Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра ІСМ



**Звіт**

до лабораторної роботи № 5

з дисципліни

​*Спеціалізовані мови програмування*

на тему:

“**Розробка ASCII ART генератора**

**для візуалізації 3D-фігур**”

Виконав студент РІ-31

**Гуледза Тарас**

Прийняв: Щербак С.С.

Львів – 2024

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

**Завдання лабораторної роботи**

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.

Файл запуску:

from Lib import FileService  
from Lab\_5.figures.Cube import Cube  
  
  
class ThreeDArtService:  
 *"""Service for generating and manipulating 3D art."""* def \_\_init\_\_(self, height: int, color: int, direction: bool):  
 *"""Initialize the 3D art service with a Cube and other parameters."""* self.art = Cube(height)  
 self.color = color  
 self.direction = direction  
  
 def change\_size(self):  
 *"""Change the size of the cube (side\_a)."""* try:  
 size = int(input("Enter new size (no less than 3): "))  
 if size >= 3:  
 self.art.side\_a = size  
 else:  
 print("Size must be at least 3. Try again.")  
 self.change\_size()  
 except ValueError:  
 print("Invalid input. Please enter a valid integer for size.")  
 self.change\_size()  
  
 def change\_color(self, color: int):  
 *"""Change the color of the art."""* self.color = color  
  
 def change\_direction(self):  
 *"""Toggle the direction of the art."""* self.direction = not self.direction  
 print("Direction was successfully changed")  
  
 def get\_art(self) -> str:  
 *"""Get the 3D art, either in original or inverted direction."""* color\_text = '\033[%dm%s\033[0m'  
 if self.direction:  
 art = self.art.get\_three\_d\_art()  
 else:  
 art = self.art.get\_three\_d\_inverted\_art()  
  
 return color\_text % (self.color, art)  
  
 def get\_2d\_art(self) -> str:  
 *"""Get the 2D art."""* color\_text = '\033[%dm%s\033[0m'  
 art = self.art.get\_two\_d\_art()  
 return color\_text % (self.color, art)  
  
 def print\_art(self):  
 *"""Print the 3D art."""* print(self.get\_art())  
  
 def print\_2d\_art(self):  
 *"""Print the 2D art."""* print(self.get\_2d\_art())  
  
 def save\_art\_into\_file(self, file\_name: str):  
 *"""Save the generated art into a file."""* try:  
 FileService.write\_into\_file(file\_name, self.get\_art())  
 except FileExistsError:  
 print(f"File {file\_name} already exists. Please choose a different name.")  
 raise FileExistsError  
  
 @staticmethod  
 def get\_art\_archive(file\_name: str):  
 *"""Retrieve and print art from a file archive."""* try:  
 print(FileService.read\_from\_file(file\_name))  
 except FileNotFoundError:  
 print(f"File {file\_name} not found.")  
 raise FileNotFoundError

**Висновок**

Виконавши ці завдання, я створив високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволить користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті. Цей проект надав мені глибоке розуміння об'єктно-орієнтованого програмування і алгоритмів графіки, сприяв творчому підходу до створення ASCII-арту.