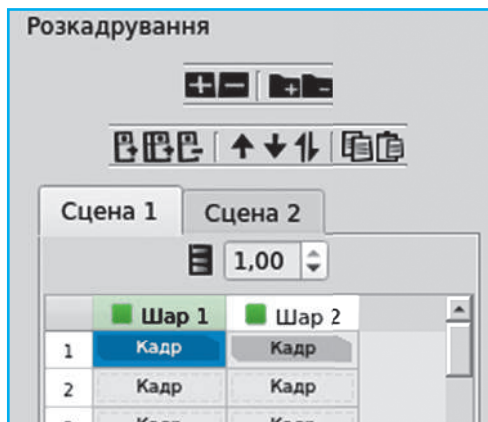




Рис. 14.4

Якщо натиснути кнопку Вставити шар , то з'явиться ще один стовпець кадрів з автоматично створеною назвою вгорі (на рис. 14.4 — Шар 2). Щоб змінити назву шару, слід двічі клацнути на ній та відредагувати.


! Шари кадру вимальовуються зліва направо, тобто об'єкти Шару 2 затулятимуть об'єкти Шару 1.



У поле  над списком шарів вводиться значення прозорості поточного шару: 0 — повна прозорість; 1 — повна непрозорість.

Порядок шарів можна змінити, перетягнувши їхні заголовки ліворуч або праворуч.

► Сцени

Анімаційний ролик або фільм може складатися з декількох **сцен** — окремих анімацій, які виводяться в певному порядку. Ви знаєте (див. § 13), що під час експортування анімації можна вибрати, які сцени і в якому порядку увійдуть до відеофайлу.

Якщо натиснути кнопку Вставити сцену , то на панелі Розкадрування з'явиться вкладка нової сцени з автоматично створеною назвою (на рис. 14.4 — Сцена 2). Сцену можна перейменувати, двічі клацнувши на її назві.

Для вилучення поточної сцени з проєкту достатньо натиснути кнопку Вилучити сцену  і в діалоговому вікні підтвердити дію кнопкою .

Слід пам'ятати, що робота з кожною сценою відбувається незалежно, з використанням усіх розглянутих засобів.





Питання для самоперевірки

1. Як створюється покадрова анімація?
2. Що таке калькування в YouTube?
3. Що на рис. 14.1, г означають три стрілки?
4. Опишіть послідовність виконання покадрової анімації.
5. Як скопіювати об'єкт на багато кадрів одночасно?
6. Як додати до анімації звук?

**Вправа 14**

■ Створити покадрову анімацію.

1. Запустіть TurbTube, розпочніть проєкт із профілем 480 (PAL DV/DVD)-25 зі тлом білого кольору.
2. Створіть анімацію з рухомою стрілкою, керуючись описом у матеріалі параграфа (приклад 1, с. 84). Збережіть проєкт у файлі з назвою Стрілка.
-  3. Створіть анімацію з появою власного імені (приклад 2, с. 85). Збережіть проєкт у файлі з назвою Ім'я.
4. Додайте до проєкту Грибний дощ звуковий супровід із файлів rain.ogg* і drop.ogg* (приклад 3, с. 87).
5. Експортуйте анімацію у файл формату MP4.
-  6. Перегляньте анімацію й обговоріть з однокласниками та однокласниками.

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106152).

**ПРАКТИЧНА РОБОТА 4
СТВОРЕННЯ АНІМАЦІЇ З ДЕКІЛЬКОХ СЦЕН**

Завдання: створити анімаційний проєкт «Весна» з використанням вивчених засобів редактора анімацій.



Сценарій. Сцена 1. Ніч. Зоряне небо. Земля вкрита снігом. Сцена 2. Сходить сонце. Сцена 3. Сніг тане. Перегляньте відео за QR-кодом або посиланням rnk.com.ua/106243.



Обладнання: комп'ютер із редактором анімацій TurbTube.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть TurbTube, розпочніть проєкт із профілем 480 (PAL DV/DVD)-25.
2. **Намалюйте небо та сніг.** Для цього додайте на статичне векторне тло темно-синій прямокутник — нічне небо, а також декілька сірих плям неправильної форми — сніг.
3. **Намалюйте зорі.** Для цього:
 - а) у *покадровому режимі* намалюйте зірку; згрупуйте її частини і додайте до бібліотеки;
 - б) із бібліотеки додайте ще декілька зірок (рис. 1).
4. **Додайте мерехтіння зірок.** Для цього:
 - а) додайте до трьох зірок анімацію часткового обертання з реверсом;
 - б) скопіюйте нерухомі зірки на наступні 100 кадрів;
 - в) видаліть 101-й кадр.
5. Додайте до проєкту сцену з назвою Сцена 2. Для цього:
 - а) на Сцені 1 скопіюйте всі об'єкти зі *статичного векторного тла*;
 - б) перейдіть на Сцену 2, зніміть прапорець Вставляти об'єкти на місце вказівника;
 - в) вставте сніг на *статичний векторний передній план*, а небо — в *покадровому режимі* на 1-й кадр.
6. Скопіюйте і вставте на цей кадр зорі. На місце відсутніх зірок (які на Сцені 1 обертаються) додайте нові з бібліотеки.
7. **Намалюйте сонце,** згрупуйте його частини. Для цього:
 - а) розташуйте сонце позаду всіх об'єктів (інструмент Вибір об'єктів  → Порядок → 
 - б) зробіть те саме з об'єктом, який зображає небо — тепер сонце видно на тлі неба, його можуть закривати інші об'єкти;
 - в) додайте зображення сонця до бібліотеки.

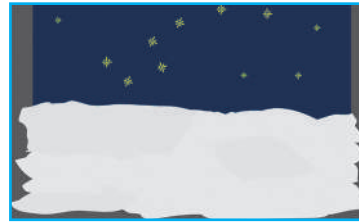


Рис. 1

8. Додайте анімацію сонця (рис. 2). Анімуйте схід сонця — переміщення на небо по кривій лінії протягом 2 с (50 кадрів). Додайте також обертання протягом 8 с (200 кадрів).
9. **Зникнення зірок.** Додайте для зірок анімацію щільності з такими параметрами:
 - початковий кадр — 1, кінцевий кадр — 20;
 - початкова щільність — 1, кінцева щільність — 0;
 - ітерацій — 20.
10. **Танення снігу** (рис. 3). Додайте до проєкту Сцену 3. На *статичному векторному тлі* розмістіть прямокутник (небо) такого самого кольору, яким закінчилась анімація кольору неба у Сцені 2. Намалюйте поверхню землі, що має з'явитися з-під снігу.



Рис. 2



Рис. 3

11. У *покадровому режимі* вставте з бібліотеки зображення сонця та додайте анімацію обертання тривалістю 200 кадрів.
12. Із *растрового тла* Сцени 1 скопіюйте об'єкти, що зображають сніг, і вставте на перший кадр Сцени 3. Додайте анімацію *зникнення* — зменшення щільності. Орієнтовні параметри такі:
 - кінцевий кадр — 200;
 - початкова щільність — 1, кінцева щільність — 0;
 - ітерацій — 80.
 Для різних частин значення зробіть дещо різними, щоб танення мало природніший вигляд. Перегляньте відео.

Зробіть висновок: що варто додати, змінити, і як це можна зробити.

РОЗДІЛ 3

МОДЕЛІ ТА МОДЕЛЮВАННЯ



§ 15. Поняття моделі. Поняття предметної галузі

§ 16. Типи моделей. Форми подання інформаційної моделі

§ 17. Побудова інформаційних моделей

§ 18. Карти знань. Сервіси для побудови карт знань

§ 19. Побудова математичної моделі

Практична робота 5. Створення карти знань

ПОВТОРЮЄМО



У курсі 6 класу ви ознайомилися з поняттям моделі. Знаєте, що *модель* (від латин. *modulus* — міра, зразок) — спрощене подання об'єкта, процесу або явища у вигляді таблиці, схеми тощо. А *моделюванням* називають заміну об'єкта на його модель.

Ви вже дізналися, чому й як використовують моделі для дослідження об'єктів навколишнього світу та перевірки різних гіпотез. Також навчилися поділяти моделі за способом подання на *матеріальні* та *інформаційні*.

Ви маєте початкове уявлення про послідовність етапів проведення комп'ютерного експерименту. Можете наводити приклади застосування комп'ютерного моделювання в різних галузях людської діяльності.

1. Що таке модель об'єкта або явища?
2. Які існують типи моделей за способом подання?
3. Що таке моделювання?
4. Які моделі ви використовуєте на уроках?
5. Наведіть приклади інформаційних моделей.
6. Що таке гіпотеза; як її можна підтвердити або спростувати?



У цьому розділі ви вдосконалили знання та навички створення інформаційних моделей, оволодієте новими засобами моделювання.

§ 15. ПОНЯТТЯ МОДЕЛІ. ПОНЯТТЯ ПРЕДМЕТНОЇ ГАЛУЗІ

Всесвіт, що нас оточує і частиною якого ми є, складається з об'єктів, з якими відбуваються певні процеси або явища. Для дослідження або споглядання чи використання об'єктів люди вдаються до моделювання.

Об'єкти навколишнього світу

Отже, об'єкти є складовими навколишнього світу. Дізнаймося, як їх класифікують, розгляньмо наведені приклади (рис. 15.1).



Рис. 15.1

Існують об'єкти *матеріальні* (наприклад, людина, собака, будинок) і *нематеріальні* (текст, формула тощо). Кожний об'єкт має властивості, що відрізняють його від інших об'єктів. Наприклад, об'єкт автомобіль має такі властивості, як назва, марка, колір, об'єм двигуна, рік випуску тощо.

Принцип і причини створення моделі

Люди часто вдаються до моделювання. Що до цього спонукає?

Модель — матеріальний або уявлюваний аналог деякого об'єкта, що зберігає його суттєві риси і здатний замінювати під час вивчення, дослідження або відтворення.

Об'єкт, що моделюється, — *прототип*, або *оригінал*.

Моделю об'єкта може бути його зменшена копія, креслення, схема, які відображають його структуру, взаємозв'язки окремих елементів. Наприклад, макет корабля, креслення будівлі, схема комп'ютера. Модель явища або процесу відтворює його хід, послідовну зміну стану, етапи розвитку. Наприклад, модель еволюції людства, модель розвитку економіки.

Моделювання — це спосіб дослідження об'єктів шляхом побудови й аналізування їх моделей.

Існують випадки, коли дослідження за допомогою моделей може бути єдиним можливим способом експериментального вивчення об'єкта.

До **створення моделі** можуть спонукати такі причини:

- надзвичайно повільний (наприклад, формування гір) або швидкоплинний (блискавка) процес;
- на момент дослідження оригіналу може не існувати (динозавр або можлива «ядерна зима»);
- об'єкт за розмірами надзвичайно великий (земна куля) або надзвичайно малий (атом);
- об'єкт або процес неможливо дослідити безпосередньо (процеси всередині зорі, виверження вулкана);
- досліджуваний процес небезпечний для життя людини або може нашкодити довкіллю (ядерний вибух);
- дослідження може нашкодити самому об'єкту (дослідження міцності моста).

Для дослідження об'єкта створюють модель, яка обов'язково має ті властивості, що й досліджуваний оригінал. Несуттєвими для дослідження властивостями нехтують.

Типи моделей за способом реалізації

Як вам відомо, за способом реалізації моделі поділяють на матеріальні (натурні, фізичні) й інформаційні (абстрактні).

Матеріальна модель відтворює фізичні й геометричні властивості об'єкта і має реальне втілення. Наприклад, глобус (рис. 15.2), лялька, аеродинамічна труба, макет літака й ін.

На рис. 15.3 наведено матеріальну модель Сонячної системи.

Інформаційна модель описує важливі для дослідження властивості досліджуваного об'єкта, але не має матеріального втілення.

На рис. 15.4 наведено інформаційну модель Сонячної системи.



Рис. 15.2

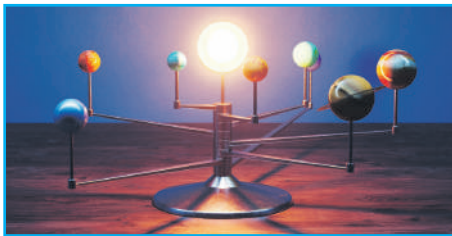


Рис. 15.3



Рис. 15.4

- Найперші згадки про земні глобуси в літературі належать до III–II ст. до н. е. (глобус Архімеда, глобус Кратеса з Пергами). Найдавніший глобус, що дійшов до нашого часу, зберігається в Нюрнберзі (Німеччина). Його виготовив німецький географ і мандрівник Мартін Бегайм (1459–1507) 1492 року. На ньому земна куля ще без Америки, Австралії та Антарктиди.



Програма Google Earth (назва українською мовою Google Планета Земля) — це віртуальний глобус, інформаційна модель Землі. У рамках проєкту в інтернеті викладено аерофотознімки та супутникові знімки більшої частини нашої планети.

2 Для одного об'єкта можна побудувати багато різних моделей. Якщо розв'язують задачу на знаходження шляху, пройденого за певний час автомобілем, що рухається зі сталою швидкістю, то враховують лише швидкість руху і час. А такими характеристиками, як колір, марка чи ціна, нехтують.

Поняття предметної галузі

Множину всіх об'єктів, що належать до певної сфери людської діяльності, називають **предметною галуззю**. Наприклад, це може бути металургійна промисловість, медицина, освіта й ін.

3 Досліджуючи об'єкти предметної галузі «Хімія», вивчають будову, склад і властивості речовин, а досліджуючи об'єкти предметної галузі «Біологія» — будову та поведінку живих організмів. Об'єкти моделювання точка, відрізок, кут, геометрична фігура належать до предметної галузі «Геометрія».

У предметній галузі можна виділяти та вивчати різні об'єкти — залежно від цілей дослідження. Предметну галузь «Географія» діти розглядають у школі, а розробники Google-карт або дослідники-мандрівники — в реальних умовах.

Отже, моделі об'єктів створюються для наочного подання матеріальних об'єктів, здійснення наукових досліджень, експериментальної перевірки гіпотез, ефективного керування процесами при розробі пристроїв і зведенні споруд.



Питання для самоперевірки

1. Що таке модель? Наведіть приклади моделей об'єктів, явищ.
2. Навіщо потрібне моделювання?
3. Наведіть приклади явищ, дослідження яких без моделювання неможливе.
4. Наведіть приклади матеріальних моделей із різних галузей.
5. Глобус — модель земної кулі. Наведіть приклади явищ (процесів), у ході моделювання яких використовується глобус.
6. Чи можна робота-андроїда вважати моделлю людини? Чому?

Вправа 15

- Визначити причини створення моделей об'єктів.
- 1. Відкрийте у вебглядачі сторінку сайту «Розмір Всесвіту».

Сайт «Розмір Всесвіту» призначений для візуалізації розмірів об'єктів Всесвіту (рис. 15.4). Він дає змогу наочно порівняти розміри об'єктів навколишнього світу, що реально існували і вже зникли: наприклад, розміри слона та тиранозавра.

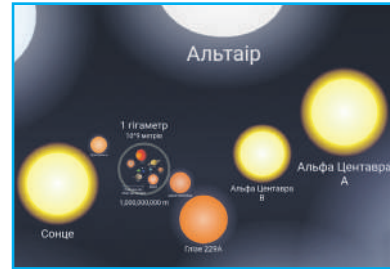


Рис. 15.4

- 2. Запустіть текстовий процесор та створіть новий документ.
- 3. Створіть наведену таблицю.

Причина	Приклад	Розмір об'єкта
Об'єкт надзвичайно великий		
Об'єкт надзвичайно малий		
Об'єкт не існує в цей момент часу		
Природне явище		

- ⚙ 4. Перейдіть у вікно «Розмір Всесвіту». Віддаляючи та наближаючи зображення мишею, знайдіть об'єкти, моделі яких створено з причин, наведених у таблиці.
- 5. Занесіть до таблиці значення властивості розмір об'єкта.
- 6. До кожного рядка відшукайте по два приклади. Збережіть документ із назвою Вправа 15. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106153).



§ 16. ТИПИ МОДЕЛЕЙ. ФОРМИ ПОДАННЯ ІНФОРМАЦІЙНОЇ МОДЕЛІ

Для об'єкта можна створити кілька моделей залежно від призначення та мети моделювання. Як можна класифікувати моделі?

Моделі та їх типи

Моделі можна поділити за наведеними ознаками (рис. 16.1).



Рис. 16.1

За **галузями використання** моделі поділяють на навчальні, дослідницькі та ігрові.

Навчальні моделі використовують для пояснення будови певного об'єкта, роботи механізму тощо (наприклад, мапа України (географія), будова квітки (біологія)). *Дослідницькі моделі* використовують для детальнішого вивчення явищ природи, взаємодії різних об'єктів тощо (дослідження блискавки). *Науково-технічні моделі*: дослідження процесів та явищ (прилад для дослідження розряду блискавки). *Ігрові моделі* застосовують, щоб проаналізувати поведінку об'єктів у непередбачуваних ситуаціях (військові навчання — це ігрова модель справжніх воєнних дій).

За **фактором часу** моделі поділяють на статичні та динамічні.

Статичні моделі показують певний незмінний стан реального об'єкта (пластиліновий будиночок є статичною моделлю

справжнього будинку). *Динамічні моделі* дають змогу перевірити, як властивості предметів або явищ змінюються з часом (метеорологи, які вивчають погоду, на основі багаторічних даних будують модель, що допомагає робити прогноз погоди).

За предметними галузями (галузями знань) моделі поділяють на фізичні, біологічні, економічні, соціальні та ін.

За способом реалізації моделі поділяють на матеріальні й інформаційні (див. § 15).

Моделі та форма їх подання

Інформаційні моделі можна класифікувати за **формою подання** (рис. 16.2).



Рис. 16.2

Словесними моделями є усні та письмові описи об'єктів.



Як приклад словесної моделі розглянемо наведену загадку: «Без вікон, без дверей — повна хата людей».

Загадка містить інформацію про одну з властивостей гарбуза (відсутність отворів) і про наявність у ньому однакових насінин. Тут «людина» є моделлю насінини.

До *графічних моделей* належать креслення, малюнки, плани приміщень, мапи тощо (наприклад, план міста).


Структурні моделі показують зв'язки між частинами об'єкта у вигляді діаграм, таблиць, схем. Модель, подану у вигляді набору команд, правил, називають *алгоритмічною*.

Спеціальні моделі відрізняються від перелічених і одна від одної. При їх побудові застосовують особливі системи позначень (наприклад, запис мелодії за допомогою нот). Модель, подану у вигляді набору формул, називають *математичною*.

Комп'ютерне моделювання

Ще одним ефективним методом наукового пізнання є комп'ютерне моделювання. Як ви знаєте, це унікальний інструмент пізнання швидкоплинних (вибух наднової зірки) або надзвичайно повільних (наприклад, геологічних) процесів. Їх можна досліджувати на комп'ютері, розтягуючи або стискаючи час або навіть зупиняючи його для вивчення певних фаз процесу.

Моделювати і вивчати за допомогою комп'ютера можна й явища, що не відбувалися, і невідомо, чи відбудуться коли-небудь у реальному житті. Наприклад, зіткнення Землі з астероїдом.

 **Комп'ютерна модель** — інформаційна модель, реалізована за допомогою програмного середовища: текстового / графічного редактора, редактора презентацій, середовища програмування.

Комп'ютерні моделі зручно досліджувати, адже вони допомагають спостерігати й досліджувати явища та процеси в динаміці їхнього розгортання; здійснювати багаторазові випробування моделі; отримувати різноманітні кількісні показники в числовому або графічному вигляді, зокрема такі, що вимагають виконання складних, багаторазових або трудомістких розрахунків.

Зважаючи на все викладене, комп'ютерні моделі теж можна поділити на такі групи.

Графічна модель — це образ, об'єкта поданий у вигляді схем, діаграм, малюнків тощо та створений за допомогою

комп'ютерних технологій. Це може бути, наприклад, модель динозавра, модель Сонячної системи (рис. 16.3, 16.4 відповідно).



Рис. 16.3



Рис. 16.4

Розрахункова модель дає змогу на основі математичної моделі об'єкта визначати числові значення його властивостей за різних вхідних даних. Аналізування цих значень допомагає прогнозувати стан об'єкта в майбутньому.

Наприклад, модель земної атмосфери дозволяє передбачити погоду, модель екосистеми — визначити межі користування природними ресурсами. На рис. 16.5 наведено розрахункову модель рівноприскореного прямолінійного руху.



Рис. 16.5

Імітаційна модель — це програма або комплекс програм, що реалізує алгоритм функціонування об'єкта за різних умов.

Імітаційні моделі створюють засобами комп'ютерної графіки. Як, наприклад, навчальні моделі для тренування, гра в шахи (рис. 16.6); ігрові моделі, що дають змогу програвати поведінку

об'єкта в ігрових ситуаціях з урахуванням реакції з боку суперника (на рис. 16.7 — авіаційний симулятор польоту).

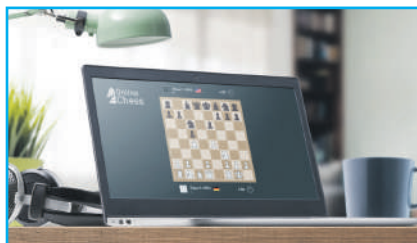


Рис. 16.6



Рис. 16.7

Комп'ютерний експеримент

Для дослідження складних об'єктів і явищ часто застосовують **комп'ютерний експеримент**. Це дослідження математичної моделі з використанням комп'ютера, при якому за одними параметрами моделі обчислюють інші й на цій основі роблять висновки про властивості об'єкта, описані математичною моделлю.

Етапи комп'ютерного експерименту наведено на рис. 16.8.

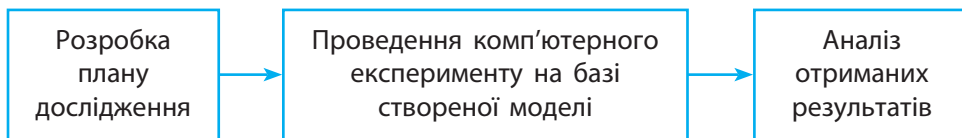


Рис. 16.8

Комп'ютерний експеримент є чисельною реалізацією математичної моделі; результат експерименту характеризує властивості саме моделі. У разі розбіжності результатів комп'ютерного і натурного експериментів з прототипом говорять про неадекватність математичної моделі.

У ході моделювання вивчаються об'єкти і явища, які неможливо, дорого або небезпечно відтворювати в реальних умовах. Проведення таких експериментів дає змогу економити матеріальні ресурси та зберігати екологічні умови існування людини, уникати можливих шкідливих наслідків випробувань.




Питання для самоперевірки

1. На які типи поділяють моделі?
2. Наведіть приклад навчальної моделі; інформаційної моделі.
3. Від чого залежить вибір інформаційної моделі?
4. Коли модель стає комп'ютерною?
5. Чи можна створити модель за допомогою різних програм?



Вправа 16

■ Провести дослідження за допомогою сервісу інтерактивних симуляцій.

1. Завантажте сервіс PhET, який надає доступ до онлайн-моделей з низки предметних областей (рис. 16.9). У списку Симуляції виберіть Вивчення Землі.
2. У переліку симуляцій виберіть Густина. Натисніть на значок , щоб почати перегляд симуляції.
3. Виберіть розділ Вступ і поекспериментуйте з різними матеріалами, змінюючи масу та об'єм блока. Опишіть, як поняття густини стосується маси тіла та об'єму.
4. Перейдіть у розділ Порівняння. Проведіть експеримент. Чому зміна маси тіла або об'єму не впливає на його густину?
5. Перейдіть у розділ Таємничі об'єкти. Проведіть експеримент:
 - а) виміряйте об'єм тіла шляхом спостереження за кількістю рідини, яку воно витісняє, і визначте його масу;
 - б) обчисліть густину тіла, порівняйте значення густини невідомого матеріалу з відомою густиною із таблиці.
6. Зробіть висновок за результатами роботи. Завершіть роботу.

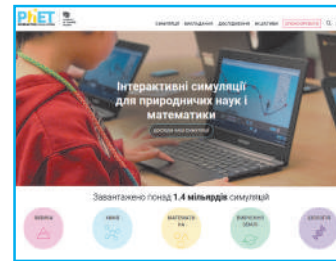


Рис. 16.9



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106154).



§ 17. ПОБУДОВА ІНФОРМАЦІЙНИХ МОДЕЛЕЙ

Створення інформаційної моделі є важливим кроком у дослідженні об'єкта (процесу, явища). Він допомагає організувати та структурувати інформацію, необхідну для досягнення мети дослідження.

Процес створення інформаційної моделі з використанням комп'ютерних технологій можна подати поетапно. А саме:

1. Визначення мети створення інформаційної моделі
2. Виділення властивостей об'єкта, що є суттєвими для дослідження
3. Виявлення взаємозв'язків між значеннями вибраних властивостей
4. Вибір форми подання інформаційної моделі
5. Реалізація моделі в програмному середовищі
6. Дослідження створеної комп'ютерної моделі



Розгляньмо етапи створення моделі на прикладі.



Задача. Обчислити час руху автомобіля, якщо відомі швидкість його руху (км/год) і шлях (км).

- 1. Визначення мети створення інформаційної моделі

Метою моделювання процесу руху об'єкта автомобіль є дослідження рівномірного руху об'єкта. Її можна віднести до типу «як зробити, щоб...»: який час потрібен автомобілю, щоб подолати певний шлях?

- 2. Виділення властивостей об'єкта, суттєвих для дослідження

Для побудови інформаційної моделі необхідно:

- зрозуміти, в якій предметній галузі шукати опис об'єктів моделювання;
- відібрати ознаки, суттєві для розв'язуваного завдання.

Процес руху автомобіля як об'єкт моделювання належить до предметної галузі «Фізика». Як обговорювалося раніше, суттєвими властивостями об'єкта автомобіль є швидкість його руху, час перебування в дорозі, пройдений шлях.

► 3. Виявлення взаємозв'язків між значеннями вибраних властивостей

Із курсів математики та фізики ви знаєте таке: якщо відомі швидкість і час рівномірного руху тіла, то можна знайти шлях, який пододало тіло. Для цього слід швидкість руху помножити на час: $l \cdot v = t$, де l — шлях; v — швидкість руху; t — час руху із зазначеною швидкістю.

► 4. Вибір форми подання інформаційної моделі

Інформаційна модель може бути подана у формі таблиці характеристик (параметрів) об'єкта. Для задач, у яких потрібно розрахувати значення параметрів об'єкта, необхідно скласти математичну модель.

Інформаційна модель задачі про рух автомобіля має такий вигляд:

Параметри	Значення
v — швидкість, км/год l — шлях, який подолав автомобіль, км	Вихідні дані
t — час руху, год	Результат

► 5. Реалізація моделі в програмному середовищі

Для задач, у яких потрібно отримати розрахункові значення, підходить середовище електронних таблиць. У ньому інформаційна та математична моделі об'єднуються в таблицю, що містить вихідні дані, проміжні розрахунки і результати. Електронна таблиця дає змогу не лише виконати розрахунки, а й побудувати графік руху автомобіля.

► 6. Дослідження створеної комп'ютерної моделі

На етапі дослідження можна провести комп'ютерний експеримент, який відповідає цілям моделювання.

Таким чином, щоб створити інформаційну модель, треба вибрати об'єкт, установити мету дослідження, виділити основні властивості й визначити форму подання інформаційної моделі.



Питання для самоперевірки

1. Опишіть порядок дій побудови інформаційної моделі об'єкта.
2. Які засоби можуть бути використані для подання інформаційних моделей?
3. Складіть інформаційну модель «піаніно» для розв'язування задач, що постають перед:
 - а) вантажником; б) директоркою магазину; в) піаністом.
4. Знайдіть в інтернеті модель вулкана. Які властивості прототипу враховано під час побудови моделі; не враховано?
5. Створіть модель Сонячної системи, яку подано в графічній формі, яка наочно відобразить суттєві властивості об'єкта дослідження.
Засіб для подання графічної моделі — графічний редактор.
6. Створіть інформаційну модель кабінету інформатики в різних програмних засобах.



Вправа 17

■ Розробити інформаційну модель руху планет Сонячної системи по орбітах.

1. З'ясуйте мету побудови моделі — визначення періоду обертання планет (рис. 17.1).
2. Виберіть для моделі структурну форму подання.



Рис. 17.1


- Засіб для подання структурної моделі — табличний процесор.
3. Запустіть табличний процесор.
Створіть новий документ із порожньою таблицею.
Заповніть таблицю значеннями властивостей планет (рис. 17.2).
 4. У клітинці D2 створіть формулу розрахунку періоду обертання Меркурія навколо Сонця (рис. 17.3).
 5. Скопіюйте формулу для решти планет Сонячної системи із використанням маркера автозаповнення.

	A	B	C	D
1	Назва планети	Радіус орбіти (км)	Швидкість руху (км/год)	Період обертання (доби)
2	Меркурій	57629350	171360	
3	Венера	107246064	125280	
4	Земля	148885361	106560	
5	Марс	226511083	86400	
6	Юпітер	774238930	46800	
7	Сатурн	1420204678	34560	
8	Уран	2941667032	24516	
9	Нептун	4493699220	19548	

Рис. 17.2

	A	B	C	D
1	Назва планети	Радіус орбіти (км)	Швидкість руху (км/год)	Період обертання (доби)
2	Меркурій	57629350	171360	=2*3,14*B14/C14/24
3	Венера	107246064	125280	

Рис. 17.3

6. Запишіть значення тривалості року на Марсі, яке отримано в результаті обчислень, результат округліть до цілого.
7.  За допомогою інформаційної моделі дослідіть:
 - а) як віддаленість від Сонця впливає на тривалість року на планеті?
 - б) чи залежить швидкість руху по орбіті від розмірів планети?
8. Збережіть документ із назвою Вправа 17. Завершіть роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106155).



§ 18. КАРТИ ЗНАНЬ. СЕРВІСИ ДЛЯ ПОБУДОВИ КАРТ ЗНАНЬ

Під час розв'язування багатьох завдань буває складно тримати в пам'яті одночасно багато чинників і зв'язків між ними. У низці випадків зручно подавати інформацію у вигляді діаграм із окремих блоків, між якими є з'єднувальні лінії.

Інформаційна модель «Інтер'єр кімнати»

Далі розглянемо матеріал на прикладі побудови інформаційної моделі Інтер'єр кімнати. На початковому етапі розроблення моделі сформульовано основне завдання — розробити інтер'єр. Після цього виділено шість напрямів, які потребують більш детального опрацювання. У ході подальшої роботи обсяг інформації зростає і модель ускладнюється (рис. 18.1).

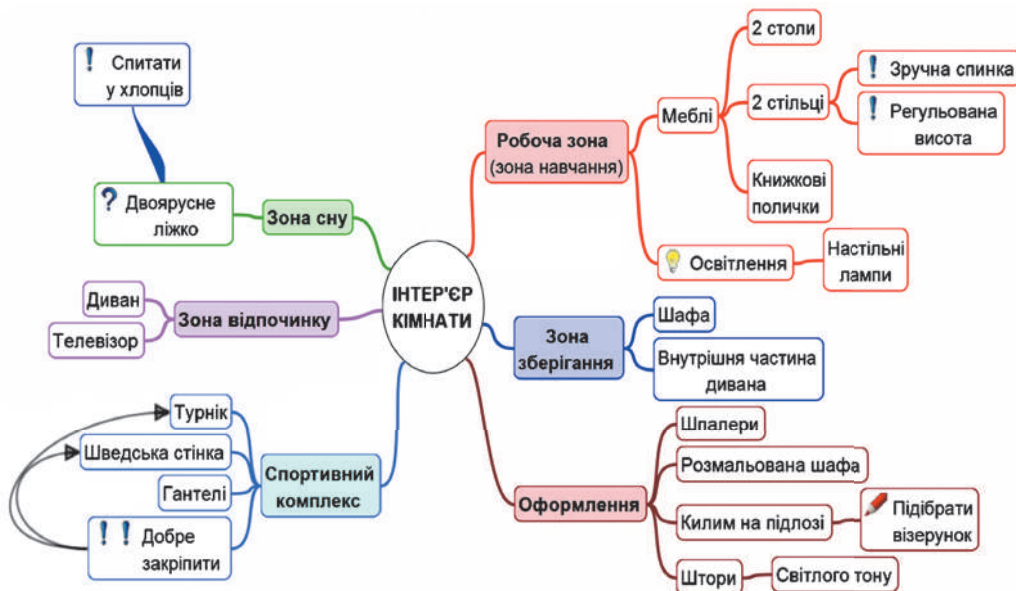


Рис. 18.1

Інформаційну модель, подану в такому вигляді, називають картою знань, асоціативною картою, картою розуму.

Карта знань (англ. *mind map*) — інформаційна модель, подана в структурній формі, яка наочно відображає взаємозв'язки між об'єктами та явищами деякої предметної галузі.

Автором теорії карт знань є британський психолог і консультант із питань освіти Тоні Б'юзен. Він розробив методику запам'ятовування й організації мислення.

Ознайомімося з **основними принципами** створення карт знань:

- об'єкт вивчення сфокусовано в центральному вузлі;
- основні теми й ідеї, пов'язані з об'єктом вивчення, розходяться від центрального вузла у вигляді гілок;
- гілки пояснено та позначено ключовими образами і словами;
- ідеї наступного порядку (рівня) також зображено у вигляді гілок, що відходять від центральних гілок і так далі;
- гілки формують зв'язану структуру (систему).

Звичайно, карту знань можна створити на папері. Проте набагато зручніше створити її за допомогою текстових процесорів, графічних редакторів, редакторів карт знань.

Для побудови карт знань існують різні інструменти (рис. 18.2).



Рис. 18.2

Користуючись такими інструментами, можна розробляти та опрацьовувати карти знань: додавати до вузлів карти знань графічні об'єкти, створювати гіперпосилання на допоміжні ресурси, керувати показом карти знань, шукати в ній дані, а також відкривати, зберігати, друкувати тощо.

Редактор карт знань FreeMind

Одним із сервісів для побудови карт знань є редактор FreeMind (англ. *free* — вільний, *mind* — розум) (рис. 18.3). Програму можна вільно завантажити з офіційного сайту розробки FreeMind.



Рис. 18.3

Під час встановлення FreeMind на робочому столі створюється ярлик, який можна використати для запуску програми. Після цього з'являється її головне вікно. Розгляньмо основні елементи керування (рис. 18.4).

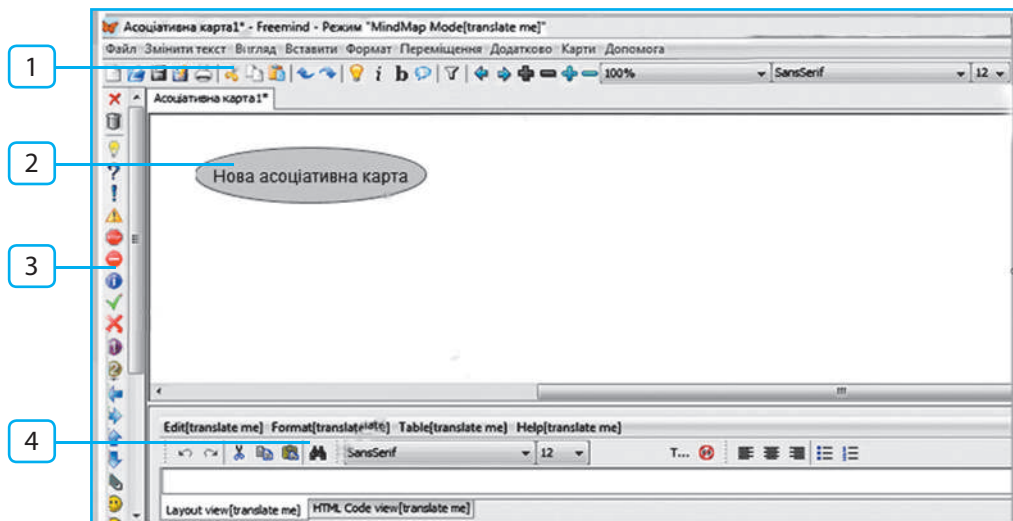


Рис. 18.4

Тут 1 — панель інструментів; 2 — центральний вузол; 3 — панель піктограм; 4 — вікно редактора приміток.


