Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №3

з дисципліни «Спеціалізовані мови програмування»

на тему

«Розробка ASCII ART генератора для візуалізації текстових даних»

Виконав:

Костюк В. О.

Перевірив:

Щербак С. С.

Львів 2024

**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.

**Реалізація:**

**Файл ascii\_art.py:**  
import pyfiglet

# Генерація ASCII-арту

def generate\_ascii\_art(text, font):

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

# Вибір шрифту

def choose\_font():

    fonts = pyfiglet.FigletFont.getFonts()

    print("Доступні шрифти:")

    for i, font in enumerate(fonts):

        print(f"{i + 1}. {font}")

    choice = int(input("\nВиберіть шрифт за номером: ")) - 1

    return fonts[choice]

# Масштабування ASCII-арту

def resize\_ascii\_art(text, font, width):

    figlet = pyfiglet.Figlet(font=font)

    ascii\_art\_lines = figlet.renderText(text).splitlines()

    resized\_ascii\_art = ""

    for line in ascii\_art\_lines:

        wrapped\_line = [line[i:i+width] for i in range(0, len(line), width)]

        resized\_ascii\_art += "\n".join(wrapped\_line) + "\n"

    return resized\_ascii\_art

# Заміна символів

def replace\_characters(ascii\_art, old\_char, new\_char):

    return ascii\_art.replace(old\_char, new\_char)

**Файл color\_output.py:**

from termcolor import colored

# Вибір кольору

def choose\_color():

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

    print("\nДоступні кольори:")

    for i, color in enumerate(colors):

        print(f"{i + 1}. {color}")

    choice = int(input("\nВиберіть колір за номером: ")) - 1

    return colors[choice]

# Виведення ASCII-арту

def print\_ascii\_art(ascii\_art, color):

    print(colored(ascii\_art, color))

# Попередній перегляд ASCII-арту

def preview\_ascii\_art(ascii\_art, color):

    print("\nПопередній перегляд вашого ASCII-арту:")

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

**Файл file\_operations.py:**

# Збереження ASCII-арту у файл

def save\_to\_file(ascii\_art):

    filename = "./test.txt"  # Вказуємо шлях без input()

    with open(filename, 'w') as file:

        file.write(ascii\_art)

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

**Файл main.py:**

from lab3.user\_input import get\_user\_input

from lab3.ascii\_art import generate\_ascii\_art, choose\_font, resize\_ascii\_art, replace\_characters

from lab3.color\_output import choose\_color, print\_ascii\_art, preview\_ascii\_art

from lab3.file\_operations import save\_to\_file

def main():

    # Введення користувача

    text = get\_user\_input()

    # Вибір шрифту

    font = choose\_font()

    # Генерація ASCII-арту

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

    # Вибір кольору

    color = choose\_color()

    # Попередній перегляд

    preview\_ascii\_art(ascii\_art, color)

    while True:

        print("\nЩо ви хочете зробити далі?")

        print("1. Змінити розмір ASCII-арту")

        print("2. Замінити символи в ASCII-арті")

        print("3. Вивести ASCII-арт")

        print("4. Зберегти ASCII-арт у файл")

        print("5. Вийти")

        choice = input("Виберіть опцію (1-5): ")

        if choice == '1':

            # Масштабування

            width = int(input("Введіть ширину: "))

            ascii\_art = resize\_ascii\_art(text, font, width)

            preview\_ascii\_art(ascii\_art, color)

        elif choice == '2':

            # Заміна символів

            old\_char = input("Введіть символ, який хочете замінити: ")

            new\_char = input("Введіть новий символ: ")

            ascii\_art = replace\_characters(ascii\_art, old\_char, new\_char)

            preview\_ascii\_art(ascii\_art, color)

        elif choice == '3':

            # Виведення ASCII-арту

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

        elif choice == '4':

            # Збереження у файл

            save\_to\_file(ascii\_art)

        elif choice == '5':

            # Вихід з програми

            print("Дякуємо за використання програми!")

            break

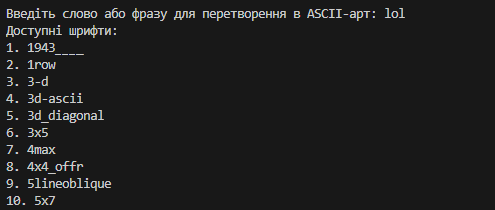
        else:

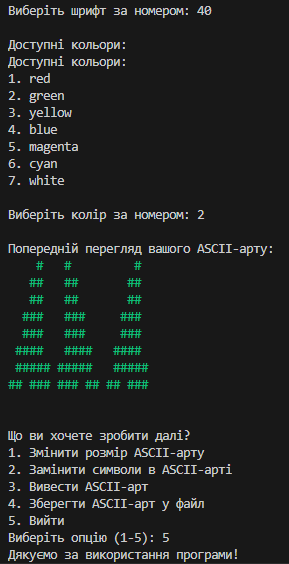
            print("Невірний вибір. Спробуйте ще раз.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Результат виконання:**





**Висновок:** на цій лабораторній роботі я створив універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами