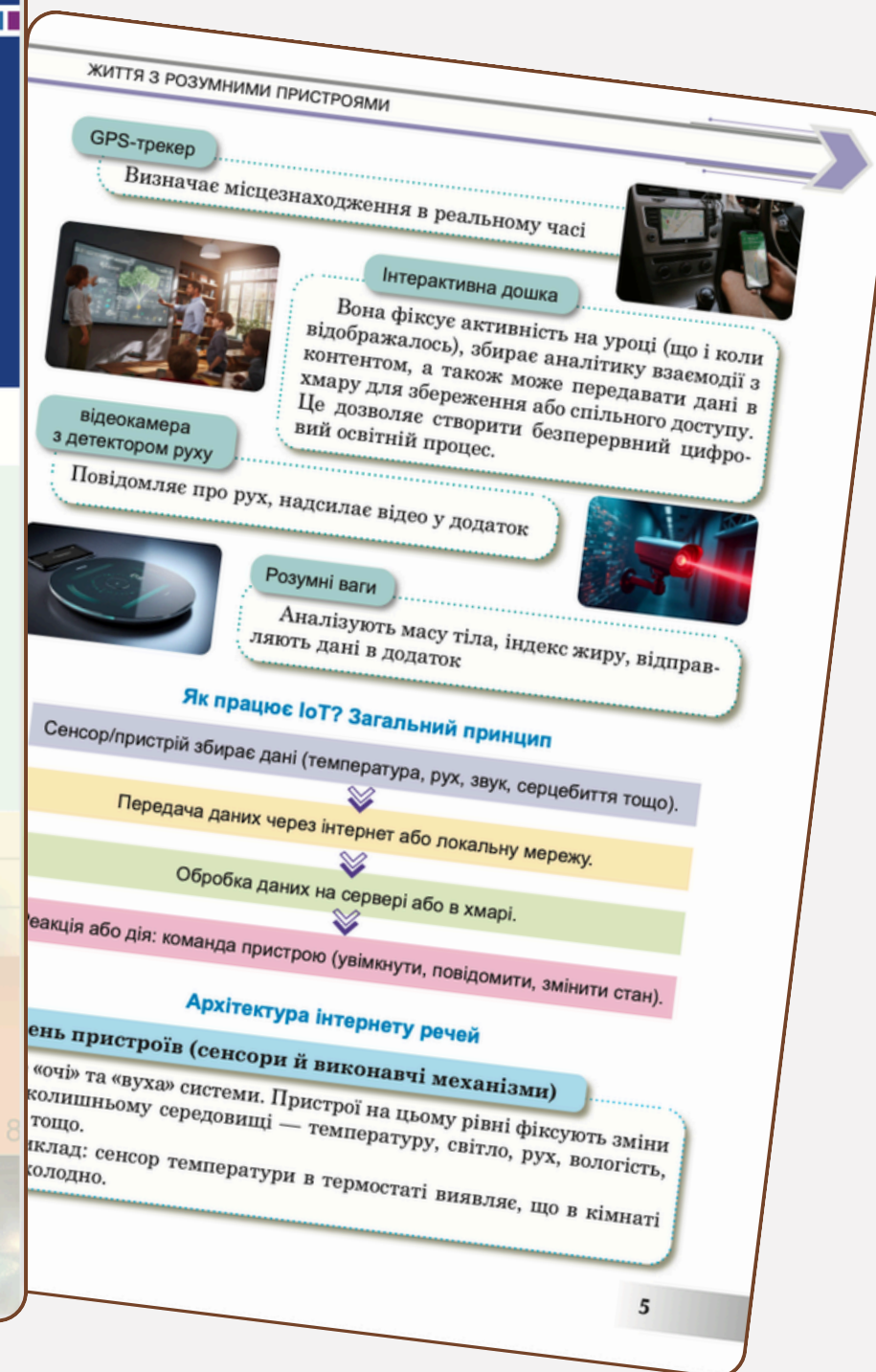
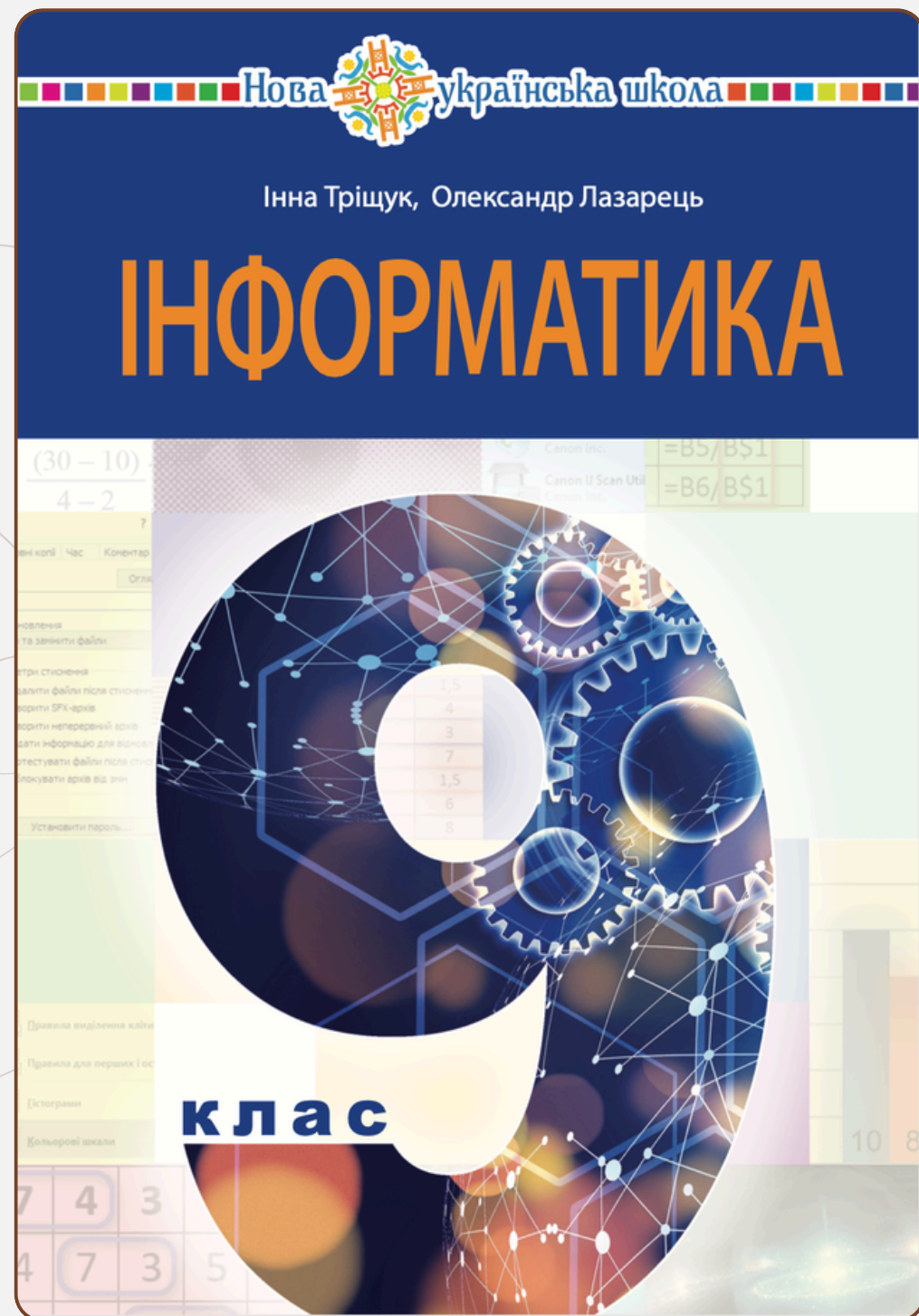


Інформатика

Підручник для 9 класів
закладів загальної
середньої освіти



Підручник,
який змінить ваше
уявлення про
інформатику!

Інна Тріщук
Олександр Лазарець

Видавництво
“Навчальна книга - Богдан”

Розділи підручника

РОЗДІЛ 1. Життя з розумними пристроями

1 Розділ

ЖИТТЯ З РОЗУМНИМИ ПРИСТРОЯМИ

Тема 1 Що таке інтернет речей (IoT)? Архітектура інтернету речей

Ключові ознаки IoT-пристроїв:

- Збір даних
- Передача даних через мережу
- Обробка та реагування на дані

Термін «Інтернет речей» (Internet of Things, IoT) був запропонований в 1999 році Кевіном Ештоном, одним з трьох засновників Центру автоматичної ідентифікації Массачусетського університету (Auto-ID Center).