Основним об'єктом карти знань є вузол. **Вузол** — це текстове поле, призначене для відображення певних даних. Для наочності об'єкти на карті знань зв'язані гілками. (Щоб карта помістилася в робочому полі, частину гілок можна приховати.)

Зазвичай на карті знань є один центральний об'єкт — *центральний вузол*. Він відображає основну ідею карти знань. Вузол вищого рівня називають *батьківським*, а нижчого — *дочірнім*. Вузли одного рівня, що мають спільний батьківський вузол, називають спорідненими.

► Робота з вузлами

Якщо вибрати команду меню Файл → Новий (або кнопку на кнопковій панелі), то на робочому полі з'явиться центральний вузол майбутньої карти знань (див. рис. 18.4).

Під час роботи з вузлами можна виконувати низку операцій. Команди для їх виконання розміщено в меню Формат, а також у контекстних меню вузлів карти знань. Розгляньмо таблицю:

Дія	Опис
Змінити текст у вузлі	Двічі клацнути ЛКМ або в контекстному меню вузла вибрати вказівку Редагувати
Додати до вибраного вузла підлеглий (дочірній)	Клавішею Insert або натисканням на кнопку 
Додати до вибраного вузла споріднений	Клавішею Enter
Додати до вузла малюнок з графічного файлу	Зберегти карту і виконати Вставити → Вибрати файл з картинкою
Об'єднати декілька вузлів	Команда Вставити → Хмара
Змінити тип вузла, параметри тексту, колір вузла або ліній	Команди меню Формат
Зберегти файл у форматі програми з розширенням .mm	Команда Файл → Зберегти
Експортувати карту в документ іншого формату	Команда Файл → Експортувати

Отже, навіщо потрібні карти знань? Використання карт знань допоможе:

- покращити запам'ятовування інформації;
- активніше генерувати ідеї щодо розвитку проєкту;
- точніше проаналізувати результати роботи, факти, події;
- проаналізувати і вдосконалити структуру твору, сайту тощо;
- ефективніше організувати роботу в групі.



Питання для самоперевірки

1. Із чого починається розроблення інформаційної моделі?
2. Наведіть приклад створення інформаційної моделі.
3. Що таке карта знань?
4. Які можливості надає використання карти знань?
5. Назвіть основні об'єкти карти знань.
6. Як додати до вузла підлеглий вузол?



Вправа 18

■ Створити карту знань «Геометричні фігури», у якій відобразити властивості основних фігур (круг, прямокутник, трикутник тощо).

1. Запустіть редактор карт знань. У центральний вузол уведіть текст «Геометричні фігури».
2. Додайте до центрального вузла підлегли вузли за кількістю складових частин і впишіть у них їхні назви (рис. 18.5).



Рис. 18.5

3. Додайте до вузла Круг підлеглий вузол і додайте до вузла зображення круга з графічного файлу.

4. Додайте до вузла з малюнком підлеглі вузли наступного рівня і зазначте в них властивості фігури Круг.
5. Об'єднайте в хмару вузол із малюнком і підлеглі до нього вузли (рис. 18.6).

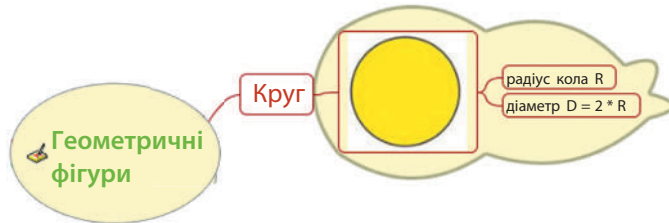


Рис. 18.6

6. Додайте до вузлів Прямокутник, Квадрат, Трикутник підлеглі вузли наступного рівня (рис. 18.7). Зазначте в них описи властивостей відповідних фігур.

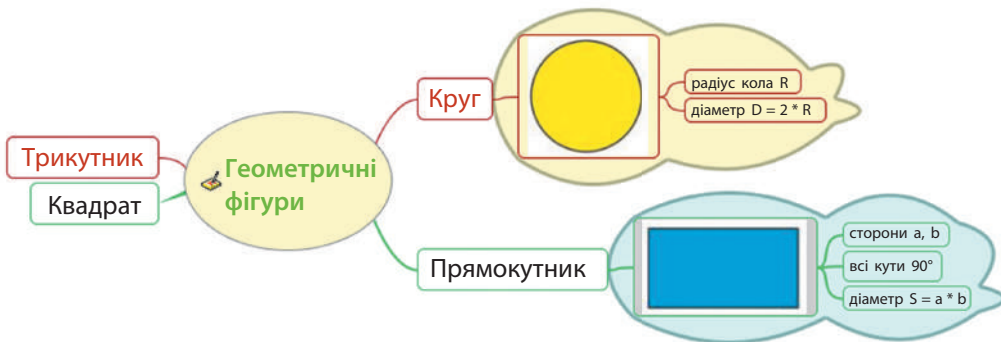


Рис. 18.7

7. Збережіть карту знань у файл із назвою Вправа 18.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106156).



§ 19. ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ

Як ви знаєте, математичне моделювання можна застосовувати у випадку, коли властивості об'єкта чи явища підлягають опису математичними формулами.

Наприклад, на уроках фізики ви ознайомилися з поняттями сили тяжіння і ваги тіла. Знаєте, що коли тіло перебуває в стані спокою або прямолінійного рівномірного руху, то його вага збігається за напрямком із силою тяжіння і дорівнює їй за значенням: $P = mg$. Це математична модель ваги тіла.

Математичне моделювання передбачає кілька етапів.

1. Постановка завдання
2. Побудова математичної моделі
3. Реалізація моделі в програмному середовищі
4. Дослідження створеної комп'ютерної моделі

Розгляньмо етапи створення моделі на прикладі.



Улітку учні й учениці 7 класу здійснили подорож на теплоході Дніпром. Шлях униз Дніпром тривав a годин, а повернення — b годин. Діти дізналися, що середня швидкість течії Дніпра на цій ділянці становить v км/год і зацікавилися, з якою швидкістю рухався теплохід. Для пошуку відповіді діти вирішили скласти комп'ютерну програму.

Проаналізуємо кроки, які потрібно виконати для створення та реалізації математичної моделі задачі.

► 1. Постановка завдання

На цьому етапі слід прийняти рішення про необхідність моделювання та чітко визначити і сформулювати мету досліджень. Бо з неї впливатиме сукупність властивостей об'єкта моделювання, що підлягатимуть відбиттю в моделі.

Потрібно також з'ясувати, до якої предметної галузі належать об'єкти, згадані в умові задачі; визначити властивості об'єктів, що є суттєвими для розв'язання задачі, та відкинути несуттєві властивості; позначити змінні для зберігання значень величин.

Усе згадане вимагає чіткого виділення вихідних даних і потрібних результатів. При цьому встановлюються обмеження на припустимі значення використаних величин.

Отже, математична постановка завдання — це точне формулювання умов і мети розв'язання. На цьому етапі потрібно чітко визначити умови задачі: *що дано; які дані допустимі; які результати, в якому вигляді мають бути отримані.*

Умову задачі тоді можна сформулювати в такий спосіб.

Теплохід пройшов a годин за течією і b годин — проти течії. Знайти швидкість руху теплохода в стоячій воді, якщо швидкість течії становить v км/год (рис. 19.1).



Рис. 19.1

► 2. Побудова математичної моделі

На цьому етапі потрібно розгорнутий опис задачі замінити математичною моделлю за допомогою математичних залежностей.

Математична модель — це опис математичних співвідношень між числовими величинами в умові задачі.

Для побудови такої моделі слід установити зв'язок між потрібними результатами і вхідними даними (записати рівняння, нерівності), який забезпечує розв'язання завдання.

Записувати математичну модель зручно в такий формі:

Що дано? <Перелік початкових даних>
Що треба знайти? <Перелік потрібних результатів>
Який зв'язок між величинами? <Математичні співвідношення, що зв'язують вхідні дані та результати>
Які є обмеження на дані? <Умови допустимості початкових даних>

Запишімо математичну модель задачі.

Що дано?	a — час руху за течією (год); b — час руху проти течії (год); v — швидкість течії (км/год)
Що треба знайти?	x — власна швидкість теплохода (км/год)
Який зв'язок між величинами?	$x = v \cdot \frac{b+a}{b-a}$
Які є обмеження на дані?	$0 \leq a \leq b$

► 3. Реалізація моделі в програмному середовищі

Раніше ми вже розглянули підходи до реалізації математичної моделі задачі в середовищі електронних таблиць. Реалізуємо математичну модель у середовищі програмування.

На цьому етапі потрібно розробити алгоритм реалізації моделі; скласти програму за розробленим алгоритмом; виконати тестування і налагодження програми. При складанні алгоритму необхідно враховувати всі його властивості (див. § 17, 5 клас).

Розробка і складання алгоритму — найважливіший етап розв'язування задачі.

Від якості алгоритму залежать правильність результатів, ефективність використання часу й оперативної пам'яті комп'ютера. Алгоритм розв'язування задачі є лінійним (рис. 19.2).

Програмування (складання програми) — кодування складеного алгоритму однією з мов програмування. Ми реалізуємо алгоритм мовою Python:

```
a = float(input('a = ?'))
b = float(input('b = ?'))
v = float(input('v = ?'))
x = v*(b+a)/(b-a)
print('x = ', x)
```

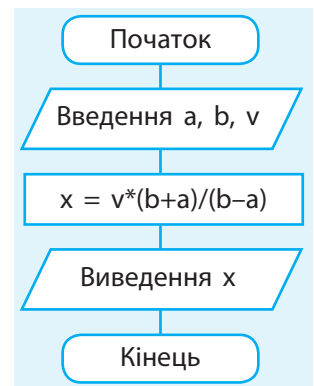


Рис. 19.2

На цьому етапі програму потрібно перевірити за допомогою тестів і виправити виявлені помилки. *Тест* — це набір спеціально дібраних вхідних даних і відповідних їм результатів.

Тестування полягає в порівнянні очікуваних результатів із результатами, отриманими під час виконання програми. Найчастіше розбіжність результатів роботи програми з тестовими спричиняють недоліки математичної моделі та помилки в алгоритмі.

Випробуємо програму на такому тесті: при $a=4.0$, $b=4.48$, $v=1.5$ результатом має бути значення $x=26.5$.

Отримаємо результат: $x=26.49999999999998 \approx 26.5$, який збігається з наведеним у тесті.

► 4. Дослідження створеної комп'ютерної моделі

На цьому етапі можна провести комп'ютерний експеримент, який відповідає цілям моделювання. Він має супроводжуватися осмисленням підсумків, аналізуванням результатів моделювання, що є основою для прийняття рішень.

Слід запуснути програму з даними умови задачі і здійснити аналіз отриманих результатів. Змінюючи вхідні дані, можна аналізувати взаємозв'язок між значеннями властивостей об'єкта.

Отже, математичні моделі можуть бути використані для розуміння природних, соціальних явищ тощо, прогнозування значень властивостей об'єктів, висунення та перевірки гіпотез.



Питання для самоперевірки

1. Яку модель називають математичною?
2. Назвіть основні етапи математичного моделювання.
3. Виведіть формулу зв'язку між величинами, наведеними на с. 118 угорі.
4. Побудуйте математичну модель задачі.

Задача. Визначити площу кільця, утвореного двома колами зі спільним центром (рис. 19.3).

5. Побудуйте математичну модель задачі.

Задача. У магазин привезли P кг яблук, а груш — у N разів більше. Скільки всього фруктів привезли в магазин?

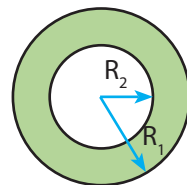


Рис. 19.3



Вправа 19

- Побудувати та реалізувати математичну модель задачі.
Задача. Визначити вагу тіла на Місяці та Марсі.

1. Постановка завдання

Суттєвими параметрами задачі є прискорення вільного падіння; маса фізичного тіла, вагу якого моделюємо. Такими параметрами, як наявність твердої поверхні, обертальний рух планети, наявність атмосфери тощо можна знехтувати.
Вхідні дані: m — маса тіла (кг); g — прискорення вільного падіння на конкретній планеті (Н/кг). Потрібно визначити значення P — ваги тіла (Н).

2. Побудова математичної моделі

Що дано?

m — маса тіла (кг);
 g — прискорення вільного падіння на конкретній планеті (Н/кг)

Що треба знайти?

P — вага тіла (Н)

Який зв'язок між величинами? $P = mg$

Які є обмеження на дані? $m > 0$; $g > 0$

3. Реалізація моделі в програмному середовищі

Відкрийте табличний процесор і створіть Нову книгу.
Створіть таблицю за зразком (рис. 19.4).

	A	B	C	D
1	Назва планети	g (Н/кг)	Маса тіла (кг)	Вага тіла (Н)
2	Земля	9,81		45
3	Місяць	1,62		45
4	Марс	3,72		45

Рис. 19.4



4. Дослідження створеної комп'ютерної моделі

- 1) У клітинку D2 уведіть формулу, що виражає зв'язок між величинами задачі: $=B2*C2$.

- 2) За допомогою маркера автозаповнення скопіюйте формулу в діапазон D3:D4.
- 3) Доповніть таблицю відомостями про інші планети Сонячної системи. Знайдіть в інтернеті та впишіть значення g на кожній планеті.
- 4) У стовпці D розрахуйте вагу тіла на кожній планеті.
- 5) Чи можна стверджувати, що об'єкти на Марсі важать третину своєї земної ваги?
Зробіть висновки за даними таблиці.
- 6) Збережіть документ із назвою Вправа 19.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106157).



ПРАКТИЧНА РОБОТА 5 СТВОРЕННЯ КАРТИ ЗНАНЬ

- Завдання:** створити карту знань «Частини мови», використовуючи програмний засіб.
- Обладнання:** комп'ютер із редактором карт знань FreeMind.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть редактор карт знань FreeMind.
У центральний вузол уведіть текст Частини мови.
2. Командами меню Формат змініть значення параметрів шрифту напису та фоновий колір вузла.


3. Проаналізуйте вигляд центрального вузла (рис. 1); позначте команди, що були виконані при форматуванні цього вузла.
4. Додайте до центрального вузла підлеглий вузол кнопкою  або клавішею Insert. Уведіть у текстове поле текст Іменник. У контекстному меню вузла задайте тип вузла — овал.
5. До вузла Іменник додайте підлеглий вузол. Упишіть у текстове поле питання, на які відповідають іменники: хто? що?
6. До вузла Хто? Що? додайте споріднений вузол і впишіть у текстове поле приклади іменників.
7. Виділіть хмарою об'єкти, що належать до групи Іменник. Змініть колір утвореної хмари.
8. За кількістю самостійних частин мови додайте до центрального вузла дочірні вузли і впишіть туди їхні назви. У контекстному меню вузлів задайте тип вузла — овал. Процес створення карти знань показано на рис. 2.



Рис. 1



Рис. 2

9. Додайте підлеглі вузли наступного рівня і зазначте в них питання, на які відповідають частини мови. Додайте до вузлів піктограму Питання.
10. Додайте вузли, споріднені вузлам, які створено в п. 9. Уведіть у текстові поля приклади частин мови.
11. Командами меню Формат змініть значення параметрів шрифту написів і фоновий колір вузлів.
12. Збережіть карту знань у графічний файл із назвою Практична 5 у форматі PNG.

Зробіть висновок: які переваги надає використання карти знань.

РОЗДІЛ 4

ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ



- § 20. Основні об'єкти електронних таблиць
- § 21. Типи даних. Введення та опрацювання даних
- § 22. Адресація в електронних таблицях
- § 23. Автозаповнення та автозавершення
Практична робота 6. Розв'язування задач на обчислення
- § 24. Функції
- § 25. Логічні функції
- § 26. Умове форматування
Практична робота 7. Використання функцій табличного процесора. Умове форматування
- § 27. Математичне моделювання в електронних таблицях
Практична робота 8. Проведення обчислювального експерименту

ПОВТОРЮЄМО



У 6 класі ви зробили перші кроки в опануванні *табличного процесора* — прикладної програми для опрацювання даних, поданих в *електронних таблицях*.

Ви ознайомилися з *інтерфейсом користувача* табличного процесора, набули практичних навичок у створенні й опрацюванні електронних таблиць.

Електронна таблиця складається з клітинок, вишикуваних у *рядки* та *стовпці*. Кожна клітинка може містити *дані різних типів*.

Ви знаєте, що таке *діапазон клітинок* і вмієте виділяти діапазони різних видів. Ви з'ясували, з чого складається *адреса* клітинки, чим *формула* відрізняється від даних.

1. Що таке табличний процесор?
2. Із чого складається адреса клітинки?
3. Чи можуть збігатися адреси різних клітинок?
4. Що можуть містити клітинки електронної таблиці?
5. Для чого призначені формули?
6. Із якого знака починається запис формули в табличному процесорі?



У цьому розділі ви докладніше ознайомитеся з табличним процесором, дізнаєтеся про способи його використання. Ви навчитеся проводити засобами цього застосунка обчислювальні експерименти.

§ 20. ОСНОВНІ ОБ'ЄКТИ ЕЛЕКТРОННИХ ТАБЛИЦЬ

Часто дані в електронних таблицях (ЕТ) потребують опрацювання. Повторімо й узагальнімо знання про об'єкти таблиці.

Об'єкти електронної таблиці

Електронна таблиця — комп'ютерне подання прямокутної таблиці, клітинки якої можуть містити дані або формули опрацювання даних.

Із клітинок (комірок) формуються рядки та стовпці таблиці. Іноді ЕТ називають програму для її опрацювання. Ми називатимемо її табличним процесором.

Табличний процесор — прикладна програма для опрацювання даних, поданих в електронних таблицях.

Документом табличного процесора є *книга*. Вона містить один або більше аркушів, на кожному з яких розташована ЕТ.

Існують різні табличні процесори. Один із них — табличний процесор Microsoft Excel (рис. 20.1, далі — Excel), який входить до складу комерційного пакунка Microsoft Office.

Завдяки відомим вам хмарним технологіям, із ЕТ можна працювати в онлайн-режимі. Маючи обліковий запис Microsoft, можна безкоштовно користуватися Excel на хмарному диску OneDrive.

ЕТ також можна створити на Google Диску (рис. 20.2), і знадобиться нам для цього лише браузер. При цьому дані з таблиць легко скопіювати чи перемістити на комп'ютер і навпаки.



Рис. 20.1

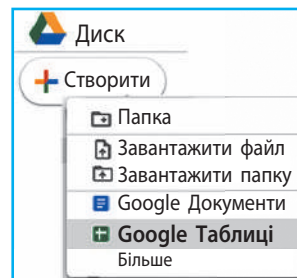


Рис. 20.2

Інтерфейс користувача

Інтерфейс табличного процесора подібний до інтерфейсу інших офісних застосунків — текстового процесора та редактора презентацій. На рис. 20.3 наведено вікно програми.

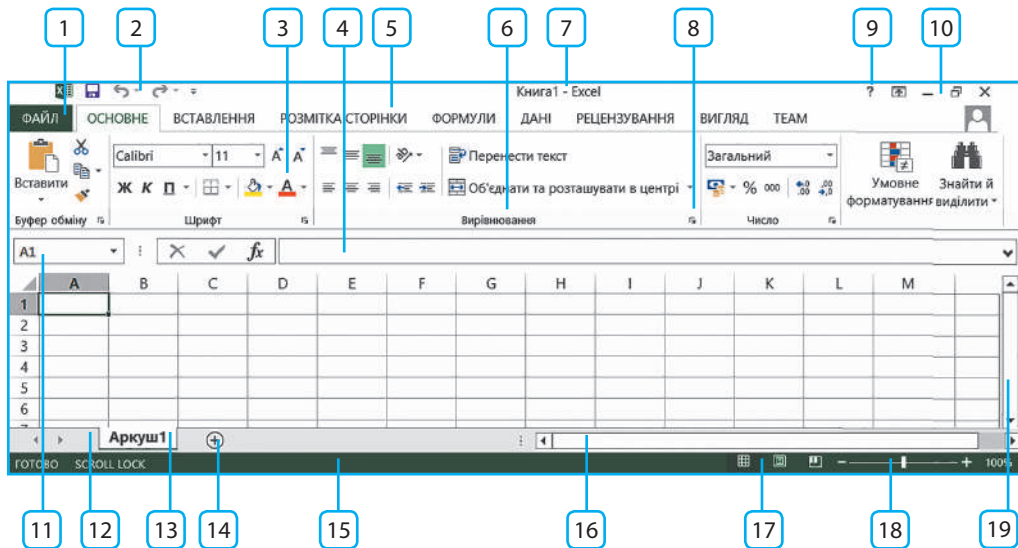


Рис. 20.3

Пригадаймо основні елементи інтерфейсу (див. рис. 20.3). Тут 1 — вкладка Файл (Офіс); 2 — панель швидкого доступу; 3 — стрічка інструментів; 4 — рядок формул; 5 — меню вкладок; 6 — група інструментів; 7 — рядок заголовка; 8 — кнопка виклику діалогового вікна; 9 — кнопка довідки; 10 — кнопки керування вікном; 11 — поле Ім'я; 12 — нижня панель; 13 — вкладка аркуша (вкладок може бути кілька); 14 — кнопка створення нового аркуша; 15 — рядок стану; 16, 19 — смуги горизонтального та вертикального прокручування; 17 — кнопки режиму перегляду; 18 — засоби масштабування.

Як бачимо, у робочому полі вікна програми зображено таблицю. Ліворуч від таблиці зазначено номери рядків, а вгорі — назви стовпців. На рис. 20.3 активна клітинка A1.

Робота з об'єктами таблиці


Об'єктами ЕТ є *клітинка, діапазон клітинок, рядок, стовпець, аркуш, книга* тощо. Із ними можна виконувати низку операцій: редагування, форматування, копіювання, видалення й ін.

► Виділення клітинок і діапазонів клітинок

Перед початком роботи з об'єктом його потрібно виділити.

Клітинки найчастіше виділяють одинарним (для уведення даних чи форматування) або подвійним клацанням (для редагування наявних даних). Клітинки також можна виділити введенням адреси чи назви клітинки в поле Ім'я (рис. 20.3, позначка 11).


Якщо потрібно опрацювати дані кількох клітинок одночасно, то працюють із діапазоном.

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

Розрізняють зв'язні й незв'язні діапазони.

Зв'язним називають діапазон, який можна виділити переміщенням указівника з натиснутою лівою кнопкою миші (ЛКМ). Щоб **виділити зв'язний діапазон**, потрібно перетягнути вказівник від однієї клітинки до іншої або ж клацнути одну клітинку, а потім із натиснутою клавішею Shift — іншу.

Виділений діапазон є прямокутним і позначається іншим кольором, окрім клітинки, з якої почалося виділення (рис. 20.4, с. 128). Початкова й кінцева клітинки містяться в протилежних кутах прямокутника.

 Окрему клітинку також вважаємо зв'язним діапазоном, — її виділяють клацанням ЛКМ.

Адреса *зв'язного* діапазону складається з адрес двох клітинок: верхньої зліва та нижньої справа.

Незв'язним називають діапазон, який складається з кількох зв'язних діапазонів. Щоб **виділити незв'язний діапазон**, слід, утримуючи клавішу Ctrl, виділити кілька клітинок і/або діапазонів (рис. 20.5, с. 128).

	A	B	C
1			
2			
3			

Рис. 20.4

	A	B	C	D
1				
2				
3				
4				

Рис. 20.5

Адреса *незв'язного* діапазону складається з окремих адрес зв'язних діапазонів, розділених крапкою з комою.



B2:C3 — прямокутний діапазон із чотирьох клітинок (див. рис. 20.4); B7 — діапазон з однієї клітинки;
C1:D2;A2;B4 — незв'язний діапазон (див. рис. 20.5).

▶ Діапазони рядків і стовпців

Щоб **виділити один рядок (стовпець)** таблиці, потрібно клацнути його номер (заголовок). А для **виділення декількох суміжних рядків (стовпців)** треба перетягнути вказівник по їхніх номерах (заголовках).

Під час виділяння в прямокутнику-підказці зазначається кількість рядків — R і стовпців — C, що входять до виділеного діапазону (рис. 20.6).


	A	B	C	D	E	F
1						
2						
3						
4						
5						

Рис 20.6





Адреса діапазону рядків (стовпців) містить адреси першого й останнього рядка (стовпця), відокремлені двокрапкою:


- 2:4 — діапазон з усіх клітинок рядків 2, 3 і 4;
- C:F — діапазон з усіх клітинок стовпців C, D, E і F.

Щоб виділити всю таблицю, потрібно в її лівому верхньому кутку клацнути значок  або натиснути сполучення клавіш Ctrl+A. А щоб зняти виділення діапазону, достатньо клацнути на будь-якій клітинці.

Збереження книги

Щоб зберегти таблицю для подальшого використання, слід:

- 1) на панелі швидкого доступу клацнути значок Зберегти  або на вкладці ФАЙЛ вибрати команду Зберегти;
- 2) зазначити місце збереження — одну з наявних у списку папок або вибрати іншу папку, скориставшись кнопкою  Огляд ;
- 3) увести назву файлу в поле Ім'я файлу і клацнути Зберегти.

Щоб зберегти копію документа в іншу папку та/або з іншим іменем, слід вибрати команду Зберегти як..., а щоб із будь-якого кроку збереження документа повернутися в режим редагування, — натиснути кнопку  або клавішу Esc.



Питання для самоперевірки

1. Що таке електронна таблиця? Для чого її застосовують?
2. Що таке табличний процесор?
3. Які є види діапазонів клітинок?
4. Скільки клітинок містить діапазон A2:B3?
5. Як виділити діапазон стовпців або рядків?
6. Як зберегти електронну таблицю у власну папку?



Вправа 20


- Пригадати основні прийоми роботи з табличним процесором; навчитися виділяти діапазони різних видів.
- 1. Запустіть текстовий процесор — для формування звіту про роботу.
Створіть у папці ярлик для запуску табличного процесора. Запустіть табличний процесор.

2. Створіть таблицю з календарем на поточний місяць (рис. 20.7). Збережіть у файл із назвою Календар.

3. Створіть порожню таблицю.

Виділіть у ній кілька суміжних рядків, запишіть у звіт кількість стовпців у них. Визначте загальну кількість рядків і запишіть у звіт.

Визначте загальну кількість клітинок у таблиці та запишіть у звіт.

 4. Виділіть незв'язний діапазон (рис. 20.8), рядки та стовпці виділяйте повністю. Зробіть знімок екрана і вставте у звіт.

	Листопад				
Пн		6	13	20	27
Вт		7	14	21	28
Ср	1	8	15	22	29
Чт	2	9	16	23	30
Пн	3	10	17	24	
Сб	4	11	18	25	
Нд	5	12	19	26	

Рис. 20.7

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										

Рис. 20.8

 5. Підрахуйте кількість клітинок виділеного діапазону (п. 4).



6. Збережіть звіт у файл із назвою Вправа 20.

Надішліть учителеві/вчительці листа, прикріпивши до нього файли Календар і Вправа 20.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106158).



§ 21. ТИПИ ДАНИХ. ВВЕДЕННЯ ТА ОПРАЦЮВАННЯ ДАНИХ

Електронні таблиці можуть опрацьовувати дані різних типів. Слід пам'ятати, що від типу даних залежить, які операції можна застосовувати для їх опрацювання. Одні й ті самі дані можуть виводитися на екран у різних форматах.

Класифікація типів даних

У клітинках таблиці можуть міститися дані та формули:

Вміст	Опис
Числові дані	Мають вигляд числа, з яким можна проводити обчислення. Можуть містити допоміжні символи: %, \$ тощо
Текстові дані	Мають вигляд рядка символів. До них здебільшого не можна застосовувати математичні операції
Формули	Арифметичні, логічні та/або інші вирази для опрацювання даних у клітинках. Можуть містити посилання, функції тощо

! Спочатку клітинки порожньої таблиці мають формат загальний. Завдяки цьому тип уведених даних визначається автоматично.

Текст може складатися з будь-яких символів, зокрема цифр. Число в клітинці з текстовим форматом виводиться, як рядок, але з ним можна виконувати обчислення.

Із числовими даними можна виконувати обчислення. Відображення цих даних на екрані залежить від вибраного формату.

Дані числового типу можна подати в наведених форматах:

Формат	Пояснення
Числовий	Виведення чисел із заданою кількістю десяткових знаків
Грошовий	До числа додається знак грошової одиниці

Продовження таблиці

Формат	Пояснення
Фінансовий	Вирівнювання за роздільником цілої і дробової частин
Дробовий	Звичайні дроби
Експоненційний	Стандартний вигляд числа
Дата	Відображення дати (не може мати від'ємного значення)
Час	Відображення часу або дати й часу (не може мати від'ємного значення)

Загальний формат

Як уже зазначалося, за замовчуванням у клітинках ЕТ встановлено формат Загальний, за якого тип уведених даних визначається автоматично. При цьому програма інколи «поміляється», сприймаючи дані не так, як сподівався користувач.



Після введення тексту –Я вчуся в клітинці побачимо: #NAME?. Це табличний процемор «вирішив», що введено від'ємне число, оскільки на початку стоїть знак «-».

Щоб уникнути помилок визначення типу даних, необхідно дотримуватися певних правил введення. Ознайомімося з деякими з них, розгляньмо таблицю:

Що вводиться	Правило
Текст	Введення розпочинати з апострофа: «'»
Відсотки	Введення закінчувати знаком відсотка: «%»
Звичайний дріб	Ціла частина чисельник/знаменник (ціла частина вводиться завжди, навіть коли дорівнює 0)
Десятковий дріб	Ціла та дробова частини відокремлюються комою
Дата	Уведення у форматі: день/місяць/рік (рік вводити не обов'язково)
Час	Уведення у форматі: година:хвилина:секунда (секунди вводити не обов'язково)

Введення даних

У клітинки можна вводити текст, числа, формули.

Якщо клітинку виділити клацанням і ввести дані, то наявні в клітинці дані буде замінено введеними. Під час уведення даних стають активними два значки: ✕ і ✓ (рис. 21.1).

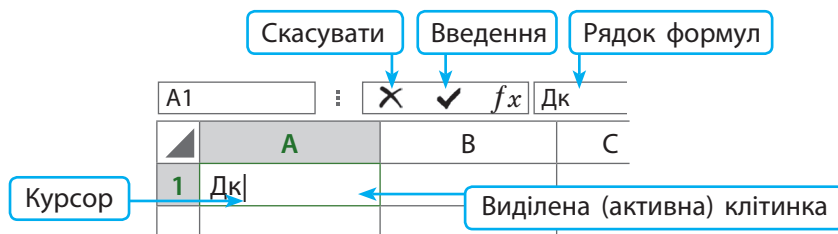


Рис. 21.1

Уводити й редагувати дані можна як у клітинці, так і в рядку формул. Якщо рядок формул відсутній, то на вкладці Вигляд слід установити позначку Рядок формул.

Уведення даних можна підтвердити клацанням значка ✓ або іншої клітинки. Щоб увести дані та перейти до сусідньої клітинки праворуч (нижче), слід натиснути клавішу Tab (Enter). Щоб скасувати розпочате введення даних, можна натиснути клавішу Esc або клацнути значок ✕.

► Особливості відображення даних

Слід пам'ятати, що вміст клітинки на екрані може відрізнятися від вигляду даних під час уведення в клітинку таблиці.

Після підтвердження введення даних текст, який не вміщується в клітинку, виступатиме на порожні клітинки праворуч (рис. 21.2). Якщо ж клітинка праворуч теж містить дані, то «зайвий» текст буде сховано за її лівою межею. Уведені дані при цьому не втрачаються, їх можна побачити в рядку формул.

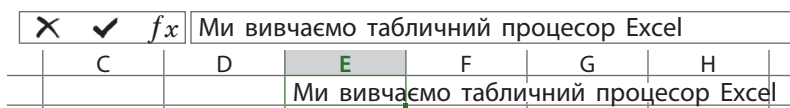


Рис. 21.2

Дробову частину, яка не вміщується в клітинку, буде автоматично округлено (рис. 21.3). Якщо ціла частина не поміщається в клітинці, то число подається в стандартному вигляді (рис. 21.4), якщо це не вдається, виводяться «решітки» (рис. 21.5).

✕	✓	<i>f</i> x	1234567,89
C	D	E	
			1234567,9

Рис. 21.3

✕	✓	<i>f</i> x	1234567,89
C	D	E	
			1E+06

Рис. 21.4

✕	✓	<i>f</i> x	1234567,89
C	D	E	
			####

Рис. 21.5

Редагування даних

Уведені в таблицю дані за потреби можна редагувати (змінювати). У цілому редагування даних в ЕТ схоже на редагування даних у текстовому документі. Для редагування наявних у клітинці даних її потрібно виділити подвійним клацанням.

Видалити символи в клітинці можна клавішами Backspace або Delete, скасувати останні дії — сполученням клавіш Ctrl+Z або кнопкою ✕.

У програмі доступні також операції копіювання, вирізання, вставлення даних відповідними сполученнями клавіш Ctrl+C, Ctrl+X, Ctrl+V або командами контекстного меню.

Форматування вмісту клітинок

Для кращого сприйняття таблиці варто приділяти увагу її оформленню. Це досягається форматуванням вмісту клітинок.

► Форматування символів

Пригадаймо призначення інструментів форматування символів (рис. 21.6): 1 — гарнітура; 2 — розмір шрифту; 3 — накреслення; 4 — встановлення меж клітинки; 5 — колір заливки; 6 — змінення розміру шрифту; 7 — колір шрифту; 8 — відкриття вікна налаштування шрифту.

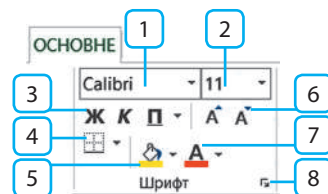


Рис. 21.6

Щоб змінити **формат вмісту клітинки** достатньо клацнути клітинку та вибрати потрібний інструмент. Якщо ж потрібно змінити формат частини тексту клітинки, слід перед цим клацнути двічі та виділити фрагмент тексту.

Форматування фрагментів вмісту клітинок із нетекстовими даними неможливе.

► Вирівнювання вмісту в клітинці

Текст у клітинці автоматично вирівнюється за лівим краєм, а числа — за правим. Щоб змінити вирівнювання вмісту, потрібно спочатку виділити клітинку (одинарним клацанням!) або діапазон, а потім скористатися відповідними інструментами вкладки Основне групи Вирівнювання.

Залежно від вибраних інструментів горизонтального й вертикального вирівнювання є 9 варіантів розташування (рис. 21.7).

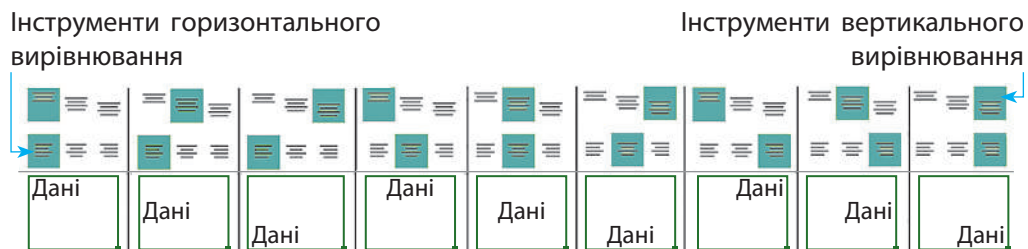


Рис. 21.7


► Орієнтація вмісту клітинок


Уміст клітинки іноді доводиться повертати. Пригадаймо сторінку щоденника, де назви днів тижня написано знизу вгору.

Щоб **повернути вміст клітинки на довільний кут**, потрібно:

- 1) на вкладці Основне в списку вибрати команду Формат вирівнювання клітинки, або розгорнути вікно налаштувань значком , або в контекстному меню клітинки вибрати команду Формат клітинок;
- 2) у вікні Формат клітинок відкрити вкладку Вирівнювання;
- 3) у блоці Орієнтація задати кут повороту або перетягнути маркер Напис на потрібний кут і натиснути кнопку ОК.

