НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №5

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав:

ст. гр. РІ-31

Назар МРАКА

Прийняв:

Сергій ЩЕРБАК

Львів-2024

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

**Завдання:**

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.

Код програми:

*class ThreeDArtService:*

*def \_\_init\_\_(self, height: int, color: int, direction: bool):*

*self.color = color*

*self.direction = direction*

*def change\_size(self):*

*try:*

*size = int(input("Enter new size (no less than 3): "))*

*if size >= 3:*

*self.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):*

*self.color = color*

*def change\_direction(self):*

*self.direction = not self.direction*

*print("Direction was successfully changed")*

*def get\_art(self) -> str:*

*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:*

*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(self.get\_art())*

*def print\_2d\_art(self):*

*print(self.get\_2d\_art())*

**Висновок:**

У рамках лабораторної роботи було розроблено додаток для малювання 3D-фігур у ASCII-арті з використанням об’єктно-орієнтованого підходу на мові Python. Програма дозволяє створювати прості 3D-фігури (наприклад, куби, піраміди, сфери) та відображати їх у вигляді ASCII-арту. Основною особливістю є використання ООП для організації коду, що забезпечує гнучкість та розширюваність додатка.