

ЗВІТ
про виконання лабораторної роботи № 5
«Розробка ASCII ART генератора для візуалізації 3D-фігур »
з дисципліни
«Спеціалізовані мови програмування»
ст. групи РІ-31
Танечник Марічки

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

Умова 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.

Текст програми:

```
import os
import sys
```

```
lab5_root = os.path.abspath(os.path.join(os.path.dirname(__file__),
".." , "..", ".."))
```

```
if lab5_root not in sys.path:
    sys.path.insert(0, os.path.join(lab5_root, "shared"))
```

```
from shared.classes.art_generator.shape_generator import
ShapeGenerator
from shared.classes.art_generator.text_generator import
TextGenerator
```

```
class ArtManager:
    def create_text_art(self, params):
        return TextGenerator(**params)
```

```
    def create_shape(self, shape_type, size, symbol):
        return ShapeGenerator(shape_type, size, symbol)
```

```
    def generate_art(self, art_instance):
        return art_instance.generate()
```

```
import os
import sys
```

```
from ..BLL.art_manager import ArtManager
from shared.classes.art_generator.console_reader import
ConsoleReader
```

```
class UserInterface:
    def start(self):
        print("Welcome to the ASCII Art 3D Generator!")
```

```
        art_type = ConsoleReader.read_art_type()
```

```
        art_manager = ArtManager()
```

```
if art_type == "text":
    params = ConsoleReader.read_text_parameters()
    art_instance = art_manager.create_text_art(params)
elif art_type == "shape":
    params = ConsoleReader.read_shape_parameters()
    art_instance = art_manager.create_shape(
        params["shape_type"], params["size"], params["symbol"]
    )

art = art_manager.generate_art(art_instance)

print("\nGenerated ASCII Art:\n")
print(art)
```

Висновки: виконуючи ці завдання, я створила високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволить користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті.