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:

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

Виконав студент РІ-31

**Шапошніков Дмитро**

Прийняв: 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.

Файл запуску:

import pyfiglet

from termcolor import colored

import os

class AsciiArtGenerator:  # Encapsulation

    def \_\_init\_\_(self):

        self.user\_input = ""

        self.selected\_font = "slant"

        self.selected\_color = "white"

        self.ascii\_art = ""

    def get\_user\_input(self):

        self.user\_input = input("Введіть слово або фразу для генерації ASCII-арту: ").strip()

        while not self.user\_input:

            self.user\_input = input("Введення не може бути порожнім. Введіть слово або фразу: ").strip()

    def choose\_font(self):

        supported\_fonts = [

            "big", "block", "bulbhead", "chunky", "coinstak", "colossal", "computer",

            "doom", "epic", "isometric1", "isometric2", "isometric3", "isometric4",

            "larry3d", "nancyj-fancy", "rectangles", "roman", "rounded", "slant", "standard"

        ]

        print("Доступні шрифти (що підтримують українські символи):")

        for i, font in enumerate(supported\_fonts):

            figlet = pyfiglet.Figlet(font=font)

            example\_text = figlet.renderText("hello")

            print(f"{i + 1}. {font}\n{example\_text}")

        while True:

            try:

                font\_choice = int(input(f"Виберіть шрифт (1-{len(supported\_fonts)}): ")) - 1

                if 0 <= font\_choice < len(supported\_fonts):

                    self.selected\_font = supported\_fonts[font\_choice]

                    break

                else:

                    print("Невірний вибір. Виберіть номер зі списку.")

            except ValueError:

                print("Будь ласка, введіть коректне число.")

    def choose\_color(self):

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

        print("Доступні кольори: ", ", ".join(colors))

        self.selected\_color = input("Виберіть колір для вашого ASCII-арту: ").lower().strip()

        while self.selected\_color not in colors:

            self.selected\_color = input("Невірний колір. Будь ласка, виберіть один із запропонованих кольорів: ").lower().strip()

    def generate\_art(self):

        try:

            figlet = pyfiglet.Figlet(font=self.selected\_font)

            self.ascii\_art = figlet.renderText(self.user\_input)

        except pyfiglet.FontNotFound:

            print("Шрифт не знайдено. Використовується стандартний шрифт.")

            self.selected\_font = "slant"

            figlet = pyfiglet.Figlet(font=self.selected\_font)

            self.ascii\_art = figlet.renderText(self.user\_input)

    def preview\_art(self):

        print("\nПопередній перегляд:")

        print(colored(self.ascii\_art, self.selected\_color))

    def save\_to\_file(self):

        while True:

            try:

                file\_name = input("Введіть назву файлу для збереження (без розширення): ").strip() + ".txt"

                if not file\_name:

                    print("Назва файлу не може бути порожньою.")

                elif os.path.exists(file\_name):

                    overwrite = input(f"Файл {file\_name} вже існує. Перезаписати його? (y/n): ").lower().strip()

                    if overwrite == 'y':

                        break

                    else:

                        continue

                else:

                    break

            except Exception as e:

                print(f"Сталася помилка: {e}")

        try:

            with open(file\_name, "w", encoding='utf-8') as file:

                file.write(self.ascii\_art)

            print(f"ASCII-арт збережено у файл {file\_name}")

        except IOError as e:

            print(f"Помилка при збереженні файлу: {e}")

class ResizableAsciiArtGenerator(AsciiArtGenerator):

    def resize\_art(self, width, height):

        resized\_art = self.ascii\_art.split('\n')

        resized\_art = [line[:width] for line in resized\_art][:height]

        self.ascii\_art = '\n'.join(resized\_art)

    def changeSymbol(self, art, symbol):

        for char in art:

            if char != '\n' and char != ' ':

                art = art.replace(char, symbol)

        return art

    def use\_custom\_symbol(self):

        use\_custom\_char = input("Чи хочете ви використовувати власні символи замість стандартних? (y/n): ").lower().strip()

        if use\_custom\_char == 'y':

            while True:

                try:

                    symbol = input("Введіть символ для створення ASCII-арту (наприклад, @, #, \*): ").strip()

                    if len(symbol) == 1 and ord(symbol) < 128:

                        self.ascii\_art = self.changeSymbol(self.ascii\_art, symbol)

                        break

                    else:

                        print("Будь ласка, введіть лише один ASCII-символ (без спеціальних символів, як ♥).")

                except ValueError:

                    print("Введення некоректне. Спробуйте ще раз.")

class AsciiArtApplication:

    def \_\_init\_\_(self, generator\_class):

        self.generator = generator\_class()

    def run(self):

        while True:

            self.generator.get\_user\_input()

            self.generator.choose\_font()

            self.generator.choose\_color()

            self.generator.generate\_art()

            resize = input("Чи бажаєте ви змінити розмір ASCII-арту? (y/n): ").lower().strip()

            if resize == 'y':

                while True:

                    try:

                        width = int(input("Введіть ширину ASCII-арту (наприклад, 80): "))

                        height = int(input("Введіть висоту ASCII-арту (наприклад, 20): "))

                        self.generator.resize\_art(width, height)

                        break

                    except ValueError:

                        print("Будь ласка, введіть числові значення для ширини та висоти.")

            use\_custom = input("Чи хочете ви використовувати власні символи? (y/n): ").lower().strip()

            if use\_custom == 'y':

                self.generator.use\_custom\_symbol()

            self.generator.preview\_art()

            save = input("Чи бажаєте ви зберегти цей ASCII-арт у файл? (y/n): ").lower().strip()

            if save == 'y':

                self.generator.save\_to\_file()

            another = input("Чи хочете ви згенерувати ще один ASCII-арт? (y/n): ").lower().strip()

            if another != 'y':

                print("Дякуємо за використання ASCII-арт генератора!")

                break

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

    app = AsciiArtApplication(ResizableAsciiArtGenerator)

    app.run()

**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.

GitHub: https://github.com/RomchykPrytula/SMP