МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №3

з дисципліни “Спеціалізовані мови програмування”

Виконала:

студентка групи ІТ-31

Юлія ДОБОШ

Прийняв:

доцент кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**Тема роботи:** Розробка ASCII ART генератора для візуалізації текстових даних

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

Хід 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.

Детально код можна переглянути тут : <https://github.com/YuliiaDobosh/Python.git>

Код програми :

import logging

import pyfiglet

import termcolor

from .interfaces import ascii\_art\_generator

from .interfaces import file\_saver

from .interfaces import input\_handler

from .interfaces import output\_handler

class ASCIIArtGenerator(ascii\_art\_generator.ASCIIArtGenerator):

"""Class for generating ASCII art."""

def \_\_init\_\_(self, input\_handler\_instance: input\_handler.InputHandler,

output\_handler\_instance: output\_handler.OutputHandler,

file\_saver\_instance: file\_saver.FileSaver):

"""Initialize the ASCIIArtGenerator."""

self.input\_handler = input\_handler\_instance

self.output\_handler = output\_handler\_instance

self.file\_saver = file\_saver\_instance

logging.info("ASCIIArtGenerator ініціалізовано")

def generate(self):

"""Generate ASCII art based on user input.

This method prompts the user for text, font, and color preferences,

generates ASCII art using pyfiglet and termcolor, and optionally

displays a preview or saves the generated art to a file.

"""

logging.info("Початок генерації ASCII арту")

text = self.input\_handler.get\_text()

font = self.input\_handler.select\_font()

color = self.input\_handler.select\_color()

ascii\_art = pyfiglet.Figlet(font=font)

art\_text = ascii\_art.renderText(text)

art\_text = termcolor.colored(art\_text, color)

if self.input\_handler.yes\_or\_no ("Do you like to see a preview? "):

self.output\_handler.display(art\_text)

if self.input\_handler.yes\_or\_no ("Do you like to save an art? "):

self.file\_saver.save\_to\_file(art\_text, f"{self.input\_handler.write\_answer('Enter file name: ')}.txt")

logging.info("ASCII арт створено")

def run(self):

"""Run the ASCII art generation process in a loop.

This method repeatedly calls the generate method and prompts the user

if they want to continue generating ASCII art.

"""

while True:

self.generate()

continue\_choice = self.input\_handler.yes\_or\_no("Do you want to continue? (yes/no): ")

if not continue\_choice:

logging.info("Користувач завершив роботу з ASCIIArtGenerator")

break

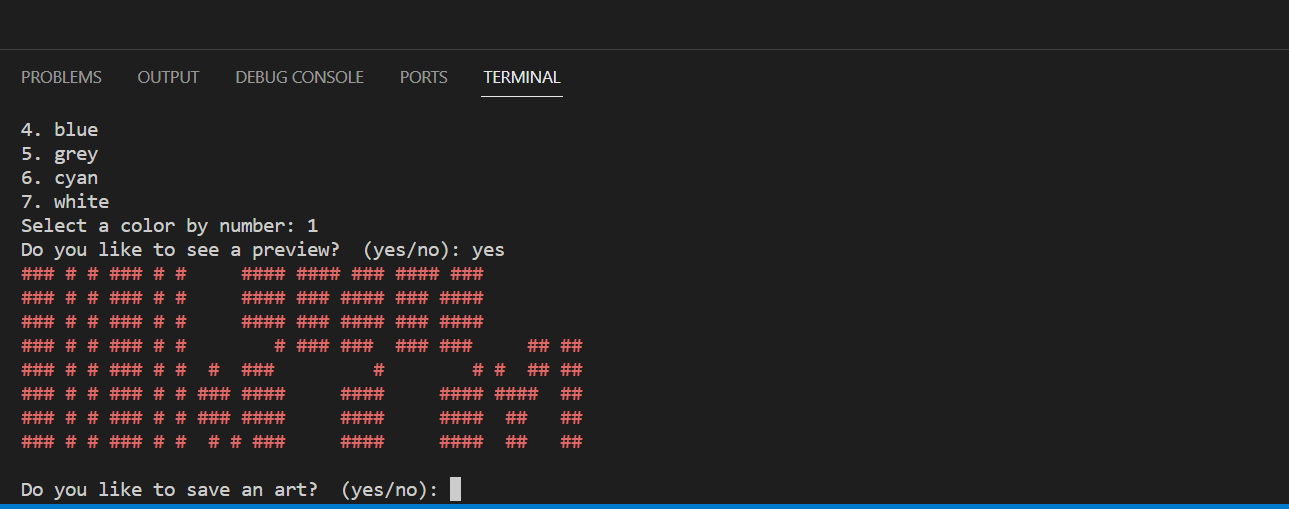


Рис.1 Результат виконання роботи

**Висновок**: Виконуючи лабораторну роботу я створила універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надав практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.