МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №4

з дисципліни Спеціалізовані мови програмування

на тему

Розробка ASCII ART генератора для візуалізації 2D-фігур

Виконав:

студент групи РІ-21сп

Дмитрій Сас

Львів – 2024

**Мета виконання лабораторної роботи:** Створення Генератора ASCII-арту без використання зовнішніх бібліотек.

**План роботи**

**Завдання 1: Введення користувача**

Створіть програму Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

**Завдання 2: Набір символів**

Визначте набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

**Завдання 3: Розміри Art-у**

Запитайте у користувача розміри (ширина і висота) ASCII-арту, який вони хочуть створити. Переконайтеся, що розміри в межах керованого діапазону

**Завдання 4: Функція генерації Art-у**

Напишіть функцію, яка генерує ASCII-арт на основі введення користувача, набору символів та розмірів. Використовуйте введення користувача, щоб визначити, які символи використовувати для кожної позиції в Art-у.

**Завдання 5: Вирівнювання тексту**

Реалізуйте опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

**Завдання 6: Відображення мистецтва**

Відобразіть створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

**Завдання 7: Збереження у файл**

Додайте можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 8: Варіанти кольорів**

Дозвольте користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

**Завдання 9: Функція попереднього перегляду**

Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їх ASCII-арту перед остаточним збереженням

**Завдання 10: Інтерфейс, зрозумілий для користувача**

Створіть інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

**Текст програмної реалізації:**

**ascii\_generator.py:**# Завдання 1-10: Розробка ASCII ART генератора для візуалізації 2D-фігур

def print\_menu():

    print("\n--- ASCII Art Generator ---")

    print("1. Створити новий ASCII-арт")

    print("2. Вийти")

def get\_user\_choice(prompt, options):

    while True:

        choice = input(prompt).strip()

        if choice in options:

            return choice

        print("Невірний вибір. Спробуйте ще раз.")

def get\_ascii\_symbols():

    # Завдання 2: Набір символів

    return ['@', '#', '\*', '=', '-', ':', '.', ' ']

def get\_canvas\_size():

    # Завдання 3: Розміри Art-у

    width = int(input("Введіть ширину ASCII-арту (10-100): "))

    height = int(input("Введіть висоту ASCII-арту (5-50): "))

    width = max(10, min(width, 100))

    height = max(5, min(height, 50))

    return width, height

def align\_text(ascii\_art, width, align):

    # Завдання 5: Вирівнювання тексту

    aligned\_art = []

    for line in ascii\_art:

        if align == "центр":

            aligned\_art.append(line.center(width))

        elif align == "право":

            aligned\_art.append(line.rjust(width))

        else:

            aligned\_art.append(line.ljust(width))

    return aligned\_art

def generate\_ascii\_art(user\_text, width, height, symbols, align="ліво"):

    # Завдання 4: Функція генерації Art-у

    ascii\_art = []

    symbol\_count = len(symbols)

    for y in range(height):

        line = []

        for x in range(width):

            char\_index = (x + y) % len(user\_text) % symbol\_count

            line.append(symbols[char\_index])

        ascii\_art.append("".join(line))

    return align\_text(ascii\_art, width, align)

def display\_ascii\_art(ascii\_art):

    # Завдання 6: Відображення мистецтва

    for line in ascii\_art:

        print(line)

def save\_ascii\_art(ascii\_art, filename):

    # Завдання 7: Збереження у файл

    with open(filename, "w") as file:

        for line in ascii\_art:

            file.write(line + "\n")

    print(f"ASCII-арт збережено у файл {filename}")

def main():

    while True:

        print\_menu()

        choice = get\_user\_choice("Виберіть опцію (1-2): ", ['1', '2'])

        if choice == '2':

            print("Дякуємо за використання ASCII Art Generator!")

            break

        # Завдання 1: Введення користувача

        user\_text = input("Введіть текст для ASCII-арту: ")

        symbols = get\_ascii\_symbols()

        # Завдання 3: Розміри Art-у

        width, height = get\_canvas\_size()

        # Завдання 5: Вирівнювання тексту

        align\_choice = get\_user\_choice("Виберіть вирівнювання (ліво, центр, право): ", ['ліво', 'центр', 'право'])

        # Завдання 4: Генерація ASCII-арту

        ascii\_art = generate\_ascii\_art(user\_text, width, height, symbols, align=align\_choice)

        # Завдання 9: Функція попереднього перегляду

        preview\_choice = get\_user\_choice("Бажаєте переглянути попередній перегляд ASCII-арту? (так/ні): ", ['так', 'ні'])

        if preview\_choice == 'так':

            display\_ascii\_art(ascii\_art)

