Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Лабораторна роботи №5

з дисципліни «Спеціалізовані мови програмування»

на тему

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

Виконав:

Костик В. Ю.

Перевірив:

Щербак С. С.

Львів 2024

**Мета:** 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.

**Реалізація:**

**Папка AppSettings, файл AppSettings.py:**  
DEFAULT\_WIDTH = 80

DEFAULT\_SYMBOL = '#'

COLORS = {

1: 'Червоний',

2: 'Синій',

3: 'Зелений',

4: 'Білий',

5: 'Жовтий'

}

**Папка classes, файл ASCIIArtApp.py:**

from functions import functions

# Словник з 2D та 3D ASCII фігурами

shapes = {

'cube': {

'3D': [

" +------+ ",

" / /| ",

" / / | ",

" +------+ + ",

" | | / ",

" | |/ ",

" +------+ "

],

'2D': [

" +------+ ",

" | | ",

" | | ",

" +------+ "

]

},

'pyramid': {

'3D': [

" /\\ ",

" / \\ ",

" /\_\_\_\_\\ ",

" /| |\\ ",

" /\_|\_\_\_\_|\_\\ "

],

'2D': [

" /\\ ",

" / \\ ",

" /\_\_\_\_\\ "

]

}

}

class ASCIIArtApp:

def \_\_init\_\_(self):

self.colors = {

1: 'Червоний',

2: 'Синій',

3: 'Зелений',

4: 'Білий',

5: 'Жовтий'

}

def get\_user\_input(self):

shape\_name = functions.get\_shape\_type() # Отримати тип фігури

# Цикл для отримання розміру фігури (2D або 3D)

while True:

dimension\_choice = input("Введіть '1' для 3D або '2' для 2D: ")

if dimension\_choice == '1':

is\_3d = True

break

elif dimension\_choice == '2':

is\_3d = False

break

else:

print("Невірний вибір. Будь ласка, спробуйте ще раз.")

color\_choice = functions.get\_color\_choice(self.colors) # Отримати вибір кольору

symbol = functions.get\_validated\_custom\_char() # Отримати символ

return shape\_name, is\_3d, color\_choice, symbol

def get\_ansi\_color(self, color):

color\_codes = {

"Червоний": "\033[91m",

"Синій": "\033[94m",

"Зелений": "\033[92m",

"Білий": "\033[97m",

"Жовтий": "\033[93m",

}

return color\_codes.get(color, "\033[97m") # За замовчуванням білий

def display\_shape(self, shape\_name, is\_3d, color, symbol):

ansi\_color = self.get\_ansi\_color(color) # Отримуємо ANSI-код кольору

reset\_color = "\033[0m" # Код для скидання кольору

shape\_type = '3D' if is\_3d else '2D'

shape = shapes[shape\_name][shape\_type]

art\_lines = [] # Список для зберігання рядків ASCII-арту

for line in shape:

# Заміна символу на вибраний символ

colored\_line = line.replace('+', symbol).replace('/', symbol).replace('\\', symbol).replace('|', symbol).replace('-', symbol).replace('\_', symbol)

art\_lines.append(f"{ansi\_color}{colored\_line}{reset\_color}") # Додаємо кольоровий рядок до списку

print(art\_lines[-1]) # Виводимо останній рядок

return art\_lines # Повертаємо список рядків ASCII-арту

def save\_to\_file(self, file\_path, shape\_name, is\_3d, symbol):

shape\_type = '3D' if is\_3d else '2D'

shape = shapes[shape\_name][shape\_type]

with open(file\_path, 'w', encoding='utf-8') as f:

for line in shape:

# Заміна символів у рядку на вибраний символ без кольору

saved\_line = line.replace('+', symbol).replace('/', symbol).replace('\\', symbol).replace('|', symbol).replace('-', symbol).replace('\_', symbol)

f.write(saved\_line + '\n') # Запис рядка без кольорових кодів

def run(self):

shape\_name, is\_3d, color\_choice, symbol = self.get\_user\_input()

