Міністерство освіти і науки України
Національний університет «Львівська політехніка»

Звіт про виконання лабораторної роботи №5
з дисципліни «Спеціалізовані мови програмування»
на тему «Розробка ASCII ART генератора для візуалізації 3D-фігур»

Виконав:
студент групи PI-32
Гусак Віктор
Прийняв:
Щербак С.С.

Мета роботи: Створення додатка для малювання 3D-фігур у ASCII-арті на основі об’єктно - орієнтованого підходу та мови Python

План роботи

Завдання 1: Проектування класів
Розробіть структуру класів для вашого генератора 3D ASCII-арту. Визначте основні компоненти, атрибути та методи, необхідні для програми.

Завдання 2: Введення користувача
Створіть методи у межах класу для введення користувача та вказання 3D-фігури, яку вони хочуть намалювати, та її параметрів (наприклад, розмір, кольори).

Завдання 3: Представлення фігури
Визначте структури даних у межах класу для представлення 3D-фігури. Це може включати використання списків, матриць або інших структур даних для зберігання форми фігури та її властивостей.

Завдання 4: Проектування з 3D в 2D
Реалізуйте метод, який перетворює 3D-представлення фігури у 2D-представлення, придатне для ASCII-арту.

Завдання 5: Відображення ASCII-арту
Напишіть метод у межах класу для відображення 2D-представлення 3D-фігури як ASCII-арту. Це може включати відображення кольорів і форми за допомогою символів ASCII.

Завдання 6: Інтерфейс, зрозумілий для користувача
Створіть зручний для користувача командний рядок або графічний інтерфейс користувача (GUI) за допомогою об'єктно-орієнтованих принципів, щоб дозволити користувачам спілкуватися з програмою.

Завдання 7: Маніпуляція фігурою
Реалізуйте методи для маніпулювання 3D-фігурою, такі масштабування або зміщення, щоб надавати користувачам контроль над її виглядом.

Завдання 8: Варіанти кольорів
Дозвольте користувачам вибирати варіанти кольорів для їхніх 3D ASCII-арт-фігур. Реалізуйте методи для призначення кольорів різним частинам фігури.

Завдання 9: Збереження та експорт
Додайте функціональність для зберігання згенерованого 3D ASCII-арту у текстовий файл

Завдання 10: Розширені функції
Розгляньте можливість додавання розширених функцій, таких як тінь, освітлення та ефекти перспективи, для підвищення реалізму 3D ASCII-арту.

Хід роботи

ascii_3d_functions.py:

```
class ASCIIArt3DGenerator:
    def __init__(self):
        self.figure = None
        self.size = (10, 10)
        self.projection = []

    def set_figure(self, figure_name):
        if figure_name.lower() == 'cube':
            self.figure = self.generate_cube()
        elif figure_name.lower() == 'pyramid':
            self.figure = self.generate_pyramid()
        elif figure_name.lower() == 'sphere':
            self.figure = self.generate_sphere()
        else:
            print("Невідома фігура. Спробуйте 'cube', 'pyramid'
або 'sphere'.")
            self.figure = None

    def generate_cube(self):
        size = self.size[0]
        # Базова проекція куба в ASCII
        return [["#" if x == 0 or x == size - 1 or y == 0 or y
== size - 1 else " " for x in range(size)] for y in range(size)]

    def generate_pyramid(self):
        size = self.size[0]
        projection = []
        for i in range(size):
            row = [" "] * (size - i) + ["#"] * (2 * i + 1) + ["
"] * (size - i)
            projection.append(row)
        return projection
```

```
    def generate_sphere(self):
        size = self.size[0]
        center = size // 2
        projection = []
        for y in range(size):
            row = []
            for x in range(size):
                distance = ((x - center) ** 2 + (y - center) **
2) ** 0.5
                if distance < center:
                    row.append("#")
                else:
                    row.append(" ")
            projection.append(row)
        return projection

    def display_ascii_art(self):
        if not self.figure:
            print("Фігура не визначена.")
            return
        for row in self.figure:
            print("".join(row))

    def run(self):
        while True:
            user_input = input("Введіть фігуру для відображення
(cube, pyramid, sphere) або 'exit' для виходу: ")
            if user_input.lower() == 'exit':
                break
            self.set_figure(user_input)
            self.display_ascii_art()
```

