МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ



ЗВІТ

Про виконання лабораторної роботи №3

«Розробка ASCII ART генератора для візуалізації текстових даних»

З дисципліни

«Спеціалізовані мови програмування»

Студента групи РІ – 31

Тураша Івана Павловича

Прийняв викладач

Щербак С.С.

**Лабораторна робота № 3. Розробка ASCII ART генератора для візуалізації текстових даних**

**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.

**Основний код програми:**

from src.user\_input import UserInput

from src.font\_manager import FontManager

from src.color\_manager import ColorManager

from src.ascii\_art\_generator import AsciiArtGenerator

from src.file\_manager import FileManager

from src.display\_manager import DisplayManager

class AsciiArtApp:

    def \_\_init\_\_(self):

        self.text = ""

        self.font = "standard"

        self.color = ColorManager.COLORS["7"]

        self.width = 80

        self.ascii\_art = ""

    def run(self):

        print("ASCII ART Генератор")

        # Введення тексту

        self.text = UserInput.get\_text()

        # Вибір шрифту

        self.font = FontManager.choose\_font()

        # Вибір кольору

        self.color = ColorManager.choose\_color()

        # Вибір ширини

        self.width = int(input("Вкажіть ширину для ASCII-арту (за замовчуванням 80): ") or 80)

        # Генерація ASCII-арту

        generator = AsciiArtGenerator(self.text, self.font, self.width)

        self.ascii\_art = generator.generate()

        # Попередній перегляд

        DisplayManager.preview(self.ascii\_art, self.color)

        # Збереження у файл

        save = input("Бажаєте зберегти результат у файл? (так/ні): ").lower()

        if save == "так":

            FileManager.save\_to\_file(self.ascii\_art)

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

    app = AsciiArtApp()

    app.run()

import pyfiglet

class AsciiArtGenerator:

    def \_\_init\_\_(self, text, font, width=80):

        self.text = text

        self.font = font

        self.width = width

    def generate(self):

        ascii\_art = pyfiglet.Figlet(font=self.font, width=self.width)

        return ascii\_art.renderText(self.text)

from colorama import Fore, Style

class ColorManager:

    COLORS = {

        "1": Fore.RED,

        "2": Fore.GREEN,

        "3": Fore.BLUE,

        "4": Fore.CYAN,

        "5": Fore.MAGENTA,

        "6": Fore.YELLOW,

        "7": Fore.WHITE

    }

    @staticmethod

    def choose\_color():

        print("Доступні кольори:")

        for key, color in ColorManager.COLORS.items():

            print(f"{key}: {color}{color.upper()}{Style.RESET\_ALL}")

        choice = input("Виберіть номер кольору (за замовчуванням білий): ") or "7"

        return ColorManager.COLORS.get(choice, Fore.WHITE)

class DisplayManager:

    @staticmethod

    def preview(ascii\_art, color):

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

        print(color + ascii\_art)

class FileManager:

    @staticmethod

    def save\_to\_file(ascii\_art):

        file\_name = input("Введіть ім'я файлу для збереження (за замовчуванням art.txt): ") or "art.txt"

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

            file.write(ascii\_art)

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

class UserInput:

    @staticmethod

    def get\_text():

        return input("Введіть текст для ASCII-арту: ")

import pyfiglet

class FontManager:

    @staticmethod

    def choose\_font():

        available\_fonts = pyfiglet.FigletFont.getFonts()  # Отримання списку шрифтів

        print("Доступні шрифти:")

        for i, font in enumerate(available\_fonts[:20], start=1):  # Відображення перших 20 шрифтів

            print(f"{i}: {font}")

        choice = int(input("Виберіть номер шрифту (за замовчуванням 1): ") or 1)

        return available\_fonts[choice - 1] if 1 <= choice <= len(available\_fonts) else "standard"

**Висновок:** Виконуючи ці завдання, я створив 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.