        # Завдання 7: Збереження у файл

        save\_choice = get\_user\_choice("Бажаєте зберегти ASCII-арт у файл? (так/ні): ", ['так', 'ні'])

        if save\_choice == 'так':

            filename = input("Введіть ім'я файлу для збереження: ")

            save\_ascii\_art(ascii\_art, filename)

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Результати тестування:**

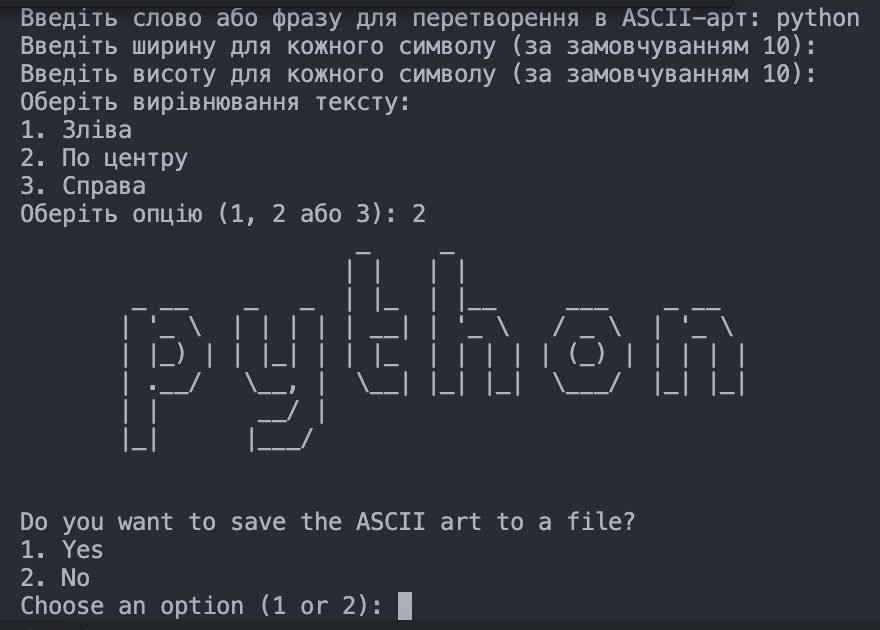


Рис. 1. Результат виведення в консоль

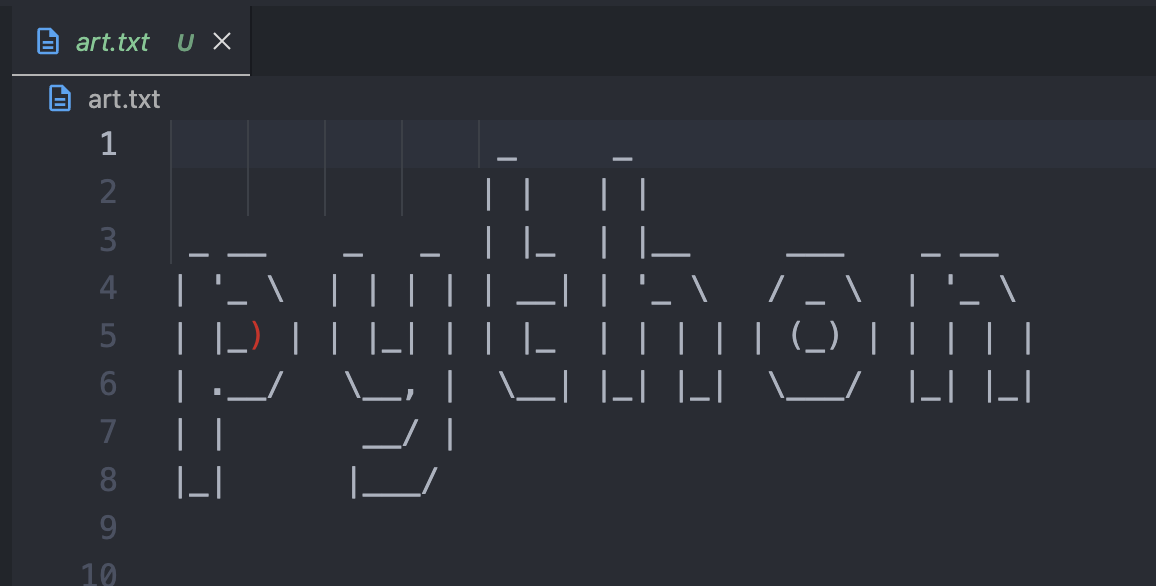


Рис. 2. Результат збережений у файл

**Висновки:** на цій лабораторній роботі було створено Генератор ASCII-арту без використання зовнішніх бібліотек.