Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 3

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“Створення додатка Генератора ASCII-арту”

Виконав:  
 студент РІ-31

Яворський О. В.

Прийняв:  
к.т.н., доцент  
 Щербак С.С.

Львів – 2024

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

**Завдання лабораторної 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.

Реалізація програми:

import pyfiglet

from termcolor import colored

from art import text2art

# Завдання 1: Введення користувача

def get\_user\_input():

return input("Введіть слово або фразу для перетворення в ASCII-арт: ")

# Завдання 2: Генерація ASCII-арту

def generate\_ascii\_art(text, font='standard', width=80):

return pyfiglet.figlet\_format(text, font=font, width=width)

# Завдання 3: Вибір шрифту

def list\_fonts():

return pyfiglet.FigletFont.getFonts()

def get\_user\_font\_choice():

available\_fonts = list\_fonts()

print("Доступні шрифти:", available\_fonts)

font = input("Виберіть шрифт для ASCII-арту: ")

return font if font in available\_fonts else 'standard'

# Завдання 4: Колір тексту

def print\_colored\_ascii\_art(ascii\_art, color='white'):

print(colored(ascii\_art, color))

def get\_user\_color\_choice():

available\_colors = ['red', 'green', 'yellow', 'blue', 'magenta', 'cyan', 'white']

print("Доступні кольори:", available\_colors)

color = input("Виберіть колір для ASCII-арту: ")

return color if color in available\_colors else 'white'

# Завдання 5: Форматування виводу

# ASCII-арт автоматично відформатований за допомогою pyfiglet.

# Завдання 6: Збереження у файл

def save\_to\_file(ascii\_art, filename='ascii\_art.txt'):

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

file.write(ascii\_art)

print(f"ASCII-арт збережено у файл: {filename}")

# Завдання 7: Вказівка розміру ARTу

def get\_user\_size():

width = input("Введіть ширину для ASCII-арту (за замовчуванням 80): ")

return int(width) if width.isdigit() else 80

# Завдання 8: Вибір символів

def generate\_custom\_ascii\_art(text, symbol='@'):

return text2art(text, chr\_ignore=True, decoration=symbol)

def get\_user\_symbol\_choice():

symbol = input("Введіть символ для створення ASCII-арту (наприклад '@'): ")

return symbol if symbol else '@'

# Основна програма

def main():

user\_input = get\_user\_input()

print("\n=== Вибір шрифту ===")

font\_choice = get\_user\_font\_choice()

print("\n=== Вибір кольору ===")

color\_choice = get\_user\_color\_choice()

print("\n=== Вибір розміру ===")

width\_choice = get\_user\_size()

print("\n=== Вибір символу ===")

symbol\_choice = get\_user\_symbol\_choice()

# Створення ASCII-арту з шрифтом, кольором, і символом

if symbol\_choice != '@': # Якщо символ обрано інший, генеруємо кастомний ASCII-арт

ascii\_art = generate\_custom\_ascii\_art(user\_input, symbol\_choice)

else:

ascii\_art = generate\_ascii\_art(user\_input, font=font\_choice, width=width\_choice)

# Виводимо ASCII-арт з кольором

print\_colored\_ascii\_art(ascii\_art, color\_choice)

# Збереження в файл

save\_choice = input("Чи бажаєте зберегти ASCII-арт у файл? (y/n): ")

if save\_choice.lower() == 'y':

filename = input("Введіть назву файлу для збереження (без розширення): ") + '.txt'

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

if name == "main":

main()

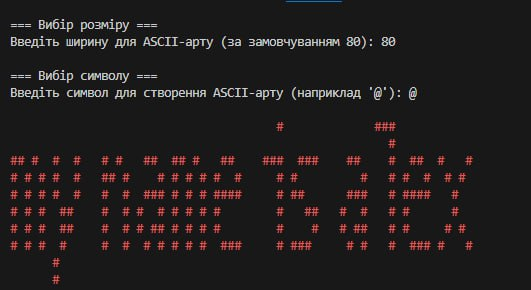


Рис. 1 (Резульати роботи програми)

**Висновок**

Виконуючи ці завдання, я створив 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.