► Встановлення формату за зразком

Щоб застосовувати **формат** певної клітинки до інших клітинок, слід її виділити, на вкладці Основне клацнути кнопку **Формат за зразком** ; протягнути вказівник-пензель по клітинках, для яких потрібно застосувати вибраний формат.

Для **багаторазового застосування формату за зразком** кнопку  необхідно клацнути двічі, а для виходу з цього режиму — натиснути клавішу Esc.

► Змінення розмірів стовпців і рядків

Щоб **змінити ширину стовпця (висоту рядка)**, достатньо перетягнути його праву (нижню) межу в рядку (стовпці) заголовків.

При цьому виводяться поточні розміри: у пунктах і пікселях — для рядків, у символах і пікселях — для стовпців за звичайного режиму перегляду (рис. 21.8).

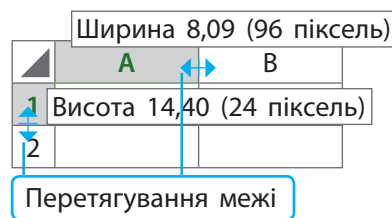


Рис. 21.8



Питання для самоперевірки

1. Які типи даних використовуються в електронних таблицях?
2. Опишіть особливості форматів числових даних для виведення дати й часу.
3. Як скасувати введення даних у клітинку? Як підтвердити?
4. Як побачити дані, якщо вміст не помістився в клітинці?
5. Як вирівняти вміст клітинки? Як його повернути?
6. Як змінити ширину стовпця; висоту рядка?



Вправа 21

■ Дослідити способи введення й редагування даних та форматування вмісту клітинок в електронній таблиці.

1. Запустіть табличний процесор і створіть новий документ. Заповніть таблицю даними за зразком (рис. 21.9).

Зверніть увагу на дані в рядку формул.

B1		✗ ✓ <i>fx</i>		Прізвище ім'я по батькові			
	A	B	C	D	E	F	
1	Номер	Прізвище	Дата				

Рис. 21.9

- Зменште приблизно втричі ширину стовпців A і C за зразком (рис. 21.10). Чому текст у стовпці A видно не повністю, а у стовпці C — повністю?
- Збільште приблизно вп'ятеро висоту першого рядка та вчетверо ширину стовпця B. Заповніть чотири клітинки даними про однокласників і однокласниць за зразком (рис. 21.11).

	A	B	C	D	
1	Но	Прізвище	Дата		

Рис. 21.10

	A	B	C
	Но м е р	Прізвище, ім'я, по батькові	Дата
1			
2		<i>Петренко Петро Петрович</i>	
3		<i>Іванчук Іван Іванович</i>	
4		<i>Сумлінна Ліна Богданівна</i>	
5		<i>Бойко Олена Вікторівна</i>	

Рис. 21.11

- Поверніть уміст клітинок верхнього рядка за зразком.
- Вирівняйте вміст за горизонталлю та вертикаллю за зразком.
- Установіть для ПІБ учнів і учениць колір — червоний; накреслення — жирний, курсив. Збережіть документ із назвою Вправа 21. Надішліть листа вчителю / вчительці з поясненням до п. 2 у вкладеному файлі Вправа 21.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106159).



§ 22. АДРЕСАЦІЯ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ

Адреси діапазонів клітинок, використані у формулах, називають посиланнями. Розгляньмо способи використання посилань.

Типи посилань

Клітинки, на які у формулах є посилання, називатимемо *адресними*. В адресах клітинок і діапазонів використовують заголовки стовпців (зазвичай це літери) і номери рядків (числа).

Посилання, яке містить тільки адреси клітинок, називають **відносним**. При копіюванні формули воно модифікується: номери рядків (заголовки стовпців) змінюються на різницю номерів рядків (стовпців) початкової й кінцевої клітинок із формулою.



Якщо формулу $=A4+B5$ *скопювати* до клітинки, що міститься на 2 стовпці праворуч і 3 рядки вище від початкової, то в новій формулі заголовки стовпців збільшаться на 2, а номери рядків зменшаться на 3, ми *отримаємо* $=C1+D2$ (рис. 22.1).

	A	B	C	D	E
1			17		
2				20	
3					
4	10		=A4+B5		
5		16			

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

Рис. 22.1

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

Щоб скопіювати формулу до суміжних клітинок, можна виділити клітинку з формулою та перетягнути маркер автозаповнення (рис. 22.2). Отримані формули будуть *модифікованими*.

Якщо у формулі є адреса, яка при копіюванні не повинна змінюватися, то використовують **абсолютне посилання**, утворене введенням знака «\$» перед заголовком стовпця і номером рядка. Якщо скопіювати до клітинки формулу $=\$A\1 , вона не змінить вигляду й посилатиметься на клітинку з адресою A1 (рис. 22.3).



Рис. 22.2

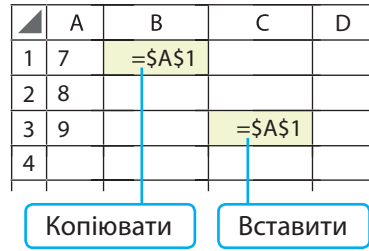


Рис. 22.3

Якщо в посиланні на клітинку абсолютним є лише заголовок стовпця або номер рядка (тобто знак «\$» міститься тільки перед заголовком стовпця або номером рядка), то таке посилання називають **мішаним**. Під час копіювання формул із мішаними посиланнями модифікуються тільки відносні частини посилань, а абсолютні залишаються незмінними.

Щоб швидко дібрати тип посилання, у формулі потрібно встановити на посиланні курсор; натискати клавішу F4 — при кожному натисканні тип змінюватиметься автоматично.

! При *копіюванні* даних з адресних клітинок до інших клітинок посилання на них у формулах не модифікуються, а при *переміщенні* — модифікуються залежно від нового розташування.

2 В електронній таблиці при переміщенні даних з адресної клітинки A1 до клітинки C3 всі посилання A1; $\$A\1 ; A\$1; $\$A1$ модифікуються відповідно таким чином: C3; $\$C\3 ; C\$3; $\$C3$ (рис. 22.4).

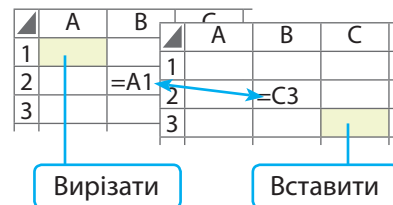


Рис. 22.4

Усі посилання також модифікуються, якщо адресна клітинка переміщується внаслідок вставлення або видалення стовпців, рядків, клітинок або діапазонів вище або ліворуч від неї.

! Видалення адресних клітинок призводить до помилок у формулах, які містять посилання на них (рис. 22.5).

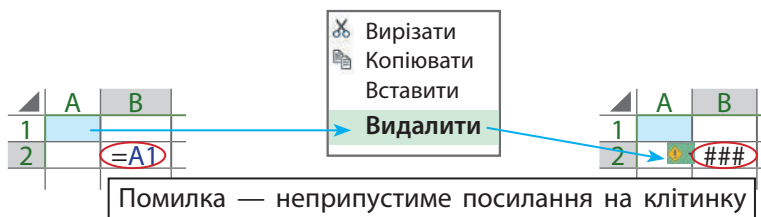


Рис. 22.5

Іменування клітинок і діапазонів

Окремим клітинкам і діапазнам можна надавати назви (іноді кажуть «імена»). Це спрощує запис складних формул, допомагає зрозуміти формули, записані раніше. Щоб надати назву виділеній клітинці або діапазону, її вводять у поле Ім'я (див. рис. 20.3).

Назви використовують у формулах, як і звичайні посилання.

3 У клітинці B1 збережено значення курсу євро відносно гривні. Надамо їй назву КУРС (рис. 22.6).

Тепер ми замість формули $=B3/B1$ у клітинку B4 можемо ввести $=B3/КУРС$.

КУРС		X	✓	<i>fx</i>
	A	B		
1	Курс євро:	42,16		
2				
3	Сума, грн:	4 356,00 €		
4	Сума, євро:	€ 340,51		

Рис. 22.6



Питання для самоперевірки

1. Яке посилання називають відносним?
2. Як записують посилання: абсолютні, мішані?
3. Після видалення клітинки в іншій клітинці з'явилося повідомлення про помилку. Поясніть причину.
4. Поясніть відмінності між посиланнями \$E\$8, E\$8, \$E8.

Вправа 22

- Розрахувати масу продуктів для приготування борщу в їдальні як добуток норми закладки на кількість порцій.
- 1. Запустіть табличний процесор, заповніть таблицю за зразком, наведеним на рис. 22.6.

	A	B	C	D	E	F
1		За нормою, г	85	172	155	57
2	Класи	Порцій	М'ясо	Картопля	Капуста	Буряк
3	7	117				
4	8	109				
5	9	107				
6	Всього	333				

Рис. 22.6

2. Уведіть у клітинку C3 формулу для розрахунку маси м'яса для приготування борщу для 7-х класів. Установіть числовий формат даних з одним десятковим знаком після коми.
3. Уведіть у клітинки D3, E3, F3 формули для розрахунку маси овочів для 7-х класів. Перетягуванням маркера автозаповнення скопіюйте відповідні формули до діапазонів C4:C5, D4:D5, E4:E5, F4:F5.
4. У клітинку B6 уведіть формулу для розрахунку загальної кількості порцій борщу для всіх класів.
- ⚙ 5. Перетягуванням маркера автозаповнення скопіюйте формулу з клітинки B6 до діапазону клітинок C6:F6. У клітинку G2 введіть текст «Маса продуктів, кг», у клітинку G3 — формулу для розрахунку загальної маси продуктів для 7-х класів.
6. Розрахуйте загальну масу продуктів у клітинках G4:G5 для решти класів, у клітинці G6 — для всіх класів. Збережіть файл із назвою Вправа 22.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106160).



§ 23. АВТОЗАПОВНЕННЯ ТА АВТОЗАВЕРШЕННЯ

Електронні таблиці надають зручні засоби автоматизації введення даних: пропонування варіантів завершення, генерування числових послідовностей та ін.

Автозаповнення

Автозаповнення — це функція, що дозволяє заповнювати суміжні клітинки однотипними даними за списками або певними правилами.

Завдяки автозаповненню можна швидко вводити, наприклад, повторювані дані, переліки днів тижня, місяців року тощо.

▶ Автозаповнення текстовими списками

Автозаповнення виконують за допомогою **маркера автозаповнення** — позначки в правому нижньому куті виділеного діапазону (рис. 23.1).

Коли в клітинку ввести початкове значення та протягти маркер уздовж стовпця (рядка), то суміжні клітинки заповняться або даними зі списку автозаповнення — якщо такий є (рис. 23.2), або текстом, скопійованим з активної клітинки, — якщо додатного списку немає (рис. 23.3).



Рис. 23.1

	A	B
1	пн	
2	вт	
3	ср	
4	чт	
5	пт	
6		

Рис. 23.2

	A	B
1	Меркурій	
2	Меркурій	
3	Меркурій	
4	Меркурій	
5	Меркурій	
6		

Рис. 23.3

Автозаповнення текстовими даними клітинок таблиці може відбуватися з певним кроком (тобто через одне, два значення і т. д.), який визначається автоматично.



На рис. 23.4 наведено процес автозаповнення скороченими назвами днів тижня через день, починаючи з вівторка.



Рис. 23.4

Якщо протягнути маркер правою кнопкою миші, то з'являється контекстне меню, в якому можна вибрати параметри автозаповнення.

► Автозаповнення обчислюваними значеннями

Числові дані та текстові дані з цілочисельними значеннями на початку (чи в кінці) рядка називатимемо *обчислюваними*.

На результат автозаповнення обчислюваними даними впливає початкова кількість виділених клітинок і клавіша Ctrl.

Якщо виділено одну клітинку, то перетягуванням отримаємо копії її даних, а утримуючи клавішу Ctrl — послідовність із кроком 1 (рис. 23.5, а). Якщо ж виділено дві клітинки, то перетягуванням побудуємо послідовність із кроком, що дорівнює різниці значень початкових клітинок, а з клавішею Ctrl — копії (рис. 23.5, б)



Рис. 23.5

Створення списку автозаповнення

Для створення списку автозаповнення потрібно:

- 1) вибрати команду Файл → Параметри → Додатково → Редагувати користувачькі списки (рис. 23.6);

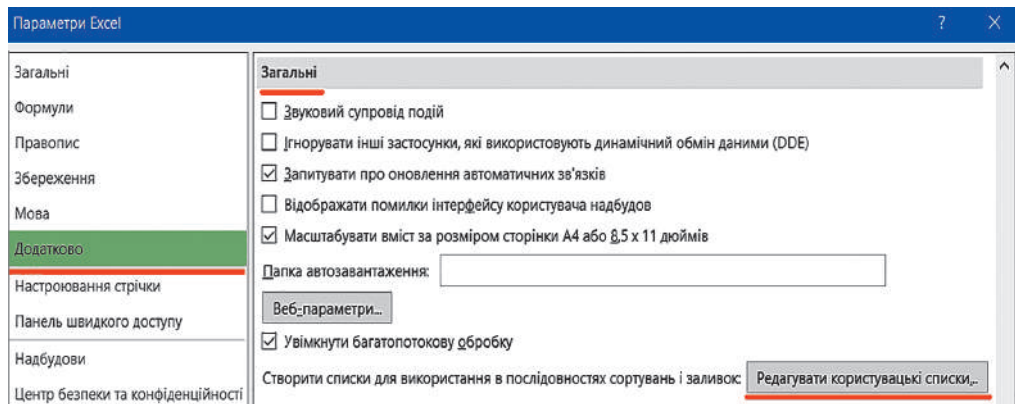


Рис. 23.6

- 2) у вікні Списки в полі Елементи списку ввести потрібні дані (у стовпець або рядок через кому);
- 3) натиснути кнопку Додати та/або ОК (рис. 23.7).

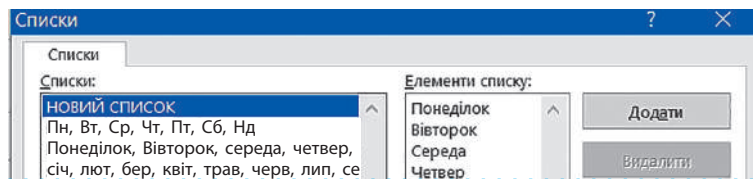


Рис. 23.7

Автозавершення

Під час набирання тексту автоматично пропонується завершення рядків із числа наявних у стовпці таблиці.

- **Автозавершення** — це функція, що забезпечує автоматичне введення повторюваних текстових даних.



На рис. 23.8 видно, що після введення в клітинку літер «карто», на сірому тлі з'явився варіант автозавершення «пля». Це відбулося тому, що в цьому ж стовпці є рядок, який починається саме з літер «карто». Натиснемо клавішу Enter або Tab — у клітинці з'явиться слово «картопля». Якщо ввести меншу кількість літер, варіанти завершення не пропонуватимуться.

карета
картопля
картуз
картопля

Рис. 23.8

Слід пам'ятати, що функція автозавершення розповсюджується на неперервну послідовність клітинок (аналізує лише клітинки, суміжні з даними) та не враховує регістр символів, що вводяться.

Автозавершення можна вимкнути. Для цього достатньо введення тексту почати з апострофа (').

Потрібне слово із суміжних у стовпці клітинок можна вибрати в контекстному меню клітинки командою Вибрати з розкритого списку (рис. 23.9).

	карета
	картуз
	картопля
	картопля
карета	
картопля	
картуз	

Рис. 23.9



Питання для самоперевірки

1. Що таке автозаповнення? Наведіть приклади.
2. Як створити новий список автозаповнення?
3. Як заповнити стовпець послідовністю чисел із певним кроком?
4. Як увести послідовність однакових чисел?
5. Що таке автозавершення?
6. Наведіть приклади автозавершення.



Вправа 23

- Ознайомитися з автозаповненням і автозавершенням під час введення даних до електронної таблиці.
1. Запустіть табличний процесор і створіть Нову книгу.

Заповніть клітинки стовпця А натуральними числами від 1 до 100 за зразком (рис. 23.10).

▲	A	B	C	D	E		Z	AA
1	1	200	півень	січень	Меркурій	Вене	урн	Уран
2	2	198	південь	лютий				
3	3	196	півкуля	березень				
4	4	194	півострів	квітень				
5	5	192	півень	травень				
6	6	190	південь	червень				
7	7	188	півострів	липень				
8	8	186	півкуля	серпень				
9	9	184	півень	вересень				
10	10	182	південь	жовтень				
11	11	180	півкуля	листопад				
12	12	178	півострів	грудень				
99	99	4		березень				
100	100	2		квітень				

Рис. 23.10

- Заповніть клітинки стовпця В парними числами від 200 до 2.
- Заповніть 12 клітинок стовпця С автозавершенням за зразком.
- Заповніть назвами місяців стовпець D. За потреби створіть список автозаповнення з назвами місяців і повторіть спробу.
- Створіть список автозаповнення з назвами планет Сонячної системи.
- Заповніть рядок 1 від стовпця E до AA назвами планет. Збережіть файл із назвою Вправа 23. Надішліть вчителю/учительці листа зі вкладеним файлом Вправа 23.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням gnk.com.ua/106161).





ПРАКТИЧНА РОБОТА 6

РОЗВ'ЯЗУВАННЯ ЗАДАЧ НА ОБЧИСЛЕННЯ

- Завдання:** розрахувати зарплату працівників за місяць, використовуючи формули й посилання.
- Обладнання:** комп'ютер із табличним процесором, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

- Запустіть табличний процесор. Створіть таблицю для розрахунку зарплати за зразком (див. рисунок).

	A	B	C	D	E
1		Робочих днів у місяці			курс €
2		22			
3	Прізвище	Відпрацьовано днів	Оклад, грн	Зарплата, грн	Зарплата, євро
4	Петренко	20	15 900,00 ₴		
5	Сидоренко	13	14 880,00 ₴		
6	Івановська	12	16 175,00 ₴		
7	Петрова	17	16 266,00 ₴		

- Знайдіть в інтернеті курси валют, уведіть дані в клітинку E2.
- У клітинці D4 за формулою розрахуйте зарплату Петренка в гривнях.

Приклад розрахунку: Оклад — це заробіток працівника за всі робочі дні в поточному місяці, зазначені в клітинці B2. Петренко заробляє на день $15900/22 \approx 722,73$ (грн). Оскільки він відпрацював 20 днів, то його зарплата становитиме $722,73 \cdot 20 \approx 14454,60$ (грн).

- Скопіюйте формулу для розрахунку зарплати Петренка в гривнях для решти працівників.
- Уведіть у клітинку E4 формулу для розрахунку зарплати Петренка в євро.
- Скопіюйте отриману формулу у відповідні клітинки для розрахунку зарплати кожного працівника в євро.


7. Уведіть у клітинку C8 формулу для розрахунку середнього окладу всіх працівників.
8. У клітинках D8, E8 розрахуйте суми зарплати всіх працівників у гривнях і євро.
9. Установіть у клітинках із грошовими сумами відповідний грошовий формат даних із двома десятковими знаками після коми з позначенням грошової одиниці (грн, євро).
10. Збережіть файл із назвою Практична робота 6 у відповідній папці. Завершіть роботу.

Зробіть висновок: від чого залежить правильність обчислень за скопійованими формулами; коли посилання у формулах повинні бути абсолютними і чому?


§ 24. ФУНКЦІЇ

Засобами табличного процесора можна виконувати складніші обчислення — для цього призначені вбудовані функції. Усі функції поділяють на математичні, статистичні, логічні тощо.

Основні поняття

 **Функція** в ЕТ — дії над даними, результатом виконання яких є певне значення. Кожна функція має назву. Дані, які опрацьовує функція, називають **аргументами**.

Аргументами функції можуть бути інші функції, вирази, посилання тощо. Аргументи функції записують у дужках після назви функції.

 Будь-яка функція в ЕТ характеризується не тільки унікальною назвою, а й кількістю аргументів, типами аргументів та результату.



Пригадаємо відому із 6 класу функцію AVERAGE (рис. 24.1):

- назва — AVERAGE;
- кількість аргументів — не менше одного;
- тип аргументів — числовий (текстові значення в адресних клітинках нехтуються);
- результат — середнє арифметичне чисел зі списку аргументів.

B	C	D	E	F	G
Оцінки					Підсумкова
12	11		9	n	=AVERAGE(B2:F2)

Рис. 24.1



Якщо функція має кілька аргументів, їх розділяють символом «;».

Функції у формули вставляють інструментами вкладки Формули або у вікні Вставлення функції, яке відкривається при натисканні кнопки fx, що в рядку формул. Потрібну функцію можна знайти за описом дії, категорією або назвою (рис. 24.2).

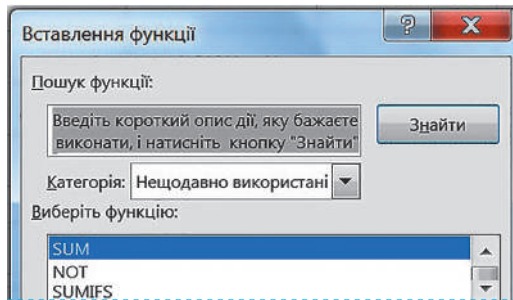


Рис. 24.2



Математичні функції

Математичні функції в ЕТ — функції, призначені для обчислень над числами та/або даними клітинок: округлення, перетворення чисел тощо. Деякі з них наведено в таблиці.

Назва	Опис дії	Кількість аргументів
ROUND	Округлення до вказаної кількості цифр після коми	Два
MOD	Остача від ділення	Два
PRODUCT	Добуток	Не менше одного
SUM	Сума	Не менше одного
SUMIF	Сума за умови, що...	Два або три

2 Знайдемо добуток чисел у діапазоні клітинок A1:B2 й округлимо його до десятих. Маємо: =ROUND(PRODUCT(A1:B2);1).

Статистичні функції

Статистичні функції в ЕТ — функції, призначені для аналізування значень діапазонів. За їх допомогою можна знайти найбільше і найменше значення, розрахувати середнє значення тощо.

3 За допомогою функції COUNTIF визначимо, скільки осіб навчаються на 12 балів. Перший аргумент — діапазон клітинок з оцінками, другий — критерій вибору:

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

Знак рівності й лапки в критерії можна опустити:

=COUNTIF(B3:B4;12)

Критерієм вибору може бути число, дата, символ тощо. За ним будується умова: чи містить поточна клітинка зазначені дані. Так, можна обчислити, скільки пропущено уроків (рис. 24.3).

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

Критерій вибору

Рис. 24.3

Розгляньмо ще деякі статистичні функції, наведені в таблиці.

Назва	Опис дії	Кількість аргументів
MIN	Найменше значення	Не менше одного
MAX	Найбільше значення	Не менше одного
AVERAGE	Середнє арифметичне	Не менше одного
COUNT	Кількість числових значень	Не менше одного
COUNTIFS	Кількість значень, які задовольняють декілька умов	Не менше двох
COUNTA	Кількість непорожніх клітинок	Не менше одного



Питання для самоперевірки

1. Що таке функція в електронній таблиці?
2. Що таке аргумент функції?
3. Назвіть кілька категорій функцій.
4. Наведіть приклади математичних функцій; статистичних.



Вправа 24

- Проаналізувати в електронній таблиці успішність учнів і учениць вашої групи (класу) з інформатики за описом.
- 1. Запустіть табличний процесор. Створіть таблицю зі стовпцями, що мають такі заголовки: А — «Прізвище»; В, С, D — «Поточні оцінки»; Е — «Підсумкова».
- 2. Уведіть у таблицю прізвища та поточні оцінки.
- 3. У стовпці Е за допомогою відповідних функцій обчисліть підсумкові оцінки як середнє арифметичне поточних оцінок, округлене до цілого значення.
- ⚙ 4. У довільній клітинці під прізвищами за допомогою функції COUNTIFS розрахуйте, скільки осіб мають підсумкові оцінки від 5 до 8 балів.
- 5. У довільній клітинці під прізвищами розрахуйте загальну кількість поточних оцінок усіх учнів і учениць.
- 6. Збережіть файл із назвою Вправа 24.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106162).



§ 25. ЛОГІЧНІ ФУНКЦІЇ

Складаючи програми мовою Python, ви використовували висловлення, які можуть бути істинними або хибними.

Логічні функції

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

Простий логічний вираз складається з двох виразів, пов'язаних операцією відношення:

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



2 > 1 — істина;
2 < 1 — хибність.

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

► Функція IF

Логічна функція IF (з англ. — якщо) призначена для обчислення одного з двох виразів залежно від умови:

IF (Умова; Вираз1; Вираз2).

Вираз1 — обчислюється, якщо умова істинна.

Вираз2 — обчислюється, якщо умова хибна.

2

У відомості навчальних досягнень біля оцінки потрібно вивести слово «Добре», якщо учень / учениця має оцінку, вищу від 6 балів, і слово «Задовільно», якщо дитина має іншу оцінку. Фрагмент алгоритму розв'язування цієї задачі можна подати у вигляді блок-схеми (рис. 25.1).

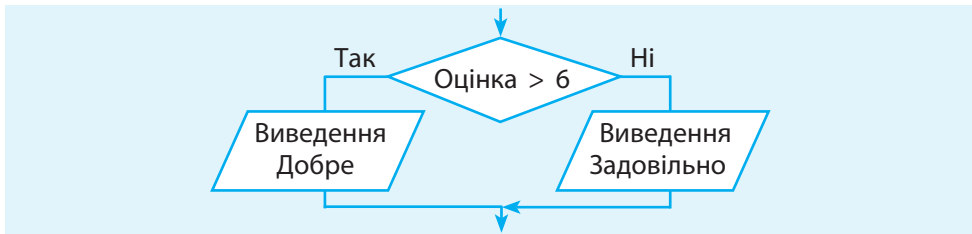


Рис. 25.1

В ЕТ цей алгоритм реалізується за допомогою функції IF (рис. 25.2):

IF (B1>6; "Добре"; "Задовільно")

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

Рис. 25.2

Логічний вираз може містити посилання на клітинки та/або арифметичні вирази. У цьому випадку, якщо значення виразу дорівнює нулю, то повертається логічне значення False, а якщо відмінне від нуля, то True.

3

2+2 трактується як істина (оскільки $2+2 \neq 0$);
2-2 трактується як хибність (оскільки $2-2 = 0$).

► Функції NOT, AND, OR

Іноді для розв'язування задачі комбінують кілька умов.

Складеним логічним виразом називають вираз, у якому кілька простих логічних виразів пов'язані логічними операціями.

Основними логічними операціями є НЕ (заперечення), І (кон'юнкція або логічне множення), АБО (диз'юнкція, логічне додавання). В ЕТ їх реалізовано за допомогою функцій NOT, AND, OR (з англ. — не, і, або відповідно).

4 Повернемося до рис. 25.2. Біля оцінки виведемо «Посередньо» для тих, хто має оцінки, вищі від 6 балів і нижчі за 10. Для інших — нічого не вводимо. Скористаємося функціями IF і AND: `=IF(AND(B1>6;B1<10); "Посередньо"; "")`

Тут останнім аргументом функції IF є лапки без символів між ними, тобто порожній рядок.

Іноді буває доцільно використати вкладені логічні функції IF; допускається вкладати одна в одну кілька функцій.

5 Нехай потрібно схарактеризувати рівень навчальних досягнень учнів і учениць за трьома умовними рівнями: «Початковий» (1–4 бали); «Середній» (5–8 балів); «Високий» (9–12 балів). В ЕТ цю задачу можна розв'язати так (рис. 25.3): `=IF(B1<5; "Початковий"; IF(B1<9; "Середній"; "Високий"))`

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

Рис. 25.3



Питання для самоперевірки

1. Чим логічні функції відрізняються від інших?
2. Як позначають операції відношень?
3. Наведіть приклад складеного логічного виразу.
4. Чи існують в електронних таблицях функції без аргументів?
5. Що спільного й відмінного у функцій AND і OR?
6. Який вираз називають складеним логічним виразом?

Вправа 25

- Створити електронну таблицю за описом.
- 1. Запустіть табличний процесор. Для вашої групи (класу) створіть і заповніть таблицю навчальних досягнень з інформатики: уведіть у стовпці А прізвища, у В — оцінки за семестр.
- 2. За допомогою функції IF створіть формулу для запису в стовпці С рівнів: «Початковий» (менше за 5 балів) і «Високий» (для решти оцінок) за зразком на рис. 25.2.
- ⚙️ 3. Створіть формулу для запису в стовпці D таких рівнів: «Юніор» (1–4 бали), «Аматор» (5–8 балів), «Експерт» (9–12 балів).
- 4. Збережіть файл із назвою Вправа 25.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106163).



§ 26. УМОВНЕ ФОРМАТУВАННЯ

💡 **Умовне форматування** — автоматична зміна формату клітинки залежно від певної умови.

Умовне форматування надає можливість оформити клітинки таблиці за стандартними правилами та стилями або використати свої (команда Інші правила).

Щоб виділений діапазон отримав умовний формат, треба вибрати команду Основне → Стилі → Умовне форматування, а потім у діалоговому вікні задати потрібне.

До клітинки можна застосувати кілька правил умовного форматування. Як умову для нього можна використати логічний вираз. Цей вираз може посилатися на інші клітинки таблиці.



Побудуємо за допомогою умовного форматування зображення шахівниці (рис. 26.1). Для побудови формул використаємо той факт, що у світлих клітинок сума номерів рядка і стовпця парна, у темних — непарна. Отже:

- 1) пронумеруємо рядки та стовпці, увівши в діапазони B1:I1 і A2:A9 послідовні натуральні числа, починаючи з 1;
- 2) для клітинки B2 задамо правила умовного форматування:
 - формула $\text{MOD}(B\$1+\$A2;2) = 0$ — жовтий колір тла;
 - формула $\text{MOD}(B\$1+\$A2;2) = 1$ — коричневий колір тла;
- 3) після закриття вікна налаштувань умовного форматування скопіюємо формат клітинки B2, застосуємо його до решти клітинок (див. рис. 26.1).

	A	B	C	D	E	F	G	H	I
1		1	2	3	4	5	6	7	8
2	1								
3	2								
4	3								
5	4								
6	5								
7	6								
8	7								
9	8								

Рис. 26.1



Якщо правила умовного форматування суперечать одне одному, то «чинним» буде правило, встановлене пізніше.



Установимо для певного діапазону червону заливку клітинок зі значеннями, більшими за 5, і зелену заливку клітинок зі значеннями, меншими за 7.

У підсумку клітинки зі значенням 6 зафарбуються зеленим кольором (рис. 26.2), оскільки відповідне правило встановлено останнім.

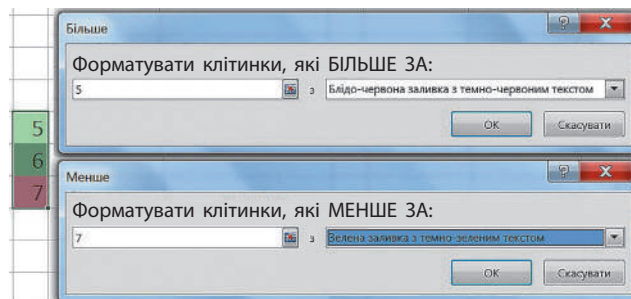


Рис. 26.2

**Питання для самоперевірки**

1. Що таке умовне форматування?
2. Як призначити клітинці умовний формат?
3. Поясніть формули, використані в прикладі 1 (с. 156).
4. Поясніть, що відбувається, коли правила умовного форматування суперечать одне одному.

**Вправа 26**

- Створити таблицю для унаочнення результатів участі школи в предметних олімпіадах за декілька років.
1. Запустіть табличний процесор і створіть таблицю (рис. 26.3).

	A	B	C	D	E
1	Досягнення (місце)				
2	Рік	2020	2021	2022	2023
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. 26.3

2. Налаштуйте умовне форматування клітинок діапазону B3:E7: 1-ше місце — червона заливка; 2-ге місце — зелена; 3-тє — жовта заливка (колір шрифту — на власний вибір).
3. Налаштуйте умовне форматування клітинок для непрیزових місць: шрифт — напівжирний; колір шрифту — синій, без заливки клітинок. Збережіть файл із назвою Досягнення.

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106164).





ПРАКТИЧНА РОБОТА 7

ВИКОРИСТАННЯ ФУНКЦІЙ ТАБЛИЧНОГО ПРОЦЕСОРА. УМОВНЕ ФОРМАТУВАННЯ

- Завдання:** створити таблицю з автоматичним обчисленням підсумкових оцінок та рівня навчальних досягнень.
- Обладнання:** комп'ютер зі встановленим табличним процесором.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть табличний процесор.
Установіть параметри сторінки: усі поля — по 1 см; колон-титули — 0 см; орієнтація — книжкова; розмір паперу — А4.
2. Створіть таблицю успішності (модель сторінки шкільного журналу) за зразком (див. рисунок, с. 159).
Об'єднайте, де потрібно, клітинки, змініть розміри стовпців та рядків. Установіть для таблиці межі середньої товщини.
3. Установіть орієнтацію тексту в клітинках відповідно до зразка, а також такі формати даних:
 - для оцінок — числовий без десяткових знаків;
 - для написів — текстовий;
 - для дат — короткий формат дати.
4. Упишіть прізвища, оцінки й інші написи.
5. Відсортуйте прізвища учнів і учениць за алфавітом.
6. За допомогою автозаповнення введіть нумерацію й дати.
7. Обчисліть відповідною функцією підсумкову оцінку кожного учня як округлене середнє арифметичне його оцінок.
8. Функцією COUNTIF розрахуйте кількість відсутніх у кожний із зазначених днів.
9. Функцією COUNTIF за підсумковими оцінками розрахуйте кількість осіб, оцінки яких відповідають початковому (менше за 4 бали) і високому (більше від 9 балів) рівням.

10. Функцією COUNTIFS за підсумковими оцінками учнів розрахуйте кількість тих, хто має достатній (від 7 до 9 балів) та середній (від 4 до 6 балів) рівні.
11. Налаштуйте умовне форматування клітинок з оцінками залежно від рівня досягнень:
 - високий — червоний колір тексту;
 - середній — зелений колір тексту;
 - достатній — синій колір тексту;
 - початковий — чорний (без заливки клітинок кольором).
12. Збережіть файл із назвою Практична робота 7.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	№ за порядком	Прізвище	Дата										Підсумкова
2			09.02.2023	16.02.2023	23.02.2023	01.03.2023	08.03.2023	15.03.2023	22.03.2023	29.03.2023	05.04.2023	12.04.2023	
3	1	Іванчук	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												

Зробіть висновок: які функції використовують для розрахунків; яка з них має найбільше аргументів.

§ 27. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ В ЕЛЕКТРОННИХ ТАБЛИЦЯХ

Як ви вже знаєте, для дослідження різних об'єктів і явищ використовують моделювання. При цьому сам об'єкт чи явище замінюють його моделлю, із якою й проводять дослідження.

Електронні таблиці — потужний засіб для обчислень, тому розгляньмо, як їх застосувати для експериментів із математичними моделями. Але спочатку пригадаймо основні поняття.

Основні поняття моделювання

Модель (від латин. *modulus* — міра, зразок) — це спрощене подання об'єкта, процесу або явища. За способом подання моделі поділяють на матеріальні й інформаційні.

Інформаційна модель — опис властивостей об'єкта, процесу чи явища, суттєвих для дослідження. З інформаційними моделями різних типів (графічних, алгоритмічних, математичних, структурних тощо) зручно працювати за допомогою комп'ютера.

Математична модель — математичні співвідношення між числовими величинами, які описують об'єкт, процес або явище. Для її побудови слід з'ясувати зв'язок між вхідними даними та очікуваними результатами, який приводить до розв'язання задачі. Цей зв'язок подають у вигляді рівнянь і нерівностей.

