‘Міністерство освіти і науки України Національний університет “Львівська політехніка” Кафедра інформаційних систем та мереж

ЗВІТ

про виконання лабораторної роботи № 3

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

з дисципліни “***Спеціальні мови програмування*** ”

Виконав студент групи IT-32

Пазюк С.А.

Прийняв: Щербак С.С

# ЛЬВІВ – 2023

**Тема роботи:** Розробка ASCII ART генератора для візуалізації текстових даних

**Мета роботи:** 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.

**Хід роботи**

**Реалізований код:**

# main.py

from user\_input import get\_user\_input, select\_font, select\_color

from output\_formatter import format\_output, colorize\_ascii\_art

from ascii\_art\_generator import generate\_ascii\_art, replace\_characters

from file\_handling import save\_to\_file

def select\_size():

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

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

    return width, height

def select\_characters():

    characters = input("Введіть символи, які ви хочете використовувати (наприклад, '@#$%'): ")

    return characters

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

    print("Попередній перегляд ASCII-арту:")

    print(ascii\_art)

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

    input\_text = get\_user\_input()

    font = select\_font()

    color = select\_color()

    width, height = select\_size()

    characters = select\_characters()

    ascii\_art = generate\_ascii\_art(input\_text, font)

    ascii\_art = replace\_characters(ascii\_art, characters)

    formatted\_ascii\_art = format\_output(ascii\_art, width, height)

    colored\_ascii\_art = colorize\_ascii\_art(formatted\_ascii\_art, color)

    preview\_ascii\_art(colored\_ascii\_art)

    save\_choice = input("Зберегти ASCII-арт у файл? (так/ні): ")

    if save\_choice.lower() == 'так':

        file\_name = input("Введіть ім'я файлу для збереження (з розширенням .txt): ")

        save\_to\_file(colored\_ascii\_art, file\_name)

# user\_input.py

def get\_user\_input():

    user\_input = input("Введіть слово або фразу, яку ви хочете перетворити в ASCII-арт: ")

    return user\_input

def select\_font():

    print("Доступні шрифти:")

    print("1. standard")

    print("2. slant")

    print("3. script")

    font\_choice = input("Виберіть номер шрифту (1/2/3): ")

    fonts = ["standard", "slant", "script"]

    if font\_choice in ['1', '2', '3']:

        return fonts[int(font\_choice) - 1]

    else:

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

        return select\_font()

def select\_color():

    print("Доступні кольори:")

    print("1. Червоний")

    print("2. Синій")

    print("3. Зелений")

    color\_choice = input("Виберіть номер кольору (1/2/3): ")

    colors = ["red", "blue", "green"]

    if color\_choice in ['1', '2', '3']:

        return colors[int(color\_choice) - 1]

    else:

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

        return select\_color()

# output\_formatter.py

from termcolor import colored

def format\_output(ascii\_art, width, height):

    lines = ascii\_art.split('\n')

    formatted\_art = []

    for line in lines:

        formatted\_line = line.center(width)

        formatted\_art.append(formatted\_line)

    formatted\_art = formatted\_art[:height]

    return '\n'.join(formatted\_art)

def colorize\_ascii\_art(formatted\_ascii\_art, color):

    return colored(formatted\_ascii\_art, color)

# file\_handling.py

def save\_to\_file(ascii\_art, file\_name):

    try:

        with open(file\_name, 'w') as file:

            file.write(ascii\_art)

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

    except Exception as e:

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

# ascii\_art\_generator.py

import pyfiglet

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

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

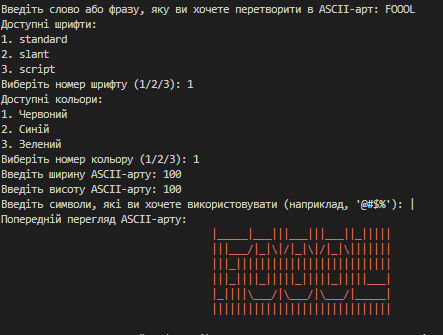
def replace\_characters(ascii\_art, characters):

    for old\_char, new\_char in zip(ascii\_art, characters):

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

    return ascii\_art

**Результат виконання программи**:



**Висновок:** Виконуючи ці завдання, я навчився створювати універсальний Генератор ASCII-арту, який дозволить користувачам налаштовувати свої творіння з різними шрифтами, кольорами, розмірами та символами. Проект надасть практичний досвід роботи з введенням користувача, зовнішніми бібліотеками, роботою з файлами та дизайном інтерфейсу користувача в Python.