Приклади IoT у повсякденному житті

- Розумний термостат: Автоматично регулює температуру
- Смарт-браслет: Вимірює пульс, сон, активність

4

РОЗДІЛ 2. Впорядкування даних

2 Розділ

ВПОРЯДКУВАННЯ ДАНИХ

Тема 11 Поняття бази даних. Основні терміни (поле, запис, типи даних, ключі)

База даних
Поле, записи
Основні типи даних
Ключове поле

Що таке база даних?
Уявіть собі шкільну бібліотеку. У ній є сотні книг. Щоб знайти потрібну, бібліотекарка користується спеціальним каталогом. У ньому зберігається інформація про всі книги: назва, автор, рік видання, жанр тощо. Таким чином бібліотекарка швидко знайде цю книгу.
Такий каталог — приклад бази даних, тобто впорядкованої структури для зберігання та обробки інформації.

База даних (БД) — це впорядкований набір інформації, яку можна зберігати, шукати, змінювати або аналізувати.

У комп'ютері база даних працює схожим чином, але замість книг там зберігаються, наприклад, списки учнів, розклади уроків чи навіть інформація про ваші вподобання (улюблений спортивний клуб, кіно, музику тощо) або додаткові потреби (безпечні продукти харчування, медична інформація, перелік та режим приймання ліків, рухової активності тощо).

Приклади баз даних:

- у магазині — накладна, де вказано назви товарів, ціну, кількість;
- у смартфоні — список контактів;
- у класному журналі — база оцінок і відомостей про учнів та учениць

Основні терміни

Щоб добре зрозуміти, як працює база даних, потрібно знати кілька основних понять.

24

РОЗДІЛ 3. Тривимірне моделювання

3 Розділ

ТРИВИМІРНЕ МОДЕЛЮВАННЯ

Тема 19 Програми для тривимірного моделювання. Етапи створення тривимірного зображення. Поняття моделі, візуалізація, віртуальний світ. Галузі використання 3D-моделей. Види моделей. Інтерфейс SketchUp, головні інструменти

3D-графіка — це особливий різновид комп'ютерної графіки — сукупність методик та інструментів, що використовуються для створення малюнків тривимірних об'єктів.
Тривимірне моделювання (3D-моделювання) — це процес створення цифрової моделі об'єкта у трьох вимірах: висота, ширина, глибина.

Ключові ознаки:

- має об'єм та форму;
- створюється у цифровому середовищі;
- доступна для перегляду з різних ракурсів.

Тривимірний малюнок легко можна відрізнити від двовимірного, бо в ньому є геометрична проекція 3D-моделі на площину, що з'являється завдяки спеціальним програмам.

Галузі використання 3D-моделей

Архітектура
У галузі архітектури 3D-моделі допомагають проектувати будинки, планувати інтер'єри та створювати візуалізації майбутніх споруд. Наприклад, так можна показати зацікавленим особам, який вигляд матиме школа ще до початку будівництва.

75

РОЗДІЛ 4. Інформаційні моделі

4 Розділ

ПРОГРАМНІ ПРОЄКТИ

Тема 33 Повторення основ Python. Структура програми

Мова програмування Python
Програма та її структура
Модулі Python

Навіщо нам Python і чому повторюємо основи?
У 7–8 класах ви вже ознайомилися з мовою програмування Python:
• малювали черепашкою графіку;
• навчилися працювати зі змінними;
• створювали анімації;
• використовували умови й цикли;
• писали прості програми та робили цікаві проєкти.
Тепер у 9 класі ми підемо далі, але для цього спершу потрібно повторити основи та розібратися, як правильно будувати програму, щоб міцно закріпити знання і перейти до складніших тем, наприклад, списки, функції чи навіть створення ігор.
Python — це універсальний інструмент у вашому рюкзаку. Він простий, як говорити з другом, але потужний, як суперкомп'ютер.
Із Python:
• створюють вебсайти (наприклад, Instagram чи YouTube);
• аналізують дані (щоб прогнозувати погоду чи рекомендувати фільми на Netflix);
• керують роботами (як у фільмах про майбутнє);
• пишуть ігри (наприклад, частини Minecraft).
Уявіть: ви можете написати програму, яка порахує, скільки грошей заощадите на новий гаджет, або симулюватиме гру «Камінь-ножиці-папір» з комп'ютером!

109

РОЗДІЛ 5. Безпека цифрових систем

5 Розділ

БЕЗПЕКА ЦИФРОВИХ СИСТЕМ

Тема 61-62 Цифрові інновації: користь і виклики. Інформація та інформаційні процеси у різних галузях. Захист особистих даних та конфіденційність у цифрових середовищах. Безпека цифрових систем

Цифрове середовище
Цифрове середовище — це віртуальний простір, що охоплює всі цифрові пристрої, мережі, онлайн-ресурси, де люди спілкуються, навчаються, працюють.
Цифрове середовище робить наше життя більш зручним і доступним. Воно дозволяє нам швидко знаходити інформацію, спілкуватися з іншими людьми, працювати та навчатися в онлайн-режимі, а також розв'язувати різноманітні завдання.
Цифрове середовище дозволяє нам оточувати себе новими можливостями, але також вимагає від нас обережності та вміння користуватися цими технологіями відповідально та безпечно.
У цифровому середовищі ми користуємося різними програмами і вебсайтами, щоб шукати інформацію, спілкуватися з іншими людьми, грати в ігри, редагувати фотографії і відео, а також виконувати багато інших завдань.

241

Складові кожної теми

Теоретична частина

РОЗДІЛ 5


Основні складові цифрового середовища

- Апаратне забезпечення (комп'ютери та цифрові пристрої)
- Мережева інфраструктура (інтернет)
- Програмне забезпечення (програми й додатки)
- Цифрові платформи та соціальні мережі
- Цифрові дані та інформаційні ресурси
- Користувачі та їхня цифрова компетентність
- Бази даних та інформаційні системи
- Інтернет речей (IoT) та смартпристрої

Цифрові інновації — це нові технології та рішення, які змінюють наше життя за допомогою комп'ютерів, смартфонів, ШІ тощо.

Приклади:

- штучний інтелект (чат-боти, перекладачі, рекомендації);
- розумні будинки та «розумні» пристрої (лампи, камери);
- онлайн-освіта (курси, платформи, відеоуроки);
- безконтактні платежі **Apple Pay/Google Pay**;
- 3D-друк, дрони, робототехніка.



242

Практична частина




БЕЗПЕКА ЦИФРОВИХ СИСТЕМ

Виклики та ризики

- Інформаційне перевантаження:** надто багато новин, контенту, реклами.
- Втрата приватності:** наші дані збирають соцмережі, сайти, додатки.
- Залежність від гаджетів:** тикток/ігри замість сну, навчання, спорту.
- Фейки та маніпуляції:** неправдиві новини, пропаганда, deepfake-відео.
- Кіберзлочинність:** шахрайство, віруси, крадіжка паролів і грошей.

Як це працює в різних сферах

- Бізнес і банки**
 - інтернет-банкінг, оплата онлайн
 - аналіз покупок (які товари популярні)
 - реклама, таргетинг (тобі показують саме ті оголошення, які «цікавлять»)
- Освіта**
 - електронні щоденники, журнали
 - платформи для навчання (**Google Classroom, Moodle, Zoom**, відеоуроки)
 - тести онлайн, автоматична перевірка
- Медицина**
 - електронні медичні картки
 - діагностичні системи (аналіз знімків, аналіз аналізів)
 - онлайн-консультації з лікарем



243

РОЗДІЛ 4

4 етап. Реалізація коду

Створення інтерфейсу

Підключення модулів:

- модуль **tkinter** для створення графічного інтерфейсу та роботи з основними віджетами.
- модуль **messagebox** для відображення інформативних повідомлень у діалогових вікнах.
- модуль **random** для випадкового вибору зі списку

```
from tkinter import *
from tkinter import messagebox
import random
```

Створення ігрового вікна

- Задання розмірів вікна ("700x650").
- Колір фону: темний (#1b263b).
- Фіксація розмірів (вимкнення масштабування).
- Встановлення заголовка: "Гра Камінь, Ножиці, Папери".
- Запустити головний цикл **mainloop()**

Створити мітку для привітання **label_greeting** текстом: "Ласкаво просимо у гру!"; шрифт: "Calibri", розмір: 14, колір фону "#1b263b" і колір тексту "#ffc000".

```
label_greeting.pack(pady=40)
```

Створити мітку для інформації вибору **label_choose** текстом: "Оберіть свій хід: "; шрифт: "Calibri", розмір: 14, колір фону "#1b263b" і колір тексту "#ffc000".

```
label_choose.pack(pady=10)
```

Створити фрейм для розміщення кнопок із заголовком "Ласкаво просимо у гру" і Паперу за допомогою методу **grid**.

```
button_frame = Frame(start, bg="#1b263b")
button_frame.pack(pady=20)
```

224

ПРОГРАМНІ ПРОЕКТИ

Практична робота №51

Завдання 1. Вік користувача

- Напишіть програму, яка запитує вік користувача.
- Перевірте, чи введено число в межах від 5 до 100.
- Якщо користувач введе текст або число поза межами, програма повинна попросити повторити введення.