Пригадаймо, в якій формі зручно описувати цю модель:

Що дано? <Перелік початкових даних>

Що треба знайти? <Перелік потрібних результатів>

Який зв'язок між величинами? <Математичні співвідношення, що зв'язують вхідні дані та результати>

Які є обмеження на дані? <Умови допустимості початкових значень>

Швидко виконання обчислень робить комп'ютер зручним інструментом для різноманітних експериментів із моделями.

Моделювання руху

На уроках математики та фізики вам зустрічалися задачі, в яких описано рух різних об'єктів: людей, човнів, космічних кораблів, пташок тощо. Для їх розв'язування ви застосовували формули, за якими обчислювали невідомі значення величин.

Величини, що характеризують рух, можна позначити так: t — час руху; l — пройдений шлях; v — швидкість руху. Якщо об'єкт рухається рівномірно, то його рух описують такі формули (саме їх ми використаємо для моделювання руху в таблиці):

$$v = \frac{l}{t}; \quad t = \frac{l}{v}; \quad l = vt.$$



Змоделюємо рух двох автомобілів, які одночасно виїхали з двох міст назустріч один одному і рухаються рівномірно. *Мета моделювання:* визначити відстань між автомобілями в різні моменти часу до їх зустрічі.

Математична модель:

<i>Що дано?</i>	L — відстань між містами;
	v_1 і v_2 — швидкості автомобілів
<i>Що треба знайти?</i>	d — відстань між автомобілями
<i>Який зв'язок між величинами?</i>	$d = L - l_1 - l_2$, де $l_1 = v_1 t$; $l_2 = v_2 t$; t — час руху
<i>Які є обмеження на дані?</i>	$L > 0$; $v_1 > 0$; $v_2 > 0$; $l_1 + l_2 \leq L$

На основі нашої моделі створимо таблицю (рис. 27.1, с. 162):

- 1) у діапазон D1:D3 введемо вхідні дані;
- 2) у клітинку D4 введемо величину проміжку часу, через який потрібно визначити відстань між автомобілями;
- 3) у клітинку A7 введемо 0; у A8 — формулу =A7+\$D\$4, яку скопіюємо в наступні клітинки цього самого стовпця;
- 4) формули для розрахунку шляхів, які автомобілі пройдуть за час, зазначений у клітинці A7, введемо в клітинки B7 (=D\$2*A7/60) і C7 (=D\$3*A7/60). Тут ділення на 60 потрібне, щоб перевести час із хвилин у години;

- 5) у клітинку D7 уведемо формулу для розрахунку відстані L або повідомлення, що автомобілі розминулися
(=IF(B7+C7<\$D\$1;\$D\$1-B7-C7;"розминулися");
- 6) скопіюємо формули діапазону B7:D7 у наступні клітинки цих самих стовпців.

	A	B	C	D
1	Відстань між містами, км			150
2	Швидкість автомобіля 1, км/год			80
3	Швидкість автомобіля 2, км/год			75
4	Крок часу, хв			6
5				
6	Час t, хв	I_1 , км	I_2 , км	L, км
7	0	0	0	150
8	6	8	7,5	134,5
9	12	16	15	119
10	16	24	22,5	103,5
11	24	32	30	88
12	30	40	27,5	72,5
13	36	48	45	57
14	42	56	52,5	41,5
15	48	64	60	26
16	54	72	67,5	10,5
17	60	80	75	розминулися
18	66	88	82,5	розминулися



Рис. 27.1

Тепер комп'ютерна модель готова для використання. Змінюючи вхідні дані можна отримувати відповідні таблиці відстаней між автомобілями. Якщо зменшити крок часу, то таблиця стане докладнішою. Рядок, де вперше з'явилося слово «розминулися», дозволяє приблизно знайти місце зустрічі автомобілів.

! Зменшення кроку часу підвищує точність результату.

У разі, коли рядків із формулами у стовпцях A:D не вистачить для простеження руху до зустрічі, достатньо скопіювати формули з останнього рядка в потрібну кількість наступних рядків.



Питання для самоперевірки

1. Що таке модель; моделювання?
2. Поясніть різницю між матеріальною й інформаційною моделями. Наведіть приклад.
3. Що таке математична модель?
4. Які величини використовують, моделюючи рух?
5. Поясніть формули математичної моделі, описаної в матеріалах параграфа.
6. Поясніть формули, використані в таблиці на рис. 27.1.



Вправа 27

- Реалізувати в електронній таблиці математичну модель руху.
1. Створіть таблицю за описом, який наведено у прикладі на с. 161. Збережіть файл із назвою Вправа 27.
 2. Перевірте правильність моделі, увівши в діапазон D1:D4 дані, як наведено на рис. 27.1. Чи відповідають отримані результати наведеним?
 3. Запустіть табличний процесор і підготуйте таблицю для результатів експерименту:

Дані	Номер досліду					
	1	2	3	4	5	6
L , км	150	150	150	210		
v_1 , км/год	80	80	80	60		
v_2 , км/год	75	75	75	40		
Крок часу, хв	6	3	1	1		
Час зустрічі, хв	60					
l_1 , км	80					

4. Дослідіть рух автомобілів за однакових відстані між містами та початкових швидкостей, але різних кроків часу (досліди 1–3). Поясніть отриманий результат.

5. Виконайте дослід 4.
6. Доберіть для дослідів 5 і 6 самостійно дані так, щоб час зустрічі та l_1 виявилися однаковими за різних значень L і v_1 . Збережіть таблицю результатів у файл із назвою Звіт 27.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106165).



ПРАКТИЧНА РОБОТА 8 ПРОВЕДЕННЯ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

Завдання: змоделювати в електронній таблиці гру Джона Конвея, що отримала назву «Життя».

Обладнання: комп'ютер зі встановленим табличним процесором.

► Теоретична частина

1970 року англійський математик Джон Конвей вигадав ігровий світ. Він розташований на полі, поділеному на однакові клітинки, кожна з яких може бути «живою» або «мертвою».

У грі людина створює початковий стан, а далі лише спостерігає за його розвитком. «Живі» клітинки, що є на полі спочатку, називають першим поколінням. Подальша доля клітинки залежить від стану її сусідів: якщо «жива» клітинка має двох або трьох «живих» сусідів, то залишається «живою», а інакше — «гине»; «мертва» клітинка, що має трьох сусідів, «оживає».

Зміна покоління відбувається одночасно в усіх клітинках поля. Гра ведеться, поки на полі є «живі» клітини або поки система не зациклюється в одному зі своїх станів.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Підготуйте модель поля: зменште ширину стовпців A:AN так, щоб клітинки були квадратними (рис. 1).

	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	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN				
1																																												
2																	0	0	0	0	0	0	0	0	0	0	0																	
3																0	0	0	0	0	0	0	0	0	0	0																		
4																0	0	0	0	1	1	1	0	0	0	0																		
5							1									0	0	0	0	2	2	3	1	0	0	0								1	1									
6						1	1									0	0	0	0	2	2	3	2	1	0	0								1	1									
7								1								0	0	0	0	1	2	4	3	3	1	0																		
8									1	1						0	0	0	0	0	2	2	2	2	1	0																		
9																0	0	0	0	0	1	2	2	1	0																			
10																0	0	0	0	0	0	0	0	0	0																			
11																0	0	0	0	0	0	0	0	0	0																			
12																0	0	0	0	0	0	0	0	0	0																			
13																0	0	0	0	0	0	0	0	0	0																			
14																0	0	0	0	0	0	0	0	0	0																			

Рис. 1

2. Позначте зовнішніми межами діапазони:
 - B2:M13 — ігрове поле;
 - O2:Z13 — для підрахунку кількості сусідів;
 - AB2:AM13 — для формування наступного покоління.
3. Налаштуйте умовне форматування клітинок діапазону B2:M13, щоб значення 1 приводило до зміни кольору тла.
4. У клітинку O2 введіть формулу для підрахунку числа «живих» сусідів у відповідній клітинки діапазону B2:M13:
 $=SUM(A1:C3)-B2$
5. Скопіюйте формулу в інші клітинки діапазону O2:Z13.
6. У клітинку AB2 введіть формулу для розрахунку вмісту відповідної клітинки ігрового поля в наступному поколінні:
 $=IF(OR(AND(B2=0;O2=3);AND(B2=1;OR(O2=2;O2=3)));1;"")$
7. Скопіюйте формулу в інші клітинки діапазону AB2:AM13. Збережіть таблицю у файл із назвою Life.

8. Уведіть одиниці в окремі клітинки діапазону B2:M13, перевірте, як змінюються значення в діапазонах O2:Z13 і AB2:AM13.
9. Скопіюйте діапазон AB2:AM13 (друге покоління).
Клацніть клітинку B2 і вставте з буфера обміну Лише значення (рис. 2).
10. Повторіть копіювання та вставлення декілька разів, спостерігаючи за розвитком «колонії» клітинок.
11. Дослідіть поведінку початкових розташувань «живих» клітинок, наведених на рис. 3.

Опишіть отриманий результат.

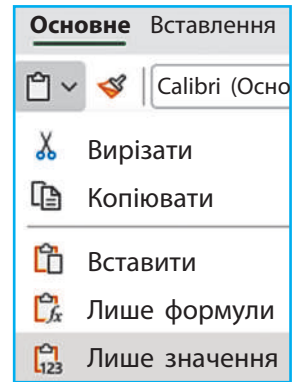


Рис. 2

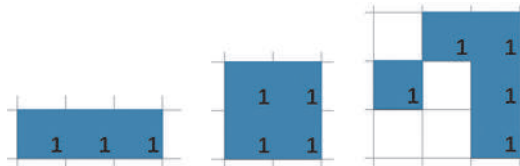


Рис. 3

12. Зобразіть «живими» клітинками першу літеру вашого імені. Дослідіть, як розвиватиметься «колонія». Збережіть 10 знімків екрана із зображеннями послідовних поколінь.

Зробіть висновок: які бувають варіанти розвитку «колонії» у грі «Життя»; завдяки чому електронна таблиця є зручним засобом для дослідження математичних моделей.

РОЗДІЛ 5

АЛГОРИТМИ ТА ПРОГРАМИ



§ 28. Етапи розв'язування задачі на комп'ютері

§ 29. Величини. Алгоритми роботи з величинами

§ 30. Опрацювання даних числових типів

§ 31. Алгоритми і програми з розгалуженнями

§ 32. Алгоритми і програми з повтореннями. Цикли

§ 33. Рядковий тип даних

§ 34. Опрацювання рядкових величин

Практична робота 9. Створення програм для опрацювання рядкових величин

§ 35. Списки — змінювані послідовності

§ 36. Функції та методи опрацювання списків

§ 37. Опрацювання елементів списку

§ 38. Створення інтерфейсу користувача

Практична робота 10. Створення програм для опрацювання списків

§ 39. Розв'язування задачі методом поділу на підзадачі

§ 40. Обчислення з використанням функцій користувача

Практична робота 11. Створення програми з використанням функцій користувача

ПОВТОРЮЄМО



Ви познайомилися з прийомами роботи в програмному середовищі Python і правилами запису алгоритмів мовою Python. Ви навчилися вести діалог в інтерактивному режимі в оболонці IDLE і створювати та зберігати програмний код для подальшого використання у вікні програми.

Ви також знайомі з поняттям величини в програмуванні і маєте уявлення про *алгоритми роботи з величинами* числового та рядкового типів.

Ви вдосконалили знання та навички зі створення інформаційних моделей, дізналися про *математичне моделювання*, проаналізували кроки, які потрібно виконати для створення та реалізації математичної моделі задачі у програмному середовищі.

1. Які види вікон існують у IDLE?
2. Які повідомлення з'являються у вікні консолі?
3. Як відкрити вікно програми?
4. Як організувати діалог людини з програмою?
5. Які типи даних вам відомі?
6. Назвіть базові алгоритмічні структури.



У цьому розділі ви навчитеся працювати з такими конструкціями мови, як рядки та списки. Ви навчитеся описувати функції користувача і використовувати метод покрокової деталізації для розв'язування складних задач.

§ 28. ЕТАПИ РОЗВ'ЯЗУВАННЯ ЗАДАЧІ НА КОМП'ЮТЕРІ

За допомогою комп'ютера можна розв'язувати безліч різноманітних задач різної складності. Наприклад, підрахувати вартість спожитої електроенергії, перекласти текст іншою мовою, розрахувати орбіту супутника зв'язку тощо. Попри те, що задачі дуже різні, можна скласти загальний план, дотримання якого дасть змогу успішно їх розв'язати (рис. 28.1).

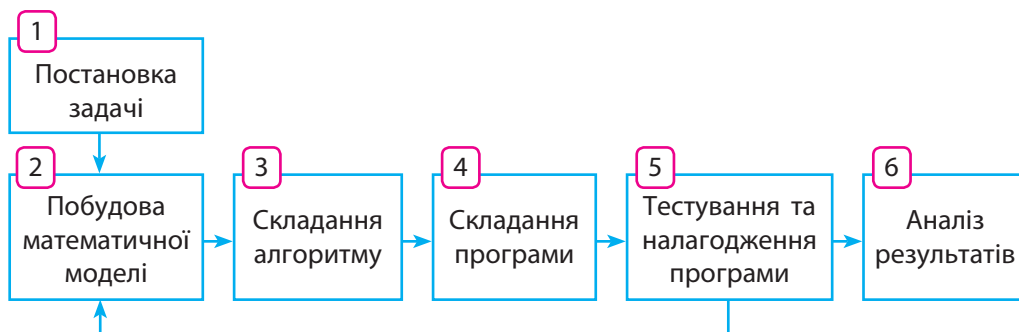


Рис. 28.1

Проаналізуймо етапи розв'язування задачі в середовищі програмування на прикладі задачі.

Задача. Діти вирішили озеленити шкільне подвір'я і засіяти насінням квітів клумбу, що має форму круга радіуса r м. Щоб засіяти 1 м^2 землі, потрібно 20 г насіння квітів. Кілограм насіння коштує K грн. Скільки коштів знадобиться, щоб засіяти клумбу насінням квітів?

► 1. Постановка задачі

На цьому етапі потрібно відкинути несуттєву для розв'язування задачі інформацію. Умову задачі тоді можна сформулювати так:

Задача. Кілограм насіння коштує K грн. На 1 м^2 землі потрібно 20 г насіння. Скільки коштує насіння, потрібне для того, щоб засіяти площу круга радіуса r м?

► 2. Побудова математичної моделі

На цьому етапі потрібно за умовою задачі побудувати математичну модель, яка включає всі суттєві дані.

Із побудовою математичної моделі ви вже ознайомилися. Отже, запишімо математичну модель нашої задачі:

<i>Дано:</i>	K — ціна 1 кг насіння (грн); r — радіус круга (м)
<i>Потрібно:</i>	C — загальна вартість насіння (грн)
<i>Зв'язок:</i>	$s = \pi r^2$ — площа клумби (м^2); $C = s \cdot 0,02 \cdot K$ (грн)
<i>Обмеження:</i>	$K \geq 0, 0 \leq r \leq 10$

► 3. Складання алгоритму

Розроблення і складання алгоритму — найважливіший етап розв'язування задачі. При складанні алгоритму необхідно врахувати всі його властивості.

! Від якості алгоритму залежать правильність результатів, ефективність використання часу та оперативної пам'яті комп'ютера.

Алгоритм розв'язування цієї задачі є лінійним, блок-схему алгоритму подано на рис. 28.2.

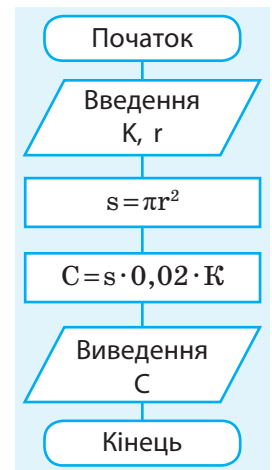


Рис. 28.2

► 4. Складання програми

Програмування (складання програми) — це кодування алгоритму мовою програмування. Реалізуємо алгоритм мовою Python. Завантажити файл для інсталяції Python можна з сайту python.org (рис. 28.3).

Отже, реалізуємо алгоритм: завантажимо Python → IDLE та виберемо команду File → New File (Файл → Новий файл).



Рис. 28.3

Згадаймо алгоритм роботи з програмою в середовищі IDLE Python (рис. 28.4).

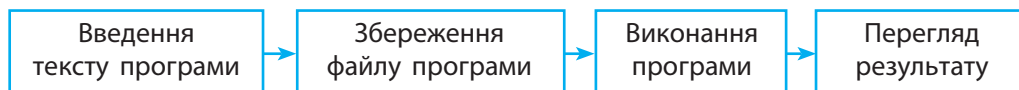


Рис. 28.4

```

У вікні програми введемо текст програми:
k = float(input('Скільки коштує 1 кг?'))
r = float(input('Чому дорівнює радіус клумби?'))
s = 3.14*r**2
c = s*0.02*k
print('c = ', c, 'грн')
  
```



► 5. Тестування та налагодження програми

На цьому етапі програму потрібно перевірити за допомогою *тестів* — наборів вхідних даних і відповідних їм результатів, і виправити виявлені помилки.

Тестування полягає в порівнянні результатів зі зразком. Часто розбіжність між ними спричиняють недоліки математичної моделі та помилки в алгоритмі.

Збережимо файл із розширенням `.py`, запустимо на виконання `Run` → `Run Module`. Випробуємо програму на такому наборі даних: для $K=150$ грн, $r=2,5$ м результатом є значення $C=58,88$ грн (рис. 28.5).

```

Скільки коштує 1 кг? 150
Чому дорівнює радіус клумби? 2.5
c = 58.875 грн
  
```

Рис. 28.5

Як бачимо, результати збігаються.

► 6. Аналіз результатів

На завершальному етапі програма виконується з даними, для яких і розпочато розв'язування задачі. Після остаточного виконання програми аналізуються отримані результати.



Питання для самоперевірки

1. Опишіть етапи розв'язування задачі на комп'ютері.
2. Дайте означення математичної моделі розв'язування задачі.
3. Опишіть алгоритм роботи з програмою в середовищі Python.
4. Поясніть мету тестування програми.
5. Побудуйте математичну модель задачі.
 - а) Уявіть, що ви вирішили протягом T годин кататися на колесі огляду. Яку відстань ви «проїдете», якщо діаметр колеса огляду D м, а один оберт воно виконує за 20 хв?
 - б) На аркуші паперу накреслені окремо квадрат та коло. Сторона квадрата A см, радіус кола — B см. Знайдіть загальну площу фігур.



Вправа 28

- Розв'язати задачу на комп'ютері.

Задача. Прямокутник, довжини сторін a і b якого задовольняють умову $\frac{a}{b} = \frac{b}{a-b}$, називають «золотим». Визначте, чи є прямокутник заданих розмірів (рис. 28.6) «золотим»?

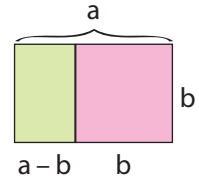


Рис. 28.6

1. Проаналізуйте постановку задачі та складіть математичну модель.

Що дано? a, b — довжини сторін.

Що треба знайти? Відповідь Так або Ні.

Який зв'язок між величинами? Якщо $\frac{a}{b} = \frac{b}{a-b}$, то вивести Так, інакше — Ні.

Які є обмеження на дані? $0 < b < a$.

2. Накресліть блок-схему алгоритму розв'язування поданої задачі.
3. У Python IDLE виберіть команду File → New File.
4. Запишіть команди введення значень змінних a і b :


```
a = float(input('a = ? '))
b = float(input('b = ? '))
```

5. Запишіть команду розгалуження для перевірки відповідності значень змінних a , b умові задачі:

```
if abs(a/b-b/(a-b))<0.001: print('Так')
else: print('Ні')
```

Примітка. Через особливості подання в пам'яті дробових чисел порівняння двох виразів за допомогою операції « $==$ » не завжди дає очікуваний результат. Вважатимемо значення рівними, якщо вони відрізняються не більше ніж на деяку досить малу величину (точність), тут 0,001. Функція `abs` повертає модуль аргументу.

6. Збережіть файл із назвою `Vprava28`. Запустіть програму та проаналізуйте результат її виконання у вікні консолі. Випробуйте програму на таких тестах:
- $a = 8.09$, $b = 5$, відповідь "Так";
 - $a = 10$, $b = 5$, відповідь "Ні".



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106416).



§ 29. ВЕЛИЧИНИ. АЛГОРИТМИ РОБОТИ З ВЕЛИЧИНАМИ

Будь-яка комп'ютерна програма оперує певними даними. У програмуванні для подання значень даних, які опрацьовує програма, застосовують різноманітні величини.

Основні характеристики величин



Величина — окремий інформаційний об'єкт (число, рядок тощо).

Кожна величина характеризується *обсягом пам'яті*, виділеної для її зберігання, *назвою (ідентифікатором)*, *типом* і *значенням*. Ці характеристики важливі для правильного використання та розуміння величин у програмуванні, вони допомагають створювати ефективні програми й уникати помилок.

Назва — це унікальне «ім'я», за яким можна звертатися до величини в програмному коді.

Назва величини в Python має починатися з літери, перед якою може стояти один (або більше) знак підкреслення, і може містити алфавітно-цифрові символи і знаки підкреслення. Крім того, назва не може збігатися з назвою ключових слів мови Python (if, def, while тощо). Назви в Python чутливі до регістру: num і Num — це різні назви.

У Python кожна величина є об'єктом певного класу. Наприклад, число 7 — це об'єкт, який належить до класу (типу) int.

! Тип величини визначає множину допустимих її значень та операції, які можна виконувати над значеннями.

Ви вже знайомі з такими типами даних, як цілі та дробові числа, логічні значення, рядки (рис. 29.1).

```
>>> a = 12      # Ціле число
>>> b = 3.14    # Дробове число
>>> c = 'Hello!' # Рядок
>>> print(a, b, c)
12 3.14 Hello!
```

Рис. 29.1

Значення величини — це конкретні дані, які вона містить у момент звернення до неї. Для оперування величинами створюються змінні. **Змінною** в програмуванні називають іменований об'єкт, який зберігає певне значення.

Змінну часто порівнюють зі скринькою, в якій зберігається значення величини. Проте змінні в мові програмування Python більше схожі на ярлики, що прикріплюються до об'єктів, які використовуються в програмі.

! Змінна в Python зберігає посилання на значення в пам'яті комп'ютера.

Літерали в Python — подання фіксованих значень у програмі. Це можуть бути числа, символи, рядки тощо, наприклад, 7, 3.14, 'Hello, World!', 'A'.

Змінна створюється в той момент, коли їй уперше присвоюють значення. Для присвоювання значень змінним використовують оператор присвоювання =. Отже, щоб **створити змінну** в Python, необхідно дати їй назву та присвоїти значення.

Загальний вигляд команди присвоювання:

$A = B$, де A — назва змінної; B — літерал, змінна або вираз.

Python автоматично визначає тип даних змінної, виходячи зі значення, яке їй присвоєно. У разі присвоювання цілого числа Python визначає тип змінної як `int`.

Якщо змінній присвоїти дробове число, вона матиме тип `float`. Цілу й дробову частини при цьому відокремлюють крапкою.

1 Створимо змінну a :

$a = 5$.

Змінна a — це посилання на об'єкт, який містить значення 5 (рис. 29.2).

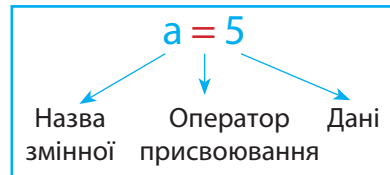


Рис. 29.2

Множинне присвоювання

У мові Python допустиме так зване множинне присвоювання, що дозволяє скоротити код. При цьому одне значення присвоюється кільком змінним одночасно. Отже, можна створити кілька змінних, потім у ході виконання програми змінити їх значення.

2 Присвоїмо змінним a , b , c значення 0:

$a = b = c = 0$

У результаті три змінні мають одне й те саме значення та зберігаються в тій самій ділянці пам'яті (рис. 29.3).

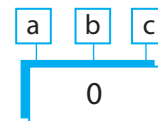


Рис. 29.3

Присвоїти значення кільком змінним можна однією командою.

3

Присвоїмо змінній `a` значення 43, змінній `b` — значення 'корови':

```
a, b = 43, 'корови'
```

4

Поміняємо місцями значення змінних `a` і `b` (рис. 29.4):

```
a = 3
```

```
b = 5
```

```
a, b = b, a
```

```
print(a, b)      # Буде надруковано 5 3
```

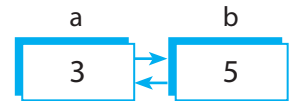


Рис. 29.4

Алгоритмічна структура слідування

Для конструювання алгоритмів використовують три базові алгоритмічні структури: слідування, розгалуження, повторення. Ви ознайомилися з ними в 5 класі.



Слідування — це така організація дій в алгоритмі, за якої дії виконуються послідовно одна за одною (без пропусків або повторень) (рис. 29.5).

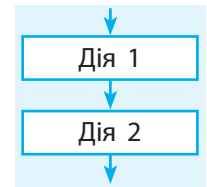


Рис. 29.5

Алгоритми, в яких використовується тільки структура слідування, називають **лінійними**.

У програмах, які реалізують лінійні алгоритми, використовують команди введення даних, присвоювання та виведення.

Команди введення та виведення даних

Діалог програми з користувачем чи користувачкою здійснюється за допомогою команд введення і виведення. Щоб такий діалог відбувся, людина має ввести за допомогою миші або клавіатури певні дані, а комп'ютер після цього — вивести результат їх опрацювання.

Функція `input()` уводить дані з клавіатури. У дужках може міститися підказка, що саме необхідно ввести (рис. 29.6).

3 Уведене значення перетворюється на ціле число

2 Підказка, що саме слід ввести

```
x = int(input('x = ?'))
```

4 Присвоювання значення змінній x

1 Зустрівши команду `input`, програма призупиняє роботу й очікує введення

Рис. 29.6

5 Уведемо з клавіатури значення змінної `a`:
`a = input('Введіть значення a: ')`

Функція `print()` виводить текст у вікно консолі. У списку виведення можуть бути константи, змінні, вирази. За замовчуванням після виведення всіх елементів списку курсор переводиться на наступний рядок. Якщо треба, щоб наступне виведення відбувалось у тому самому рядку, до списку виведення потрібно додати параметр `end = "` (замість `"` можна ввести текст-роздільник).

6 Виведемо значення змінних `x = 5` і `y = 7` та їх суми:
`print('x = ', x, ', y = ', y, end = ', ')` # Роздільник — кома і пропуск
`print('сума дорівнює ', x+y)`
 Буде виведено: `x = 5, y = 7, сума дорівнює 12.`

Значення, яке повертає функція `input()`, має рядковий тип (`str`), що може спричинити помилки при опрацюванні числових даних.

7 Введене за допомогою функції `input()` число зберігається як рядок. У разі спроби подвоїти значення змінної `a` буде виконано не множення чисел, а повторення рядкового значення.
`>>> a = input('a = ? ')`
`a = ? 5`
`>>> a*2`
`'55'`

Для правильного опрацювання числа при введенні даних слід перетворити рядкове значення на число за допомогою функції `int()` (рис. 29.7) (див. далі).

```
>>> a = int(input('a = ? '))
a = ? 5
>>> a*2
10
```

Рис. 29.7

Таким чином, для розв'язування будь-якої задачі в середовищі програмування потрібно правильно організувати введення, опрацювання і виведення даних.



Питання для самоперевірки

1. Які характеристики величин відповідають змінним:
 - а) $a = 12$;
 - б) $b = \text{True}$;
 - в) $c = 3.14$;
 - г) $d = "8.67"$?
2. Поясніть, як виконується команда `input()`.
3. Назвіть літерали та змінні в списку виведення:


```
print('a = ', a, 5, '3*b', 3*b)
```
4. Знайдіть значення x після виконання послідовності присвоювань:
 - а) $y = 2$; $x = y$;
 - б) $x = 8$; $x = x + 2$;
 - в) $x = 5$; $x = -x$.
5. Запишіть команди присвоювання, які реалізують такі дії:
 - а) змінній c присвоїти суму значень змінних a і b ;
 - б) подвоїти значення змінної a ;
 - в) значення змінної a збільшити на 10.



Вправа 29

- Розв'язати задачу на комп'ютері.

Задача. Вартість поїздки на таксі тривалістю до 5 хв — 20 грн, а кожної наступної хвилини — 10 грн. Складіть програму для розрахунку вартості поїздки тривалістю t хв (для $t \geq 5$).

1. Проаналізуйте задачу та складіть математичну модель.

Що дано? t — тривалість поїздки (хв).

Що треба знайти? c — вартість поїздки (грн).

Який зв'язок між величинами? $c = 20 + 10(t - 5)$.

Які є обмеження на дані? $t \geq 5$.

2. У Python IDLE виберіть команду File → New File.
Запишіть команду введення тривалості поїздки й присвоювання цього значення змінній t :


```
t = int(input('t = ? '))
```
3. Запишіть команду присвоювання значення змінній c .

4. Запишіть команду виведення значення змінної s , додавши до списку виведення текстову константу "грн".
5. Збережіть файл із назвою `Uprava29`. Випробуйте програму.
6. Доповніть програму перевіркою правильності введеного значення ($t \geq 5$). Випробуйте програму для різних значень t .



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106417).



§ 30. ОПРАЦЮВАННЯ ДАНИХ ЧИСЛОВИХ ТИПІВ

Цілі та дробові числа

Цілі числа належать до типу `int` (рис. 30.1) і можуть бути як завгодно великими та обмежуються лише доступною пам'яттю.

```
>>> a = 5
>>> type(a)
<class 'int'>
```

← Тип величини можна визначити за допомогою функції `type()`

Рис. 30.1

Дробові числа, або числа з рухомою комою, належать до типу `float`. Величина типу `float` може мати до 18 значущих цифр.



Знайдемо частку від ділення числа 18 на 3. Як ми бачимо, 18 кратне 3, але результат ділення належить до типу `float` (рис. 30.2).

```
>>> a = 18/3
>>> a
6.0
>>> type(a)
<class 'float'>
```

Рис. 30.2

Якщо числа цілого або дійсного типу досить великі або занадто малі, то їх записують як добуток числа й степе­ня числа 10. При цьому замість знака множення та числа 10 вказується велика літера E або мала e, а потім — показник степе­ня 10. Наприклад, 1.496E11 або 1.496e11.

2 Число після літери «e» — це показник степе­ня числа 10, на який треба помножити число 3.9, записане перед e:

$x = 3.9e3$

`print(x)` # 3900.0

$x = 3.9e-3$

`print(x)` # 0.0039

Перетворення типу даних

Під час розроблення програми може виникнути потреба в перетворенні значення одного типу даних на значення іншого типу.

Функція `input()` повертає значення рядкового типу (`str`), а для виконання над введеними даними математичних операцій вони повинні мати числовий тип. У цьому та інших подібних випадках виникає необхідність змінити тип даних.

Для подальшого опрацювання числа при введенні даних потрібно перетворити рядкове значення на числове, а при виведенні може знадобитися перетворення числа на рядок. Ці операції називаються **перетворенням типів**.

Для **перетворення типів** призначені такі функції:

- `str()` — створює рядок із числа;
- `float()` — створює дробове число з рядка або цілого числа;
- `int()` — створює ціле число з рядка або дробового числа.

! Зверніть увагу: дробове число при перетворенні на ціле не округлюється, а лише відкидається дробова частина.

Проаналізуймо приклади застосування функцій перетворення типів для різних аргументів, наведені в таблиці:

Вираз	<code>int('25')</code>	<code>int(4.9)</code>	<code>int('4abc56')</code>	<code>str(25)</code>	<code>float(25)</code>
Результат виконання	25	4	П о м и л к а	'25'	25.0

3

Складемо програму для розв'язування задачі. **Задача.** Скільки контейнерів можна розмістити на вантажній платформі розмірами $a \times b$ м, якщо сторона квадратної основи контейнера x м (рис. 30.3)? Сторони контейнерів паралельні краям платформи.

Поміркуємо, скільки контейнерів можна поставити вздовж сторін платформи: $k_1 = \frac{a}{x}$, $k_2 = \frac{b}{x}$.

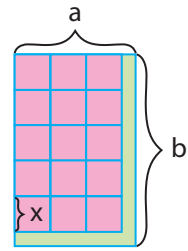


Рис. 30.3

Але число контейнерів має бути цілим, тобто $k_1 = \left\lfloor \frac{a}{x} \right\rfloor$, $k_2 = \left\lfloor \frac{b}{x} \right\rfloor$ (квадратні дужки позначають цілу частину числа). Тоді загальну кількість знайдемо як добуток k_1 і k_2 .

Що дано?

a, b, x

Що треба знайти?

k — число контейнерів

Який зв'язок між величинами?

$k_1 = \left\lfloor \frac{a}{x} \right\rfloor$, $k_2 = \left\lfloor \frac{b}{x} \right\rfloor$

Які є обмеження на дані?

$a > 0, b > 0, x > 0$

```
a = float(input('a = ?'))
b = float(input('b = ?'))
x = float(input('x = ?'))
k = int(a/x)*int(b/x)
print(k)
```

Якщо ми запусимо програму на виконання і введемо значення $a = 12.6$, $b = 7.5$, $x = 2.2$, то буде виведено значення $k = 15$.

4

Розглянемо приклади запису арифметичних виразів:

Арифметичний вираз	Лінійний запис виразу
$\frac{7-x}{5+3(x+4)} - 6x$	$(7-x)/(5+3*(x+4))-6x$
$\left \frac{x-1}{2x} \right $	$abs((x-1)/(2*x))$

Згадаймо математичні оператори Python, розгляньмо таблицю:

Операція	Оператор	Приклад	Результат
Додавання	+	a = 15+3	a = 18
Віднімання	-	b = 15-3	b = 12
Множення	*	c = 8*2	c = 16
Ділення	/	d = 7/2	d = 3.5
Обчислення неповної частки від ділення	//	e = 7//2	e = 3
Обчислення остачі від ділення	%	f = 7%2	f = 1
Піднесення до степеня	**	j = 7**2	j = 49

Арифметичні вирази записують в один рядок, а порядок дій, зокрема для дробів, визначають дужками.



Присвоїмо змінній *a* значення виразу $\frac{2x-5}{3+x}+7x$. Маємо:

$$a = (2*x-5)/(3+x)+7*x$$

► Стандартні математичні функції

У мові Python для розв'язування різних задач існують вбудовані (стандартні) функції. Їх можна використовувати без підключення модулів. Деякі з них описано в таблиці:

Функція	Що повертає	Приклад	Результат
abs(x)	x — модуль <i>x</i>	y = abs(-2.5)	y = 2.5
divmod(a, b)	Неповну частку й остачу від ділення <i>a</i> на <i>b</i>	c, d = divmod(7, 3)	c = 2 d = 1
round(num,[n])	Число <i>num</i> , округлене до <i>n</i> знаків після коми; якщо аргументу <i>n</i> немає, повертає найближче ціле	x = 1.5278 y = round(x) z = round(x, 2)	y = 2 z = 1.53

Закінчення таблиці

Функція	Що повертає	Приклад	Результат
eval(s)	Значення арифметичного виразу, запис якого зберігається в рядку <code>s</code>	<code>a = eval("(3+21)/3+4")</code>	<code>a = 12.0</code>

Випадкові числа

Ви вже застосовували випадкові числа для створення ігрових або тестових програм. Згадаймо: щоб використовувати функції для отримання випадкових чисел, слід завантажити модуль `random`:

```
from random import*
```

Щоб отримати випадкове число, необхідно викликати одну з функцій:

- `randint(x1, x2)` — ціле з діапазону від `x1` до `x2` ($x1 \leq x2$);
- `random()` — дробове з проміжку від 0.0 до 1.0.

