Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Звіт про виконання лабораторної роботи №3

з дисципліни «Спеціалізовані мови програмування»

на тему «Розробка ASCII ART генератора для візуалізації текстових даних»

Виконала:  
студентка групи РІ-31  
Заплетнюк Ю.І.

Прийняв:  
Щербак С.С.

**Мета роботи:** створення додатка генератора ASCII-арту.

**План роботи:**

Завдання 1: Введення користувача

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

Завдання 2: Бібліотека ASCII-арту

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

Завдання 3: Вибір шрифту

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

Завдання 4: Колір тексту

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

Завдання 5: Форматування виводу

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

Завдання 6: Збереження у файл

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

Завдання 7: Розмір ARTу

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

Завдання 8: Вибір символів

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

Завдання 9: Функція попереднього перегляду

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

Завдання 10: Інтерфейс, зрозумілий для користувача

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

**Код програмної реалізації**

**art\_generatop.py**

import pyfiglet

from art import text2art

from colorama import Fore, Style

from classes.user\_input import UserInput

from classes.art import Art

class ArtGenerator:

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

self.available\_fonts = pyfiglet.FigletFont.getFonts()

def run(self):

user\_input = UserInput.get\_input("Enter a word or phrase to convert to ASCII art: ")

font = UserInput.select\_font(self.available\_fonts)

color = UserInput.select\_color()

width = UserInput.get\_input("Enter the width for the ASCII art: ")

height = UserInput.get\_input("Enter the height for the ASCII art: ")

custom\_char = UserInput.get\_input("Enter a character to use for ASCII art: ")

ascii\_art = Art(user\_input, font, color, width, height, custom\_char)

colored\_art = ascii\_art.get\_colored\_art()

Art.preview\_art(colored\_art)

if UserInput.get\_input("Do you want to save ASCII art? (y/n): ").lower() == 'y':

ascii\_art.save\_to\_file("ascii.txt")

**art.py**

import pyfiglet

from art import text2art

from colorama import Fore, Style

class Art:

def \_\_init\_\_(self, user\_input, font, color, width, height, custom\_char):

self.user\_input = user\_input

self.font = font

self.color = color

self.width = int(width)

self.height = int(height)

self.custom\_char = custom\_char

self.art = self.generate\_art()

def generate\_art(self):

return pyfiglet.figlet\_format(self.user\_input, font=self.font, width=self.width)

return art.replace(' ', self.custom\_char)

def get\_colored\_art(self):

return self.color + self.art + Style.RESET\_ALL

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

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

file.write(self.art)

print(f"ASCII-art saved у '{filename}'")

@staticmethod

def preview\_art(ascii\_art):

print("\nASCII art preview:\n")

print(ascii\_art)

**user\_input.py**

from colorama import Fore

class UserInput:

@staticmethod

def get\_input(prompt):

return input(prompt)

@staticmethod

def select\_font(available\_fonts):

print("Available fonts:")

for i, font in enumerate(available\_fonts[:10]):

print(f"{i + 1}. {font}")

choice = int(UserInput.get\_input("Choose a font number: ")) - 1

return available\_fonts[choice]

@staticmethod

def select\_color():

print("Available colors:")

print("1. Red")

print("2. Blue")

print("3. Green")

choice = int(UserInput.get\_input("Choose fonr color number: "))

if choice == 1:

return Fore.RED

elif choice == 2:

return Fore.BLUE

elif choice == 3:

return Fore.GREEN

else:

return Fore.WHITE

**runner.py**

import sys

import os

sys.path.append(os.path.join(os.path.dirname(os.path.abspath(\_\_file\_\_)), '..'))

from classes.user\_input import UserInput

from classes.art\_generatop import ArtGenerator

from classes.art import Art

artgen = ArtGenerator()

artgen.run()

На рис. 1 показано результат виконання завдання.



Рис. 1. Результат виконання завдання

**Висновки:** виконавши лабораторну роботу, було cтворено універсальний генератор ASCII-арту, який дозволить користувачам налаштовувати творіння з різними шрифтами, кольорами, розмірами та символами. Проєкт надав практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.