Завдання 2. Калькулятор

- Створіть програму, яка запитує у користувача два числа та обчислює їхню суму.
- Використайте **try-except**, щоб обробити помилки введення (наприклад, якщо введено текст замість числа).

Завдання 3. Ділення чисел

- Програма повинна запитувати дільник і ділене.
- Використайте **try-except** для обробки **ZeroDivisionError**.
- Якщо дільник = 0, вивести дружнє повідомлення: **На нуль ділити не можна!**

Завдання 4. Перевірка списку

- Створіть список `fruits = ["яблуко", "банан", "груша"]`.
- Запитайте користувача, який фрукт він хоче вибрати.
- Використайте **try-except** для обробки **IndexError** або **ValueError** (наприклад, якщо користувач введе індекс, що не існує).

Завдання 5. Перевірка користувацьких даних

- Створіть програму, яка запитує:
 - Вік користувача
 - Оцінку за тест

Кроки:

- Перевірте, що введено саме **число**.
- Перевірте, що вік — **від 5 до 120**, а оцінка — **від 0 до 12**.
- Якщо введено неправильні значення — повідомте користувача та запропонуйте спробувати ще раз.

Підказка: використайте цикл **while True** разом із **try-except**.

211

Інтерактивна частина

ВПОРЯДКУВАННЯ ДАНИХ

Практична робота №4

Індивідуальна робота

Завдання 1.

Складіть та заповніть таблицю з 4-5 учнями свого класу. Поля: ID, Ім'я, Прізвище, Клас, Середній бал.

- Укажіть, які типи даних підходять для кожного поля у вашій таблиці.
- У таблиці оберіть поле, яке можна зробити ключовим. Чому?

Завдання 2.


Створіть таблицю для зберігання інформації про улюблені фільми своїх друзів. Визначте:

- Які поля будуть у таблиці (наприклад, «Назва фільму», «Рік», «Жанр»).
- Який тип даних матиме кожне поле.
- Яке поле буде ключовим.
- Заповніть таблицю мінімум трьома записами.

Завдання 3.

Створіть базу даних для парку розваг, щоб вести облік атракціонів. Виконайте:

- Визначте мінімум 5 полів (наприклад, «Назва атракціону», «Тип», «Мінімальний вік», «Ціна квитка», «Час роботи»).
- Вкажіть тип даних для кожного поля.
- Виберіть ключове поле.
- Заповніть таблицю 3–4 записами.
- Напишіть два запитання, які можна розв'язати за допомогою цієї бази даних (наприклад, «Які атракціони підходять для дітей до 10 років?»).



27

Реляційна база даних. Визначення об'єктів предметної області Тема 12

Ключові поняття
Реляційна база даних
Предметна область
Зв'язки між таблицями

Що таке реляційна база даних?

Реляційна база даних — це спосіб організації інформації, де дані зберігаються у вигляді таблиць, які пов'язані між собою.

Кожна таблиця містить інформацію про певний тип об'єктів, наприклад, про учнів/учениць, книги чи товари. Таблиці з'єднуються за допомогою спеціальних полів, які називаються ключами, щоб можна було легко знаходити потрібну інформацію.

Приклад із життя: уявіть шкільну їдальню. У них є таблиця з меню (страви, ціни) і таблиця з учнями (імена, класи). Якщо хочете знати, хто замовив борщ, ці таблиці можна пов'язати через унікальний номер замовлення. Це і є реляційна база даних — інформація розбита на окремі таблиці, але їх можна об'єднати.

ID	Назва страви	Ціна (грн)
1	Борщ	30
2	Вареники	25
3	Каша гречана	20

Номер замовлення	Ім'я учня/учениці	Клас	ID страви
2001	Марія	7-В	1
2002	Іван	9-А	3

Із запису 2001 відомо, що Марія замовила борщ, бо ID страви — 1. Це і є зв'язок між таблицями.

28

Інтерактивна презентація містить:

- елементи керування презентацією (головне меню, вихід, вперед, назад);
- інтерактивні кнопки з додатковою інформацією;
- інтерактивний вміст акцентує увагу, систематизує та узагальнює теоретичні відомості;
- інтерактивна презентація з теоретичною інформацією розроблена для кожної теми.

СЕРЕДОВИЩЕ ВИКОНАННЯ ПРОЄКТУ

Google Документи – для текстів, сценаріїв, описів

Trello – для планування кроків проєкту

Google Диск – для збереження всіх матеріалів команди

Створення текстового документу для:

- формулювання проблем і цілей проєкту;
- опису кожного IoT-пристрою;
- підготовки тексту презентації (хто що говорить);

Візуальна дошка задач з картками:

- "Зробити" – що заплановано;
- "У процесі" – над чим зараз працюють;
- "Готово" – виконані завдання.

До кожної задачі можна прикріпити відповідального учасника.

Спільна папка команди, де зберігаються:

- ескізи, чернетки, скріншоти;
- Canva-посилання;
- макети з Tinkercad;

Powered by genially EDUCATION

БЕЗПЕКА ІОТ ОЗНАЧАЄ ЗАХИСТ ДАНИХ, ЯКІ ПРИСТРОЇ ЗБИРАЮТЬ І ПЕРЕДАЮТЬ.

ПОТЕНЦІЙНІ ЗАГРОЗИ:

- Хакерські атаки
- Витік персональних даних
- Використання пристроїв як "ботів"

Powered by genially EDUCATION

Практична робота №5

Завдання 1. Оберіть одну з тем для бази даних:

- Шкільна їдальня
- Онлайн-магазин
- Домашні тварини

Опишіть предметну область.

Визначте 3–4 об'єкти.

Для кожного об'єкта — придумайте 3–5 полів.

Укажіть зв'язки між об'єктами.

Завдання 2. На основі попереднього завдання:

Складіть таблиці для кожного об'єкта.

Укажіть ключові поля.

Придумайте 2–3 записи для кожної таблиці.

Завдання 3.

БД для кабінету (курсу, уроків тощо)

Створіть базу даних «Рослини шкільного двору».

Таблиці:

- Рослини (назва, тип, висота, цвітіння).
- Місце посадки (ділянка, сторона школи, тип ґрунту).
- Чергові (відповідальні особи тощо), які доглядають рослини.

Опиши зв'язки:

- Одна рослина — одне місце посадки.
- Одна особа може доглядати за кількома рослинами.



Видалення елементів зі списку

Є кілька зручних способів:

☞ Метод `remove()` — видаляє за значенням: `fruits.remove("груша")`

Якщо такого елемента немає — буде помилка.

☞ Метод `remove()` — видаляє за значенням: `fruits.pop(2)`

Якщо індекс не вказати, `pop()` видаляє останній елемент.

☞ Оператор `del` — видаляє елемент або навіть увесь список:

```
del fruits[0] # видаляє перший елемент
# або
del fruits   # видаляє повністю список
```

☞ Метод `clear()` — очищає список (залишає порожній)

```
fruits.clear()
print(fruits) # []
```

Розширена інформація: `remove` — для значення, `pop/del` — для індексу. Якщо кілька однакових — `remove` видаляє тільки перше. Для видалення всіх — цикл (про це в наступних темах). Зрізи для видалення: `del список[1:3]` — видаляє шматок.

Комбінування з функціями та модулями

☞ Приклад коду з функцією:

Наприклад, функція, яка приймає список і додає елемент.

```
def add_to_list(my_list, item):
    my_list.append(item)
    return my_list # Повертаємо оновлений список

friends = ["Аня", "Богдан"]
new_friends = add_to_list(friends, "Віка")
print("Друзі:", new_friends) # ['Аня', 'Богдан', 'Віка']
```

☞ Приклад коду з модулями:

`random.choice(список)` — випадковий елемент, як у грі. В даному прикладі обираємо випадково колір зі списку.

Практична робота

- має чіткий алгоритм виконання;
- розроблена для кожної теми;
- активізує пізнавальну та практичну діяльність учнів;
- розроблено шаблони заголовків для практичних робіт.

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

for row in matrix:
    for item in row:
        print(item, end=" ")
    print()
```

1 2 3
4 5 6
7 8 9

Пояснення:

- Перший цикл перебирає кожен рядок (тобто підписок).
- Другий цикл — кожен елемент у цьому рядку.

Коли застосовувати: якщо потрібно працювати з таблицями, сітками або будь-якими даними, організованими у вигляді рядків і стовпців (наприклад, розклад уроків, шахівниця, пікселі зображення).

Підрахунок елементів

Підрахунок — це ітерація з лічильником.



Підрахунок всіх елементів: для знаходження кількості усіх елементів використовують `len(список)`

Підрахунок з умовами

Іноді потрібно дізнатися, скільки разів певний елемент трапляється у списку. Це можна зробити вручну через цикл:

```
animals = ["кіт", "собака", "кіт", "папуга", "кіт"]
count = 0

for a in animals:
    if a == "кіт":
        count += 1
```

Кількість котів: 3

```
print("Кількість котів:", count)
```

Метод `count(елемент)`

```
grades = [10, 12, 9, 12, 11]
twelves = grades.count(12)
print("Кількість 12:", twelves)
```

Кількість: 12: 2

МІНІ ПРОЕКТИ

Сумування елементів

```
Сумування за допомогою циклу
pocket_money = [50, 30, 40, 60]
total = 0
for money in pocket_money:
    total = total + money
print("Всього грошей:", total)
```

Використована функція `sum()`

```
pocket_money = [50, 30, 40, 60]
total = sum(pocket_money)
print("Швидка сума:", total)  Всього грошей: 180
```

Сумування елементів за умовами

```
pocket_money = [50, 30, 40, 60]
big = 0
for money in pocket_money:
    if money > 40:
        big += money
print("Великі суми:", big)  Великі суми: 110
```

Сумування елементів

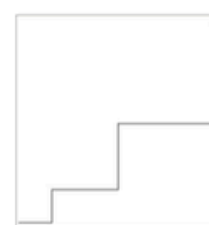
```
Сумування елементів для знаходження середнього
grades = [9, 10, 7, 8, 12]
total = 0
for i in grades:
    total = total + i
average = total / len(grades)
print("Середній бал =", average)  Середній бал = 9.2
```

Імпортування `turtle`

```
turtle import *
t = turtle.Turtle()
positions = [50, 100, 150]

for dist in positions:
    t.forward(dist) # Рух уперед
    t.left(90)      # Поворот вліво
    t.forward(dist) # Рух уперед
    t.right(90)     # Поворот праворуч
```

Приклад з `turtle`:



Приклад. Створення нового списку:

```
numbers = [2, 3, 4, 5]
squares = []

for n in numbers:
    squares.append(n ** 2)

print("Квадрати:", squares)  Квадрати: [4, 9, 16, 25]
```



Коллективні творчі практичні роботи

- має чіткий алгоритм виконання;
- розроблена для кожної теми;
- активізує пізнавальну та практичну діяльність учнів;
- розроблено шаблони заголовків для практичних робіт.

Конкатенація

Конкатенація — це операція з'єднання (об'єднання) рядків або інших послідовностей в один рядок.

Простими словами: конкатенація — це коли ми «склеюємо» два або більше текстів (рядків) разом.

Важливо:

- Конкатенація не виконує обчислень — вона не додає, а з'єднує текст.
- Якщо потрібно об'єднати числа, спочатку треба перетворити їх на текст.

```
name = 'Nazar'
print('Hello, ' + name)
# Hello, Nazar
```

Порожній рядок

"" — як порожня сторінка

Множення

Множення рядків — це операція, при якій рядок повторюється кілька разів підряд.

```
laugh = "ха" * 5
print(laugh)
# хахахаха
```

Рядок — це теж список!

Уявімо, що рядок — це список символів, де кожен символ має свій порядковий номер (індекс).

Це означає, що ми можемо:

- звертатися до окремих символів;
- зробити зрізи (отримувати частини рядка);
- проходити по рядку циклом, як по списку.

Індексація починається з 0. Тобто перший символ має індекс 0, другий — 1, і т.д. Якщо з кінця, то індекси йдуть -1, -2 і т.д.

Приклад:

```
word = "Книга"
print(word[0]) # К — перший символ
print(word[1]) # н — другий символ
print(word[2]) # и
print(word[3]) # г
print(word[4]) # а
print(word[-1]) # а — останній символ
```

153

Створення таблиць у Microsoft Access

У Microsoft Access таблиці можна створювати кількома способами.

Імпорт таблиці

Імпорт таблиці з даних із автоматичним визначенням структури

Конструктор таблиці

Конструктор таблиці (визначення структури таблиці)

Майстер таблиці

Майстер таблиці (визначене створення на основі шаблонів)

Імпорт

Імпорт даних або зв'язком із зовнішніми джерелами даних

Створення таблиці в режимі таблиці

Режим таблиці найбільш підходить для швидкого створення таблиці з автоматичним визначенням структури.

Кроки

1. Відкрити Microsoft Access і створити нову базу даних.

2. Натиснути **Створити (Create)** у групі **Таблиці** та натиснути **Таблиця (Table)**.

3. З'явиться порожня таблиця з полем **Ідентифікатор** (з автоматичною нумерацією).

4. Натиснути кнопку **Клацніть, щоб додати**, щоб додати нові поля, обравши тип даних цього поля.

5. Подвійним натисканням змінити назву поля.
6. Ввести дані безпосередньо в таблицю, змінюючи назви полів і типи даних за потреби.
7. Після введення даних зберегти таблицю, присвоївши їй ім'я.

Особливості

- Поле **Ідентифікатор** автоматично створюється як первинний ключ із типом **Автонумерація**.
- Для зміни типів даних або структури потрібно перейти до режиму конструктора.
- Цей спосіб зручний для початківців, але менш гнучкий, ніж режим конструктора.

37

2

Перегляд даних

Форми дозволяють переглядати записи по одному, а не всі одразу, як у таблиці. Це зручно, якщо ти хочеш перевірити інформацію про конкретну особу чи книгу.

3

Редагування даних

Можна змінювати дані через форму, наприклад, оновити ціну страви в їдальні чи змінити адресу людини в базі.

4

Фільтрація та пошук

Форми можуть мати кнопки чи поля для пошуку даних. Наприклад, можна знайти всі книги певного жанру, ввівши «фентезі» у поле пошуку.

5

Зручний дизайн

Форми можна налаштувати, щоб вони мали привабливий вигляд: додати кольори, змінити розмір полів чи додати логотип школи.

Технології No Code

888

Технології No Code — це інструменти, які дозволяють створювати програми, бази даних чи навіть вебсайти без програмування (написання коду).

Microsoft Access є прикладом такої технології, бо ви можете створювати форми, запити та звіти, просто використовуючи мишку та готові інструменти програми. Це як конструктор LEGO: складаєте потрібні деталі (поля, кнопки), щоб отримати бажаний результат.

No Code інструменти, такі як **Access**, дозволяють:

- Створювати бази даних і форми за допомогою графічного інтерфейсу.
- Налаштовувати вигляд форм (наприклад, додавати кольори чи підписи).
- Автоматизувати завдання, такі як пошук чи сортування даних.

47

- подано ключові поняття до кожної теми;
- терміни позначені спеціальними маркерами;
- алгоритми процесів розміщені у візуальних блоках, що сприяє активізації уваги учня;
- представлення інформації у вигляді інфографіки, що сприяє кращому запам'ятовуванню та розумінню матеріалу.

- теоретична інформація з програмування подана чітко та логічно, відповідаючи віковим особливостям учнів;
- додано багато скріншотів коду для кращого розуміння.

Теоретична інформація з програмування

Модулі Python: ваші помічники в програмуванні

Модулі — це як набори інструментів, які додають нові можливості до Python.

У 7-му та 8-му класах ми вже використовували модулі `turtle` і `tkinter` для графіки, `math` для математичних функцій та `random` для випадкових чисел.

Щоб використовувати модуль, його потрібно імпортувати командою `import`. Уявіть, що це ніби відкрити коробку з новими фломастерами для малювання.



Математика стає простішою

Модуль `math` — це ваш калькулятор для складних обчислень. Він потрібен, коли хочете знайти квадратний корінь, округлити число чи використати константи, як число π (пі). У 8-му класі ми використовували його для математичних функцій, наприклад, при розв'язанні квадратних рівнянь.

Модуль `math`

Основні функції модуля `math`:

- `math.sqrt(x)` — квадратний корінь числа `x`.
Наприклад, `math.sqrt(16)` повертає 4.
- `math.pow(x, y)` — підносить `x` до степеня `y`.
Наприклад, `math.pow(2, 3) = 8`.
- `math.pi` — константа π (≈ 3.14159), корисна для обчислень із колами.
- `math.floor(x)` — округлює вниз (до меншого цілого).
Наприклад, `math.floor(3.7) = 3`.
- `math.ceil(x)` — округлює вгору (до більшого цілого).
Наприклад, `math.ceil(3.7) = 4`.

Приклад з життя. Ви плануєте вечірку і хочете спекти круглий торт. Щоб знати, скільки крему потрібно, обчислимо площу торта ($S = \pi r^2$).