Дробові числа з інших діапазонів отримують за допомогою математичних операцій:

```
number1 = 10+random()*50    # Значення від 10.0 до 60.0
print(number1)              # Може бути виведено 52.493371860628294
```



Питання для самоперевірки

- Обчисліть значення, яке повертає функція:
 - `abs(-6.9)`;
 - `divmod(16, 2)`;
 - `int(6.9)`;
 - `round(6.9)`.
 Результат перевірте в консолі Python.
- Поясніть, як отримати випадкове дробове число в діапазоні від 0 до 100.
- Обчисліть значення виразів:
 - `123//100`;
 - `123%10`;
 - `123//10%10`.
- Запишіть наведені вирази за правилами Python:
 - $10^5d - 4\frac{3}{7}c^3 + \frac{2}{3}$;
 - $\frac{35x-4y}{b} - \frac{8c+5}{4b} \cdot \frac{b}{x+y}$.
- Складіть програму для розв'язування задачі.

Задача. Потрібно обклеїти шпалерами стіну завдовжки a м і завширшки b м. Скільки коштуватимуть шпалери, якщо ціна рулону k грн? Довжина рулону 10 м, ширина — 80 см.



Вправа 30

- Написати програму Конвертер миль для переведення значення відстані, заданої в кілометрах, у милі.

Примітка. Мілья (від лат. *mille passuum* — тисяча подвійних кроків римських солдат) — одиниця вимірювання відстані, введена в Стародавньому Римі. Розрізняють сухопутну і морську милі: 1 сухопутна миля $\approx 1,609$ км; 1 морська миля $\approx 1,852$ км. В Україні до метричної системи були відомі велика та мала милі: велика = 12057 м; мала = 8956 м.

1. У Python IDLE виберіть команду File \rightarrow New File. Запрограмуйте введення значення змінної km:
`km = int(input('Уведіть відстань в км: '))`
2. Обчисліть відстань у сухопутних милях: `mile = km/1.609`
 Запишіть команду виведення значення змінної mile із трьома знаками після коми:
`print(km, ' км = ', round(mile, 3), ' миль.')`
3. Обчисліть значення заданої відстані в морських милях. Виведіть результат із трьома десятковими знаками.
4. Обчисліть значення заданої відстані у великих милях.
5. Обчисліть значення заданої відстані в малих милях.
6. Збережіть файл із назвою Vprava30. Випробуйте програму для km = 10, переконайтеся в правильності обчислень (рис. 30.4).

```
Уведіть відстань в км: 10
10 км = 6.215 миль.
10 км = 5.4 миль.
10 км = 0.829 миль.
10 км = 1.117 миль.
>>>
```

Рис. 30.4



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106418).



§ 31. АЛГОРИТМИ І ПРОГРАМИ З РОЗГАЛУЖЕННЯМИ

Логічними є вирази, при обчисленні яких одержується логічне значення True або False («істина» або «хиба»). Результат їх обчислення можна зберігати в змінну, яка буде мати тип bool.

Прості і складені вирази

Простим логічним виразом називають вираз, побудований за допомогою операцій порівняння. Розгляньмо таблицю:

==	дорівнює	>	більше	>=	більше або дорівнює
!=	не дорівнює	<	менше	<=	менше або дорівнює



Проаналізуємо результати обчислення логічних виразів:

>>> print(2*2 == 4) True	Оскільки $2 \cdot 2 = 4$ — істинна рівність, то значення відповідного логічного виразу дорівнює True
>>> NUM = 10 >>> NUM%3 == 0 False	Оскільки остача від ділення 10 на 3 не дорівнює 0, значення логічного виразу дорівнює False

Складений логічний вираз складається з одного або більше простих логічних виразів, до яких застосовано логічні операції and, or, not. Розгляньмо таблицю:

Операція	Назва	Приклади	Значення
or	або логічна сума	$2*2 == 7$ or $4 <= 5$ $3 > 8$ or $12 < 5$ or $9 == 1$	True False
and	і логічний добуток	$2*2 == 7$ and $4 <= 5$ $3 <= 8$ and $12 > 5$ and $9 != 1$	False True
not	не логічне заперечення	not $2*2 == 4$ not $8 < 3$	False True

2 Перевіримо, чи належить значення NUM проміжку [7; 20]:
 NUM = 12
 print (NUM >= 7 and NUM <= 20) # Буде виведено True

Якщо змінній присвоїти логічне значення, вона належатиме до типу bool:

```
>>> a = 3>7
>>> a
False
```

▶ Вбудовані функції all і any

Іноді одночасну перевірку декількох умов можна записати коротше за допомогою вбудованих функцій all (з англ. *всі*) і any (з англ. *хоч одна*).

Функція all(<список логічних виразів>) повертає True, якщо істинні всі перелічені в списку умови.

3 З'ясуємо, чи є значення NUM парним числом із проміжку від 7 до 20. Для цього слід одночасно перевірити три умови:
 >>> NUM = 8
 >>> all([NUM >= 7, NUM <= 20, NUM%2 == 0])
 True

Функція any(<список логічних виразів>) повертає True, якщо хоча б один елемент списку дорівнює True.

4 Перевіримо, чи дорівнює 0 хоча б одне зі значень x, y, z:
 >>> x, y, z = 4, 0, 5
 >>> any([x == 0, y == 0, z == 0])
 True

Алгоритмічна структура розгалуження

Розгалуження — це така організація дій в алгоритмі, за якої, залежно від істинності деякої умови, виконується певна послідовність дій.

Пригадаймо синтаксис та семантику команд розгалуження.

► Команда перевірки умови if

Виконання команди if (з англ. *якщо*) можна описати так: «якщо умова істинна, то виконати дії» (рис. 31.1). Команда if реалізує алгоритмічну конструкцію неповне розгалуження.

Синтаксис команди if:

```
if <умова>:
    ____<команди>
```

! Команди записують з обов'язковим однаковим відступом від лівого краю. Якщо команда лише одна, її можна записати після двокрапки в одному рядку з умовою.

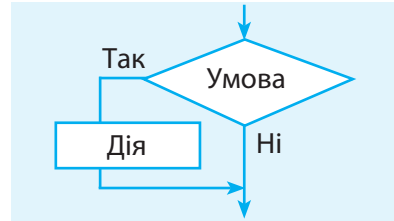


Рис. 31.1

5 Складемо програму для впорядкування двох уведених із клавіатури значень за неспаданням:

```
a, b = map(int, input().split())
if a>b:
    a, b = b, a
print(a, b)
```

► Команда розгалуження if...else

Алгоритмічній конструкції повне розгалуження (рис. 31.2) відповідає команда розгалуження if...else (з англ. *якщо...інакше*).

Синтаксис команди if...else:

```
if <умова>:
    ____<команди 1>
else:
    ____<команди 2>
```

Якщо <умова> має значення True, то виконується блок <команди 1>. В іншому випадку виконується блок <команди 2>, записаний після службового слова else. Команди, вкладені в гілки if і else, записують з однаковим відступом.

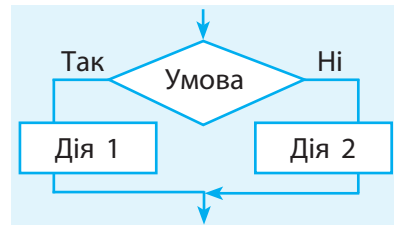


Рис. 31.2

Вкладення розгалужень

Якщо після перевірки деякої умови для прийняття рішення потрібно знову перевіряти умову, утворюють вкладені розгалуження. У програмі при цьому в одній або й обох гілках команди розгалуження записують нові команди розгалуження.



Складемо програму, яка б за введеним цілим числом виводила слово «рік» у правильному відмінку (рис. 31.3).

Проаналізувавши форми слова «рік», ми помічаємо, що форма слова залежить від останньої цифри числа. Проте значення від 11 до 14 слід опрацювати окремо (блок-схема алгоритму на рис. 31.4).

1 рік	11 років	21 рік
2 роки	12 років	22 роки
3 роки	13 років	23 роки
4 роки	14 років	24 роки
5 років	15 років	25 років
6 років	16 років	26 років
7 років	17 років	27 років
8 років	18 років	28 років
9 років	19 років	29 років
10 років	20 років	30 років

Рис. 31.3

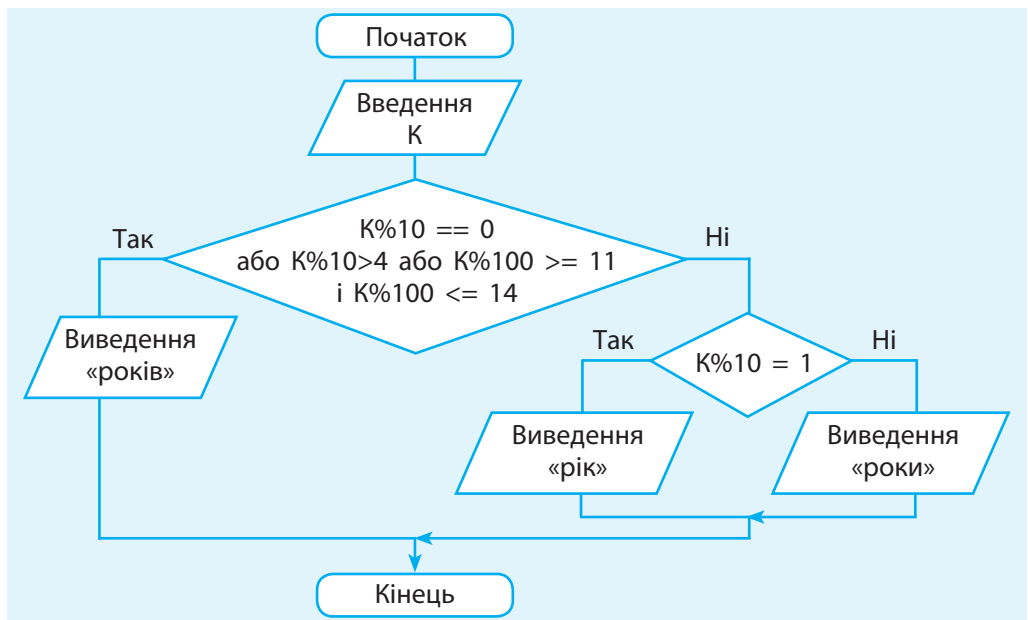


Рис. 31.4

Цей алгоритм реалізує така програма:

```
K = int(input())
if K%10 == 0 or K%10>4 or K%100 >= 11 and K%100 <= 14:
    print('років')
else:
    if K%10 == 1: print('рік')
    else: print('роки')
```

Багатоваріантне розгалуження

Вкладені розгалуження можна записати у компактнішому вигляді, скориставшись командою `if...elif...else` (англ. *else if* — інакше, якщо). Ця команда дає змогу перевірити послідовно кілька умов і виконати блок коду, що відповідає першій умові, яка виявиться істинною.

Синтаксис команди	Приклад
<pre>if <умова1>: <команди 1> elif <умова2>: <команди 2> else: <запасний блок команд></pre>	<pre>if a>b : print("a>b") elif a<b: print("a<b") else: print("a = b")</pre>

Згадаймо, які є правила виконання команди `if...elif...else`:

- якщо умова в гілці `if` *істинна* (True), виконується відповідний блок коду, а інші гілки команди пропускаються;
- якщо умова в гілці `if` *хибна*, послідовно перевіряються умови в гілках `elif`.

Якщо умова `elif` *істинна*, то виконується відповідний блок коду, і всі подальші гілки пропускаються; далі виконується команда, записана після `if-elif`;

- якщо всі попередні умови (`if` і `elif`) були *хибними*, виконується блок коду в гілці `else`. Гілка `else` є *необов'язковою*. Якщо гілки `else` немає й умови в усіх гілках `if-elif` *хибні*, жодного блоку команд виконано не буде.



Програма запитує назву річки і виводить повідомлення про те, куди річка впадає (рис. 31.5).

```
name = input('Уведіть назву річки: ')
if name = 'Дніпро' or name = 'Дністер': print('впадає в Чорне море.')
elif name = 'Ворскла': print('впадає в Дніпро.')
elif name = 'Збруч': print('впадає в Дністер.')
else: print('Відомостей немає.')
```

Рис. 31.5

Розглянуті команди розгалуження та логічні операції дозволяють реалізувати вибір дій у найрізноманітніших випадках.

Олександр Летичевський — видатний український учений у галузі кібернетики та інформатики (рис. 31.6). Результати, які він отримав у галузях теорії автоматів і алгоритмів, комп'ютерної алгебри, штучного інтелекту та теорії проектування обчислювальних систем, широко відомі світовій науковій спільноті.



Рис. 31.6



Питання для самоперевірки

- Початкові значення змінних $a = -3$; $b = 5$. Визначте, чому дорівнюють a і b після виконання таких команд:

а) if $a > b$: $a = a - b$	в) if $a < 0$:	г) if $a \neq b$:
else: $b = b - a$	$a = -a$	$a = b$
б) if $a < 0$:	г) if $a < 5$:	
if $a > b$: $a = b$	$b = a + 1$	
else: $b = a$	else:	
else:	if $a < 10$:	
if $a > b$: $b = a$	$b = a * 2$	
else: $a = b$		
- Складіть програму, яка за введеним значенням температури t виводить повідомлення про стан, у якому перебуває вода за такої температури (твердий, рідкий, газоподібний).

Вправа 31

- Для цілого числа K від 1 до 99 надрукувати фразу «Мені K років», урахувавши, що за деяких значень K слово «років» треба замінити на «рік» або «роки».
- 1. У Python IDLE виберіть команду File → New File.
- 2. Запрограмуйте введення значення змінної k :
`k = int(input('Введіть число від 1 до 99: '))`
- 3. Проаналізуйте, як залежить відповідь від числа K :

$11 \leq K \leq 14$	Мені K років
$K \% 10 = 1$	Мені K рік
$K \% 10 = 0$	Мені K років
$2 \leq K \% 10 \leq 4$	Мені K роки
$5 \leq K \% 10 \leq 9$	Мені K років

Із таблиці видно, що слід відокремити значення K від 11 до 14. Алгоритм розв'язування задачі наведено в прикладі 6.
`if(k >= 11) and (k <= 14): print('Мені ', k, ' років')`

- 4. Для інших значень K форма відповіді залежить від числа одиниць у десятковому записі числа K :
`if(k >= 11) and (k <= 14): print('Мені ', k, ' років')`
`elif k%10 == 0: print('Мені ', k, ' років')`
`elif k%10 == 1: print('Мені ', k, ' рік')`
`elif k%10 >= 5 and k%10 <= 9: print('Мені ', k, ' років')`
- ⚙️ 5. Допишіть гілку `elif` для перевірки умови $2 \leq k \leq 4$.
 Перевірте правильність роботи програми для різних значень K .
- 6. Збережіть програму у файлі з назвою `Vprava31`.
 Випробуйте програму, якщо: а) $k = 25$; б) $k = 12$.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106419).



§ 32. АЛГОРИТМИ І ПРОГРАМИ З ПОВТОРЕННЯМИ. ЦИКЛИ

Як відомо, однією з базових алгоритмічних структур є цикл.

Повторення (цикл) — це алгоритмічна структура, за допомогою якої певна послідовність дій виконується декілька разів.

Ви маєте досвід запису алгоритмів із повторенням (циклів) мовою Python і вмієте використовувати два види команд циклу: із параметром (`for`) та з умовою (`while`).

Згадаймо: команди, що повторюються під час виконання циклу, називають **тілом циклу**, а кожне виконання тіла циклу — **ітерацією**.

Цикл із параметром

Цикл із параметром (або цикл `for`) використовують, якщо кількість повторень визначається певною послідовністю значень. Таку послідовність можуть утворювати символи рядка, елементи списку тощо. Блок-схему циклу з параметром подано на рис. 32.1.

Синтаксис циклу `for`:

```
for x in <послідовність>: # Заголовок циклу
    ____<команди>
```

Тут змінна `x` — параметр циклу — послідовно набуває значень кожного члена послідовності.



Застосуємо цикл `for` для перебору символів рядка. Виведемо всі символи рядка, що містяться в змінній `s`, окрім «зірочок»:

```
s = 'AB*C*D*EF*G'
```

```
for C in s: # Перебір у змінній C символів рядка s
    if not C == '*': print(C, end = " ")
```

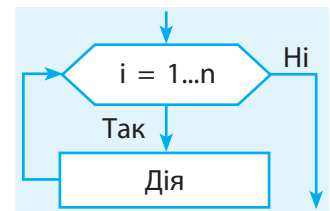


Рис. 32.1

2

Дано список днів тижня.

З'ясуємо, скільки з них є вихідними (сб, нд).

```
D = ['пн', 'нд', 'чт', 'сб', 'пт', 'сб', 'вт', 'ср', 'нд', 'чт', 'чт', 'нд', 'вт', 'пн', 'ср']
```

```
K = 0
```

```
for day in D:      # Перебір у змінній day елементів списку D
```

```
    if day in ['сб', 'нд']: K += 1
```

```
print(K)
```

Числові послідовності можна задавати як діапазон range():

```
range ([<початок>] <кінець>[, <крок>])
```

Такий об'єкт генерує послідовність цілих чисел від значення початок до значення кінець із заданим кроком, при цьому значення кінець у послідовність не входить. Чергове число генерується за кожного звертання до об'єкта, тому навіть для довгих послідовностей він не займає багато пам'яті.

3

Надрукуємо числа від 20 до 24.

Якщо крок дорівнює 1, цей параметр можна не зазначати:

```
for i in range(20, 25): print(i)
```

Надрукуємо числа від 0 до 3.

Якщо початок дорівнює 0, цей параметр можна не зазначати:

```
for i in range(4): print(i)
```

Надрукуємо числа від 5 до 1. Оскільки потрібно вести відлік у зворотному порядку, то крок має бути від'ємним:

```
A = range(5, 0, -1)
```

```
for i in A: print(i)
```

Цикл із передумовою

Цикл із передумовою (або цикл while) буде повторюватися доти, поки виконється деяка умова. Блок-схему команди while наведено на рис. 32.2.

Синтаксис команди:

```
while <умова>:      # Заголовок циклу
```

```
    ____<команди>
```

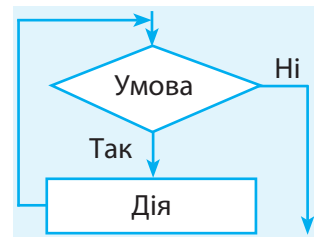


Рис. 32.2

Виконання команди циклу `while` починається з обчислення значення логічного виразу — умови повторення циклу.

Якщо *умова істинна*, то виконуються команди тіла циклу і керування повертається на перевірку умови.

Якщо *умова хибна*, то виконання циклу припиняється і виконується команда, записана після команди `while`.

! Якщо за першої перевірки умова виявиться хибною, то тіло циклу `while` не виконається жодного разу.

4

У багатьох математичних задачах виникає послідовність чисел, названа на честь італійського математика Фібоначчі: перші два числа дорівнюють 1, а кожне наступне дорівнює сумі двох попередніх. Математично це можна записати так:

$$Fib_1 = Fib_2 = 1; \quad Fib_n = Fib_{n-1} + Fib_{n-2}.$$

Визначимо номер першого числа Фібоначчі, більшого за 1000.

`a = b = 1` # Перші два числа Фібоначчі

`n = 2` # Змінна — лічильник чисел

`while b <= 1000:` # Поки останнє число менше або дорівнює 1000:

`a, b = b, a+b` # — обчислюємо нове число

`n += 1` # — збільшуємо лічильник

`print('число', b, 'має номер', n)`

Вкладені цикли

Вкладений цикл — цикл, який міститься в тілі іншого циклу.

Вкладений цикл також називають **внутрішнім**, а цикл, у якому він міститься, — **зовнішнім**.

Усередину вкладеного циклу може бути вкладено наступний цикл і так далі. У мові Python підтримується до 20 рівнів вкладеності.

Розгляньмо порядок виконання вкладених циклів.

Під час *першої ітерації* зовнішній цикл викликає внутрішній, який виконується до свого завершення, після чого керування повертається в тіло зовнішнього циклу.

Під час *другої ітерації* зовнішній цикл знову викликає внутрішній. І так доти, поки не завершиться зовнішній цикл.

5

Обчислимо значення змінної $y = 2^k + n$ при значеннях змінних $n = 1, 2, 3$ і $k = 2, 4, 7, 9$.

Якщо перебрати всі пари значень n і k , отримаємо 12 значень змінної y . Скласти програму можна так: для кожного значення $n = 1, 2, 3$ перебрати всі значення k зі списку $[2, 4, 7, 9]$, тобто n використати як параметр зовнішнього циклу, k — як параметр внутрішнього циклу (рис. 32.3):

```
print('n k y')
for n in range(1, 4):
    for k in [2, 4, 7, 9]:
        y = 2*k+n
        print(n, k, y)
```

n	k	y
1	3	5
1	4	9
1	7	15
1	9	19
2	2	6
2	4	10
2	7	16
2	9	20
3	2	7
3	4	11
3	7	17
3	9	21

Рис. 32.3

6

Книга коштує 240 грн. У покупця є купюри номіналами 20, 50 і 100 грн. Складемо програму для визначення всіх можливих варіантів оплати.

Переберемо всі можливі варіанти кількостей купюр. Щоб набрати названу суму, потрібно 20-гривневих купюр — не більше 12, 50-гривневих купюр — не більше 6 і 100-гривневих купюр — не більше 2. Позначимо кількості купюр кожного номіналу змінними k_{20} , k_{50} , k_{100} і переберемо у вкладених циклах їхні значення зі вказаних діапазонів. Відповіддю будуть такі набори значень, за яких $k_{20} \cdot 20 + k_{50} \cdot 50 + k_{100} \cdot 100 == 240$.

```
for k20 in range(13):           # Кількість 20-гривневих купюр
    for k50 in range(6):        # Кількість 50-гривневих купюр
        for k100 in range(3):  # Кількість 100-гривневих купюр
            if k20*20+k50*50+k100*100 == 240:
                print("20 грн - ", k20, ", 50 грн - ", k50, ", 100 грн - ", k100)
```

Результат виконання програми наведено на рис. 32.4.

20 грн - 2, 50 грн - 0, 100 грн - 2
20 грн - 2, 50 грн - 2, 100 грн - 1
20 грн - 2, 50 грн - 4, 100 грн - 0
20 грн - 7, 50 грн - 0, 100 грн - 1
20 грн - 7, 50 грн - 2, 100 грн - 0
20 грн - 12, 50 грн - 0, 100 грн - 0

Рис. 32.4

Отже, цикли в програмуванні є потужним інструментом для виконання завдань, пов'язаних із повтореннями дій. Почергове опрацювання ходів людини і комп'ютера під час гри, послідовне виведення на екран питань тесту або програвання нот мелодії — це все задачі, де стануть в нагоді цикли.



Питання для самоперевірки

- У вікні консолі введіть команду:
`for i in range(65, 90): print(chr(i))`
 Обговоріть результати виконання команди `for`.
 Поекспериментуйте зі зміною параметрів функції `range()`.
- З'ясуйте, що буде виведено в результаті виконання фрагмента програми:

а) <code>for i in range(5):</code>	б) <code>for x in 'рядок':</code>	в) <code>for x in range(5, 1, -1):</code>
<code>print(i*i)</code>	<code>print(2*x)</code>	<code>print(x)</code>
- Із клавіатури вводиться 10 довільних цілих чисел. Запишіть команди для виконання таких дій:

а) знайдіть суму та середнє арифметичне;	б) знайдіть суму тільки від'ємних чисел;	в) знайдіть найменше число.
--	--	-----------------------------
- Визначте значення змінної `s` після виконання операторів:

а) <code>s = 0</code> <code>i = 0</code> <code>while i<5:</code> <code> i = i+1</code> <code> s = s+i</code>	б) <code>s = 0</code> <code>i = 5</code> <code>while i>1:</code> <code> s = s+i</code> <code> i = i-1</code>	в) <code>s = 0</code> <code>i = 0</code> <code>while i<5:</code> <code> i = i+1</code> <code> s = s+10</code>
---	---	--
- Складіть програму для розв'язування задачі.
Задача. 20 л соку потрібно розлити в банки місткістю 3, 2, 1 л. Як отримати всі варіанти розливу? Підрахуйте кількість варіантів.
- Складіть програму обчислення площі прямокутних трикутників, катети яких набувають значень:
`a = 10; 10.15; 10.3; 10.45; 10.6; 10.75 (м)`
`b = 7; 7.4; 7.8; 8.2; 8.6; 9.0 (м)`
 Скільки значень площі трикутників буде виведено?

Вправа 32

- Написати програму розв'язування задачі.

Задача. Завод має відвантажити замовникам n шт. цегли ($n \geq 1000$). Цеглу можна перевозити пакунками по 100, 300 і 500 шт. Отримати всі можливі варіанти пакувань. Підрахувати кількість варіантів.

1. У Python IDLE виберіть команду File \rightarrow New File.
2. Запрограмуйте введення значення змінної n :
`n = int(input('Задайте кількість цегли (n >= 1000): '))`
3. Нехай значення кількості пакунків по 100, 300 і 500 шт. зберігатимуться в змінних $p1$, $p3$, $p5$. Пакунків по 100 шт. може знадобитися не більше $n//100$; тому значення змінної $p1$ слід перебрати в діапазоні `range(n//100+1)`:
`for p1 in range(n//100+1):`

Запишіть заголовки вкладених циклів для перебору значень змінних $p3$, $p5$.

4. Відповіддю будуть такі значення змінних $p1$, $p3$, $p5$, за яких $p1*100+p3*300+p5*500 == n$. Для їх виведення в тілі внутрішнього циклу запишіть команду розгалуження:
`if p1*100+p3*300+p5*500 == n: print(p1, p3, p5)`

- ⚙ 5. Додайте до програмного коду команди для визначення кількості варіантів пакування.

6. Збережіть програму у файлі з назвою `Vprava32`.

Випробуйте програму для $n = 1200$. Порівняйте отримані результати з наведеними на рис. 32.5.

```
0 4 0
1 2 1
2 0 2
3 3 0
4 1 1
6 2 0
7 0 1
9 1 0
12 0 0
```

Кількість варіантів: 9

Рис. 32.5



Комп'ютерне тестування


Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106420).



§ 33. РЯДКОВИЙ ТИП ДАНИХ

Багато задач програмування пов'язані з опрацюванням текстових даних: це задачі шифрування, перекладу з однієї мови на іншу тощо. Для опрацювання текстових даних у Python існує тип даних `str` (від англ. *string* — рядок).

Найпростіші дії з рядками ми вже виконували під час організації введення та виведення даних. Розгляньмо складніші.

 **Значення величини типу `str`** — це довільна послідовність (рядок) символів.

Рядкові константи в Python записують в одинарних або подвійних лапках. Початок і кінець рядка позначають лапками одного виду.

Коди символів


Усі символи (літери, цифри, знаки пунктуації тощо) утворюють так звану таблицю символів. Кожний символ у ній має свій номер — код.

Код символу можна визначити за допомогою функції `ord()`:

```
>>> ord('A')      # Велика англійська літера 'A' (її код – 65)
65
```

Дізнатися, який символ відповідає певному коду, можна за допомогою функції `chr()`:

```
>>> chr(1040)    # 1040 – код великої української літери 'А'
'А'
```

 Великі й малі літери, літери англійського й українського алфавітів із подібним написанням мають різні коди.

Мова програмування Python підтримує для рядків таблицю символів стандарту Юнікод (англ. *Unicode*; рис. 33.1).



Рис. 33.1

Нумерація символів у рядку

Символи в рядку впорядковані: кожний із них має порядковий номер — **індекс**. Нумерація символів починається з нуля.

До символу в рядку можна звертатися за індексом, зазначивши його у квадратних дужках. Розгляньмо приклади.

1 Надрукуємо першу літеру рядка Червона калина:

```
>>> a = 'Червона калина'
>>> a[0]
'ч'
```

Відлік символів можна починати з кінця рядка. У цьому випадку індекси починаються з -1 (останній символ).

2 Надрукуємо третю з кінця літеру рядка Червона калина:

```
>>> a[-3]
'и'
```

3 Присвоїмо змінній `b` символ рядка `a` з індексом 8:

```
>>> b = a[8]
>>> b
'к'
```

Функція `len()` повертає кількість символів, із яких утворено рядок:

```
k = len(s),
```

де `k` — змінна для збереження значення, яке повертає функція; `s` — рядок, довжину якого потрібно визначити.

4 Визначимо довжину рядка Червона калина (див. приклад 1):

```
>>> len(a)
14
```

Пропуск теж є символом, тому довжина рядка дорівнює 14.

Отримання зрізів

Із рядка можна отримати не лише символ, а й зріз (частину символів рядка). Зріз із суміжних символів — це *підрядок*.

Використання зрізів дозволяє отримувати підрядки за одну дію.

Команда отримання зрізу з рядка:

```
z = s[i:j:step],
```

де z — змінна, в яку зберігається зріз рядка s ; i — індекс початку зрізу; j — індекс його закінчення (причому символ із номером j у зріз не входить); $step$ — крок, з яким вибираються символи.

Якщо символи вибираються підряд, то параметр $step$ можна опустити. Якщо відсутній індекс початку зрізу, то зріз береться від 0-го до $(j-1)$ -го символу. За відсутності індексу закінчення зрізу зріз береться від i -го символу до кінця рядка.

5

Скопіюємо з рядка a підрядок Червона:

```
>>> a = 'Червона калина'
>>> a[:7]
'Червона'
```

6

Скопіюємо з рядка a підрядок калина:

```
>>> a[8:]
'калина'
```

7

Скопіюємо з рядка a символи з парними індексами:

```
>>> a[::2]
'Чроаклн'
```



!

Рядки — це незмінювані послідовності. При спробі змінити символ у рядку виникне помилка (рис. 33.2).

```
>>> s[3] = 'a'
Traceback (most recent call last):
  File "<pyshe 11#3>", line 1, in <module>
    s[3] = 'a'
TypeError: object does not support item assignment
```

Рис. 33.2

8

Запишемо символи рядка a у зворотному порядку:

```
>>> a[::-1]
'анилак ановреЧ'
```


Перегляд символів у рядку

Під час опрацювання рядкових величин буває потрібно перевірити, чи є певний підрядок у рядку. Для цього в Python існує оператор `in`, який повертає `True`, якщо підрядок наявний у рядку, і `False`, якщо ні.


```
>>> print('y' in 'Python')
True
```

За допомогою циклу `for` можна вивести рядок по літерах:

```
>>> for c in a: print(c)
```

Операції конкатенації і дублювання

Об'єднання двох або більше рядків в один рядок називають **конкатенацією**. Цю операцію позначають символом «+».

 Об'єднаємо рядкові змінні `a` і `b`, додавши між ними пропуск.

```
>>> a = 'Богдан'
>>> b = 'Хмельницький'
>>> c = a + ' ' + b           # c = 'Богдан Хмельницький'
```


Дублювання — це повторення текстового фрагмента задану кількість разів. Цю операцію позначають символом «*».

 Повторимо символ «-» 20 разів.

```
>>> c = '-'*20               # c = '-----'
```

 Потроїмо кожний символ рядка `s`.

```
s = '12345'
s1 = ""
for i in s:
    s1 = s1+i*3
print(s1)                   # Буде надруковано 111222333444555
```

 Американська розробниця програмного забезпечення Радія Джей Перлман однією з перших почала навчати програмуванню дітей молодшого віку. Вона розробила **TORTIS** — дитячу версію навчальної робототехнічної мови **LOGO**.

Біля витоків української школи програмування стояла Катерина Ющенко, докторка фізикоматематичних наук, авторка однієї з перших у світі мов програмування високого рівня, в якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера (рис. 34.5).



Рис. 34.5



Питання для самоперевірки

- $s = \text{'Вивчайте мову Python!'}$
Визначте довжину рядка s і індекс літери 'P'.
- $s = \text{'Сонечко'}$
Якого значення набуває змінна a в результаті виконання команд:

а) $a = s[2:5];$	в) $a = s[5:];$	г) $a = 5*s[2];$
б) $a = s[:3];$	г) $a = s[::2];$	д) $a = s[:: -1];$
- Запишіть команди, що реалізують присвоєння змінній b таких зрізів рядка a :
 - перші вісім символів;
 - останні чотири символи;
 - символи з індексами, кратними трьом.
- Запишіть команду, за якою з рядка S отримано рядок $S1$:
 - $S = \text{'інформатика'} \rightarrow ? \rightarrow S1 = \text{'форма'}$
 - $S = \text{'інформатика'} \rightarrow ? \rightarrow S1 = \text{'інтика'}$
- Розгадайте фразу, що зашифрована в рядку $s1$.
Який зріз із рядка $s1$ потрібно присвоїти рядку $s2$ для розшифрування цієї фрази:
 $s1 = \text{'я клеюгбмлщюй кіеннфгошрщмзастхиткуур'}$
 $s2 = s1[?]$
- За допомогою звернень до символів у рядку за їхніми індексами та операції конкатенації отримайте рядки "програміст" і "користування" з рядка
 $a = \text{"Тестуванням можна довести наявність помилок у програмі, але не їх відсутність. Едсгер Дейкстра"}$

Вправа 33

- Зашифрувати повідомлення шляхом вставляння заданого символу після кожної літери повідомлення.
- 1. Запишіть команду введення з клавіатури відкритого повідомлення і збереження рядкового значення у змінній `s_open`.
`s_open = input('Уведіть відкрите повідомлення: ')`
 Запишіть команду введення з клавіатури символу для вставляння і збереження його значення у змінній `symbol`.
- 2. Створіть змінну `s_shifr` для збереження зашифрованого повідомлення: `s_shifr = ''`
- 3. Реалізуйте алгоритм шифрування:
 - 1) послідовно перебираємо всі символи рядка `s_open`;
 - 2) при кожній ітерації циклу додаємо до рядка `s_shifr` черговий символ рядка `s_open` і символ `symbol`.

```
for c in s_open:
    s_shifr = s_shifr+c+symbol
```

 Запишіть команду виведення зашифрованого повідомлення.
- 4. Розшифруйте зашифроване повідомлення, отримавши зріз зі значення змінної `s_shifr`:
`s_deshifr = s_shifr[::2]`
 Виведіть значення змінної `s_deshifr`.
- 5. Зашифруйте рядок `s_open`, кожен символ замінивши його кодом і додавши пропуск:
`s_shifr = ''`

```
for c in s_open:
    s_shifr = s_shifr+ord(c)+' '
```

 Запишіть команду виведення зашифрованого повідомлення.
- ⚙️ 6. Збережіть програмний код у файл із назвою `Vprava33`. Випробуйте програму, ввівши відкрите повідомлення 'Вивчайте мову Python!'.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106421).



§ 34. ОПРАЦЮВАННЯ РЯДКОВИХ ВЕЛИЧИН

Ви вже знайомі з прийомами роботи з текстом у текстовому редакторі, наприклад із пошуком і заміною символів. Мова Python має засоби для того, щоб запрограмувати різні дії з рядковими величинами.

Усі змінні в Python є об'єктами, а відповідні операції доступні у вигляді методів.

Синтаксис виклику методу для змінної:

<назва змінної>.<назва методу> ([<параметри>])

! Рядкові методи в Python не змінюють рядка, для якого викликані, а утворюють новий.

Якщо в програмі потрібно використовувати результат виклику методу, його слід зберегти в змінній (у тій самій або іншій).

Рядкові методи

Розгляньмо в інтерактивному режимі деякі методи опрацювання рядків. У наведених далі прикладах методи викликаються для змінної `s`, яка містить рядок, що опрацьовується.

► Пошук символів у рядку

Метод `find()` шукає в рядку `s` заданий рядок (підрядок) і повертає індекс першого символу цього підрядка в рядку.

Якщо підрядок не знайдено, метод повертає значення `-1`:

```
номер = s.find(subs),
```

де `номер` — змінна цілого типу, яка набуває значення індексу першого символу підрядка `subs` у рядку `s`; `subs` — шуканий підрядок.



З'ясуємо, чи є підрядок `'ох'` у рядку `s`:

```
>>> s = 'Торох, торох, розсипався горох'
>>> s.find('ох')           # Буде виведено 3
```

Отже, в рядку `s` є підрядок `'ох'` і він починається з символу з індексом 3.

▶ **Заміна символів у рядку**

Метод `replace()` шукає й замінює підрядок у заданому рядку:
`s_rez = s.replace(s_find, s_ins)`,

де `s_rez` — змінна для збереження результату заміни; `s_find` — підрядок, який потрібно замінити в рядку `s`; `s_ins` — рядок, яким буде замінено підрядок `s_find`.

