МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Лабораторна робота №4

з дисципліни “Спеціалізовані мови програмування”

Виконав:

студент групи ІТ-31

Олександр ГАМАЮНОВ

Прийняв:

доцент кафедри ІСМ

Сергій ЩЕРБАК

Львів-2023

**Тема роботи:** Розробка ASCII ART генератора для візуалізації 2D-фігур **Мета роботи:** Створення Генератора ASCII-арту без використання зовнішніх бібліотек

Хід 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.

Код:

Main.py:  
""" Main file"""

from Data.data import color\_mapping

from .Classes.generator import ASCIIArtGenerator

from .Classes.input import InputHandler

def main() -> None:

    """ Main method"""

    handler = InputHandler()

    user\_language = handler.get\_language(input("Enter language"))

    user\_input = handler.get\_user\_input(input("Введіть слово або фразу для ASCII-арту: "))

    size = handler.get\_art\_size(input("Введіть ширину (1-100):"),input("Введіть висоту (1-100):"))

    width = size[0]

    height = size[1]

    alignment = handler.get\_alignment(input("Виберіть вирівнювання тексту (left, center, right): "))

    color = color\_mapping[handler.get\_color(input("Оберіть колір @(чорний), #(сірий), \*(білий)]:"))]

    ascii\_art = ASCIIArtGenerator(user\_input,color,width,height,alignment,user\_language)

    ascii\_art.generate\_ascii\_art()

    ascii\_art.display\_ascii\_art()

    ascii\_art.save\_to\_file()

    user\_control = handler.get\_answer(input('Чи бажаєте Ви продовжити? Y N: '))

    if(user\_control in ['N','n','no']):

        exit()

if \_\_name\_\_ == '\_\_main\_\_':

    while 1:

        main()

Generator.py:  
""" Generator file"""

import logging

class ASCIIArtGenerator:

    """ Class for generatin ASCII arts"""

    def \_\_init\_\_(self, user\_input, art\_color, width, height, alignment,language):

        self.user\_input = user\_input

        self.art\_color = art\_color

        self.width = width

        self.height = height

        self.alignment = alignment

        self.ascii\_art = ""

        self.language = language

    def generate\_ascii\_art(self):

        """ Generate arts"""

        input\_lines = self.user\_input.split('\n')

        max\_length = max(len(line) for line in input\_lines)

        self.ascii\_art = [''] \* len(self.language[' '])

        for i in range(len(self.language[' '])):

            for line in input\_lines:

                if self.alignment == 'center':

                    line = line.center(max\_length)

                elif self.alignment == 'right':

                    line = line.rjust(max\_length)

                else:

                    line = line.ljust(max\_length)

                for char in line:

                    char = char.upper()

                    if char in self.language:

                        self.ascii\_art[i] += f'\x1b[{self.art\_color}m{self.language[char][i]}\x1b[0m'

                    else:

                        self.ascii\_art[i] += ' '

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

        logging.info('ASSCII-art is generated')

    def display\_ascii\_art(self):

        """Display arts """

        print(self.ascii\_art)

        logging.info('ASCII-art is displeyed')

    def save\_to\_file(self):

        """ Save arts"""

        save\_option = input("Зберегти ASCII-арт у файлі? (Так/Н і): ").lower()

        if save\_option == 'так':

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

            filepath = 'Data/' + filename

            with open(filepath,'w',encoding='utf-8') as file:

                file.write(self.ascii\_art)

                print(f"ASCII-арт було збережено у файлі {filename}")

                logging.info('ASCII-art is saved')

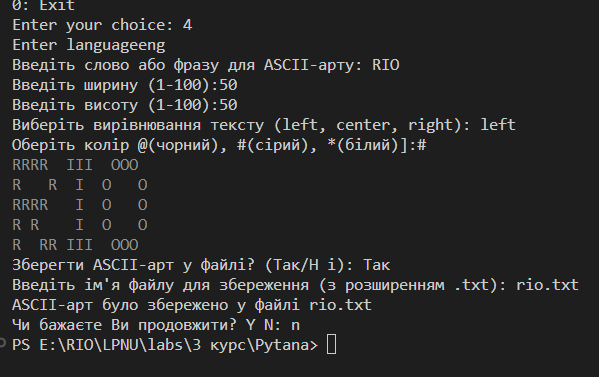


Рис.1 Результат виконання роботи

**Висновок**: Виконуючи лабораторну роботу я створив генератор ASCII-арту без використання постороніх бібліотек, та надав користувачам можливість налаштовувати розміри, вирівнювання та кольори.