```
from math import *

radius = float(input("Введи радіус торта (см):")) # Наприклад, 15 см
area = pi * pow(radius, 2) # Площа = π * r²
print("Площа торта:", round(area, 2), "см²") # Округлення до 2 знаків
```



Додаємо випадковість

Модуль `random` робить програми непередбачуваними, як гра в кубик чи лотерею.

Основні функції модуля `random`:

- `random.randint(a, b)` — випадкове ціле число від `a` до `b` (включно).

Модуль `random`

У Python теж є вбудовані функції:

- `print()` — виводить інформацію на екран
- `input()` — отримує дані від користувача
- `len()` — обчислює довжину рядка чи списку
- `int()` — перетворює на ціле число



Функція — це блок коду, який виконує певну дію. Його можна викликати скільки завгодно разів, в будь-якому місці програми, не переписуючи знову всього коду.

Щоб створити свою функцію, у Python використовують ключове слово `def`.

Синтаксис:

```
def ім'я_функції(параметри):
    #Тіло функції
    #Дії, які виконує функція
    return результат
```

- `def` — означає «визначаємо функцію»;
- `ім'я_функції` — придумуємо самі. Повинно бути англійською мовою, з маленької літери, та має описувати, що саме вона робить (наприклад, `calculate_area` — обчислити площу). Не використовуйте пробілів, замість них робіть підкреслення `_`;
- `параметри` — це «вхідні дані», які ми передаємо для функції, як інгредієнти в рецепті. Їх пишуть у дужках. Якщо параметрів немає, дужки пусті `()`;
- `тіло функції` — код, який виконується. Він відступає на 4 пробіли (індентація — пам'ятає з циклів чи умов?);
- `return` — те, що функція повертає назад.

Приклад простої функції без параметрів:

```
def say_hello(): # Визначення функції
    print("Привіт, друзі! Готові програмувати?") # Тіло
```

Щоб викликати функцію, достатньо звернутися до неї за ім'ям:

```
say_hello() #Виведе: "Привіт, друзі! Готові програмувати?"
```

Впорядкованість. Елементи йдуть у послідовності, як у черзі в шкільній їдальні. Перший елемент — на позиції 0 (так, у Python нумерація починається з 0!).

Змінність. Списки можна змінювати — додавати, видаляти, змінювати елементи.

Дублікати. Можна мати однакові елементи, наприклад, два «яблука» в списку покупок.

Різноманітність. Списки можуть містити елементи різних типів.

Як створити список

Список створюють за допомогою квадратних дужок `[]`, а елементи розділяють комами.

Він буде змінною, що зберігає кілька значень одразу.

Порожній список:	<code>empty = []</code> — як порожня коробка, куди потім складаєте речі.
З елементами:	<code>numbers = [1, 2, 3]</code> <code>fruits = ["яблуко", "банан", "груша", "персик"]</code>
З повтореннями:	<code>colors = ["червоний", "синій", "червоний"]</code>
Змішаний:	<code>mixed = [10, "kit", True, 3.14]</code>
Вкладені списки:	<code>matrix = [[1, 2], [3, 4]]</code> — список списків, як таблиця

Приклади списків

Ситуація з життя	Список у Python
Уроки на тиждень	<code>lessons = ["математика", "інформатика", "історія", "англійська"]</code>
Температура за тиждень	<code>temps = [15, 17, 20, 22, 18, 16, 14]</code>
Продукти в холодильнику	<code>food = ["молоко", "хліб", "яйця", "сир"]</code>
Улюблені ігри	<code>games = ["Minecraft", "Brawl Stars", "Roblox"]</code>

Довжина списку

У Python список може містити багато елементів: числа, слова або навіть інші списки.

Іноді потрібно дізнатися, скільки елементів у списку — наприклад, скільки учнів у класі, скільки оцінок у журналі чи скільки покупок у кошику.

Для цього використовують вбудовану функцію `len()`.

```
fruits = ["яблуко", "апельсин", "груша"]
print(len(fruits)) # 3
```

- практична інформація з програмування подана чітко та логічно, відповідаючи віковим особливостям учнів;
- додано багато скріншотів коду для кращого розуміння.

Практична інформація з програмування

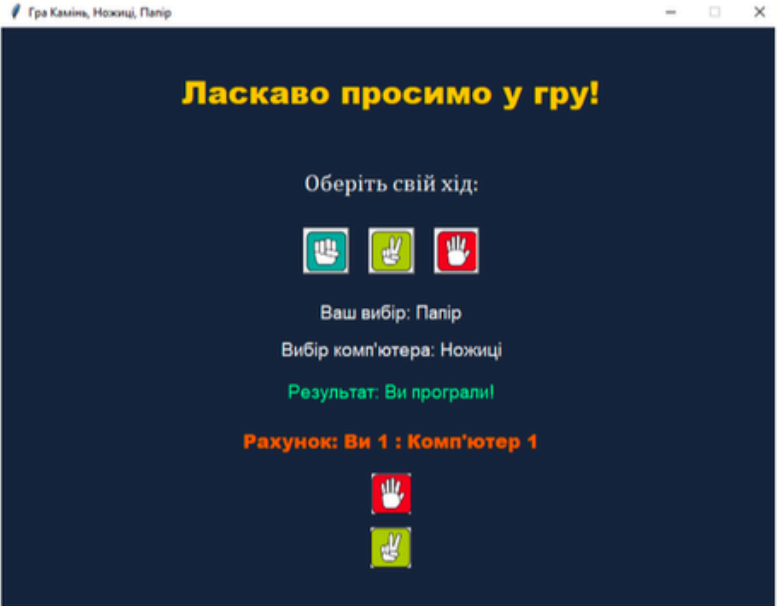
Проект 1
Додаток «Камінь, Ножиці, Папір»

Теми 55-57

1 етап. Організаційний

Завдання: Розробити програму з графічним інтерфейсом, яка дозволяє користувачеві:

- Обрати свій хід (Камінь, Ножиці або Папір).
- Порівняти свій вибір з комп'ютером і дізнатися результат (виграш, програш або нічия).
- Динамічно відобразити рахунок гри.



Мета програми:

- Створити стартове вікно з привітанням та кнопками для вибору ходу.
- Показувати вибір гравця і комп'ютера у вигляді тексту та зображень.
- Відображати результат кожного раунду і оновлювати рахунок.
- Забезпечити зручний та інтуїтивно зрозумілий інтерфейс.

222

ПРОГРАМНІ ПРОЄКТИ

2 етап. Підготовчий

Мета програми:


- Стартове вікно:
 - Привітання користувача.
 - Мітка з інструкцією.
 - Кнопки для вибору ходу.
- Вікно гри:
 - Показ вибору гравця та комп'ютера.
 - Зображення вибору.
 - Мітка результату.
 - Мітка рахунку.

Зображення:

- Потрібно підготувати **png**-файли для кожного ходу (зображення однакового розміру):
 - rock.png** – камінь
 - scissors.png** – ножиці
 - paper.png** – папір
- Масштабування через **subsample** для зручного відображення.

Збереження зображень до проекту.

В інтернеті знайдіть тематичні зображення каменя, ножиць та паперу. Для цього в пошуковому рядку введіть ключові слова, віднайдіть потрібні зображення та збережіть їх у папці з проектом і переназвіть їх зручним іменем.



Зауваження! Візьміть до уваги, що розширення має бути **png** або **gif**. Для зручності — зберігайте зображення, роблячи скріншоти потрібного зображення, а також з прозорим фоном.

3 етап. Проектний

Структура інтерфейсу:

Вікно гри:

- Колір фону: темний (**#1b263b**).
- Мітки для відображення вибору гравця і комп'ютера.
- Мітка для результату раунду (**#00ff99**).
- Мітка для рахунку (**#ff6600**).
- Кнопки з картинками Камінь, Ножиці, Папір.

223

РОЗДІЛ 4

4 етап. Реалізація коду

Створення інтерфейсу

Підключення модулів:

- модуль **tkinter** для створення графічного інтерфейсу та роботи з основними віджетами.
- модуль **messagebox** для відображення інформативних повідомлень у діалогових вікнах.
- модуль **random** для випадкового вибору зі списку

