МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра ІСМ



ЗВІТ

Про виконання лабораторної роботи №4

«Розробка ASCII ART генератора для візуалізації 2D-фігур»

З дисципліни

«Спеціалізовані мови програмування»

Студентки групи РІ – 31

Нащецької Яни Вікторівни

Прийняв викладач

Щербак С.С.

**Лабораторна робота № 4. Розробка ASCII ART генератора для візуалізації 2D-фігур**

**Мета:** 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.

**Основний код програми:**

def get\_ascii\_font(symbol="@"):

    return {

        "A": [

            f"  {symbol}  ",

            f" {symbol} {symbol} ",

            f"{symbol}{symbol}{symbol}{symbol}{symbol}",

            f"{symbol}   {symbol}",

            f"{symbol}   {symbol}"

        ],

        "B": [

            f"{symbol}{symbol}{symbol} ",

            f"{symbol}   {symbol}",

            f"{symbol}{symbol}{symbol} ",

            f"{symbol}   {symbol}",

            f"{symbol}{symbol}{symbol} "

        ],

        "C": [

            f" {symbol}{symbol}{symbol}",

            f"{symbol}    ",

            f"{symbol}    ",

            f"{symbol}    ",

            f" {symbol}{symbol}{symbol}"

        ],

        "D": [

            f"{symbol}{symbol}{symbol} ",

            f"{symbol}   {symbol}",

            f"{symbol}   {symbol}",

            f"{symbol}   {symbol}",

            f"{symbol}{symbol}{symbol} "

        ],

        "E": [

            f"{symbol}{symbol}{symbol}{symbol}{symbol}",

            f"{symbol}    ",

            f"{symbol}{symbol}{symbol} ",

            f"{symbol}    ",

            f"{symbol}{symbol}{symbol}{symbol}{symbol}"

        ],

        "F": [

            f"{symbol}{symbol}{symbol}{symbol}{symbol}",

            f"{symbol}    ",

            f"{symbol}{symbol}{symbol} ",

            f"{symbol}    ",

            f"{symbol}    "

        ],

        "G": [

            f" {symbol}{symbol}{symbol} ",

            f"{symbol}    ",

            f"{symbol}  {symbol}{symbol}{symbol}",

            f"{symbol}   {symbol}",

            f" {symbol}{symbol}{symbol} "

        ],

//…

import os

import sys

shared\_root = os.path.abspath(os.path.dirname(\_\_file\_\_))

sys.path.insert(0, shared\_root)

from .ascii\_art\_generator import ArtGenerator

from shapes import Cube, Sphere

class ShapeGenerator(ArtGenerator):

    def \_\_init\_\_(self, shape\_type, size=5, symbol="\*"):

        super().\_\_init\_\_(symbol)

        self.shape\_type = shape\_type

        self.size = size

    def generate(self):

        if self.shape\_type == "cube":

            shape = Cube(self.size, self.symbol)

        elif self.shape\_type == "sphere":

            shape = Sphere(self.size, self.symbol)

        else:

            raise ValueError("Unsupported shape type.")

        return shape.draw()

**Вмсновок:** Виконуючи цю лабораторну роботу,я створила генератор ASCII-арту з нуля, та надала можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило мені глибше розібратися як створюється ASCII-арт.