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

Данило ДЯЧОК

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:**import os

class ASCIIGenerator:

def \_\_init\_\_(self):

self.char\_set = {}

self.load\_characters()

self.width = 0

self.height = 0

self.text = ""

self.alignment = "left"

def load\_characters(self):

for char in "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&\*()\_+-=[]{}|;':\",.<>?/ ":

if char.isalpha():

filename = f"lab\_4/alphabets/{char}.txt"

elif char.isdigit():

filename = f"lab\_4/numbers/{char}.txt"

else:

filename = f"lab\_4/symbols/{char}.txt"

if os.path.exists(filename):

with open(filename, 'r') as file:

self.char\_set[char] = [line.rstrip('\n') for line in file.readlines()]

else:

self.char\_set[char] = [' ' \* 10] \* 10 # Default to blank if file doesn't exist

def get\_user\_input(self):

self.text = input("Введіть слово або фразу для перетворення в ASCII-арт: ")

self.width = int(input("Введіть ширину для кожного символу (за замовчуванням 10): ") or 10)

self.height = int(input("Введіть висоту для кожного символу (за замовчуванням 10): ") or 10)

# Покращений UX для вирівнювання тексту

print("Оберіть вирівнювання тексту:")

print("1. Зліва")

print("2. По центру")

print("3. Справа")

alignment\_option = input("Оберіть опцію (1, 2 або 3): ").strip()

if alignment\_option == '1':

self.alignment = 'left'

elif alignment\_option == '2':

self.alignment = 'center'

elif alignment\_option == '3':

self.alignment = 'right'

else:

print("Невірний вибір. За замовчуванням використано вирівнювання зліва.")

self.alignment = 'left'

# Завдання 2-5: Генерація ASCII-арту

def generate\_art(self):

ascii\_lines = ['' for \_ in range(self.height)]

for char in self.text:

if char in self.char\_set:

char\_lines = self.char\_set[char]

else:

# Обробка невизначених символів за допомогою порожнього простору

char\_lines = [' ' \* self.width] \* self.height

# Перевірка, чи є достатньо рядків у char\_lines

if len(char\_lines) < self.height:

char\_lines += [' ' \* self.width] \* (self.height - len(char\_lines))

for i in range(self.height):

ascii\_lines[i] += char\_lines[i] + ' ' # Додаємо один пробіл між символами

return ascii\_lines

# Завдання 6: Виведення арту

def display\_art(self, ascii\_art):

for line in ascii\_art:

aligned\_line = self.\_align\_line(line, self.width \* len(self.text))

print(aligned\_line)

def \_align\_line(self, line, width):

if self.alignment == 'left':

return line.ljust(width)

elif self.alignment == 'center':

return line.center(width)

elif self.alignment == 'right':

return line.rjust(width)

else:

return line

# Новий метод: Зберегти арт у файл

def save\_art\_to\_file(self, ascii\_art):

filename = input("Введіть ім'я файлу для збереження ASCII-арту (наприклад, art.txt): ")

with open(filename, 'w') as file:

for line in ascii\_art:

file.write(line + '\n')

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

**Результати тестування:**

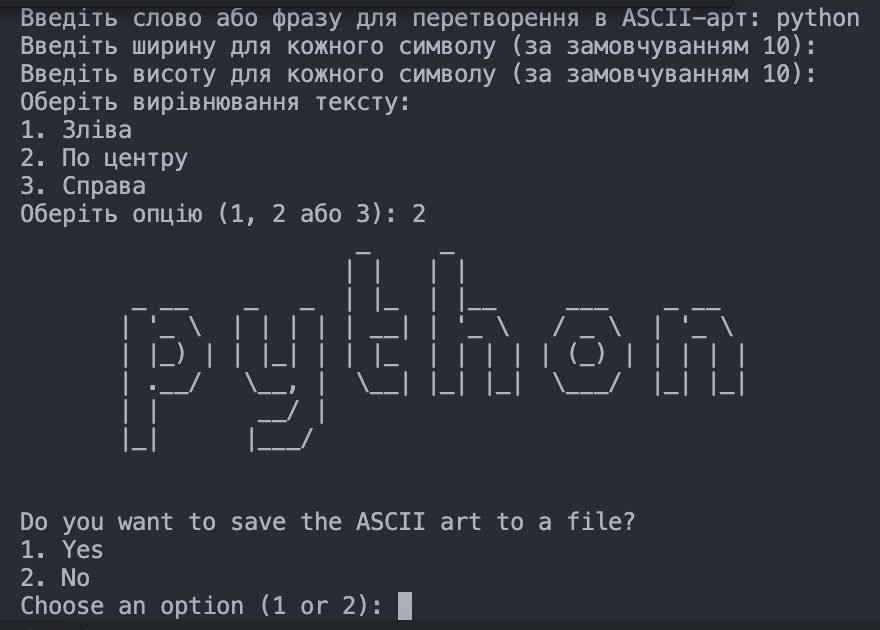


Рис. 1. Результат виведення в консоль

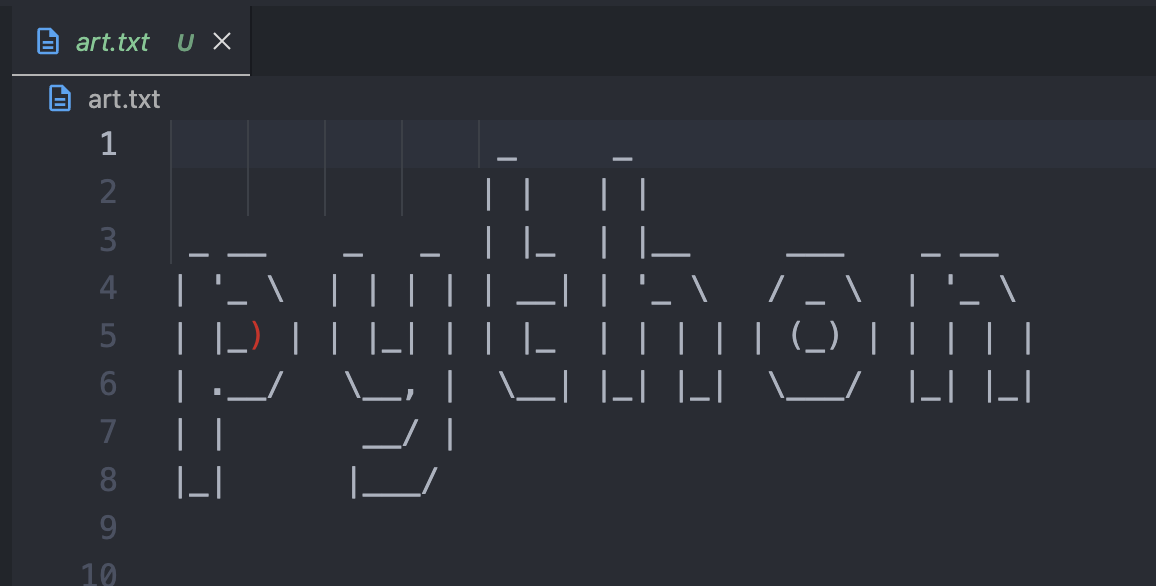


Рис. 2. Результат збережений у файл

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