```
from tkinter import *
from tkinter import messagebox
import random
```

Створення ігрового вікна:

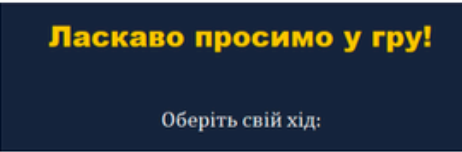
- Задання розмірів вікна ("700x650").
- Колір фону: темний (**#1b263b**).
- Фіксація розмірів (вимкнення масштабування).
- Встановлення заголовка: "Гра Камінь, Ножиці, Папір".
- Запустити головний цикл **mainloop()**

Створити мітку для привітання **label_greeting** з такими властивостями: текст: "Ласкаво просимо у гру!"; шрифт: Arial Black, 24; колір фону **#1b263b** і колір тексту **#ffcc00**.

```
label_greeting.pack(pady=40)
```

Створити мітку для інформації вибору **label_choose** з такими властивостями: текст: "Оберіть свій хід:"; шрифт: "Cambria", 18; колір фону **#1b263b** і колір тексту **#ffffff**.

```
label_choose.pack(pady=10)
```



Створити фрейм для розміщення кнопок із зображенням Каміня, Ножиць і Паперу за допомогою методу **grid**

```
button_frame = Frame(start, bg="#1b263b")
button_frame.pack(pady=20)
```

224

- теоретична інформація з моделювання подана чітко та логічно, відповідаючи віковим особливостям учнів;
- додано багато скріншотів коду для кращого розуміння.

Теоретична інформація з тривимірного моделювання

Програми для 3D-моделювання

Blender – безкоштовна програма з відкритим кодом, яка містить широкий набір функцій для 3D-моделювання, скульптингу, рендерингу та анімації. Завдяки активній спільноті та постійним оновленням Blender є відмінним вибором як для початківців, так і для професіоналів.



Tinkercad – простий у використанні вебдодаток для 3D-моделювання, який ідеально підходить для новачків. Він дозволяє створювати моделі за допомогою базових геометричних форм та працює одразу в браузері, що робить його доступним для широкого кола користувачів.



SketchUp – зручна програма для швидкого 3D-моделювання, яка активно використовується в архітектурі та дизайні. Її простий інтерфейс дозволяє легко створювати 3D-моделі без необхідності глибоких технічних знань.



Види 3D-моделей

Фізичні моделі

Це моделі, які імітують форму та геометрію реальних об'єктів, без додаткової інформації про властивості матеріалу.



Приклад: у SketchUp можна змоделювати стілець або будинок, щоб побачити їхню форму з усіх боків.

Де використовується:

- У навчанні (наприклад, моделювання простих предметів на уроках).
- В архітектурі (первинні форми будівель до деталізації).

Інформаційні моделі

Ці моделі містять додаткові параметри — матеріали, текстури, вагу, міцність, теплопровідність тощо.

Де використовується:

- У будівництві (інформаційне моделювання будівель).
- У виробництві (моделювання деталей машин із точними матеріалами).
- В інженерії — аналіз поведінки конструкцій при навантаженнях.

Схематичні моделі

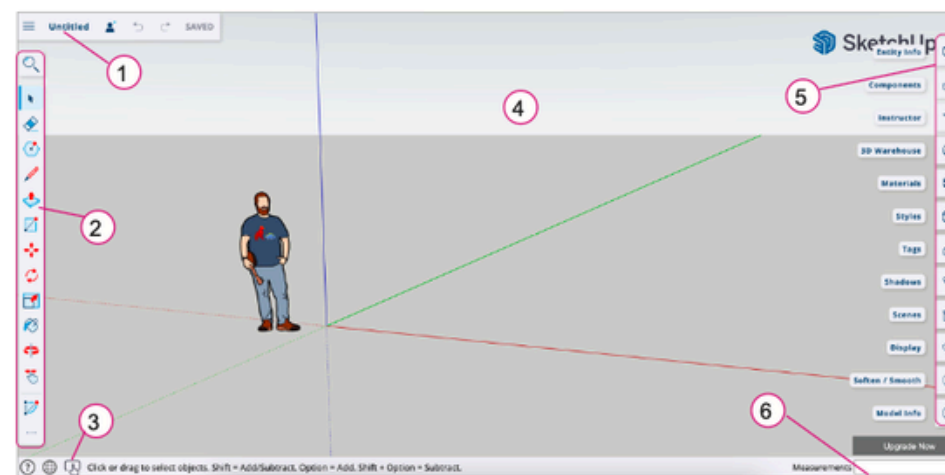
Це спрощені геометричні моделі, які показують загальну структуру, логіку або процес, але без деталей.

Приклад: блок-схема вентиляційної системи будівлі або спрощена 3D-модель маршруту труб.

Де використовується:

- У плануванні та аналізі.
- У технічній документації.
- У навчанні, коли потрібно пояснити принцип роботи системи (наприклад, рух електронів у колі).

SketchUp for Web — це браузерна версія популярного 3D-редактора, яка дозволяє створювати тривимірні моделі без встановлення програми. Все, що потрібно — комп'ютер та доступ до інтернету.



1 **Панель зверху** — містить опції збереження, відкриття файлів, експорту моделі, вибору шаблону та налаштувань проекту.

2 **Панель інструментів ліворуч (Toolbar)** — містить основні інструменти для малювання та редагування об'єктів. Усі інструменти мають піктограми, які зручно розміщені вертикально.

3 **Палець з курсором** — підказки щодо дій з мишею. Показує коротку підказку щодо поточного активного інструмента. Наприклад: **Click or drag to select objects** — натисніть або перетягніть, щоб вибрати об'єкти.

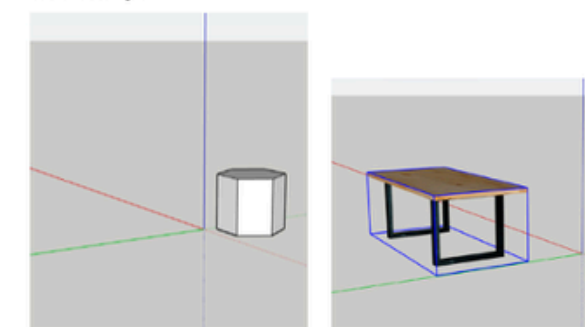
Якщо обрати інший інструмент — підказка зміниться відповідно.

У SketchUp і будь-якому іншому 3D-редакторі об'єкти створюють у тривимірному просторі, який має три осі.

Червона вісь (X) позначає горизонтальний напрямок зліва направо. Вона допомагає визначити, наприклад, довжину об'єкта або будівлі.

Зелена вісь (Y) також є горизонтальною, але йде в напрямку вперед-назад. Зазвичай її використовують для створення глибини, наприклад, ширини кімнати.

Синя вісь (Z) — вертикальна. Вона визначає напрямок вгору-вниз і використовується для задання висоти об'єкта, наприклад, висоти стіни або даху.



Гарячі клавіші

- R — Прямокутник
- C — Коло
- P — Push/Pull
- M — Переміщення
- S — Масштабування
- Q — Обертання
- L — Лінія
- E — Гумка (Erase)
- Space — Виділення

Select (Вибір) — використовується для вибору об'єктів. Фішка: **Shift** + клацання — додає або знімає об'єкт з вибору. Двічі клацання — вибір об'єкта + його поверхні; тричі — вибір усієї групи.

Eraser (Гумка) — стирає лінії та об'єкти.

