ЗВІТ

про виконання лабораторної роботи № 5

«Розробка ASCII ART генератора для візуалізації 3D-фігур»

з дисципліни

«Спеціалізовані мови програмування»

студентки групи РІ-31

Заяць Анастасії Назарівної

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

**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:**

from BLL.classes.threeD\_art\_generato import ThreeDArtService

import sys

# gen = ThreeDArtService()

# white = MenuItem("1", "White.", gen.change\_color, [0])

# red = MenuItem("2", "Red.", gen.change\_color, [91])

# blue = MenuItem("3", "Blue.", gen.change\_color, [94])

# change\_color\_menu = MenuBuilder([white, red, blue])

# create\_item = MenuItem("1", "Create a cube.", gen.print\_art)

# change\_color = MenuItem("2", "Change art color.", change\_color\_menu.initialize)

# change\_size = MenuItem("3", "Change art size.", gen.change\_size)

# change\_direction = MenuItem("4", "Change art direction.", gen.change\_direction)

# save\_art = MenuItem("5", "Save art.", gen.save\_art\_into\_file, ["./Data/art.txt"])

# see\_art = MenuItem("6", "See previous arts.", gen.get\_art\_archive, ["./Data/art.txt"])

# exit\_menu = MenuItem("9", "Exit.", sys.exit)

# main\_menu = MenuBuilder([create\_item, change\_color, change\_size, change\_direction, save\_art, see\_art, exit\_menu])

def menu():

art\_service = ThreeDArtService(3, 0, False)

while True:

print("\n--- 3D Art Menu ---")

print("1. Change Cube Size")

print("2. Change Color")

print("3. Toggle Art Direction")

print("4. Display Art")

print("5. Display 2D Art")

print("6. Save Art to File")

print("7. Load and Display Art from File")

print("0. Exit")

choice = input("Choose an option: ")

if choice == "1":

art\_service.change\_size()

elif choice == "2":

new\_color = int(input("Enter color code (e.g., 31 for red, 32 for green, 34 for blue): "))

art\_service.change\_color(new\_color)

elif choice == "3":

art\_service.change\_direction()

elif choice == "4":

art\_service.print\_art()

elif choice == "5":

art\_service.print\_2d\_art()

elif choice == "6":

file\_name = input("Enter file name to save: ")

art\_service.save\_art\_into\_file(file\_name)

print(f"Art saved to {file\_name}")

elif choice == "7":

file\_name = input("Enter file name to load: ")

try:

ThreeDArtService.get\_art\_archive(file\_name)

except FileNotFoundError:

print("File not found. Please check the file name.")

elif choice == "0":

print("Exiting program.")

break

else:

print("Invalid choice. Please try again.")

def main():

menu()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**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. Цей проект надав мені глибоке розуміння об'єктно-орієнтованого програмування і алгоритмів графіки, сприятиме творчому підходу до створення ASCII-арту.