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,  
Пятницька Вікторія

Прийняв:

Щербак С.С.

**Мета.** 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.

**Код програми:**

***./ascii.py****:*

import os

import pyfiglet

from termcolor import colored

def user\_input():

return input("Enter the text to convert to ASCII art: ")

def generate\_ascii\_art(text, font, custom\_char=None):

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

art = ascii\_art.renderText(text)

if custom\_char:

for old\_char in set(art):

if old\_char not in ['\n', ' ']:

art = art.replace(old\_char, custom\_char)

return art

def choose\_font():

fonts = ['slant', '3-d', '5lineoblique', 'standard', 'big', 'block', 'bubble', 'digital', 'isometric1', 'banner']

print("Available fonts:")

for i, font in enumerate(fonts):

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

font\_choice = int(input("Choose a font by number: "))

return fonts[font\_choice - 1]

def choose\_color():

colors = ['red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']

print("Available colors:")

for i, color in enumerate(colors):

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

color\_choice = int(input("Choose a color by number: "))

return colors[color\_choice - 1]

def choose\_symbols():

custom\_char = input("Enter the character you'd like to use instead of default ASCII characters (optional): ")

return custom\_char if custom\_char else None

def display\_ascii\_art(art, color):

print(colored(art, color))

def save\_to\_file(art, filename="ascii\_art"):

filename += '.txt'

while os.path.exists(filename):

choice = input(f"The file '{filename}' already exists. Overwrite it? (Enter 'yes' or any character if no): ").strip().lower()

if choice == 'yes':

break

else:

filename = input("Please enter a new filename : ").strip()

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

file.write(art)

print(f"ASCII art saved to {filename}")

def resize\_ascii\_art(art, width\_scale, height\_scale):

lines = art.splitlines()

scaled\_art = []

for line in lines:

scaled\_line = ''.join([char \* width\_scale for char in line])

scaled\_art.append(scaled\_line)

final\_art = []

for line in scaled\_art:

for \_ in range(height\_scale):

final\_art.append(line)

return "\n".join(final\_art)

def preview\_art(art, color):

print("\nPreview of your ASCII art:")

display\_ascii\_art(art, color)

def run\_ascii():

text = user\_input()

font = choose\_font()

color = choose\_color()

custom\_char = choose\_symbols()

ascii\_art = generate\_ascii\_art(text, font, custom\_char)

try:

width\_scale = int(input("Enter width scale factor (1 for normal, 2 for double width): "))

height\_scale = int(input("Enter height scale factor (1 for normal, 2 for double height): "))

if width\_scale not in [1, 2]:

raise ValueError("Invalid choice for width scale!")

if height\_scale not in [1, 2]:

raise ValueError("Invalid choice for height scale!")

except ValueError as e:

print(f"Error: {e}")

return

resized\_art = resize\_ascii\_art(ascii\_art, width\_scale, height\_scale)

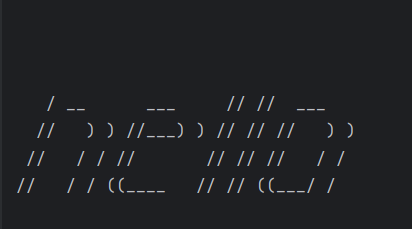
preview\_art(resized\_art, color)

save\_choice = input("Do you want to save the ASCII art to a file? (Enter 'yes' or any character if no): ").strip().lower()

if save\_choice == 'yes':

filename = input("Enter filename: ")

save\_to\_file(resized\_art, filename)

**

*Рис.1.Приклад виконання програми*

**Висновки:** внаслідок виконання лабораторної роботи я створила додаток Генератора ASCII-арту.