Rectangle / Polygon / Circle — малюють базові фігури: прямокутник, багатокутник, коло.

Push/Pull (Витягування) — витягує 2D-форми у 3D-об'єкти.

Line (Лінія) — малює прямі лінії. Основний інструмент для побудови. Фішка: утримуй **Shift** для фіксації напрямку по осях (X, Y, Z).

Move (Переміщення) — пересуває об'єкти або копіює. Фішка: утримуй **Ctrl (Cmd)** під час переміщення — створення копії. Стрілками фіксує рух по конкретній осі.

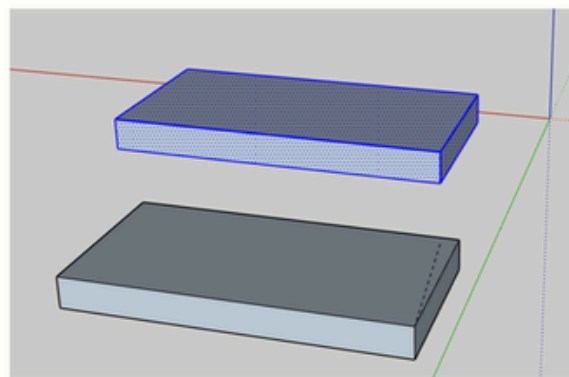
- практична інформація з моделювання подана чітко та логічно, відповідаючи віковим особливостям учнів;
- додано багато скріншотів коду для кращого розуміння.

Практична інформація з тривимірного моделювання

РОЗДІЛ 3

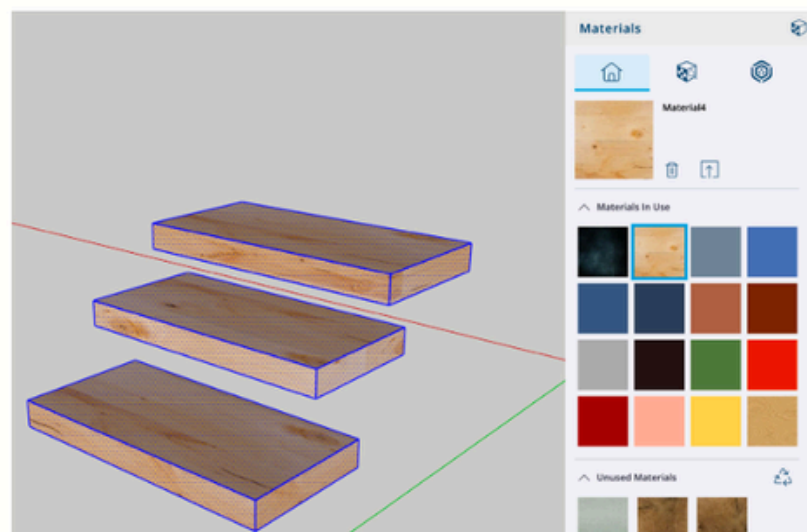
Створіть наступну сходинку.

- Оберіть **Move Tool**.
- Утримуйте **Ctrl** (або **Option** на **Mac**), щоб зробити копію ступеня.
- Перетягніть копію трохи назад і вгору (наприклад, на 30 см назад і на 15 см вгору).
- Якщо з'являється зелена і синя підсвітка — ви все робите правильно по осях.



Повторіть для наступних сходинок.

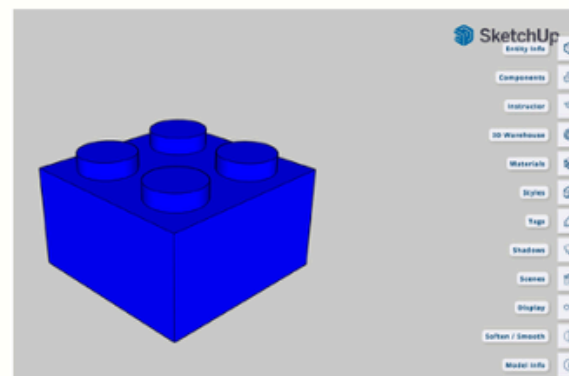
- Ще раз скопіюйте попередню сходинку.
- Створіть 3–5 ступенів.



84

ТРИВИМІРНЕ МОДЕЛЮВАННЯ

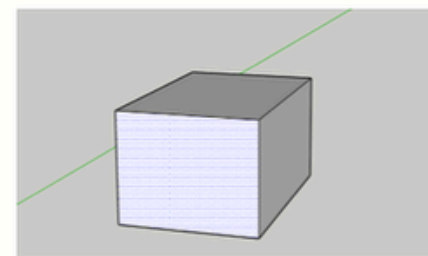
Завдання 2. Створити LEGO-подібну цеглинку з 4 круглими виступами.



1. Створіть основу (куб).

Інструмент: **Rectangle Tool**

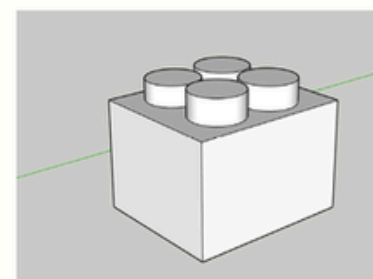
Клацніть на площині й створіть прямокутник (наприклад, 4x4 см).
Push/Pull Tool → витягніть вгору (наприклад, 2 см).



2. Створіть перший виступ.

Circle Tool → виберіть верхню грань куба.

Намалюйте коло (радіусом близько 0.6 см) у верхньому лівому куті.
Push/Pull Tool → витягніть вгору (наприклад, 0.4 см).



85

РОЗДІЛ 3

3. Зробіть копії виступу.

Виділіть круглий виступ (**Select Tool**).

Оберіть **Move Tool** → наведіть на центр циліндра.

Утримуйте **Ctrl** (або **Option** на **Mac**) — це задіє режим копіювання.

Потягніть копію в інший кут → клацніть.

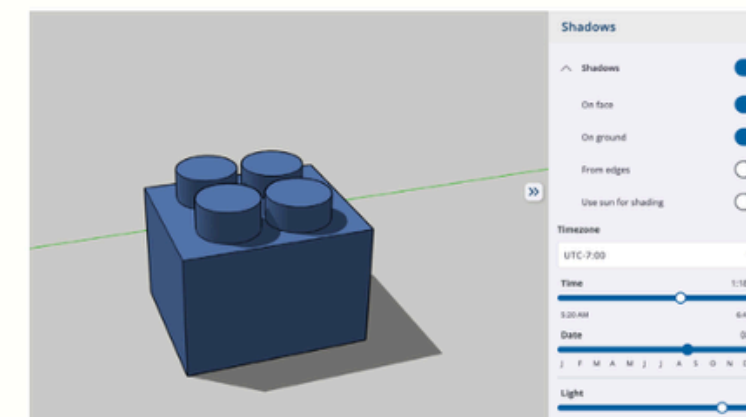
У полі введення напишіть *3 → натисніть **Enter** — отримаєте 3 додаткові копії = всього 4 виступи.

4. Змініть колір.

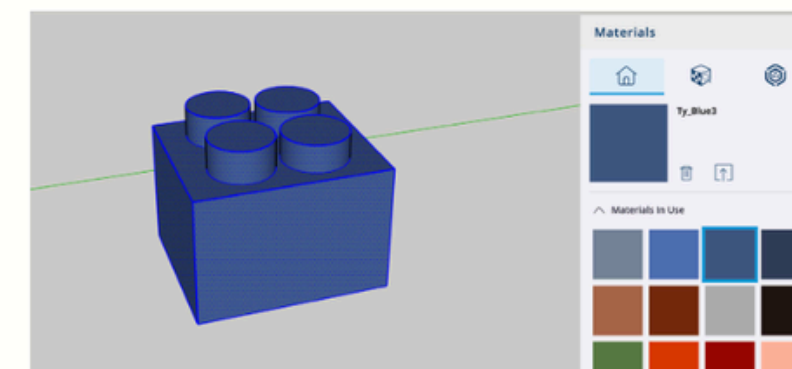
Відкрийте **Materials** (праворуч).

Оберіть синій колір.

Залейте всю фігуру (куб + виступи).



Можна додати тінь.



86

Це ключова особливість підручника для 9 класу — майже кожен розділ завершується реальним проєктом. «Розумний клас майбутнього», «IoT у моєму житті», «Будинок моєї мрії», «Камінь, Ножиці, Папір», «Хто хоче стати мільйонером» — це не просто завдання, а повноцінні міні-продукти, які учні створюють власноруч.

Проектне навчання

Дослідницький проєкт
«IoT у моєму житті»

Теми 6–7

QR-код

Індивідуальна дослідницька робота

Дослідити, як інтернет речей уже впливає на ваше життя, проаналізувати приклади застосування, переваги, ризики та особливості конфіденційного використання.

Скласти план 5-7 речей IoT, які ви хочете використовувати у своєму житті.

Приклади IoT-пристроїв:

- Розумна колонка
- Розумний термостат
- Браслет здоров'я
- Розумне освітлення
- QR-коди на транспорті
- GPS-трекер для домашніх тварин

Середовище виконання проєкту

- Canva** інтуїтивно зрозумілий сервіс для створення презентацій, макетів, схем
- Google Slides** прості у використанні інструменти для графічної схеми або презентації
- Google Документи** для текстів виступу

Результат індивідуального дослідницького проєкту

Презентація в Canva

- Опис 3–5 IoT-пристроїв, які використовуєте або плануєте використовувати (що робить, яку проблему усуває, які дані збирає).
- План: 5–7 IoT-речей, які ви хочете мати в майбутньому, з поясненням вибору.
- Аналіз впливу на ваш цифровий слід: які дані зберігаються, які ризики можуть бути.
- Рекомендації щодо конфіденційності: як захищати свої дані, що варто врахувати.

20

Креативний/маркетинговий проєкт
«IoT для майбутнього: ідея, реклама, презентація»

Теми 8–10

QR-код

Колективна робота

Робота у парах або трійках.

Мета проєкту: розробити ідею інноваційного IoT-пристрою (опис, яку проблему розв'язує, як працює).

Результат вашої командної роботи:

- Ідея інноваційного IoT-пристрою: опис, яку проблему розв'язує, як працює.
- 3D-модель пристрою (ThinkerCad або SketchUp).
- Назва, логотип і слоган пристрою (візуальний бренд).
- Рекламний продукт: або відеореклама (30–60 с.) або рекламний постер / банер.
- Презентація команди (до 5 хв).

Середовище виконання проєкту

- Tinkercad**
- SketchUp**
- Canva**
- Google Slides**
- Google Документи**
- Trello**
- Google Диск**

22

ЖИТТЯ З РОЗУМНИМИ ПРИСТРОЯМИ

Генерація ідей + проєктування

- Визначення проблеми – Вправа: «Що мене щодня турбує? Що я хочу поліпшити?»
- Мозковий штурм рішень – Які функції має пристрій? Які сенсори? Зв'язок зі смартфоном? Збір даних?
- Розробка бренду – Назва, слоган, логотип.