# Відображення фігури

art\_lines = self.display\_shape(shape\_name, is\_3d, color\_choice, symbol)

save\_choice = functions.get\_save\_choice()

if save\_choice.lower() == "так":

file\_path = r"C:\\Users\\Vlad\\Desktop\\унік\\пітоній\\ASIIART\_Pro\_Max\_beta\_v05.1\\art.txt"

self.save\_to\_file(file\_path, shape\_name, is\_3d, symbol) # Зберігаємо файл

print(f"ASCII-арт збережено у файл {file\_path}")

**Папка functions, файл functions.py:**from AppSettings.AppSettings import COLORS

def get\_shape\_type():

shapes = ["cube", "pyramid"]

print("Доступні фігури:")

for i, shape in enumerate(shapes, 1):

print(f"{i}. {shape}")

while True:

choice = input("Виберіть фігуру: ")

try:

return shapes[int(choice) - 1]

except (IndexError, ValueError):

print("Помилка: Виберіть правильний номер фігури.")

def get\_shape\_dimension():

while True:

dimension = input("Якщо 3D - введіть 1, якщо 2D - введіть 2,: ").strip().lower()

if dimension in ['1', '2']:

return dimension

print("Неправильний ввід. Спробуйте ще раз.")

def get\_color\_choice(colors):

while True:

print("Виберіть колір:")

for num, color in colors.items():

print(f"{num}: {color}")

try:

color\_choice = int(input("Введіть номер кольору: "))

if color\_choice in colors:

return colors[color\_choice]

else:

print("Помилка: Виберіть номер кольору з наявних.")

except ValueError:

print("Помилка: Введіть число.")

def get\_validated\_custom\_char():

while True:

custom\_char = input("Введіть символ для ASCII-арту (Enter для використання #): ")

if not custom\_char:

return '#'

if len(custom\_char) == 1:

return custom\_char

print("Помилка: Ви повинні ввести рівно один символ.")

def get\_save\_choice():

while True:

save\_choice = input("Бажаєте зберегти ASCII-арт у файл? (так/ні): ").strip().lower()

if save\_choice in ["так", "ні"]:

return save\_choice

print("Помилка: Введіть 'так' або 'ні'.")

**Папка interface, файл interface.py:**

def display\_menu():

print("Вітаємо в Генераторі ASCII-арту!")

print("Оберіть необхідну дію:")

**Папка logs, файл logs.py:**

def log\_action(action):

with open("log.txt", "a") as log\_file:

log\_file.write(action + "\n")

**Файл main.py:**

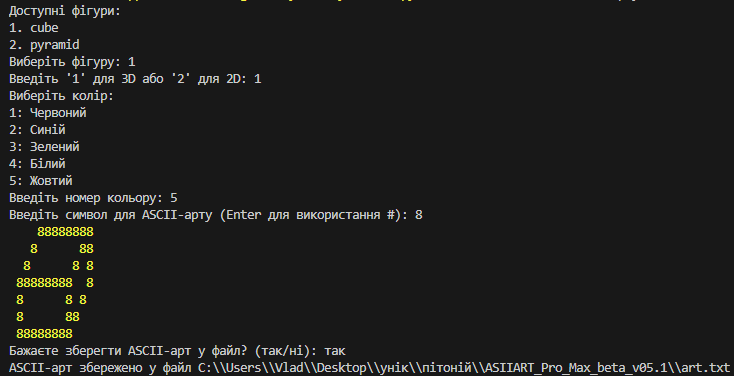
from classes.ASCIIArtApp import ASCIIArtApp

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

app = ASCIIArtApp()

app.run()

**Результат виконання:**

****

**Висновок:** на цій лабораторній роботі я створив високорівневий об'єктно-орієнтований генератор 3D ASCII-арту, який дозволить користувачам проектувати, відображати та маніпулювати 3D-фігурами в ASCII-арті