Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 3

**“ Розробка ASCII ART генератора для візуалізації текстових даних ”**

з дисципліни **“** **Спеціалізовані мови програмування”**

Виконав:

студент групи ІТ-32

ОЛЕНЮК О. М.

Прийняв:

ЩЕРБАК С. С.

**Львів – 2023**

**Мета**: 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 програми:**

function.py

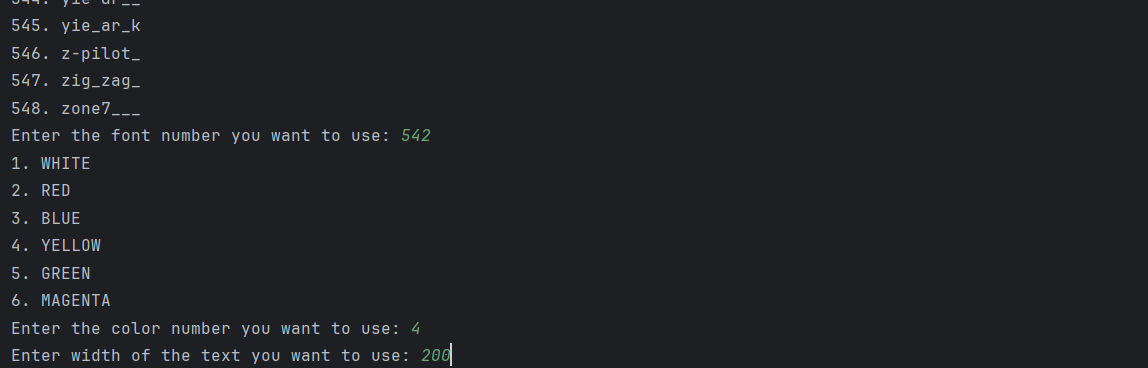
import pyfiglet # Import the library to generate ASCII-Art  
from colorama import Fore # Import for Color Leaning Text  
  
# Creating a vocabulary with all the fonts available  
fonts = dict(enumerate(sorted(pyfiglet.FigletFont.getFonts())))  
# Creating a vocabulary with some colors for the text  
colors = {  
 1: 'WHITE',  
 2: 'RED',  
 3: 'BLUE',  
 4: 'YELLOW',  
 5: 'GREEN',  
 6: 'MAGENTA'  
}  
  
  
# Function to output all the fonts available  
def display\_fonts() -> None:  
 for i in fonts:  
 print(str(i) + ". " + fonts[i])  
  
  
# Function to output all colors available  
def display\_colors() -> None:  
 for i in colors:  
 print(str(i) + ". " + colors[i])  
  
  
# Function to generates ASCII-Art from the text entered  
def get\_text(text, font, color\_position, width) -> str:  
 fig = pyfiglet.Figlet(font)  
 fig.width = width  
 formatted\_text = fig.renderText(text)  
 return getattr(Fore, colors[color\_position]) + formatted\_text  
  
  
# Function to write text in file  
def write\_in\_file(file\_path, text) -> None:  
 with open(file\_path, "w", encoding="utf-8") as file:  
 file.write(text)  
  
  
# Function to read text from file  
def read\_from\_file(file\_path) -> str:  
 with open(file\_path, "r", encoding="utf-8") as file:  
 return file.read()

console.output.py

import functions  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 while True:  
 try:  
 while True:  
 initial\_text = str(input("Enter a word or phrase you want to turn into ASCII-Art: "))  
 if not initial\_text.isascii():  
 print("Error! The text should only contain ASCII characters")  
 continue  
 else:  
 break  
  
 while True:  
 functions.display\_fonts()  
 font\_position = int(input("Enter the font number you want to use: "))  
 if font\_position not in functions.fonts:  
 print("Error! Invalid font number. Please enter again.")  
 continue  
 else:  
 break  
  
 while True:  
 functions.display\_colors()  
 color\_position = int(input("Enter the color number you want to use: "))  
 if color\_position not in functions.colors:  
 print("Error! Invalid color number. Please enter again.")  
 continue  
 else:  
 break  
  