ascii_3d_runner.py:

```
import sys
import os

sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(
__file__))))

from functions.ascii_3d_functions import ASCIIArt3DGenerator

generator = ASCIIArt3DGenerator()
generator.run()
```

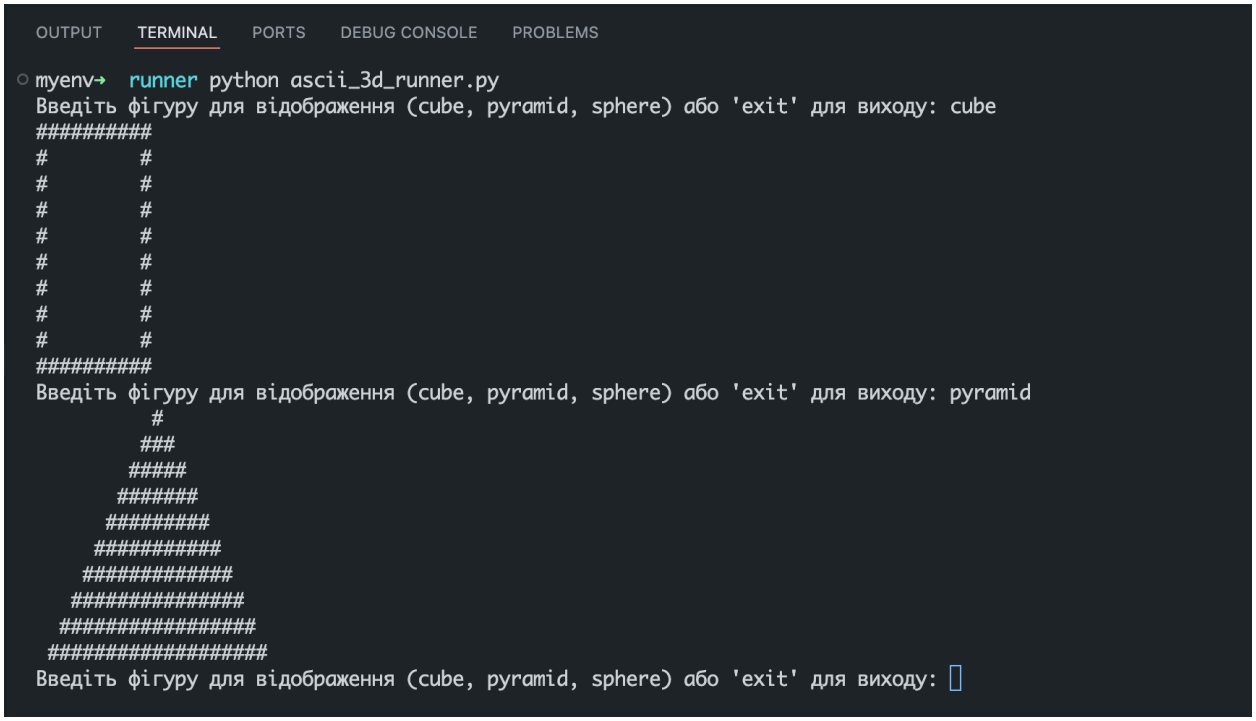

Рис. 1. Приклад роботи програми

Висновки: Виконавши ці завдання, було створено генератор ASCII-арту з нуля, та надано можливість налаштовувати символи, розміри, вирівнювання та кольори, що дозволило глибше розібратися як створюється ASCII-арт.