Пам'ятайте про врахування потреб вразливих груп, уникнення їх дискримінації та забезпечення інклюзії

Створення прототипу пристрою

- У середовищі Tinkercad або SketchUp.
- Простий 3D-макет або модель, що демонструє зовнішній вигляд та елементи пристрою.

Рекламний продукт

- Відеореклама (30-60 секунд).
- Рекламний постер / банер.

Все робимо в Canva

Презентація проєкту

- Представте свій проєкт за 3–5 хвилин.
- Поясніть, чому саме цей пристрій вибраний.

23

Завершальний розділ підручника охоплює теми, які є критично важливими саме зараз: цифровий слід, конфіденційність, електронна демократія, великі дані та штучний інтелект. Учні вивчають не лише теорію, а й правові аспекти роботи з інформацією та порівнюють сервіси електронного навчання.

Цифрова безпека та ШІ

Електронна демократія. Електронне урядування. Цифровий підпис Тема 64

Ключові поняття

Сучасна держава активно використовує цифрові технології для взаємодії з громадянами. Електронна демократія, електронне урядування та цифровий підпис є ключовими елементами цієї трансформації.

Цифрова трансформація держави — це систематичне впровадження інформаційно-комунікаційних технологій у роботу органів влади.

Її мета — зробити публічні послуги більш доступними, прозорими та ефективними. Громадянин отримує можливість взаємодіяти з державою дистанційно, без фізичної присутності в установах. Основними інструментами електронної демократії є електронні петиції, онлайн-консультації, публічні обговорення та сервіси «бюджету участі».

Вони дають можливість громадянам висувати пропозиції, коментувати проекти рішень і контролювати використання публічних коштів. Електронна демократія в Україні реалізується через:

- ♦ платформи електронних петицій (місцеві та загальнодержавні);
- ♦ публічні обговорення проектів рішень органів влади онлайн тощо.

Електронне урядування

Електронне урядування забезпечує цифрову інфраструктуру та електронні послуги держави. Електронна демократія використовує ці можливості для участі громадян у прийнятті рішень. Електронне урядування втілюється через:

- ♦ єдині портали державних послуг та мобільні застосунки;
- ♦ Електронні реєстри (населення, бізнесу, нерухомості тощо);

254

БЕЗПЕКА ЦИФРОВИХ СИСТЕМ

- ♦ онлайн-сервіси: реєстрація ФОП, довідки, витяги, соцвиплати, запис до черги.

Переваги

- Доступність послуг 24/7 незалежно від місця проживання.
- Менше черг, менше суб'єктивного впливу чиновників.
- Прозорість рішень та руху публічних коштів.

Ризики

- Залежність від стабільності ІТ-систем та кіберзахисту.
- Можливість витоку персональних даних у разі слабкої безпеки.

Електронні послуги — це адміністративні та інші публічні послуги, які надаються в онлайн-форматі через спеціальні портали чи додатки.

Громадянин може подати заяву, отримати довідку, зареєструвати бізнес або сплатити податки дистанційно.

Електронний цифровий підпис

Електронний цифровий підпис (або скорочено — **ЕЦП**) за правовим статусом прирівняний до власноручного підпису або печатки.

ЕЦП — це дані в електронній формі, отримані за результатами криптографічного перетворення, які додаються до інших даних або документів і забезпечують їх цілісність та ідентифікацію автора.

За допомогою послуг ЕЦП можна підписувати електронні документи, користуватися електронними послугами, реєструватися на державних порталах тощо. Документи, підписані за допомогою ЕЦП, мають таку ж саму юридичну силу, як і звичайні.

Цифровий підпис дає змогу:

- Підписувати заяви, договори та інші документи без паперу й особистої присутності.
- Подавати електронну звітність, реєструвати бізнес, оформлювати послуги онлайн.
- Підтверджувати свою особу в державних та банківських сервісах.

255

БЕЗПЕКА ЦИФРОВИХ СИСТЕМ

Криптовалюти, блокчейн. NFT

Криптовалюта — це цифрові гроші, які існують у вигляді записів у блокчейні й не мають фізичної форми.

Транзакції підтверджуються учасниками мережі за допомогою криптографічних алгоритмів.
Переваги: децентралізація, прозорість операцій, швидкі міжнародні перекази.

NFT — це унікальні криптографічні токени в блокчейні, які засвідчують право власності на конкретний цифровий об'єкт.

На відміну від криптовалют, NFT не є взаємозамінними: кожен токен має свій ідентифікатор. Використовуються для продажу цифрового мистецтва, колекційних предметів, ігрових активів, але викликають дискусії щодо реальної цінності та спекуляцій.

Блокчейн — це розподілений реєстр транзакцій, який зберігається одночасно на багатьох комп'ютерах мережі.

Дані організовані в блоки, що послідовно з'єднані криптографічними хешами.

Кодування та шифрування даних

Кодування — це подання даних у певному форматі або системі запису (наприклад, кодування тексту в ASCII чи Unicode).

Основна мета кодування — забезпечити коректне зберігання й передавання інформації між системами.
Формати тексту: UTF-8, Unicode.
Коди символів (таблиці ASCII, emoji-коди).
Кодування зображень і відео (JPEG, MP4).

259

Дякуємо

за увагу!