Якщо `s_ins = ""` (порожній рядок), то метод `replace` просто вилучить знайдені підрядки.

2 Замінімо в рядку `s` усі підрядки `'cry'` на підрядки `'smile'` і результат запишемо в змінну `s1`:

```
>>> s = 'Why do you cry, Willy, why do you cry?'
>>> s1 = s.replace('cry', 'smile')
>>> s1
'Why do you smile, Willy, why do you smile?'
```

▶ **Визначення кількості входжень підрядка**

За допомогою методу `count()` можна визначити кількість входжень заданого підрядка в рядок:

```
kilk = s.count(subs),
```

де `kilk` — змінна, що набуває значення кількості входжень підрядка `subs` у рядок `s`; `subs` — шуканий підрядок.

3 Підрахуємо кількість повторень підрядка `'ох'` у заданому рядку `s`. Результат присвоїмо змінній `k`:

```
>>> s = 'Торох, торох, розсипався горох'
>>> k = s.count('ох')          # k = 3
```

▶ **Перевірка складу рядка**

Іноді буває потрібно перевірити, чи складається рядок тільки з літер або тільки з цифр. Для цього в Python є спеціальні методи, які повертають логічне значення (True або False). А саме:

- `s.isdigit()` — повертає True, якщо рядок `s` містить тільки цифри;
- `s.isalpha()` — повертає True, якщо рядок `s` містить тільки літери.

4 Запишемо фрагмент коду програми, який пропонує створити пароль, що складається тільки з літер:

```

parol = ""
while not(parol.isalpha()):
    parol = input('Введіть пароль: ')
print('Ваш пароль: '+parol)

```

Приклад виконання програми наведено на рис. 34.1.

```

Введіть пароль: par123ol
Введіть пароль: 20-20-20
Введіть пароль: secret
Ваш пароль: secret

```

```
>>>
```

Рис. 34.1

Розгляньмо приклади програм для опрацювання рядків.

5

Дано рядок: 'Ми програмуємо мовою Python!'. Визначимо довжину рядка, код букви з індексом 10, кількість букв 'м' у рядку, позицію, з якої починається слово 'Python' (рис. 34.2).

```

s = 'Ми програмуємо мовою Python!'
print(len(s))           # Довжина рядка s           28
print(ord(s[10]))       # Код букви з індексом 10           1091
print(s.count('м'))     # Кількість букв 'м' у рядку s       3
print(s.index('Python')) # Позиція початку слова 'Python'     21

```

Рис. 34.2

6

Вставимо підрядок subs у заданий рядок s після n-го символу (рис. 34.3):

```

s = '1234'
subs = 'abc'
n = 2
snew = s[0: n+1]+subs+s[n+1: len(s)]
print(snew)   # Буде надруковано '123abc4'

```

Рис. 34.3

7

Перевіримо, чи є рядок s паліндромом (рис. 34.4):

```

s = 'козак з казок'
s1 = s[::-1]   # Запис рядка у зворотному порядку
if s == s1: print('ТАК')
else: print('НІ')

```

Рис. 34.4



Задано текстовий рядок. Вилучимо з нього всі символи, що не є цифрами (рис. 34.5):

```
s = 'd3fgh56as67'
cifra = '0123456789'
snew = " # Створено порожній рядок snew
for i in range(len(s))
    if not (s[i] in cifra) # Якщо символ не є цифрою,
        snew = snew+s[i] # додати символ до рядка snew
print(snew) # Буде надруковано 'dfghas'
```

Рис. 34.5

Методи опрацювання рядкових величин широко застосовуються в програмуванні — у програмах, які призначено для шифрування і дешифрування текстів, для літературознавчого аналізування та визначення авторської приналежності твору тощо.



Питання для самоперевірки

1. Як можна видалити підрядок із заданого рядка? Поясніть на прикладі.
2. Як у заданий рядок уставити інший рядок? Поясніть це на прикладі.
3. `s = 'мова програмування Python'`
Яких значень набуває змінна `x` після виконання команд:
а) `x = len(s)`; в) `x = s.find('o')`;
б) `x = s.replace('o', 'A')`; г) `x = s.count('v')`?
4. Розгадайте фразу, що зашифрована в рядку
`s1 = 'lfin lofinvefin youfin'`.
Запишіть команду для розшифрування цієї фрази.
5. Дано рядок: 'Добрідень тобі, Україно моя!'. Визначте довжину рядка, код букви з індексом 5, кількість букв 'o' в рядку, позицію, з якої починається слово Україно.
6. Рядок містить дефіс. Поміняйте місцями частини рядка до та після дефісу. Наприклад, для рядка 'comp-uter' результатом має бути рядок 'uter-comp'.




**Вправа 34**

- Опрацювати рядкові величини.

Задача. У XIX ст. для пересилання важливих повідомлень використовували шифри, в яких літери в повідомленні переставляють за певним правилом. Написати програму, яка шифрує повідомлення, міняючи місцями кожні два знаки.

Наприклад: Введення: Я чекаю вас завтра
Виведення: Ячакитумв саз варт а

1. У Python IDLE виберіть команду File → New File. Запрограмуйте введення значення змінної `s`.
2. Створіть змінну `s1` для формування результату і надайте їй порожнє значення.
3. Кількість символів у рядку `s` має бути парним числом, тому запишіть команду розгалуження для реалізації такої дії: якщо довжина рядка `s` є непарним числом, то додати до рядка пропуск:

```
if (len(s)%2 != 0): s = s+' '
```
4.  Запишіть команду циклу для послідовного звернення до парних індексів (0 опрацьовуємо як парне число). У тілі циклу до результуючого рядка `s1` додаються символи `s[i+1]` та `s[i]`:

```
for i in range(0, len(s), 2):
    s1 = s1+s[i+1]+s[i]
```
5. Виведіть зашифроване повідомлення.
6. Збережіть програмний код у файл із назвою `Vprava34`. Виконайте програму.

**Комп'ютерне тестування**

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106422).





ПРАКТИЧНА РОБОТА 9

СТВОРЕННЯ ПРОГРАМ ДЛЯ ОПРАЦЮВАННЯ РЯДКОВИХ ВЕЛИЧИН

- Завдання:** скласти програму для розв'язування задачі.
- Задача.** Дано номер мобільного телефону і прізвище власника. Потрібно:
- зашифрувати прізвище власника, використовуючи заміну символів їх кодами;
 - визначити PIN-код, який містить суму цифр у номері телефону і кількість символів у прізвищі власника;
 - вивести у зашифрованому вигляді прізвище власника та його PIN-код.
- Обладнання:** комп'ютер із середовищем програмування мовою Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виберіть команду File → New File.

- Запишіть команди для введення прізвища власника та номера його мобільного телефону:

```
s = input('Прізвище? ')
number = int(input('Телефон? '))
```

- Замініть символи рядка s їх кодами. Створіть змінну shifr і надайте їй значення порожнього рядка. У циклі for для кожного символу рядка s отримайте код, перетворіть числове значення коду в тип string і додайте до рядка shifr (з додаванням пропуску після кожного коду):

```
shifr = ""
for i in range(len(s)):
    shifr = shifr+str(ord(s[i]))+' '
```

Запишіть команду виведення рядка shifr.

3. Знайдіть суму цифр у номері телефону. Для обчислення суми цифр числа `number` використайте такий алгоритм:
поки `number > 0`:
 - додати до суми `pin` останню цифру числа `number`;
 - здійснити цілочисельне ділення числа `number` на 10.

```
pin = 0
while number > 0:
    pin = pin + number % 10
    number = number // 10
```
4. Знайдіть кількість символів у прізвищі власника:

```
lens = len(s)
```
5. Щоб сформувати рядок PIN-коду власника, потрібно значення змінної `pin` перетворити до типу `string` і додати перетворене до типу `string` значення змінної `lens`. Але довжина прізвища може бути менше 10 символів, а PIN-код має бути 4-цифровим. Тому у випадку `lens < 10` при формуванні PIN-коду додамо символ '0':

```
lens = len(s)
if lens < 10:
    pincode = str(pin) + '0' + str(lens)
else:
    pincode = str(pin) + str(lens)
```
6. Збережіть програмний код у файлі з назвою `Pract9`.
Випробуйте програму.
Приклад результату виконання програми наведено на рисунку.

```
Прізвище? Іваненко
Телефон: 0681234567
Зашифроване прізвище: 1030 1074 1072 1085 1077 1085 1082 1086
PIN-код: 4208
```


Зробіть висновок: як створити програму для опрацювання рядкових величин.


§ 35. СПИСКИ — ЗМІНЮВАНІ ПОСЛІДОВНОСТІ

Ми вже розглянули прості типи даних — цілі числа, дробові числа, рядки, логічні значення. Величина простого типу містить одне значення. Але в житті ми постійно стикаємося з наборами значень, об'єднаних за певними ознаками, наприклад, список гравців футбольної команди, значення середньодобових температур за тиждень тощо.

Поняття списку

Для опрацювання сукупностей даних у програмуванні існують структуровані типи, один з яких — список. Величина структурованого типу може містити множину значень.

 **Список (list)** — це упорядкований набір об'єктів різних типів (чисел, рядків, списків тощо), який можна змінювати.

 У тексті програми список записують у квадратних дужках [], а його елементи відокремлюють комами.

Ви вже використовували списки колірних констант під час малювання Черепашкою:

```
colors = ['red', 'yellow', 'green', 'blue'].
```

Як бачимо, створити змінну типу список можна за допомогою команди присвоювання.

 Створимо змінні типу список:

```
list1 = [] # Порожній список
list2 = [3.14, 9.8, 2.4, 1.0] # Список із дробових чисел (тип float)
list3 = ['робот', 'машина', 'ведмедик', 'курка'] # Список із рядків
# (тип str)
list4 = ['Дмитрук', 2008, 1.64, 'футбол'] # Список із даних
# різних типів
```

Список можна уявити як полицю, на якій розташовано деякі об'єкти (рис. 35.1).



Рис. 35.1

Додати елемент до списку можна за допомогою методу `append()`. Цей елемент займе місце в кінці списку.

2

Додамо до списку `list3` елемент 'равлик':

```
list3.append('равлик')
```

```
print(list3)
```

Буде виведено: ['робот', 'машина', 'ведмедик', 'курка', 'равлик'].

!

Додати елемент можна тільки до вже створеного списку (можливо, порожнього).

3

Створимо список із натуральних чисел від 1 до 10:

```
a = []
```

```
for i in range(1, 11):
```

```
    a.append(i)
```

```
print(a)          # Буде надруковано [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Операція `in`

Перевірити, чи є в списку певний елемент, допоможе операція `in`.

Синтаксис використання оператора `in`:

```
<значення> in <послідовність>
```

де послідовність — рядок, список, діапазон тощо, в якому виконується пошук значення.

Результат операції — значення логічного типу: `True`, якщо послідовність містить значення, або `False`, якщо не містить.

4 Визначимо, чи є в списку `letters` елемент `'a'`:
`if 'a' in letters: print('Так')`
`else: print("Елемента 'a' в списку немає")`

Логічний вираз `'a' in letters` набуває значення `True`, якщо в списку `letters` є рядок `'a'`, або значення `False` — якщо немає.

Нумерація елементів списку

Списки є впорядкованими послідовностями, тому можна звертатися до їх елементів за індексами, отримувати зрізи, визначати довжину списку. Нумерація елементів починається з нуля (рис. 35.2).

```
lst = ['a' 'b' 'c' 'd' 'e' 'f']
lst[0] = 'a'
```

Рис. 35.2

5 Дано список `lst = ['a', 'b', 'c', 'd', 'e', 'f']`. Проаналізуємо результати виконання дій зі списком.

№ з/п	Дія	Оператор	Результат
1	Визначити довжину списку	<code>>>> len(lst)</code>	6
2	Вивести елемент списку з індексом 3	<code>>>> lst[3]</code>	'd'
3	Вивести перші 3 елементи списку	<code>>>> lst[0:3]</code>	['a', 'b', 'c']
4	Вивести елементи списку починаючи з елемента з номером 3 і до кінця	<code>>>> lst[3:]</code>	['d', 'e', 'f']
5	Вивести зріз списку довжиною 1 елемент	<code>>>> lst[3:4]</code>	['d']

Зверніть увагу на те, що зріз списку теж є списком. У прикладі 5 (2) для отримання елемента списку вказано його індекс, тому отримано значення типу `string`. Але якщо вказати суміжні індекси через двокрапку (приклад 5 (5)), отримаємо зріз у вигляді списку з одного елемента.

На відміну від рядків, списки — це *змінювані* послідовності: елементи списку можна змінювати, вилучати, додавати нові.

6 Змінимо значення 1-го елемента списку `mylist` на "м'яч" (рис. 35.3). (У слові «м'яч» є апостроф, тому слово потрібно взяти в подвійні лапки.)

```
>>> mylist = ['робот', 'машина', 'ведмедик', 'курка']
>>> mylist[1] = "м'яч"
>>> mylist
['робот', "м'яч", 'ведмедик', 'курка']
```

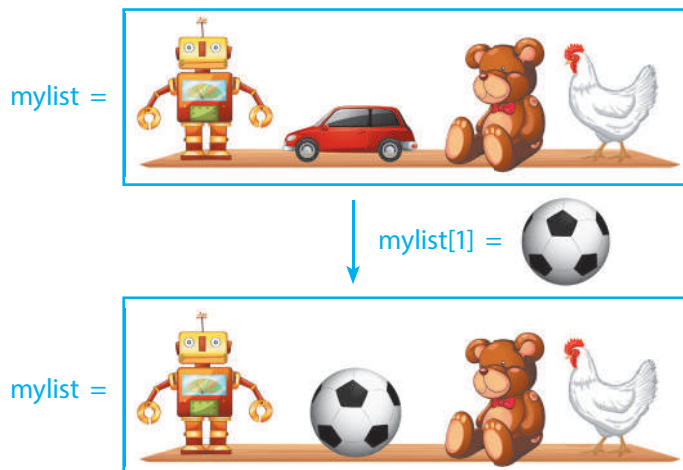


Рис. 35.3

7 Замінімо перші три елементи списку нулями:

```
>>> lst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> lst[0:3] = [0, 0, 0]
>>> lst
[0, 0, 0, 4, 5, 6, 7, 8, 9, 10]
```

Об'єднання й повторення списків

Як і над рядками, над списками можна виконувати операції об'єднання (+) та повторення (*).

8 Подвоїмо список [1, -2, 5]:

```
>>> a = [1, -2, 5]
>>> a = a*2
>>> a
[1, -2, 5, 1, -2, 5]
```

9 Об'єднаємо списки [1, -2, 5] і [2, 3, 0]:

```
>>> c = [1, -2, 5]+[2, 3, 0]
>>> c
[1, -2, 5, 2, 3, 0]
```



Списки дуже корисні в програмах, у яких потрібно працювати з сукупностями даних (наприклад, таблиця результатів спортивних змагань). Як бачимо, використання списків дозволяє опрацьовувати в циклі всі елементи за єдиним алгоритмом.



Питання для самоперевірки

1. Які об'єкти можуть бути елементами списків?
2. Як додати елемент до списку?
3. Як отримати частину заданого списку? Наведіть приклади.
4. Як виконується операція об'єднання списків? Поясніть на прикладі.
5. Дано список $a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$. Яких значень набувають елементи списку b , якщо:

а) $b = a[0:3]$;	в) $b = a$;	г) $b = a[2:3]+[1, -2, 0]$;
б) $b = a[6:]$;	г) $b = [-1, 0, 1]$;	д) $b = [-1, -2, 5]+[2, 3, 0]$?
6. Заповніть список a десятьма цілими випадковими числами ($0 \leq a_i < 10$). Перевірте, чи є у списку число 0. Створіть список b з елементів списку a , що мають парні індекси.



Вправа 35

- Скласти ігрову програму Відгадай столицю. Занести назви країн і їхніх столиць до двох списків. Програма повинна почергово запитувати назву столиці та перевіряти правильність відповіді, а після трьох помилок припиняти роботу, повідомляючи результат.

У Python IDLE виберіть команду File → New File.

1. Створіть список назв країн і список їхніх столиць:
`kr = ['Франція', 'ОАЕ', 'Нідерланди', 'Туреччина', 'Швейцарія']`
`st = ['Париж', 'Абу-Дабі', 'Амстердам', 'Анкара', 'Берн']`



2. Створіть змінну — лічильник помилок: `k = 0`.
Надайте значення 0 змінній `i` для збереження індексу поточного елемента.

Запитання пропонуються доти, поки не закінчився список країн (`i < 5`) і людина не припустилася третьої помилки (`k < 3`).

```
while k < 3 and i < len(kr):
    print('Країна: ', kr[i])
    s = input('Назвіть столицю країни: ')
    if s == st[i]: print('Правильно!')
    else:
        k = k + 1
        print('Помилка!')
    i = i + 1
```



3. Напишіть команду розгалуження, у якій (залежно від значення `k`) виводиться відповідне повідомлення:
`if k == 0: print('Вітаю! Ви – знавець географії!')`
`elif k == 1: print('Непогано! Ви можете вирушати в подорож!')`
`elif k == 2: print('Почитайте підручник географії!')`
`else: print('Вам краще сидіти вдома!')`



4. Додайте до списків назви ще декількох країн і столиць. Внесіть до програмного коду відповідні зміни.
5. Збережіть програму у файлі з назвою `Vprava35`.
6. Випробуйте програму.
Проаналізуйте результати роботи програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106423).



§ 36. ФУНКЦІЇ ТА МЕТОДИ ОПРАЦЮВАННЯ СПИСКІВ

Для розв'язування різноманітних задач опрацювання даних за допомогою списків, поряд зі вже розглянутим методом для додавання елемента `append()`, Python має низку методів списків і вбудованих функцій.

Методи списків

Як ви знаєте, **синтаксис виклику методів** такий:

<назва об'єкта>.<назва методу>([параметри])

Методи списків змінюють список, тому результат не потрібно зберігати в іншу змінну.

Розгляньмо на прикладі списку `a = [1, 5, 7, 5, 31, -5]` деякі методи списків:

Метод	Призначення	Приклад
<code>list.insert(i, x)</code>	Вставляє на <i>i</i> -те місце елемент зі значенням <i>x</i>	<pre>>>> a.insert(2, 78) >>> a [1, 5, 78, 7, 5, 31, -5]</pre>
<code>list.pop(i)</code>	Видаляє зі списку елемент з індексом <i>i</i> та повертає його значення	<pre>>>> a.pop(4) 31 >>> a [1, 5, 7, 5, -5]</pre>
<code>list.remove(x)</code>	Видаляє зі списку перший елемент зі значенням <i>x</i>	<pre>>>> a.remove(5) >>> a [1, 7, 5, 31, -5]</pre>
<code>list.index(x)</code>	Повертає індекс першого елемента зі значенням <i>x</i>	<pre>>>> a.index(31) 4</pre>
<code>list.count(x)</code>	Повертає кількість елементів зі значенням <i>x</i>	<pre>>>> a.count(5) 2</pre>

Продовження таблиці

Метод	Призначення	Приклад
list.sort() list.sort(reverse = True)	Упорядковує список за зростанням (або за спаданням, якщо параметр reverse = True)	>>> a.sort() >>> a [-5, 1, 5, 5, 7, 31]
list.reverse()	Переставляє елементи списку у зворотному порядку	>>> a.reverse() >>> a [-5, 31, 5, 7, 5, 1]



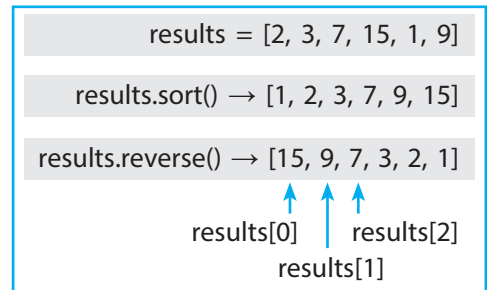
Видалимо зі списку letters всі елементи 'a':

```
letters = ['a', 'd', 'a', 'a', 'b']
while 'a' in letters:
    letters.remove('a')
print(letters) # Буде виведено: ['d', 'b']
```



Надрукуємо три найбільших елементи списку results:

```
results = [2, 3, 7, 15, 1, 9]
results.sort()
results.reverse()
for i in range(3):
    print(results[i])
```



Списки зручно використовувати під час опрацювання рядкових даних. Деякі призначені для цього функції та методи наведено в таблиці:

Функція або метод	Призначення	Приклад
s.split(символ)	Розбиття рядка s на список рядків за заданим символом-роздільником	>>> s = 'Python – це цікаво' >>> a = s.split(' ') >>> a ['Python', '-', 'це', 'цікаво']

Продовження таблиці

Функція або метод	Призначення	Приклад
list(рядок)	Розбиття рядка на список символів	<pre>>>> b = list(s) >>> b ['P', 'y', 't', 'h', 'o', 'n', ' ', '-', ' ', 'ц', 'e', ' ', 'ц', 'i', 'k', 'a', 'v', 'o']</pre>
c.join(список)	Об'єднання елементів списку в рядок із роздільником c між елементами	<pre>>>> a = ['q', 'w', 'e', 'r', 'tttt', 'y'] >>> '-'.join(a) 'q-w-e-r-tttt-y'</pre>

 Запишемо рядок c у зворотному порядку:

```
c = 'Без труда нема плода'
a = list(c)      # Створення списку a з символів рядка c
a.reverse()     # Перестановка елементів списку a у зворотному
                # порядку
c1 = ".join(a)  # Об'єднання списку в рядок із порожнім
                # роздільником
print(c1)      # Надруковано 'адоп амен адурт зеБ'
```

Функції для роботи зі списками

Функції для роботи зі списками не змінюють сам список, а повертають певне значення. Із функціями len() і list() ви вже знайомі. Деякі інші корисні функції описано в таблиці:

Функція	Призначення	Приклад
max(list)	Повертає значення найбільшого елемента	<pre>>>> a = [1, 5, 7, 31, -5] >>> max(a) 31</pre>
min(list)	Повертає значення найменшого елемента	<pre>>>> min(a) -5</pre>
sum(list)	Повертає суму елементів	<pre>>>> sum(a) 39</pre>



Дано список $a = [1, 5, 7, 31, -5]$.

Знайдемо індекс найбільшого елемента в списку:

```
>>> a = [1, 5, 7, 31, -5]
>>> n = a.index(max(a))           # n = 3
```

Видалити будь-який об'єкт, зокрема й елемент списку, дозволяє функція `del`:

```
>>> a = ['зима', 'весна', 'літо', 'осінь']
>>> del(a[2])
>>> a
['зима', 'весна', 'осінь']
```

Таким чином, комбінуючи функції та методи роботи зі списками, ми можемо, як із цеглинок, скласти програму для розв'язування складної задачі.



Питання для самоперевірки

1. Які ви знаєте способи видалення елемента зі списку? Назвіть ці способи.
2. Назвіть два способи перевірити, чи є в списку певний елемент.
3. Наведіть приклад ситуації з повсякденного життя, в якій потрібно було відсортувати певні дані.
4. Дано список $a = [2, -7, -1, 8, 2, 4]$.

Що буде надруковано після виконання наведених фрагментів програми?

- | | |
|--|---|
| <p>а) <code>m = min(a)</code>
 <code>n = a.index(m)</code>
 <code>del(a[n])</code>
 <code>print(a)</code></p> | <p>г) <code>a.sort()</code>
 <code>print(a)</code>
 <code>s = sum(a[1:5])</code>
 <code>print(s)</code></p> |
| <p>б) <code>a.sort()</code>
 <code>a.reverse()</code>
 <code>for i in range(3):</code>
 <code> print(a[i])</code></p> | <p>г) <code>n = a.index(max(a))</code>
 <code>a[0], a[n] = a[n], a[0]</code>
 <code>print(a)</code></p> |
| <p>в) <code>k = a.count(2)</code>
 <code>print(k)</code></p> | |

5. Складіть програму для заміни нулем найбільшого елемента списку з десяти чисел.
6. У фігурному катанні загальна оцінка за виступ обчислюється за такими правилами: відкидаються найвища і найнижча оцінки, а з решти обчислюється середнє арифметичне. Складіть програму для визначення оцінки за цими правилами.

Вправа 36

- Скласти програму опрацювання списку.
Задача. Дано список оцінок, виставлених за проходження тесту. Складіть програму для визначення:
 - а) середнього арифметичного оцінок;
 - б) кількості учнів і учениць, які отримали оцінку, вищу за середній бал.
1. У Python IDLE виберіть команду File → New File.
Створіть список оцінок:
`a = [11, 8, 6, 8, 12, 9, 7, 6]`
 2. Обчисліть s — суму елементів списку a .
 3. Обчисліть середнє арифметичне елементів списку a , збережіть це значення в змінній av .
 4. Виведіть значення середнього арифметичного, округленого до цілого за допомогою функції `round()`:
`print(round(av))`
 5. Запишіть програмний код для визначення кількості елементів списку, більших за значення середнього арифметичного.
 - ⚙️ 6. Збережіть програму у файлі з назвою `Vprava36`.
Випробуйте програму, проаналізуйте результати її роботи. Доповніть програму командами для введення списку з клавіатури та випробуйте її.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106424).



§ 37. ОПРАЦЮВАННЯ ЕЛЕМЕНТІВ СПИСКУ

Методи опрацювання списків у Python надають багато корисних можливостей для роботи зі списками, дозволяючи додавати, видаляти, сортувати та здійснювати інші операції над елементами списку. Але є такі види задач, для розв'язування яких цих методів недостатньо. Наприклад, якщо нам потрібно змінити значення всіх елементів списку за певним правилом або виконати певні дії над елементами списку, які відповідають деякій умові, слід у циклі перебрати всі елементи списку.

Перебір елементів списку

Опрацювання елементів списку зручно виконувати в циклі, де параметр циклу послідовно набуває значень елементів списку або їх індексів.

Розгляньмо це на прикладі виведення списку на екран:

№ з/п	Опис	Приклад
1	Послідовно звертаємося до елементів за їхніми індексами. У такому випадку потрібно знати кількість елементів у списку або визначити її функцією <code>len()</code>	<pre>a = [2, 5, 14, 62, -5] for i in range(len(a)): a[i] += 2 print(a[i])</pre>
2	Перебираємо всі наявні в списку елементи. Застосовується, коли не потрібно змінювати значення елементів списку	<pre>for item in a: print(item)</pre>

Як бачимо, лише перший спосіб — звернення за індексом — дозволяє *змінювати* значення елементів списку.

Опрацювання елементів списку

Розгляньмо алгоритми розв'язування задач, пов'язаних із опрацюванням списків на наведених далі прикладах.

▶ Зміна значень елементів списку

Задача. У ботанічному саду ростуть 3 магнолії, 5 яблунь, 4 груші, 6 каштанів, 2 сливи і 1 персикове дерево. Число дерев кожного виду занесено до списку `tree`.

1) Садівникам треба збільшити кількість дерев кожного виду вдвічі. Як змінити значення елементів списку `tree`?

Алгоритм розв'язування задачі: перебрати елементи списку; змінити значення кожного елемента за заданим правилом.

```
tree = [3, 5, 4, 6, 2, 1].
```

```
for i in range(len(tree)): tree[i] = tree[i]*2
```

Після виконання команди циклу змінна `tree` набуває значення `[6, 10, 8, 12, 4, 2]`.

2) Садівникам треба збільшити на 3 кількість дерев, число яких менше за 4. Як змінити значення елементів списку `tree`?

Алгоритм розв'язування задачі: перебрати елементи списку; якщо черговий елемент або його індекс відповідає деякій умові, змінити його значення за заданим правилом.

```
tree = [3, 5, 4, 6, 2, 1].
```

```
for i in range(len(tree)):
```

```
    if tree[i]<4: tree[i] = tree[i]+3
```

Після виконання команди циклу змінна `tree` набуває значення `[6, 5, 4, 6, 5, 4]`.



▶ Визначення кількості елементів списку із заданою властивістю

3) Як садівники можуть з'ясувати, скільки видів дерев у саду представлено в недостатній кількості — не більше 3 дерев?

Алгоритм розв'язування задачі: надати змінній-лічильнику значення 0; перебрати елементи списку; якщо черговий елемент відповідає деякій умові, то значення лічильника збільшити на 1.

```
tree = [3, 5, 4, 6, 2, 1].
```

```
k = 0; # Лічильник елементів із заданою властивістю
```

```
for i in range(len(tree)):
```

```
    if tree[i] <= 3:
```

```
        k = k+1
```

```
        print(i)
```

```
print ('k = ', k)
```

```
# Якщо дерев i-го виду мало,
```

```
# то збільшити лічильник на 1
```

```
# і вивести індекс елемента
```

```
# Вивести кількість виявлених видів
```

► Знаходження суми елементів із заданою властивістю

4) Садівникам треба знайти загальну кількість дерев, вид яких представлений не менш ніж 4 деревами.

Алгоритм розв'язування задачі: надати змінній для підрахунку суми значення 0; перебрати елементи списку; якщо черговий елемент відповідає умові, додати його значення до значення суми. Опрацюємо елементи списку `tree = [3, 5, 4, 6, 2, 1]`:

```
suma = 0; # Змінна для збереження значення суми
for item in tree
    if item >= 4: suma = suma+item
print('suma = ', suma) # Буде надруковано suma = 15
```



Під час перебору значень елементів списку може перевірятися, чи є значення кратним деякому числу, чи є число додатним або від'ємним, чи потрапляє значення у вказаний числовий проміжок та інші умови залежно від поставленої задачі.

Поділимо на два парні елементи списку `a = [2, 5, 14, 62, -5]`:

```
for i in range(len(a)):
    if a[i]%2 == 0: a[i] = a[i]//2
print(a) # Буде надруковано [1, 5, 7, 31, -5]
```



Під час перебору індексів елементів списку може перевірятися також і значення індексу елемента списку (наприклад, змінюються елементи, що стоять на парних місцях).

Замінімо нулем елементи списку `a = [2, 5, 14, 62, -5]`, що мають парні індекси:

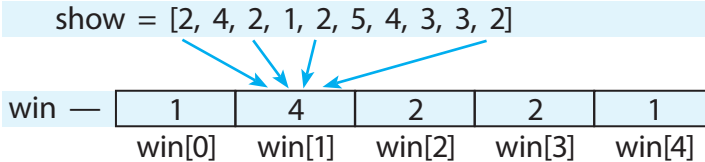
```
for i in range(len(a)):
    if i%2 == 0: a[i] = 0
print(a) # Буде надруковано [0, 5, 0, 62, 0]
```



Складемо програму для розв'язування задачі.

На телевізійному шоу глядацька аудиторія голосує за 5 учасників і учасниць шоу, відправляючи СМС-повідомлення з відповідним номером. Скласти програму для визначення особи, яка переможе в шоу, на основі 10 надісланих СМС-повідомлень.

Для розв'язування задачі створимо список `show` і занесемо в нього номери, що вказані в СМС-повідомленнях. Створимо додатковий список `win` із 5 елементів — за числом учасників і учасниць. В i -му елементі списку `win` будемо зберігати інформацію про те, скільки разів зустрівся певний номер у списку `show` (див. рисунок).



Алгоритм визначення переможця: припустити, що найбільшим є $k = \text{win}[0]$, після чого перебрати елементи списку з індексами від 1 до 4; якщо черговий елемент $\text{win}[i] > k$, то k при своїти $\text{win}[i]$ та запам'ятати індекс i в змінній n .

```
show = [2, 4, 2, 1, 2, 5, 4, 3, 3, 2]
```

```
win = [0, 0, 0, 0, 0]
```

```
for item in show:
```

```
    win[item-1] = win[item-1]+1
```

```
k = win[0]
```

```
n = 0
```

```
for i in range(1,5):
```

```
    if k < win[i]:
```

```
        k = win[i]
```

```
        n = i
```

```
print('Переміг учасник із номером ', n+1)
```



Питання для самоперевірки

- Поясніть відмінності між способами перебору елементів списку:
 - `for i in range(len(a)): print(a[i]);`
 - `for i in a: print(i).`
- Поясніть алгоритм змінення значень усіх елементів списку за певним правилом.

3. На складі зберігається 20 видів товарів. До списку *a* занесено кількість одиниць кожного товару, до списку *b* — ціни цих товарів. Обчисліть загальну вартість товарів на складі.



Вправа 37

- Розробити програму опрацювання списку значень середньодобової температури за один тиждень місяця. Визначити, скільки разів за тиждень температура змінювала знак і в які за порядком дні це відбувалося.

У Python IDLE виберіть команду File → New File.

1. Завантажте модуль для генерування випадкових чисел:
`from random import*`
2. Створіть порожній список *t*.
3. Заповніть список випадковими значеннями в діапазоні від -10 до $+5$. Виведіть утворений список у консоль.
4. Створіть змінну *k* для збереження кількості змін знака температури.
5. Запишіть команду циклу для перебору елементів списку з індексами від 0 до 5. Якщо добуток поточного і наступного елементів від'ємний, потрібно вивести індекси цих елементів і збільшити значення лічильника *k*:

`k = 0`

`for i in range(6):`

`if(t[i]*t[i+1]<0):`

`print(i, ' ', i+1)`

`k += 1`

6. Запишіть команду виведення значення *k*. Збережіть програму у файлі з назвою *Vprava37*. Випробуйте програму. Внесіть зміни, щоб окремо підраховувалося число днів, коли середньодобова температура дорівнювала 0.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106425).



§ 38. СТВОРЕННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

Ви вже маєте досвід написання консольних програм. У сучасних програмах зазвичай використовується графічний інтерфейс, який є наочнішим і зручнішим, ніж консоль.

Графічний інтерфейс — вигляд вікна програми, в якому для взаємодії людини й комп'ютера застосовуються графічні компоненти (вікна, меню, кнопки тощо). Людина має довільний доступ до всіх видимих екранних об'єктів — елементів інтерфейсу.

Ознайомлення з EasyGUI

Швидко створити графічний інтерфейс із використанням вікон виведення повідомлень, кнопочових меню, діалогових вікон та зображень допоможе бібліотека EasyGUI. Її слід установити на комп'ютер із офіційного сайту easygui.sourceforge.net.

Перед викликом функцій із бібліотеки EasyGUI на початку програмного коду слід імпортувати модуль `easygui`:

```
from easygui import*
```

► Вікно повідомлень

Функція `msgbox()` модуля `easygui` створює діалогове вікно з повідомленням і єдиною кнопкою.

Формат функції:

```
msgbox(message[, title, button, image]),
```

де `message` — текст повідомлення; `title` — заголовок діалогового вікна; `button` — напис на кнопці; `image` — ім'я графічного файлу (у форматі GIF) для додавання зображення.

 Створимо діалогове вікно для привітання (рис. 38.1).

```
msgbox('Вітаю!', 'Привітання', 'Привіт!',  
'python.gif')
```

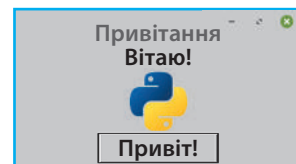


Рис. 38.1

Обов'язковим є тільки параметр `message`.

Якщо певний параметр потрібно пропустити, то значення решти параметрів задавайте за допомогою присвоєнь: параметр = значення.



Створимо вітальне діалогове вікно з малюнком і кнопкою ОК, не вказуючи заголовка вікна.

```
msgbox('Вітаю!', image = 'python.gif')
```

Опрацювання дій користувача

Дослідимо, як віджети з бібліотеки `EasyGUI` реагують на дії користувача. Введемо у вікні консолі команди:

```
>>> from easygui import*
>>> msgbox('Вітаю!')
```

і отримаємо реакцію: 'OK'.

Рядком 'OK' інтерпретатор Python повідомляє, що користувачем/користувачкою натиснуто кнопку ОК.

Функції модуля `easygui` завжди підтримують зворотний зв'язок, інформуючи про дії користувача, — яку кнопку натиснуто, які дані уведено тощо. Інформацію про дії користувача можна зберегти в змінній для подальшого опрацювання.

Збережемо інформацію у змінній `my_click`:

```
>>> my_click = msgbox('Вітаю!')
```

Клацнемо кнопку ОК, щоб закрити діалогове вікно, й у вікні консолі переглянемо значення змінної `my_click`:

```
>>> my_click
'OK'
```

Ми переконалися, що дії користувача — клацання кнопки ОК — зафіксовано в змінній `my_click`. Це дозволяє програмувати реакцію на дії користувача під час використання складніших функцій із бібліотеки `EasyGUI`.

Діалогове вікно з набором кнопок

Функція `buttonbox()` модуля `easygui` створює діалогове вікно з повідомленням і набором кнопок.

Формат функції:

`buttonbox(message, title, choices)`

де `message` — текст повідомлення; `title` — заголовок діалогового вікна; `choices` — список написів для кнопок.



Створимо програму для вибору відповіді на загадку:

```
from easygui import*
```

```
s = 'Фарбоване коромисло над річкою зависло. Що це? '
```

```
puzzle = buttonbox(s, 'Загадка', ['Міст', 'Веселка', 'Електродроти'])
```

```
if (puzzle == 'Веселка'): msgbox ('Молодець! Це '+puzzle, 'Відповідь')
```

```
else: msgbox ('Ні, це не '+puzzle, 'Відповідь')
```

Запустивши програму, побачимо вікно з трьома кнопками (рис. 38.2), а після натискання однієї з них — вікно з відповідним коментарем.

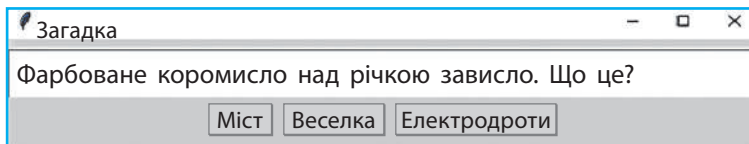


Рис. 38.2

Проаналізуємо хід виконання програми. Замість уведення значення змінної `puzzle` за допомогою оператора `input()` програма пропонує клацнути кнопку. Це клацання сприймається як уведення даних. Текст напису з кнопки (значення типу `string`) зберігається в змінній `puzzle`.

Вікно `choicebox`

Коли потрібно зробити вибір із двох-трьох варіантів, зручно використовувати функцію `buttonbox()`. Але якщо варіантів більше, вікно буде перевантажене кнопками, тому краще скористатися функцією `choicebox()`.

Вікно, створене цією функцією, містить список варіантів. Людина може вибрати один із них або клацнувши його двічі, або клацнувши його в списку й натиснувши кнопку ОК.

Формат функції choicebox():

choicebox (message, title, choices),

де message — текст повідомлення;

title — заголовок діалогового вікна;

choices — список варіантів.



4

Виберемо персонажа для гри за допомогою діалогового вікна зі списком варіантів. Зображення персонажів зберігаються у файлах 1.gif, 2.gif, 3.gif, 4.gif. Присвоїмо змінній вибраний елемент списку:

```
from easygui import*
```

```
msg = "Виберіть персонажа гри"
```

```
title = "Вибір героя"
```

```
choices = ["Птах", "Заєць", "Злюка", "Кажан"] # Список варіантів
```

```
choice = choicebox(msg, title, choices) # Присвоєння змінній choice
```

```
if (choice == "Птах"): image = "1.gif" # вибраного елемента  
# списку
```

```
elif (choice == "Заєць"): image = "2.gif"
```

```
elif (choice == "Кажан"): image = "3.gif"
```

```
else: image = "4.gif"
```

```
msgbox ("Ваш герой "+choice, "Ваш вибір", "ОК", image)
```

Функція choicebox() виводить діалогове вікно зі списком (рис. 38.3). Після натискання кнопки ОК змінна choice набуває значення, вибраного в списку, яке й виводиться функцією msgbox() разом із зображенням (рис. 38.4).

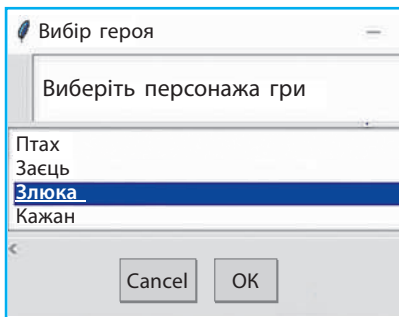


Рис. 38.3

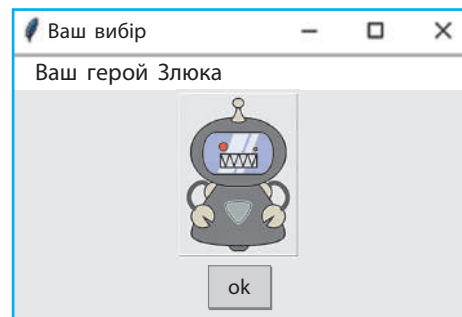


Рис. 38.4

Введення даних

Розглянуті функції дозволяють робити вибір з обмеженої кількості запропонованих варіантів. А як надати можливість уводити власний варіант відповіді? Для цього в бібліотеці EasyGUI є функція `enterbox`, яка відкриває вікно введення:

```
змінна = enterbox(message, title, default),
```

де `message` — текст підказки; `title` — заголовок діалогового вікна; `default` — варіант, що пропонується за замовчуванням.

5 Створимо діалогове вікно без заголовка для введення відповіді; запропонуємо за замовчуванням варіант Ескімо (рис. 38.5).
`favor = enterbox('Твоє улюблене морозиво? ', default = 'Ескімо')`

При запуску програми побачимо, що в полі введення вже є варіант Ескімо. Замість нього можна ввести іншу відповідь, але якщо нас усе влаштовує, слід клацнути кнопку ОК.

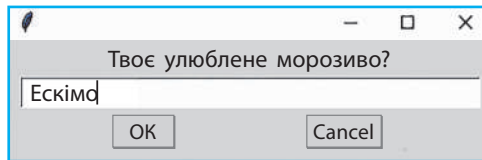


Рис. 38.5

Значення, отримане за допомогою функції `enterbox`, належить до рядкового типу. Якщо потрібно опрацювати числове значення, отриманий рядок слід перетворити на число за допомогою відомих вам функцій `int()` або `float()`.

Як бачимо, можливостей бібліотеки EasyGUI достатньо, щоб зробити, наприклад, програму для перевірки знань із предмета.

● Мову Python використовують для розв'язування багатьох як наукових, так і бізнес-завдань. Завдяки простоті вивчення та наявності додаткових потужних бібліотек Python широко застосовують науковці різних галузей, наприклад, фізики, математики, біологи й інші.



Питання для самоперевірки

1. Як викликати вікно з повідомленням і єдиною кнопкою з модуля EasyGUI?
2. Який вигляд має діалогове вікно, створене з параметрами: `choice = buttonbox('Виберіть рівень', 'Гра', ['1', '2', '3'])`?
3. Який вигляд має діалогове вікно, створене з параметрами: `choice = choicebox('Скільки гравців? ', 'Гра', ['1', '2', '3'])`?
4. Як засобами модуля EasyGUI ввести рядок тесту?
5. Як засобами модуля EasyGUI ввести ціле число?



Вправа 38

- Скласти програму Калькулятор для обчислення суми, різниці та добутку двох уведених чисел.
1. У Python IDLE виберіть команду File → New File.
Запишіть оператор для завантаження модуля `easygui`:
`from easygui import*`
 2. Уведення чисел і обчислення виразу відбуватимуться в циклі `while(rep)`, тобто поки значення змінної `rep` дорівнює `True`.
Задайте початкове значення змінної `rep`:
`rep = True`
 3. У тілі циклу `while(rep)`: введіть два числа за допомогою діалогового вікна `enterbox()` (рис. 38.6):
`while(rep):`
 `n1 = float(enterbox('Введіть перше число: '))`
 `n2 = float(enterbox('Введіть друге число: '))`
 4. Реалізуйте вибір арифметичної дії за допомогою відповідних кнопок діалогового вікна `buttonbox()` (рис. 38.7).

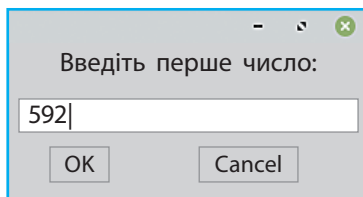


Рис. 38.6

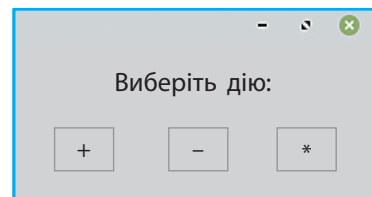


Рис. 38.7

Вибраний елемент списку присвойте змінній `operation`:

```
ch = ['+', '-', '*']
```

```
operation = buttonbox('Виберіть дію: ', choices = ch)
```

Проаналізуйте значення змінної `operation` та організуйте виконання потрібної арифметичної дії в операторі `if...elif...else` і виведення результату (рис. 38.8):

```
if operation == '+':
```

```
    msgbox('Результат '+str(n1+n2))
```

```
elif operation == '-':
```

```
    msgbox('Результат '+str(n1-n2))
```

```
else:
```

```
    msgbox('Результат '+str(n1*n2))
```

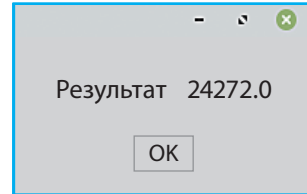


Рис. 38.8

5. Щоб запитати в користувача / користувачки, чи слід повторити обчислення, створіть діалогове вікно з кнопками Так, Ні; вибраний елемент списку присвойте змінній `rovtor`:

```
rovtor = buttonbox('Ще раз?', choices = ['Так', 'Ні'])
```

Якщо змінна `rovtor` набуває значення Ні, змінна `rep` отримує значення `False`, і цикл `while(rep)`: припиняє роботу:

```
if rovtor == 'Ні':
```

```
    rep = False
```

6. До списку написів для кнопок діалогового вікна вибору дії додайте кнопки «/», «//», «%» і запрограмуйте відповідні обчислення.

Запропонуйте власне розширення можливостей калькулятора. Збережіть файл із назвою `Vprava38`.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106426).





ПРАКТИЧНА РОБОТА 10

СТВОРЕННЯ ПРОГРАМ ДЛЯ ОПРАЦЮВАННЯ СПИСКІВ

- Завдання:** скласти програму для розв'язування задачі.
Задача. На автобазі 10 вантажівок. Список [42, 28, 53, 39, 54, 40, 35, 26, 51, 38] містить значення маси вантажів (у тоннах), перевезених за добу. Визначити: а) загальну масу перевезеного вантажу; б) номер у списку автомобіля, що перевіз вантаж найбільшої маси; в) кількість автомобілів, що перевищили денну норму перевезення (50 т).
- Обладнання:** комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виберіть команду File → New File.

1. Створіть змінну `v` типу список і надайте цій змінній значення, що вказані в умові задачі.
2. Запишіть команду присвоєння змінній `s` значення суми елементів списку `v` (скористайтесь функцією `sum(list)`). Запишіть команду для виведення значення змінної `s`.
3. Запишіть команду присвоєння змінній `m` значення найбільшого елемента списку (скористайтесь функцією `max(list)`). Запишіть команду для виведення значення змінної `m`.
4. Запишіть команду присвоєння змінній `n` індексу найбільшого елемента списку (скористайтесь методом `list.index()`). Запишіть команду для виведення значення змінної `n`.
5. Установлено денну норму перевезення 50 т. За алгоритмом визначення кількості елементів із заданою властивістю з'ясуйте, скільки автомобілів перевищили норму.
6. Збережіть файл із назвою `Pract10`.

Зробіть висновок: як опрацювати числові дані, що зберігаються в списках.

§ 39. РОЗВ'ЯЗУВАННЯ ЗАДАЧ МЕТОДОМ ПОДІЛУ НА ПІДЗАДАЧІ

Припустімо, нам потрібно створити малюнок. Спочатку ми маємо визначити, що саме потрібно намалювати (будинок, дерево, сонце тощо), та скласти алгоритм малювання (рис. 39.1).

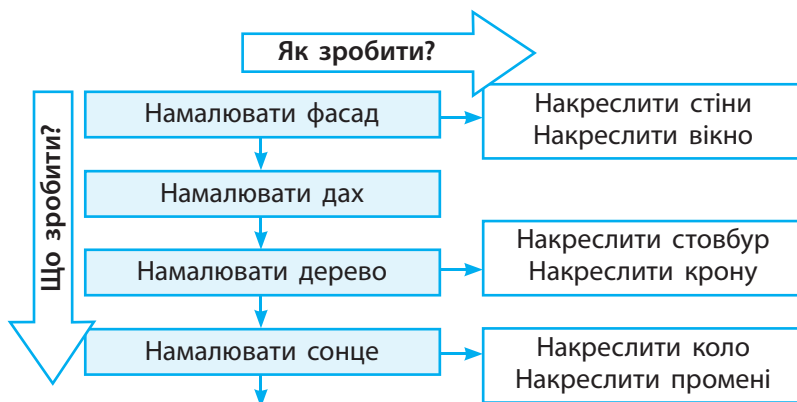


Рис. 39.1

Визначивши основні кроки алгоритму, почнімо планувати дії щодо їх реалізації: як намалювати стіни, дах та ін.

Метод поділу складної задачі на підзадачі

Метод поділу на підзадачі полягає в тому, щоб у ході розроблення алгоритму розв'язування складної задачі виконати такі дії:

- 1) визначити основні кроки, тобто розробити стратегію розв'язування задачі (*що зробити?*);
- 2) розробити шляхи та методи реалізації намічених кроків (*як зробити?*).



Алгоритм розв'язування кожної підзадачі можна оформити як **функцію** — окремий іменованний блок програмного коду (рис. 39.2).

Функції реалізують алгоритми розв'язування підзадач складної задачі. Їх можна порівняти з невеликими програмами, що вбудовані в основну програму (підпрограмами).

Для малювання зображення, наведеного на рис. 39.3, перед основним програмним кодом створюються чотири функції з назвами `draw_wall`, `draw_roof`, `draw_tree`, `draw_sun` для розв'язування підзадач основної задачі. В основній програмі ці функції викликаються послідовно за їхніми назвами.

Функції роблять програмний код коротшим, зручнішим для читання і, головне, — придатним для повторного використання. Наприклад, якщо ми захочемо намалювати ще одне дерево біля будиночку, то повторно викличемо функцію `draw_tree` з іншими координатами початку малювання.

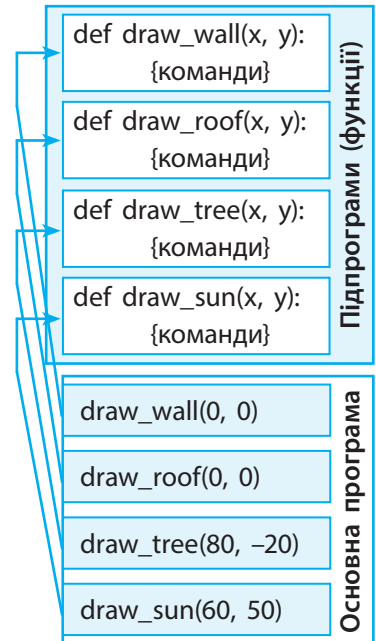


Рис. 39.2

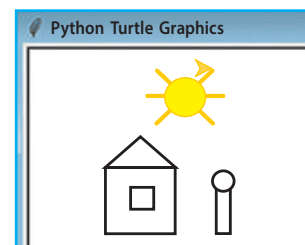


Рис. 39.3

Правила створення і виклику функцій у Python

Ви вже ознайомилися зі стандартними функціями різного призначення. Наприклад, функція `randint()` повертає випадкове число, функція `print()` виводить список значень у вікно консолі, функція `forward()` пересуває Черепашку. Ці функції вбудовані, тобто описані в модулях і бібліотеках мови Python.

У Python є можливість створювати й використовувати власні функції, так звані функції користувача. Згадані раніше функції `draw_roof`, `draw_tree`, `draw_sun` є функціями користувача.

Функція складається з трьох частин: **назви**, **параметрів** і **тіла**.

У Python функції визначаються за допомогою зарезервованого слова `def`.

Опис функції:

```
def <назва функції>(<перелік параметрів>):
    <тіло функції>
```



Тіло функції є послідовністю команд, які будуть виконані після викликання функції.

Ознайомімося з **правилами створення функцій** у Python.

- Блок функції починається з ключового слова `def`, після якого пишуть назву функції й круглі дужки `()`.
- У середині дужок пишуть параметри — назви змінних, які отримують значення під час виклику функції. Їх ще називають *формальними параметрами*. Якщо функції не треба передавати значення, необхідно записати порожні дужки.
- Після дужок ставлять двокрапку і з нового рядка з відступом записують команди тіла функції.

Після того як функцію створено, її можна викликати з іншої функції або з основної програми.

Щоб **викликати функцію**, потрібно:

- 1) увести назву функції й додати дужки;
- 2) у дужках перелічити аргументи функції — значення, які буде присвоєно формальним параметрам.

Якщо формальних параметрів немає, то при виклику після назви функції потрібно писати порожні дужки.

Після виклику виконуються команди тіла функції.

- ! Опис функції повинен міститися вище від виклику функції. Це пояснюється тим, що інтерпретатор зчитує код рядково, і про те, що міститься в рядках, які розташовані далі, йому невідомо.

Якщо виклик функції записано раніше, ніж її опис, то виникає помилка `NameError`.

1 Створимо і викличемо функцію, яка виводить у консоль квадрат числа. Призначення команд у цьому програмному коді пояснено на рис. 39.4.

```
def kv(x):
    print(x*x)
kv(5)
```

→ заголовок функції
→ тіло функції
→ виклик функції

Рис. 39.4

2 Опишемо і викличемо функції для малювання будинку (див. рис. 39.3). Під час виклику функцій у дужках зазначають значення координат x , y , із яких буде починатися малювання:

```
def draw_wall(x, y): # Фасад
    up()
    goto(x, y)
    down()
    for n in range(4): # Стіни
        forward(50)
        right(90)
    up()
    goto(x+15, y-15)
    down()
    for n in range(4): # Вікно
        forward(20)
        right(90)

def draw_roof(x, y): # Дах
    up()
    goto(x, y)
    down()
    left(50)
    forward(40)
    right(100)
    forward(40)
    up()
    left(50)

# Основна програма
width(3)
draw_wall(0, 0)
draw_roof(0, 0)
```

Таким чином, за допомогою функцій можна поділити програму на окремі частини, кожна з яких виконує конкретне завдання.

Функції також надають можливість багаторазово викликати один і той самий код із різних місць програми.





Питання для самоперевірки

1. На які підзадачі можна поділити задачу малювання жабки (рис. 39.5)?
2. Що буде, якщо описану функцію жодного разу не викликати?
3. Напишіть функцію для малювання квадрата. Як її можна використати для малювання квадратів, наведених на рис. 39.6?
4. Чому простіше експериментувати з програмою, в якій виконання підзадач оформлено як функції?
5. У програмному коді (рис. 39.7) назвіть: заголовок функції; тіло функції; команду виклику функції.



Рис. 39.5

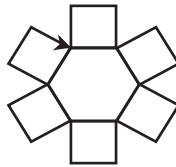


Рис. 39.6

```
def sum(a, b):
    print(a+b)
sum(3, 5)
```

Рис. 39.7

Вправа 39

- Написати програму малювання вулиці з кількома будинками з огорожею (рис. 39.8).

1. У написанні коду малювання вулиці використовуйте функції `draw_wall`, `draw_roof` для малювання будинку.
2. Опишіть функцію `draw_fence(x, y)` для малювання фрагмента огорожі.
3. Напишіть програму малювання вулиці з трьох будинків:



Рис. 39.8

```
from turtle import*
# Додайте сюди описи функцій
width(3)
x = 0
for n in range(3):
    draw_wall(x, 0)
    draw_roof(x, 0)
    x = x+60 # Відстань між початками малювання будинків
```

4. Додайте до програми малювання 20 фрагментів (рис. 39.9).
 $x = -10$
 $y = -50$
 for n in range(20):
 draw_fence (x, y)
 $x = x+10$ # Відстань між фрагментами огорожі
5. Збережіть файл із назвою Vprava39 і випробуйте програму.
6. Поекспериментуйте з кількістю будинків і довжиною огорожі.



Рис. 39.9



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106427).



§ 40. ОБЧИСЛЕННЯ З ВИКОРИСТАННЯМ ФУНКЦІЙ КОРИСТУВАЧА

Ви вже вмiєте викликати функції, передаючи значення для їх формальних параметрів. Розгляньмо, як створити функцію, яка після виклику не лише виконує певні дії, а й повертає на місце виклику певне значення.

Повернення результату виконання функції

Припустимо, що в тілі функції виконуються деякі обчислення. Якщо результат обчислень у подальшому треба використувати в основній програмі, у тілі функції потрібно записати команду: `return <вираз>`.

Команда `return` припиняє виконання функції й повертає значення виразу на місце виклику функції (рис. 40.1).

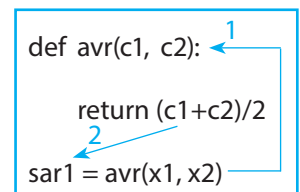


Рис. 40.1

Назву і параметри функції записують у правій частині команди присвоєння або у виразі. Можна також викликати функцію зі списку виведення команди `print()`.



Складемо програму для розв'язування задачі.

Задача. У першому магазині мобільні телефони видів А і Б коштують a_1 , b_1 грн, а в другому — a_2 , b_2 грн. Визначимо, у якому магазині середня ціна телефонів нижча.

```
def avr():
    a = int(input('Скільки коштує телефон А?'))
    b = int(input("Скільки коштує телефон Б?"))
    sar = (a + b)/2
    return sar
sar1 = avr() # Виклик функції avr()
sar2 = avr() # Повторний виклик функції avr()
if sar1<sar2: print('Нижчі ціни у 1-му магазині')
elif sar1>sar2: print('Нижчі ціни у 2-му магазині')
else: print('Ціни однакові')
```



Проаналізуємо програмний код.

- У тілі функції `avr()` уводяться значення локальних змінних a , b ; обчислюється середнє арифметичне значень a , b . Команда `return` повертає значення змінної `sar` в основну програму.
- Змінним `sar1` і `sar2` присвоюються значення, які повертає функція без параметрів `avr()`.
- Порівнюються значення змінних `sar1` і `sar2`.

Область видимості змінних

Не можна використати змінні, створені в тілі функції, після того, як ця функція завершить роботу. Вони існують лише під час її виконання. У таких випадках кажуть, що область видимості змінних обмежена.



Область видимості змінних — та частина програмного коду, у якій змінна доступна для використання.

Змінні, створені всередині тіла функції, є **локальними** — їх «не видно» з інших функцій і з основної програми.

Якщо змінну створено в основній програмі, вона є **глобальною** і її можна використовувати в усіх наступних командах та у викликаних функціях. Глобальна змінна не втрачає свого значення при переході керування до функції і при поверненні керування в основну програму.

Якщо потрібно змінити глобальну змінну всередині функції, то її слід додатково оголосити в тілі функції, використовуючи ключове слово `global` (рис. 40.2).

```
def info():
    global a
    a += 1
    b = 22
a = 44
info()
print('a =', a) # Буде виведено a = 45
```

Рис. 40.2

У програмах, де потрібно підраховувати кількість деяких подій (кількість правильних відповідей у тестовій програмі, кількість влучань в ігровій програмі й ін.), зазвичай в основній програмі створюють глобальну змінну-лічильник. Проте змінюватися значення лічильника має в тілі функції, яка аналізує подію, що відбулася (чи правильна відповідь, чи зроблено вдалий хід тощо). Тому потрібно командою `global <назва глобальної змінної>` дозволити змінювати її значення в тілі функції.

Розгляньмо приклад програми для перевірки знання користувачем/користувачкою таблиці множення.



В основній програмі створено змінну `k` для підрахунку кількості правильних відповідей. У тілі функції значення лічильника збільшується, якщо уведене число дорівнює добутку випадкових чисел `a` і `b`.

```

from random import*
def check(a,b):
    global k
    print(a, '*', b, '=')
    vidp = int(input('Твоя відповідь? '))
    if vidp == a*b: k = k+3
k = 0
for i in range(4):
    a, b = randint(2, 9), randint(2, 9)
    check(a, b)
print ('Оцінка ', k)

```



Алгоритм створення функції в Python

У III ст. до н. е. в трактаті «Начала» давньогрецький математик Евклід описав алгоритм знаходження найбільшого спільного дільника (НСД) двох чисел m і n НСД(m, n): поки $m \neq n$, від більшого числа віднімати менше.

Розгляньмо алгоритм створення функції для обчислення деякого значення на прикладі пошуку НСД двох чисел (рис. 40.3).

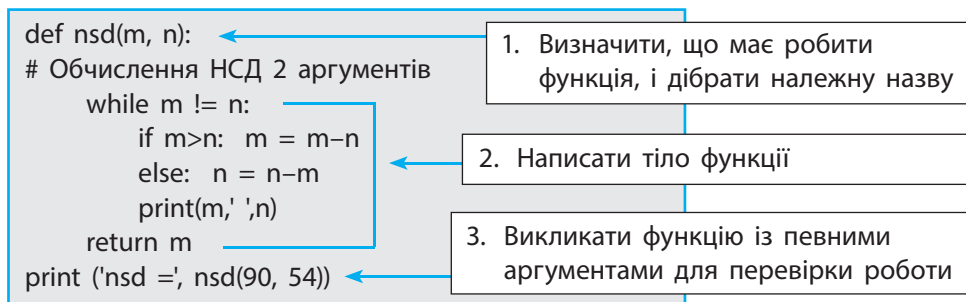


Рис. 40.3

3 Випробуємо програму для $n = 90$; $n = 54$. Результат виконання програми наведено на рис. 40.4.

```

36 54
36 18
18 18
nsd = 18

```

Варто супроводжувати всі функції коментарями.

Рис. 40.4



Питання для самоперевірки

1. Що буде виведено, якщо $x = 123$; $x = 54321$?

```
def f(x):
    k = 0
    while x > 0:
        x = x // 10
        k += 1
    return k
x = int(input('x = ?'))
print(f(x))
```



2. Описано функцію для визначення більшого з двох чисел. Як використати цю функцію для визначення найбільшого з трьох чисел x , y , z ?

```
def m(a, b):
    if a > b: k = a
    else: k = b
    return k
```

3. Проаналізуйте програму для знаходження площі прямокутної пластини розмірами $a \times b$ см із круглим отвором радіуса r см (рис. 40.5). Додайте до програмного коду описи функцій $s(a, b)$, $c(r)$.

```
a = int(input('a = ?'))
b = int(input('b = ?'))
r = int(input('r = ?'))
pl = s(a, b) - c(r)
print("Площа дорівнює ", pl, " кв. см")
```

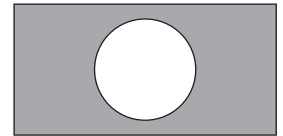


Рис. 40.5

4. Створіть функцію, яка повертає добуток двох чисел.



Вправа 40

- Дано функцію для обчислення найбільшого спільного дільника $\text{nsd}(m, n)$. Користуючись нею, знайти найменше спільне кратне (nsk) чисел a, b, c, d , якщо відомо, що $\text{nsk}(m, n) = m * n // \text{nsd}(m, n)$.

1. На рис. 40.3 (с. 243) описано функцію $\text{nsd}(m, n)$ для обчислення НСД чисел m, n . Запишіть у вікні програми код функції.

2. Запишіть у кодї основної програми команди для введення значень змінних a , b .
Змінній nab надайте значення НСК чисел a , b :
 $nab = a*b//nsd(a, b)$
3. Запишіть у кодї програми команду для введення значень змінних c , d . Змінній $nabc$ надайте значення НСК чисел nab , c :
 $nabc = nab*c//nsd(nab, c)$.
4. Змінній $nabcd$ надайте значення НСК чисел $nabc$, d .
5. Виведіть отримане значення $nabcd$.
6. Збережіть файл із назвою `Vprava40`. Випробуйте програму:
 - а) для набору чисел 25, 5, 10, 2;
 - б) для власного набору чисел.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/106428).



ПРАКТИЧНА РОБОТА 11 СТВОРЕННЯ ПРОГРАМИ З ВИКОРИСТАННЯМ ФУНКЦІЙ КОРИСТУВАЧА

- Завдання:** скласти програму для розв'язування задачі.
Задача. На стадіоні місця класу А коштують a грн, місця класу В — b грн, а місця класу С — c грн. Обчислити загальну суму прибутку від продажу квитків.
- Обладнання:** комп'ютер із середовищем програмування мовою Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виберіть команду File → New File.

1. Запишіть команди для введення значень змінних a , b , c .
2. Опишіть функцію `kvitok(level, v)`, яка запитує, скільки продано квитків класу $level$, і повертає суму коштів, отриманих від продажу квитків цього класу.

```
def kvitok(level, v):
    k = int(input('Скільки продано квитків класу '+level))
    suma = v*k
    return suma
```

3. Для обчислення суми коштів від продажу квитків класу A функцію `kvitok(level, v)` слід викликати з фактичними параметрами $'A'$, a , де $'A'$ — позначення класу; a — вартість квитків цього класу.

```
s_a = kvitok('A', a)
```

4. Запишіть команди виклику функції `kvitok` для обчислення сум коштів від продажу квитків класів B , C .
5. В основній програмі обчисліть загальну суму, отриману від продажу квитків: $suma = s_a + s_b + s_c$
6. Запишіть команду виведення значення загальної суми.
7. Збережіть програму у файлі `Pract11`. Випробуйте роботу програми для різних початкових значень. Приклад результату виконання програми наведено на рисунку.

```
Вартість квитка класу A: 100
Вартість квитка класу B: 150
Вартість квитка класу C: 200
Скільки продано квитків класу A 500
Скільки продано квитків класу B 300
Скільки продано квитків класу C 200
Загальна сума 135000 грн
```

Зробіть висновок: як створювати функції для підзадач основної задачі.

РОЗДІЛ 6

ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Практична робота 12. Створення анімаційного оформлення для відео

Практична робота 13. Моделювання роботи автозаправної станції

Практична робота 14. Розробка інформаційної моделі. Реалізація математичної моделі задачі в середовищі програмування

Практична робота 15. Складання програми для обчислення чисел Фібоначчі

Практична робота 16. Опрацювання рядкових даних

Практична робота 17. Створення програми з розгалуженнями і повтореннями

Практична робота 18. Розробка застосунку «Відгадай загадку»

ПОВТОРЮЄМО



Ви вже здобули певні знання та навички:

- навчилися користуватися електронною поштою, дізналися, що таке хмарні технології, ознайомилися з роботою деяких інших сервісів; з'ясували, що таке штучний інтелект і як його застосовують;
- ознайомилися з редактором двовимірної анімації та навчилися розробляти за його допомогою прості мальовані анімації;
- продовжили роботу зі створення інформаційних моделей, оволоділи новими засобами моделювання;
- продовжили роботу з табличним процесором, дізналися про нові способи його використання; навчилися проводити обчислювальні експерименти;
- продовжили вивчення мови програмування Python, навчилися працювати з рядками та списками, використовувати метод покрокової деталізації для розв'язування складних задач.

1. Що таке хмарний диск і як його відкрити?
2. Для чого складають сценарій анімаційного фільму?
3. На яких уроках і як ви використовуєте моделі об'єктів?
4. Що можуть містити клітинки електронної таблиці?
5. Як організувати діалог користувача з програмою?



У цьому розділі ви створите інформаційні продукти, користуючись інструментами програмних середовищ, із якими ознайомилися в 7 класі.



ПРАКТИЧНА РОБОТА 12

СТВОРЕННЯ АНІМАЦІЙНОГО ОФОРМЛЕННЯ ДЛЯ ВІДЕО

Завдання: створити анімацію з номером учасника/учасниці конкурсу (рис. 1) для відеофільму про шкільний конкурс ерудитів.

Обладнання: комп'ютер із редактором двовимірної анімації.



Рис. 1

► Теоретична частина

Переглядаючи різні телепрограми, можна помітити, що іноді на відео накладають написи або малюнки: таблички з прізвищами, номери учасниць/учасників конкурсу тощо. Ці елементи можуть бути анімаційними.

Щоб підготувати таке оформлення для відео за допомогою редактора анімації TurbTube, потрібно створити анімацію та експортувати її як послідовність зображень (наприклад, формату PNG) із прозорим тлом. Потім за допомогою спеціальної програми (відеоредактора) таку послідовність зображень можна накласти на інше відео.

Плануючи сценарій, слід урахувати, що в TurbTube один об'єкт не може бути задіяний більш, ніж в одній автоматичній анімації певного типу. Тому, наприклад, якщо об'єкт має спочатку збільшитися, а через деякий час зменшитися то другу анімацію масштабування доведеться застосувати не до того об'єкта, який збільшувався, а до його копії.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть редактор двовимірної анімації TurbTube. Створіть новий проєкт: розміри кадру — 720×720 ; частота кадрів — 25 кадр/с. Збережіть проєкт із назвою Номер.

2. В анімації використаємо два об'єкти: фігуру неправильної форми на тлі (далі — фігура) і число.

Сценарій анімації може бути такий:

Тривалість (кадри)	На шарі Фігура		На шарі Число
0,5 с (1–12)	Обертається перший примірник фігури	Зростає щільність фігури	
0,5 с (13–25)			Збільшується розмір першого примірника числа
3 с (26–100)			
0,5 с (101–112)			Зменшується розмір другого примірника числа
0,5 с (113–125)	Продовжує обертання другий примірник фігури	Спадає щільність фігури	

3. Назвіть наявний на панелі Розкадрування шар Фігура. Намалюйте на ньому фігуру — тло і додайте до бібліотеки.

4. Додайте до фігури анімаційні ефекти:

- обертання з такими параметрами: назва — Обертання1, кінцевий кадр — 100, швидкість — 3,6 градусів/кадр;
- анімацію щільності з такими параметрами: назва — Щільність1; початковий кадр — 1; кінцевий кадр — 100; початкова щільність — 0; кінцева щільність — 1; ітерацій — 12.

5. На *100-му кадрі* вставте з бібліотеки другий примірник фігури. Розташуйте його так само, як, розташована на цьому кадрі перша фігура.

6. Додайте до другого примірника фігури анімаційні ефекти:

- обертання з такими параметрами: назва — Обертання2; кінцевий кадр — 125; швидкість — 3,6 градусів/кадр;

- анімацію щільності з такими параметрами: назва — Щільність2; початковий кадр — 112; кінцевий кадр — 125; початкова щільність — 1; кінцева щільність — 0; ітерацій — 14.
- У налаштуваннях анімаційного ефекту Обертання1 змініть номер кінцевого кадру на 99. (Тепер на *100-му кадрі* залишився тільки другий примірник фігури.)
 - Додайте шар із назвою Число. На *кадрі 13* цього шару намалюйте число малого розміру, згрупуйте його частини і розташуйте в центрі кадру (рис. 2). Додайте число до бібліотеки під назвою Номер.
 - Додайте до числа анімацію масштабування із такими параметрами: назва — Збільшення; кінцевий кадр — 100; коефіцієнт масштабування — 1,25; число ітерацій — 13. (Завдяки цьому протягом 13 кадрів число збільшиться і залишатиметься на екрані до *100-го кадру* включно.)
Перегляньте анімацію і за потреби доберіть коефіцієнт масштабування, щоб на *25 кадрі* число не було надто великим або малим.
 - На *100-му кадрі* вставте з бібліотеки ще один об'єкт Номер, збільште до розмірів першого примірника. Для анімаційного ефекту Збільшення замініть номер кінцевого кадру на 99.
 - Налаштуйте зменшення другого примірника числа на *кадрах 101–112*.
 - Експортуйте анімацію в послідовність зображень формату PNG (префікс назви файлу — номер_4; установіть прапорець Увімкнути прозорість тла).
Перегляньте результат — послідовність кадрів із прозорістю, збережених у окремих файлах.

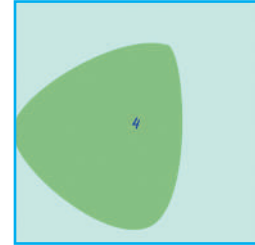


Рис. 2

Зробіть висновок: як забезпечити участь об'єкта послідовно в декількох анімаційних ефектах одного типу; як використання шарів спрощує маніпуляції з об'єктами.



ПРАКТИЧНА РОБОТА 13 МОДЕЛЮВАННЯ РОБОТИ АВТОЗАПРАВНОЇ СТАНЦІЇ

Завдання: створити електронну таблицю, що моделює роботу автозаправної станції: поповнення та витрати пального різних видів, облік коштів.

Обладнання: комп'ютер із табличним процесором.

► Теоретична частина

Розгляньмо таку модель автозаправної станції (АЗС) (рис. 1). Спочатку в касі є S_0 грн, а в цистерні — V_0 л пального. Час від часу відбуваються такі події:

- купівля (К) об'єму пального V_k л, за яке сплачують за гуртовою ціною P_k грн/л;
- продаж (П), тобто заправлення автомобіля об'ємом пального V_p за роздрібною ціною P_p грн/л.



Рис. 1

Після кожної події сума в касі змінюється. Якщо в касі недостатньо коштів для купівлі пального або не вистачає пального, щоб заправити черговий автомобіль, оператор отримує повідомлення.

► Математична модель

Що дано? S_0, V_0 ; для кожної операції: V_k, P_k або V_p, P_p

Що треба знайти? S, V — суму та об'єм пального після кожної операції

Який зв'язок між величинами?

Після купівлі: $S = S_0 - V_k P_k$; $V = V_0 + V_k$;

після продажу: $S = S_0 + V_p P_p$; $V = V_0 - V_p$

Які є обмеження на дані? $V_k > 0$; $P_k > 0$; $V_p > 0$; $P_p > 0$;

$S_0 \geq V_k P_k$; $V_0 \geq V_p$

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

- Відкрийте табличний процесор і створіть електронну таблицю (ЕТ) одним із відомих вам способів.
У перші 5 рядків таблиці уведіть написи відповідно до рис. 2 (с. 254).
- Уведіть в ЕТ такі початкові дані:
 - у клітинку А6 — значення грошової суми, наявної в касі АЗС;
 - у клітинку Н6 — об'єм бензину, наявний на АЗС.
- У клітинки С7:F7 уведіть зразок даних про операцію купівлі/продажу (у стовпці Операція літера П означає «продаж», літера К — «купівля»).
- Значення грошової суми в клітинці G7 має бути від'ємним у разі купівлі, і додатним у разі продажу. Тому для його розрахунку перевірте значення клітинки D7 функцією IF:
=IF(D7="П";E7*F7;-E7*F7)
- Так само залишок палива (клітинка H7) після купівлі має збільшитися, а після продажу — зменшитися на величину об'єму з клітинки F7:
=IF(D7="К";H6+F7;H6-F7)
- У клітинку A7 уведіть формулу для розрахунку суми в касі після купівлі або продажу:
=A6+G7
- Скопіюйте формули з клітинок A7, G7 і H7 у наступні 50 рядків.
Після копіювання формул у деяких клітинках з'явилися значення, що не становлять практичного інтересу (нулі у стовпці G, повтори результатів останньої операції в стовпцях A і H).
Приховайте такі значення за допомогою умовного форматування (див. п. 8–10).
- Виділіть клітинки з формулами в стовпці G і задайте умовний формат для значення 0 — світло-сірий колір тексту на рис. 2.

9. Після кожної операції сума в касі має змінюватися. Використаймо це: для клітинки A7 задайте умовний формат для значення, що дорівнює значенню клітинки A6 — світло-сірий колір тексту на рис. 2.
10. Скопіюйте формат клітинки A7 і застосуйте до наступних клітинок стовпця A. Цей формат застосуйте також до клітинок стовпця H, починаючи з H7.
11. Оскільки не можна продати більше пального, ніж є на АЗС, для клітинки F7 задайте умовний формат за формулою $\text{AND}(D7="П";F7>H7)$ — червоне тло клітинки. Застосуйте цей формат до наступних клітинок цього стовпця.
12. Оскільки не можна витратити більше грошей, ніж є в касі, налаштуйте умовне форматування клітинок із формулами стовпця G, подібно до клітинки F7 (див. п. 11). Збережіть таблицю у файл із назвою АЗС. Перевірте таблицю, вводючи дані про різні операції.

	A	B	C	D	E	F	G	H
1	Автозаправна станція «Автомото-плюс»							
2	Дата		11.02.24					
3								
4	Каса	Бензин						
5		Час	Операція	Ціна, грн/л	Об'єм, л	Сума, грн	Залишок, л	
6	50 000,00 грн	*	*	*	*	*	5000	
7	55 000,00 грн	08:00:00	П	50	100	5 000,00 грн	4 900	
8	15 000,00 грн	08:15:00	К	20	2000	-40 000,00 грн	6 900	
9	22 500,00 грн	08:22:00	П	50	150	7 500,00 грн	6 750	
10	22 500,00 грн					0,00 грн	6 750	
11	22 500,00 грн					0,00 грн	6 750	

Рис. 2

Зробіть висновок: як за допомогою електронної таблиці можна моделювати роботу торговельних підприємств; які засоби допомагають уникнути помилок під час уведення даних.



ПРАКТИЧНА РОБОТА 14

РОЗРОБКА ІНФОРМАЦІЙНОЇ МОДЕЛІ. РЕАЛІЗАЦІЯ МАТЕМАТИЧНОЇ МОДЕЛІ ЗАДАЧІ В СЕРЕДОВИЩІ ПРОГРАМУВАННЯ

- Завдання:** скласти програму для розв'язування задачі.
Задача. Діти вирішили облаштувати на шкільному подвір'ї круглу клумбу діаметром d м. Відомо, що на 1 м^2 потрібно посіяти $1,5$ г насіння квітів. Скільки грамів насіння квітів знадобиться, щоб засіяти створену клумбу? Скільки коштуватиме насіння, якщо вартість 1 г становить 71 грн 60 к?
Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Постановка задачі. Дано діаметр круга, вагу насіння, необхідну для засіювання насінням 1 м^2 . Знайдіть площу круга, масу насіння для посіву на цій площі та вартість насіння.
2. Побудова математичної моделі

<i>Що дано?</i>	d — діаметр круга, м
<i>Що треба знайти?</i>	s — площа круга, м^2 ; $masa$ — загальна маса насіння, г; $vart$ — вартість насіння, грн

<i>Який зв'язок між величинами?</i>	$s = \pi \frac{d^2}{4}$; $masa = s \cdot 1.5$; $vart = masa \cdot 71.6$
-------------------------------------	---

Які є обмеження на дані? $d > 0$

3. Розробіть блок-схему алгоритму розв'язування задачі.
4. Завантажте програму Python → IDLE і створіть нове вікно програми.

Реалізуйте алгоритм мовою Python:

