‘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 2D-фігур”***

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 2D-фігур***

**Мета роботи:** Створення Генератора ASCII-арту без використання зовнішніх бібліотек

**План роботи**

Завдання 1: Введення користувача

Створіть програму Python, яка отримує введення користувача щодо слова або фрази, яку вони хочуть перетворити в ASCII-арт.

Завдання 2: Набір символів

Визначте набір символів (наприклад, '@', '#', '\*', тощо), які будуть використовуватися для створення ASCII-арту. Ці символи будуть відображати різні відтінки.

Завдання 3: Розміри Art-у

Запитайте у користувача розміри (ширина і висота) ASCII-арту, який вони хочуть створити. Переконайтеся, що розміри в межах керованого діапазону

Завдання 4: Функція генерації Art-у

Напишіть функцію, яка генерує ASCII-арт на основі введення користувача, набору символів та розмірів. Використовуйте введення користувача, щоб визначити, які символи використовувати для кожної позиції в Art-у.

Завдання 5: Вирівнювання тексту

Реалізуйте опції вирівнювання тексту (ліво, центр, право), щоб користувачі могли вибирати, як їх ASCII-арт розміщується на екрані.

Завдання 6: Відображення мистецтва

Відобразіть створений ASCII-арт на екрані за допомогою стандартних функцій друку Python.

Завдання 7: Збереження у файл

Додайте можливість зберігати створений ASCII-арт у текстовий файл, щоб користувачі могли легко завантажувати та обмінюватися своїми творіннями.

Завдання 8: Варіанти кольорів

Дозвольте користувачам вибирати опції кольорів (чорно-білий, відтінки сірого) для свого ASCII-арту.

Завдання 9: Функція попереднього перегляду

Реалізуйте функцію попереднього перегляду, яка показує користувачам попередній перегляд їх ASCII-арту перед остаточним збереженням

Завдання 10: Інтерфейс, зрозумілий для користувача

Створіть інтерфейс для користувача у командному рядку, щоб зробити програму легкою та інтуїтивно зрозумілою для використання.

**Хід роботи**

**Реалізований код:**

import random

# Символи ASCII-арту

characters = {

    'a': [

        "  AA  ",

        " A  A ",

        " AAAAA",

        " A  A ",

        " A  A "

    ],

    'b': [

        " BBB  ",

        " B  B ",

        " BBBB ",

        " B  B ",

        " BBB  "

    ],

    'c': [

        " CCCC ",

        " C    ",

        " C    ",

        " C    ",

        " CCCC "

    ],

    'e': [

        " EEEE ",

        " E    ",

        " EEEE ",

        " E    ",

        " EEEE "

    ],

    'g': [

        " GGGG ",

        " G    ",

        " G GG ",

        " G  G ",

        " GGGG "

    ],

    'i': [

        " III ",

        "  I  ",

        "  I  ",

        "  I  ",

        " III "

    ],

    'r': [

        " RRR  ",

        " R  R ",

        " RRR  ",

        " R R  ",

        " R  R "

    ],

    's': [

        " SSSS ",

        " S    ",

        " SSSS ",

        "    S ",

        " SSSS "

    ],

    'y': [

        " Y  Y ",

        "  YY  ",

        "   Y  ",

        "   Y  ",

        "   Y  "

    ]

}

# Завдання 8: Варіанти кольорів

color\_option = input("Виберіть опцію кольорів (чорно-білий/відтінки сірого): ").lower()

# Завдання 1: Введення користувача

input\_text = input("Введіть слово або фразу для ASCII-арту: ")

# Завдання 5: Вирівнювання тексту

alignment = input("Виберіть вирівнювання (ліво, центр, право): ").lower()

if alignment not in ['ліво', 'центр', 'право']:

    alignment = 'ліво'

# Завдання 3: Розміри Art-у

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

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

# Завдання 4: Функція генерації Art-у

def generate\_ascii\_art(input\_text, characters, alignment, width, height, color\_option):

    lines = []

    if alignment == 'центр':

        lines.append(input\_text.center(width))

    elif alignment == 'право':

        lines.append(input\_text.rjust(width))

    else:

        lines.append(input\_text.ljust(width))

    for i in range(5):

        line = ""

        for char in input\_text:

            if char.lower() in characters:

                line += characters[char.lower()][i]

            else:

                line += " " \* 6  # Пробіли, якщо символ не знайдено

            line += " "  # Додатковий пробіл між буквами

        if color\_option == 'чорно-білий':

            line = "\033[30m" + line + "\033[0m"  # Чорний колір тексту

        elif color\_option == 'відтінки сірого':

            line = "\033[90m" + line + "\033[0m"  # Сірий колір тексту

        lines.append(line)

    return lines

# Генерація ASCII-арту

art = generate\_ascii\_art(input\_text, characters, alignment, width, height, color\_option)

# Завдання 6: Відображення мистецтва

for line in art:

    print(line)

# Завдання 7: Збереження у файл

save\_option = input("Бажаєте зберегти ASCII-арт у файл? (так/ні): ").lower()

if save\_option == 'так':

    filename = input("Введіть ім'я файлу: ")

    with open(filename, 'w') as file:

        for line in art:

            file.write(line + '\n')

# Завдання 9: Функція попереднього перегляду

def preview\_ascii\_art(art):

    print("Попередній перегляд ASCII-арту:")

    for line in art:

        print(line)

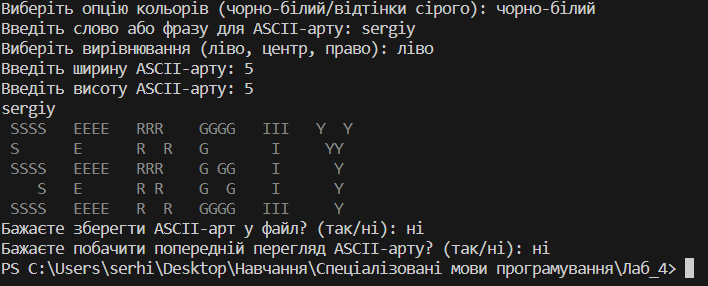
# Попередній перегляд ASCII-арту

preview\_option = input("Бажаєте побачити попередній перегляд ASCII-арту? (так/ні): ").lower()

if preview\_option == 'так':

    preview\_ascii\_art(art)

**Результат виконання программи**:



**Висновок:** Виконуючи ці завдання я навчився створювати генератор ASCII-арту з нуля, та надати можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволить мені глибше розібратися як створюється ASCII-арт