МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж



Лабораторна робота №3

з дисципліни «Спеціалізовані мови програмування»

на тему «Розробка ASCII ART генератора для візуалізації текстових даних»

Виконала студентка

групи РІ-32

Титиш О.І.

Прийняв:

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.

**Код програми**

art\_generator.py

import pyfiglet  
from colorama import Fore, Style  
  
*# Доступні кольори з бібліотеки colorama*colors = {  
 'red': Fore.RED,  
 'green': Fore.GREEN,  
 'blue': Fore.BLUE,  
 'yellow': Fore.YELLOW,  
 'magenta': Fore.MAGENTA,  
 'cyan': Fore.CYAN,  
 'white': Fore.WHITE  
}  
  
def generate\_ascii\_art(text, font, color, width=80, height=None):  
 *"""Генерація ASCII-арту з використанням вибраного шрифту, кольору, ширини і висоти"""* ascii\_art = pyfiglet.figlet\_format(text, font=font, width=width)  
 color\_code = colors.get(color.lower(), Fore.WHITE) *# Встановлюємо білий за замовчуванням* colored\_ascii\_art = f"{color\_code}{ascii\_art}{Style.RESET\_ALL}"  
 return colored\_ascii\_art  
  
def generate\_custom\_symbol\_art(text, font, width, symbol):  
 *"""Генерація ASCII-арту з кастомними символами"""* ascii\_art = pyfiglet.figlet\_format(text, font=font, width=width)  
 custom\_art = ascii\_art.replace('#', symbol) *# Замінюємо всі '#' на вибраний символ* return custom\_art

color\_manager.py

available\_colors = ['red', 'green', 'blue', 'yellow', 'magenta', 'cyan', 'white']  
  
def get\_available\_colors():  
 *"""Отримання списку доступних кольорів"""* return available\_colors  
  
def is\_valid\_color(color):  
 *"""Перевірка, чи є колір у списку доступних"""* return color in available\_colors

font\_manager.py

import pyfiglet  
  
def get\_available\_fonts():  
 *"""Отримання списку доступних шрифтів"""* return pyfiglet.FigletFont.getFonts()  
  
def is\_valid\_font(font):  
 *"""Перевірка, чи шрифт існує в списку доступних"""* return font in pyfiglet.FigletFont.getFonts()

file\_handler.py

import os  
  
def save\_to\_file(content, filename):  
 *"""Збереження ASCII-арту у файл"""* if not os.path.exists('assets'):  
 os.makedirs('assets')  
  
 with open(f'assets/{filename}.txt', 'w') as file:  
 file.write(content)  
  
  
def load\_from\_file(filename):  
 *"""Завантаження ASCII-арту з файлу"""* with open(f'assets/{filename}.txt', 'r') as file:  
 return file.read()

utils.py

def get\_user\_input(prompt, valid\_options=None):  
 *"""Отримання введення користувача з валідацією"""* while True:  
 user\_input = input(prompt)  
 if valid\_options and user\_input not in valid\_options:  
 print(f"Невірний вибір. Доступні варіанти: {', '.join(valid\_options)}")  
 else:  
 return user\_input

console\_interface.py

from BLL.art\_generator\_LAB3 import generate\_ascii\_art, generate\_custom\_symbol\_art  
from BLL.font\_manager\_LAB3 import get\_available\_fonts, is\_valid\_font  
from BLL.color\_manager\_LAB3 import get\_available\_colors, is\_valid\_color  
from DAL.file\_handler\_LAB3 import save\_to\_file  
from Shared.utils\_LAB3 import get\_user\_input  
  
  
def start\_console\_interface():  
 print("ASCII ART Генератор")  
  
 *# Введення тексту* text = input("Введіть текст для ASCII-арту: ")  
  
 *# Вибір шрифту* fonts = get\_available\_fonts()  
 print("Доступні шрифти:", fonts[:20]) *# Показуємо 20 перших шрифтів* font = get\_user\_input("Виберіть шрифт: ", fonts)  
  
 *# Вибір кольору* colors = get\_available\_colors()  
 print("Доступні кольори:", colors)  
 color = get\_user\_input("Виберіть колір: ", colors)  
  
 *# Введення ширини та висоти* width = int(input("Введіть бажану ширину (наприклад, 100): "))  
 height = input("Введіть бажану висоту: ") or None  
  
 *# Вибір символа для ASCII-арту* use\_custom\_symbol = get\_user\_input("Використовувати кастомний символ замість '#' для ASCII-арту? (y/n): ",  
 ['y', 'n'])  
 if use\_custom\_symbol == 'y':  
 symbol = input("Введіть символ, який буде використовуватися для заміни '#': ")  
 ascii\_art = generate\_custom\_symbol\_art(text, font, width, symbol)  
 else:  
 ascii\_art = generate\_ascii\_art(text, font, color, width=width, height=height)  
  
 *# Попередній перегляд ASCII-арту* print("Попередній перегляд ASCII-арту:")  
 print(ascii\_art)  
  
 *# Збереження у файл* save\_choice = get\_user\_input("Зберегти результат у файл? (y/n): ", ['y', 'n'])  
 if save\_choice == 'y':  
 filename = input("Введіть ім'я файлу: ")  
 save\_to\_file(ascii\_art, filename)  
 print(f"ASCII-арт збережено у файл: assets/{filename}.txt")

runner.py

from UI.console\_interface\_LAB3 import start\_console\_interface

if \_\_name\_\_ == "\_\_main\_\_":  
 start\_console\_interface()

**Результат виконання програми**

****

*Рисунок 1. Результат виконання*

Посилання на GitHub:

**Висновок:** При виконанні даної лабораторної роботи, я створила універсальний Генератор ASCII-арту, який дозволяє користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надає практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.