 except ValueError as e:  
 print("Error! Please enter a number.")  
 continue  
  
 try:  
 while True:  
 width = int(input("Enter width of the text you want to use: "))  
 if width <= 0:  
 print("Width must be a positive value. Please enter a positive value.")  
 else:  
 break  
 except ValueError as e:  
 print("Error! Please enter correct width (It must be number).")  
 continue  
  
 modified\_text = functions.get\_text(initial\_text, functions.fonts[font\_position], color\_position, width)  
 print("The result of ASCII art: ")  
 print(modified\_text)  
  
 save\_to\_file = input("Save the text in the file? (yes/no): ")  
 if save\_to\_file.lower() == "yes":  
 filename = input("Enter the file name to save the text: ")  
 functions.write\_in\_file(filename, modified\_text)  
 print(f"Text saved in the file {filename}")  
  
 if input(  
 "Continue to write? (yes/no): ").lower() == "yes":  
 continue  
 else:  
 break

**Результат програми:**

На рис.1, 2, 3 зображений результат виконання роботи у програмі.

Рис.1

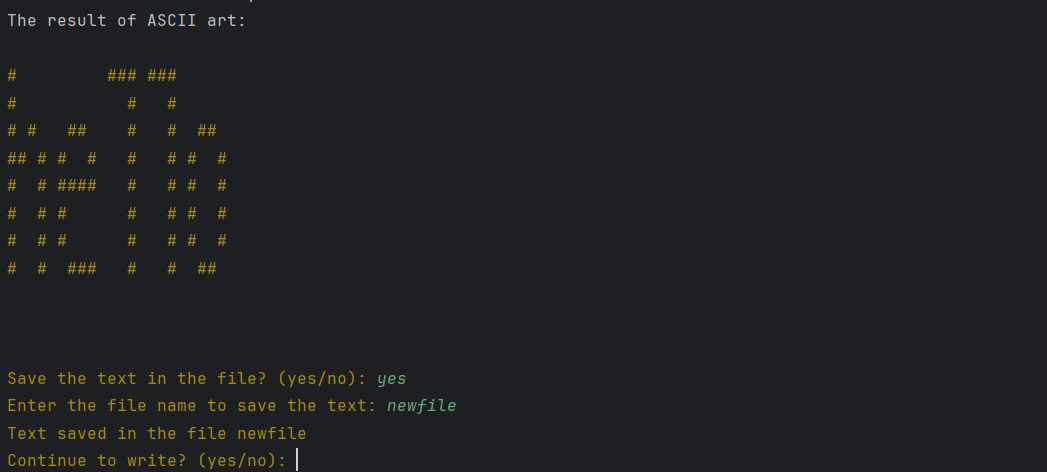


Рис.2

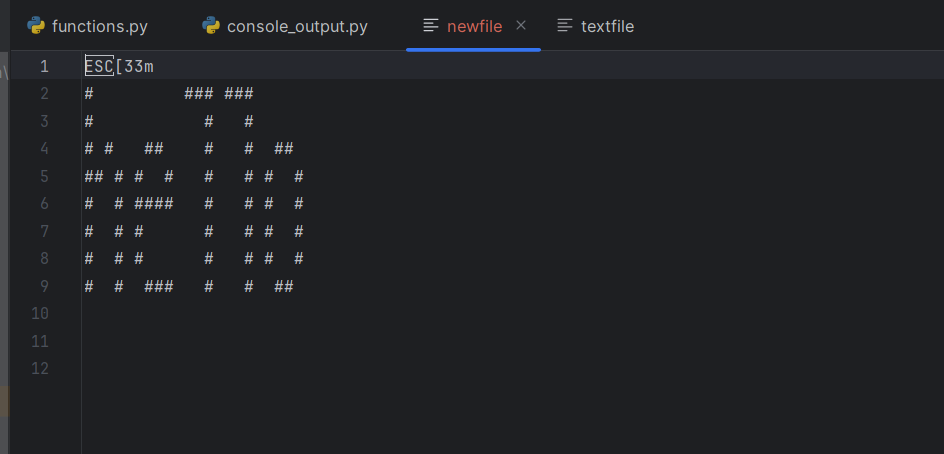


Рис.3

**Посилання на репозиторій**: <https://github.com/Oleksandr2004Oleniuk/SMP.git>

**Висновок**. у цій лабораторній роботі я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Я отримав практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.