МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж



Лабораторна робота №3

з дисципліни Спеціалізовані мови програмування

на тему

Розробка ASCII ART генератора для візуалізації текстових даних

Виконав:

студент групи РІ-21сп

Владислав ДМИТРЕНКО

Львів – 2024

**Мета:** Створення додатка Генератора ASCII-арту.

**План роботи**

**Завдання 1:** Створіть Python-програму, яка приймає введення користувача для слова або фрази, яку треба перетворити в ASCII-арт.

**Завдання 2:** Інтегруйте бібліотеку ASCII-арту (наприклад, pyfiglet або art) у вашу програму для генерації ASCII-арту з введення користувача

**Завдання 3:** Дозвольте користувачам вибирати різні стилі шрифтів для свого ASCII-арту. Надайте список доступних шрифтів та дозвольте їм вибрати один.

**Завдання 4:** Реалізуйте опцію вибору користувачем кольору тексту для їхнього ASCII-арту. Підтримуйте основний вибір кольорів (наприклад, червоний, синій, зелений).

**Завдання 5:** Переконайтеся, що створений ASCII-арт правильно відформатований та вирівнюється на екрані для зручності читання.

**Завдання 6:** Додайте функціональність для збереження створеного ASCII-арту у текстовому файлі, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

**Завдання 7:** Дозвольте користувачам вказувати розмір (ширина і висота) ASCII-арту, який вони хочуть створити. Масштабуйте текст відповідно.

**Завдання 8:** Дозвольте користувачам вибирати символи, які вони хочуть використовувати для створення ASCII-арту (наприклад, '@', '#', '\*', тощо).

**Завдання 9:** Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їхнього ASCII-арту перед остаточним збереженням.

**Завдання 10:** Створіть зручний для користувача інтерфейс командного рядку для додатка, щоб зробити його інтуїтивно зрозумілим та легким у використанні.

**Результати тестування:**

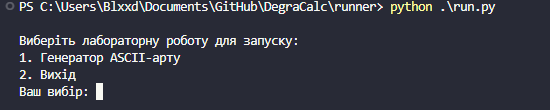


Рис. 1. Користувацький інтерфейс

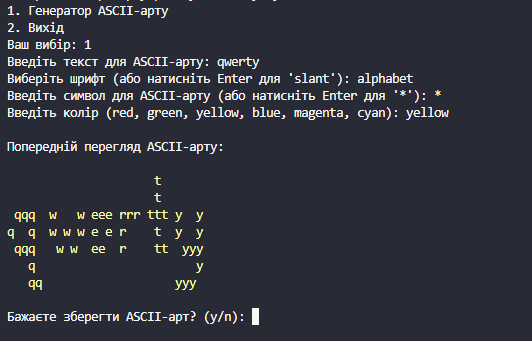


Рис. 2. Приклад роботи генератора артів

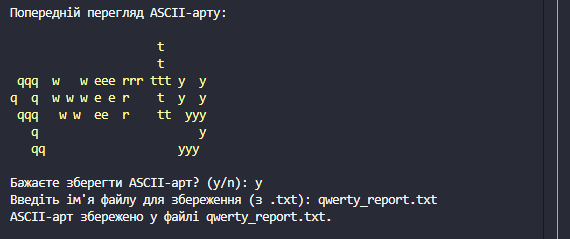
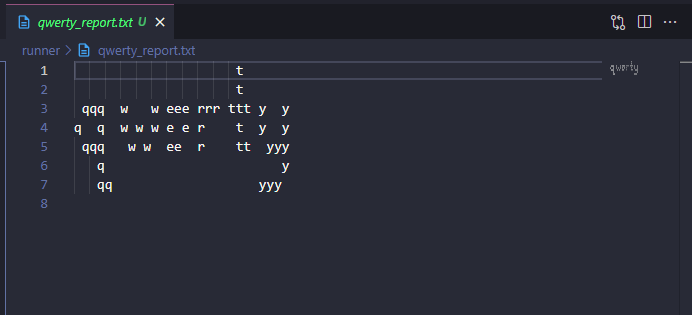


Рис. 3. Зберігання арту

  
Рис. 4. Результат збереження

Текст генератора артів

import pyfiglet

class AsciiArtGenerator:

def \_\_init\_\_(self, text, font='slant', symbol='\_'):

self.text = text

self.font = font

self.symbol = symbol

def generate\_ascii\_art(self):

try:

ascii\_art = pyfiglet.figlet\_format(self.text, font=self.font)

return ascii\_art.replace('\_', self.symbol)

except Exception as e:

raise ValueError("Помилка генерації ASCII-арту: " + str(e))

def available\_fonts(self):

return pyfiglet.FigletFont.getFonts()

Текст функції збереження

class FileManager:

@staticmethod

def save\_to\_file(content, filename):

try:

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

file.write(content)

except Exception as e:

raise IOError(f"Помилка збереження файлу: {filename}. Причина: {str(e)}")

**Висновки:** В ході виконання лабораторної роботи було створено універсальний генератор ASCII-арту, який дозволяє користувача налаштовувати свої творіння різними шрифтами, кольорами та символами.