```
d = float(input('d = ?'))
s = 3.14*d**2/4
masa = s*1.5
vart = masa*71.6
print('Потрібно ', round(masa, 2), ' г')
print('Вартість насіння ', round(vart, 2), ' грн')
```

5. Аналіз результатів. Перевірте роботу програми, якщо $d=3$ м (рис. 1).

```
d = ? 3
Потрібно 10.6 г
Вартість насіння 758.78 грн
```

Рис. 1

- Оцініть повноту та вірогідність результатів.
- Знайдіть в інтернеті інформацію про види однорічних квітів. Знайдіть відповіді на запитання: які квіти вибрати, щоб клумба була квітучою і різнобарвною протягом 3 літніх місяців; скільки насіння потрібно, щоб засіяти 1 м^2 ?
 - Запустіть редактор карт знань. Створіть карту знань «Квіткова клумба», у якій відобразить властивості квітів.
 - У центральний вузол уведіть текст «Квіти для клумби».
 - Додайте до центрального вузла підлеглі вузли за кількістю квітів, інформацію про які ви знайшли, упишіть їхні назви.
 - Додайте до кожного вузла 1-го рівня підлеглі вузли, що відображають властивості квітки, наведені на рис. 2.



Рис. 2

Зробіть висновок: як створити інформаційну модель об'єкта.



ПРАКТИЧНА РОБОТА 15

СКЛАДАННЯ ПРОГРАМИ ДЛЯ ОБЧИСЛЕННЯ ЧИСЕЛ ФІБОНАЧЧІ

Завдання: скласти програму для розрахунку N -го числа Фібоначчі.

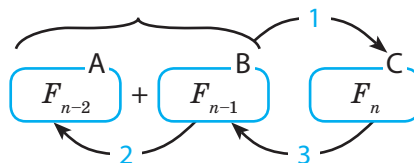
Обладнання: комп'ютер із середовищем програмування Python.

▶ Теоретична частина

Леонардо Пізанський, відоміший як Фібоначчі, — це італійський математик XIII ст., автор математичних трактатів, завдяки яким Європа довідалася про вигадану індійцями позиційну систему числення, відому зараз як арабські цифри. *Числами Фібоначчі* називають послідовність чисел, які обчислюють за таким правилом: $F_1 = F_2 = 1$; $F_n = F_{n-1} + F_{n-2}$. Інакше кажучи, перші два члени послідовності — одиниці, а кожний наступний — сума значень двох попередніх чисел: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55...

При діленні числа з ряду Фібоначчі на попереднє частка зі збільшенням номера числа наближається до так званого *золотого перетину* ($\approx 1,618$; таке співвідношення розмірів вважають естетично привабливим). Для обчислення n -го числа виконується така послідовність дій (див. рисунок):

- 1) виділяються змінні a і b для зберігання двох поточних чисел Фібоначчі F_{n-1} і F_{n-2} ;
- 2) сума чисел $a+b$ заноситься в змінну c (на рисунку — 1);
- 3) на наступній ітерації циклу:
 $b = F_{n-1}$ стає $(n-2)$ -м членом ряду, тому $A=B$ (на рисунку — 2);
 $c = F_n$ стає $(n-1)$ -м членом, тому $B=C$ (на рисунку — 3).



Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте IDLE Python і створіть файл із назвою Pract15. Запишіть команду введення значення змінної n для збереження номера шуканого числа:

```
n = int(input('Уведіть номер числа:'))
```

2. Запишіть програмний код, що реалізує алгоритм обчислення n -го числа.

```
a = b = 1 # Перші два числа послідовності
          # дорівнюють 1
for i in range(3, n+1): # F1 = F2 = 1, тому обчислення
                       # починається з n = 3.
    c = a+b           # i-те число дорівнює сумі двох
                     # попередніх чисел
    a = b             # b стає (i-2)-м членом ряду
    b = c             # c стає (i-1)-м членом ряду
```

3. Запишіть команду виведення значення c після завершення роботи циклу.

4. Знайдіть числа Фібоначчі з номерами 10; 20; 30.

5. Запишіть програму для визначення номера першого числа Фібоначчі, більшого за 1000. Кількість повторень циклу невідома, тому використовуємо цикл із умовою:

```
a = b = 1 # Перші два числа послідовності дорівнюють 1
n = 2
while c < 1000:
    n = n+1 # Номер числа збільшується на 1
    c = a+b # n-е число дорівнює сумі двох попередніх
           # чисел
    a = b   # b стає (i-2)-м членом ряду
    b = c   # c стає (i-1)-м членом ряду
```

6. Запишіть команду виведення значення n після завершення роботи циклу. Випробуйте програму.

Зробіть висновок: як виконуються циклічні алгоритми для обчислення чисел у певній послідовності.



ПРАКТИЧНА РОБОТА 16 ОПРАЦЮВАННЯ РЯДКОВИХ ДАНИХ

Завдання: скласти програму шифрування повідомлення шифром Цезаря.

Обладнання: комп'ютер із середовищем програмування Python.

► Теоретична частина

Давньоримський імператор Юлій Цезар був першим, про кого відомо, що він придумав власне шифр. Абетка розміщується на колі за годинниковою стрілкою (при цьому в українському алфавіті після А йде Б, а після Я — А (рис. 1)). Для шифрування кожна літера в тексті замінюється літерою, віддаленою від неї по колу на задане число позицій за годинниковою стрілкою.



Рис. 1

Отже, якщо вибрано зсув на 3, то замість i -тої використовується $(i + 3)$ -тя літера. Розшифрування виконується у зворотному порядку: беруть літеру, розміщену на задане число позицій проти годинникової стрілки.

На початку шифрування вибирають значення зсуву — ціле число k . Для того щоб імітувати зсув по колу на k , літера з номером n замінюється на літеру з номером new_n , що обчислюється за формулою $new_n = (n + k) \% len(alphabet)$, де $alphabet$ — рядок, що містить всі літери алфавіту.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму Python → IDLE і створіть нове вікно.
2. Створіть змінну `alphabet`, у якій зберігатиметься рядок, що містить літери українського алфавіту й апостроф.
`alphabet = "абвгдежзиййклмнопрстуфхцчшщъя"`

3. Програма запитує, яку дію слід виконати — зашифрувати чи розшифрувати повідомлення.
Запишіть команду введення значення для змінної `vidp`:
`vidp = input('Зашифрувати - 1, розшифрувати - 2 ')`
4. Шифрування відкритого повідомлення оформте у вигляді функції `shifr()`, а розшифровування — `deshifr()`.
Запишіть команду розгалуження, яка викликає потрібну функцію залежно від вибору користувача / користувачки:
`if vidp == '1': shifr()`
`else: deshifr()`
5. Опишіть функцію `shifr()`. Створіть потрібні змінні:
`code = " # Змінна для збереження зашифрованого тексту`
`vidkr = input('Уведіть відкрите повідомлення: ')`
`k = int(input('Уведіть значення зсуву: '))`
6. Запишіть команду циклу для заміни символів рядка `vidkr`:
`for bukva in vidkr: # Переглядаються символи рядка vidkr`
`if bukva in alphabet: # Якщо символ є в рядку alphabet,`
`n = alphabet.find(bukva) # обчислюється індекс`
`# символу в рядку alphabet`
`new_k = (n + k) % len(alphabet) # до індексу додається`
`# значення зсуву`
`code += alphabet[new_k] # до рядка code додається`
`# символ alphabet[new_k]`
`else: # Якщо символ відсутній у рядку alphabet,`
`code += bukva # символ додається до рядка code`
`# без змін`
7. Запишіть команду виведення зашифрованого повідомлення.
8. Опишіть функцію `deshifr()`. Для розшифрування до програми:
 - уводиться значення змінної `code` (зашифроване повідомлення);
 - створюється змінна `vidkr` для збереження розшифрованого;
 - переглядаються символи рядка `vidkr`; якщо символ є в рядку `alphabet`, визначається індекс символу в рядку `alphabet` та обчислюється нове значення індексу за формулою `new_k = (n-k)%len(alphabet)` (після цього символ `alphabet[new_k]`

додається до рядка `vidkr`; якщо символ відсутній у рядку `alphabet`, він додається до `vidkr` без змін);

- виводиться значення змінної `vidkr` (розшифроване повідомлення).

9. Збережіть файл із назвою `React16`.

10. Випробуйте програму, проаналізуйте її роботу для різних значень зсуву (рис. 2).

```
Шифрувати – 1, розшифрувати – 2: 1
Уведіть відкрите повідомлення: повідомляю тобі таємницю
Уведіть значення зсуву: 3
тсджкжспоба хчгк хгипрйща
>>>
Шифрувати – 1, розшифрувати – 2: 2
Уведіть зашифроване повідомлення: тсджкжспоба хчгк хгипрйща
Уведіть значення зсуву: 3
повідомляю тобі таємницю
```

Рис. 2

Зробіть висновок: як організувати опрацювання даних рядкового типу.



ПРАКТИЧНА РОБОТА 17 СТВОРЕННЯ ПРОГРАМИ З РОЗГАЛУЖЕННЯМИ І ПОВТОРЕННЯМИ

Завдання: скласти програму для розв'язування задачі.
Задача. Вкладниця поклала до банку S грн за умови місячної відсоткової ставки 5% . Починаючи з 4-го місяця, відсоткова ставка збільшилася до $6,5\%$. Яка сума буде на рахунку через n місяців? Через скільки місяців сума внеску подвоїться?

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Проаналізуйте математичну модель задачі.

<i>Що дано?</i>	v_0 — початкова сума внеску; n — кількість місяців; p — величина відсоткової ставки.
<i>Що треба знайти?</i>	v_n — величина внеску через n місяців.
<i>Який зв'язок між величинами?</i>	$v_n = v_{n-1} + v_{n-1} \frac{p}{100}$.

Завантажте програму Python → IDLE і створіть нове вікно.

2. Запишіть команду введення значення змінної vn :
`vn = int(input('Уведіть початкову суму внеску: '))`
3. Запишіть команду введення значення змінної n .
4. Запишіть команду циклу для обчислення значення внеску через n місяців з урахуванням того, що, починаючи з 4-го місяця, ставка збільшилася до 6,5% :
`for i in range(n):`
 `if i<3: p = 5`
 `else: p = 6.5`
 `vn = vn+vn*p/100`
5. Запишіть команду виведення значення змінної vn .
6. Збережіть файл із назвою Pract17.
7. Випробуйте програму для $n = 24$, $vn = 1000$. Проаналізуйте результат.
8. Внесіть зміни до програмного коду з тим, щоб з'ясувати, за якого значення n сума внеску на рахунку подвоїться.
9. Запишіть команду виведення значення змінної n .
10. Випробуйте програму для $vn = 1000$.
 Проаналізуйте результат.

Зробіть висновок як створити програму з повтореннями та розгалуженнями всередині циклу.



ПРАКТИЧНА РОБОТА 18

РОЗРОБКА ЗАСТОСУНКУ «ВІДГАДАЙ ЗАГАДКУ»

Завдання: скласти програму, яка пропонує користувачеві / користувачці розгадати кілька загадок. Програма повинна по чергово задавати загадки, пропонувати варіанти та перевіряти правильність відповідей.

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

У Python IDLE виберіть команду File → New File.

1. Запишіть команду завантаження модуля `easygui`:
`from easygui import*`
2. Створіть список `riddles`, елементами якого є тексти загадок:
`riddles = ['Фарбоване коромисло над річкою зависло. Що це?',
'Біла рілля, чорне насіння. Хто його вміє, той і розуміє.
Хто його знає, той і позбирає',
'Чорний Іван, дерев'яний каптан: де носом проведе, там
помітку кладе',
'І червона, й соковита, та гірка вона все літо. Припече
мороз – вона стала добра і смачна.']`
3. Створіть список `answer`, елементами якого є відповіді на загадки:
`answer=['Веселка', 'Книжка', 'Олівець', 'Калина']`
4. Знайдіть в інтернеті зображення, що відповідають змісту загадок. Збережіть їх у файлах у форматі GIF у вашій папці. Пам'ятайте, що графічні файли повинні міститися в одній папці з файлом програми.
5. Створіть список `pictures`, елементами якого є назви графічних файлів:
`pictures=['Веселка.GIF', 'Книжка.GIF', 'Олівець.GIF', 'Калина.GIF']`
6. Створіть змінну — лічильник правильних відповідей: `bal = 0`.

7. У списку 4 загадки, тому в циклі `for i in range(4)`: запрограмуйте такі дії:
 - створити діалогове вікно `buttonbox`, у якому виводиться текст загадки та пропонуються варіанти відповіді;
 - присвоїти змінній `puzzle` значення, що повертає функція `buttonbox`;
 - перевірити, чи дорівнює значення змінної `puzzle` відповідному елементу списку `answer` і створити діалогове вікно з повідомленням про правильність відповіді.

```
for i in range(4):
    puzzle = buttonbox(riddles[i], 'Загадка', answer)
    if(puzzle == answer[i]):
        msgbox('Молодець! Це ' + puzzle, 'Ваша відповідь', 'ok', pictures[i])
        bal = bal+1
    else:
        msgbox('Ні, це не ' + puzzle, 'Ваша відповідь')
```
8. У змінній `message` збережіть текст повідомлення про результат:


```
message = 'Правильних відповідей – ' + str(bal)
```
9. Створіть діалогове вікно для виведення повідомлення `message`:


```
msgbox(message, 'Твій результат', 'OK')
```
10. Додайте до списків тексти ще декількох загадок і відповідей на них. Внесіть до програмного коду відповідні зміни.
11. Збережіть файл із назвою `Pract18`.
12. Випробуйте програму. Проаналізуйте результати (рис. 1–3).

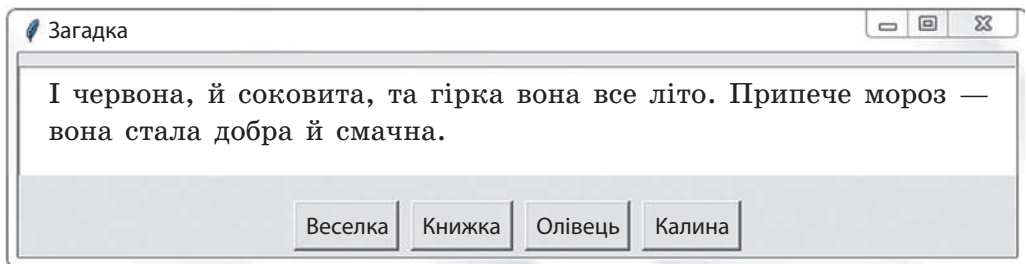


Рис. 1

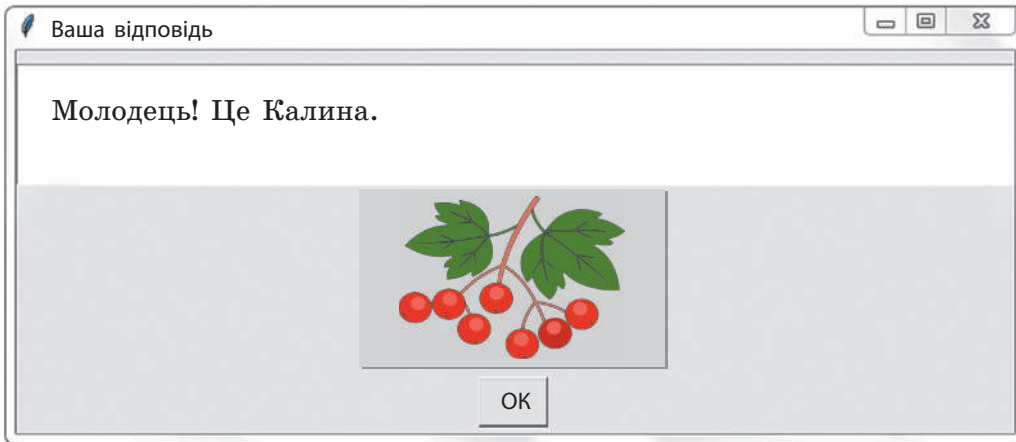


Рис. 2

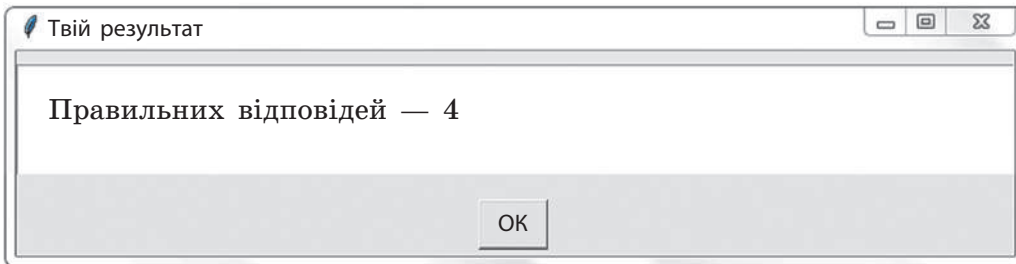


Рис. 3

Зробіть висновок: як розробити програму, яка пропонує варіанти відповідей та перевіряє їх правильність.

КОМП'ЮТЕРНИЙ СЛОВНИК

Величина — інформаційний об'єкт, основними характеристиками якого є назва, вид, тип і значення.

Вкладений цикл — цикл, який міститься в тілі іншого циклу.

Гіперпосилання — адреса ресурсу, пов'язана з елементом сторінки, клацання якого приводить до переходу до цього ресурсу.

Діапазон клітинок (зв'язний) — прямокутна частина таблиці, яка знаходиться на перетині декількох стовпців та рядків.

Електронна пошта — служба обміну повідомленнями через комп'ютерну мережу з використанням поштової системи адресації.

Електронна таблиця — комп'ютерне подання прямокутної таблиці, клітинки якої можуть містити дані або формули опрацювання даних.

Експертна система — програма, що допомагає людині приймати рішення в певній галузі на основі збереженої в базі знань інформації, отриманої від фахівців (експертів).

Змінна — величина, значення якої може змінюватися в ході виконання програми.

Значення — дані, що зберігає величина; може змінюватися під час виконання програми.

Інтерфейс користувача — сукупність засобів операційної системи для взаємодії з людиною.

Карта знань — інформаційна модель, подана у структурній формі, яка наочно відображає взаємозв'язки між об'єктами та явищами деякої предметної галузі.

Комп'ютерна модель — інформаційна модель, реалізована за допомогою певного програмного середовища.

Комп'ютерний експеримент — дослідження математичної моделі з використанням комп'ютера, за якого за одними параметрами моделі обчислюють інші й на цій основі роблять висновки про властивості об'єкта, описані математичною моделлю.

Константа — величина, значення якої не може змінюватися в ході виконання програми.

Модель — це спрощене подання об'єкта у вигляді таблиці, схеми тощо.

Моделювання — спосіб дослідження об'єктів шляхом побудови й аналізування їх моделей.

Повторення (цикл) — алгоритмічна структура, за допомогою якої певна дія виконується декілька разів.

Предметна галузь — множина всіх об'єктів, що належать до певної сфери людської діяльності.

Профіль користувача — сукупність персональних даних і налаштувань робочого середовища.

Розгалуження — алгоритмічна структура, в якій виконання певної дії залежить від істинності умови.

Слідування — організація дій в алгоритмі, за якої дії виконуються послідовно одна за одною без пропусків або повторень.

Список розсилки — засіб електронної пошти, який надає можливість об'єднати певну кількість контактів у єдину групу для спрощення процесу розсилання листів.

Табличний процесор — прикладна програма для опрацювання даних, поданих в електронних таблицях.

Тіло циклу — серія команд, які повторюються під час виконання циклу.

Умова — запитання, сформульоване так, що допускає лише одну з двох відповідей: «так» або «ні».

Фішинг — шахрайські методи отримання особистих даних або коштів через інтернет.

Формат файлу — набір правил, за якими дані записуються у файл.

Функція — поймає блок програмного коду, до якого можна звернутися з будь-якого місця програми необхідну кількість разів.

Хмарний диск — сховище даних, розташоване на віддаленому сервері й доступне через інтернет.

Хмарні сервіси — сервіси, пов'язані з наданням користувачам постійного доступу до віддалених інтернет-ресурсів (серверів, застосунків, сховищ тощо).

Чат-бот — програма, що імітує відповіді людини в діалозі.

Штучний інтелект — розділ інформатики, присвячений розробці машин, які моделюють інтелект людини.

АЛФАВІТНИЙ ПОКАЖЧИК

- А**
Автозавершення 144
Автозаповнення 142
Адресна книга 25
Аргумент 148
- В**
Величина 173
Вкладений цикл 194
- Г**
Гіперпосилання
(посилання) 33
Графічний інтерфейс 227
- Д**
Динамічне тло 59
- Е**
Електронна пошта 7
Експертна система 43
- З**
Змінна 174
Значення величини 174
- І**
Інтерфейс користувача 126
Інтернет речей 47
Інформаційна модель 97
- К**
Карта знань 111
Комп'ютерна модель 102
Комп'ютерний
експеримент 104
- М**
Математична модель 117
Матеріальна модель 97
Метод ключових кадрів 64
Метод поділу на підзадачі 235
- О**
Область видимості
змінних 241
- П**
Предметна галузь 98
Повторення (цикл) 192
Покадрова анімація 83
Поштовий сервер 8
Профіль користувача 19
- Р**
Рівень доступу 38
Розгалуження 186
- С**
Слідування 176
Спам 29
Список 211
Список розсилки 26
Статичне тло 58
- Т**
Табличний процесор 125
Типи даних 131
Типи посилань 138
- У**
Умовне форматування 155
- Ф**
Функція 148
Функція користувача 236
- Х**
Хмарний диск 35
Хмарний сервіс 34
- Ч**
Чат-бот 45
- Ш**
Штучний інтелект 42

ЗМІСТ

Передмова.....	3
----------------	---

Розділ 1. СЛУЖБИ ІНТЕРНЕТУ

§ 1. Поштові служби інтернету.....	7
§ 2. Робота з електронною поштою	12
Практична робота 1.....	17
§ 3. Налаштування поштової скриньки	19
§ 4. Особливості електронного листування.....	25
§ 5. Етичне та безпечне користування електронною поштою.....	29
§ 6. Використання інтернет-ресурсів. Гіперпосилання в листах	33
§ 7. Колективна робота з документами.....	38
Практична робота 2.....	41
§ 8. Штучний інтелект. Інтернет речей	42

Розділ 2. КОМП'ЮТЕРНА АНІМАЦІЯ

§ 9. Принцип анімації. Редактор двовимірної анімації TurbTube.....	51
§ 10. Тло анімації. Керування кольором	57
§ 11. Автоматичне створення анімації	64
§ 12. Створення анімаційних ефектів.....	71
§ 13. Бібліотека об'єктів. Імпортування зображень. Експортування проекту	76
Практична робота 3.....	81
§ 14. Покадрова анімація.....	83
Практична робота 4.....	90

Розділ 3. МОДЕЛІ ТА МОДЕЛЮВАННЯ

§ 15. Поняття моделі. Поняття предметної галузі.....	95
§ 16. Типи моделей. Форми подання інформаційної моделі	100

§ 17. Побудова інформаційних моделей.....	106
§ 18. Карти знань. Сервіси для побудови карт знань.....	110
§ 19. Побудова математичної моделі	116
Практична робота 5.....	121

Розділ 4. ОПРАЦЮВАННЯ ТАБЛИЧНИХ ДАНИХ

§ 20. Основні об'єкти електронних таблиць.....	125
§ 21. Типи даних. Введення та опрацювання даних.....	131
§ 22. Адресація в електронних таблицях	138
§ 23. Автозаповнення та автозавершення	142
Практична робота 6.....	147
§ 24. Функції.....	148
§ 25. Логічні функції	152
§ 26. Умове форматування	155
Практична робота 7.....	158
§ 27. Математичне моделювання в електронних таблицях	160
Практична робота 8.....	164

Розділ 5. АЛГОРИТМИ ТА ПРОГРАМИ

§ 28. Етапи розв'язування задачі на комп'ютері	169
§ 29. Величини. Алгоритми роботи з величинами	173
§ 30. Опрацювання даних числових типів	179
§ 31. Алгоритми і програми з розгалуженнями	185
§ 32. Алгоритми і програми з повтореннями	192
§ 33. Рядковий тип даних.....	198
§ 34. Опрацювання рядкових величин.....	204
Практична робота 9	209
§ 35. Списки — змінювані послідовності.....	211
§ 36. Функції та методи опрацювання списків.....	217
§ 37. Опрацювання елементів списку	222
§ 38. Створення інтерфейсу користувача	227
Практична робота 10	234
§ 39. Розв'язування задачі методом поділу на підзадачі	235

§ 40. Обчислення з використанням функцій користувача	240
Практична робота 11	245

Розділ 6. ПРАКТИКУМ ІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Практична робота 12	249
Практична робота 13	252
Практична робота 14	255
Практична робота 15	257
Практична робота 16	259
Практична робота 17	261
Практична робота 18	263
Комп'ютерний словник	266
Алфавітний покажчик	268



Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	у кінці року
1				
2				
3				
4				
5				

Навчальне видання

БОНДАРЕНКО Олена Олександрівна
ЛАСТОВЕЦЬКИЙ Василь Васильович
ПИЛИПЧУК Олександр Павлович
ШЕСТОПАЛОВ Євген Анатолійович

«ІНФОРМАТИКА»

Підручник для 7 класу закладів загальної середньої освіти

Рекомендовано

Міністерством освіти і науки України

Видано за рахунок державних коштів.

Продаж заборонено

Підручник відповідає Державним санітарним нормам і правилам
«Гігієнічні вимоги до друкованої продукції для дітей»

Редактор *Л. А. Каюда*. Технічний редактор *А. В. Пліско*.
Художнє оформлення *В. І. Труфена, Т. В. Задорожної*.
Комп'ютерна верстка *О. Л. Піковець*. Коректор *Н. В. Красна*.

Підписано до друку 22.04.2024. Формат 70×90/16. Папір офсетний.
Гарнітура Шкільна. Друк офсетний. Ум. друк. арк. 19,89. Обл.-вид. арк. 20,29.
Наклад 46825 пр. Зам. № 7604-2024.

ТОВ Видавництво «Ранок»,
вул. Космічна, 21а, Харків 61145; вул. Деревлянська, 13, к. 3316, Київ 04119.
Свідоцтво суб'єкта видавничої справи ДК № 7548 від 16.12.2021.
E-mail: office@ranok.com.ua

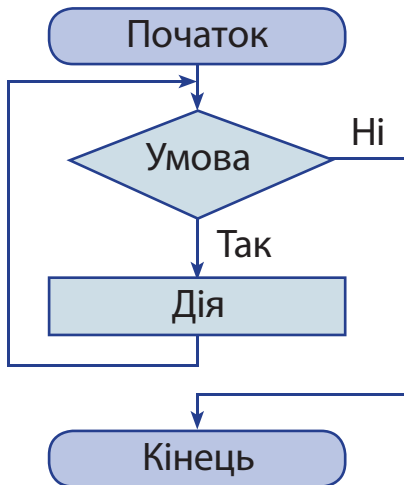
Надруковано у друкарні ТОВ «ТРИАДА-ПАК»,
пров. Сімферопольський, 6, Харків 61052.
Свідоцтво суб'єкта видавничої справи ДК № 5340 від 15.05.2017.
Тел. +38 (057) 712-20-00. E-mail: sale@triada.kharkov.ua

БАЗОВІ АЛГОРИТМІЧНІ

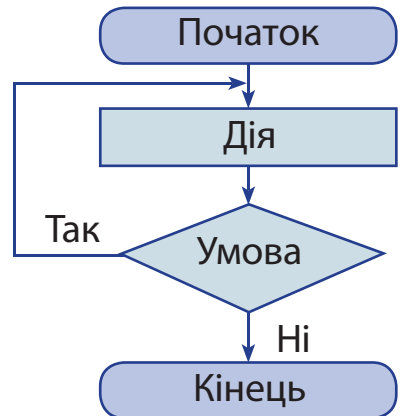
Лінійний алгоритм



Алгоритм із повторенням (цикл із передумовою)

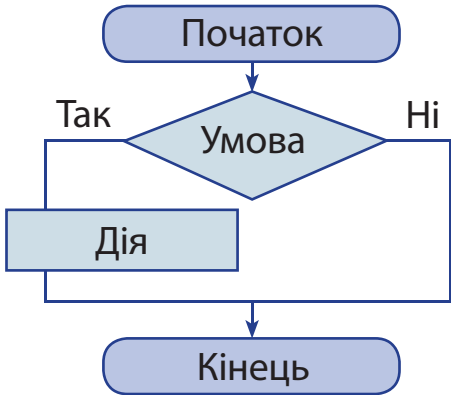


Алгоритм із повторенням (цикл із післяумовою)

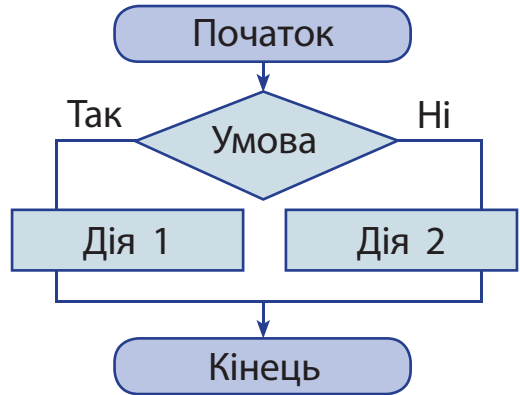


СТРУКТУРИ

Алгоритм із розгалуженням (неповне розгалуження)



Алгоритм із розгалуженням (повне розгалуження)



Алгоритм із повторенням (цикл із параметром)

