

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №5

з дисципліни

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

на тему

РОЗРОБКА ASCII ART ГЕНЕРАТОРА ДЛЯ ВІЗУАЛІЗАЦІЇ 3D-ФІГУР

Виконав:

ст. гр. РІ-31

ЛАЗАР В.С.

Прийняв:

ЩЕРБАК С.С.

Львів-2024

Мета роботи:

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

Хід роботи:

Завдання 1: Проектування класів

Розробити структуру класів для генератора 3D ASCII-арту. Визначити основні компоненти, атрибути та методи, необхідні для програми.

Завдання 2: Введення користувача

Створити 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

Визначити 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

Реалізувати метод, який перетворює 3D-представлення фігури у 2D-представлення, придатне для ASCII-арту.

Завдання 5: Відображення ASCII-арту

Написати 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

Створити 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

Реалізувати методи для маніпулювання 3D-фігурою, такі масштабування або зміщення, щоб надавати користувачам контроль над її виглядом.

Завдання 8: Варіанти кольорів

Дозволити користувачам вибирати варіанти кольорів для їхніх 3D ASCII-арт-фігур. Реалізувати методи для призначення кольорів різним частинам фігури.

Завдання 9: Збереження та експорт

Додати функціональність для зберігання згенерованого 3D ASCII-арту у текстовий файл.

Завдання 10: Розширені функції

Розглянути можливість додавання розширених функцій, таких як тінь, освітлення та ефекти перспективи, для підвищення реалізму 3D ASCII-арту.

Код програми:

```
""""The console module of this lab work""""  
from Data.Shared.classes.validators import Validators  
  
class Console:
```

```
"""The console class of this lab work"""
```

```
instance = None
```

```
def __new__(cls):
```

```
    if cls.instance is None:
```

```
        cls.instance = super(Console, cls).__new__(cls)
```

```
    return cls.instance
```

```
def __init__(self):
```

```
    self.main()
```

```
    self.ascii = None
```

```
def main(self):
```

```
    """The main menu of this lab work"""
```

```
    self.create_shape()
```

```
    while True:
```

```
        prompt = input("1 - Create a new shape\n"
```

```
                        "2 - Change shape's size\n"
```

```
                        "3 - Move shape\n"
```

```
                        "4 - Change shape's color\n"
```

```
                        "5 - Print shape\n"
```

```
                        "6 - Export shape\n"
```

```
                        "Your choice: ")
```

```
        if not Validators.main_prompt(prompt, self):
```

```
            return
```

```
def create_shape(self):
```

```
    """The UI for the shape creation"""
```

```
    while True:
```

```
        shape_prompt = input("Select shape:\n"
```

```
                              "1 - Cube\n"
```

```
                              "2 - Pyramid\n"
```

```
                              "3 - Sphere\n"
```

```
                              "Your choice: ")
```

```
        try:
```

```

        Validators.validate_value(shape_prompt, 0, 4)
except ValueError as e:
    print(e)
    continue
size_prompt = input("Enter shape's size (5-100): ")
try:
    Validators.validate_value(size_prompt, 4, 101)
    size = int(size_prompt)
except ValueError as e:
    print(e)
    continue
self.ascii = Validators.create_shape(shape_prompt, size)
print("Shape was created successfully")
return

```

```

def change_size(self):
    """The UI for the shape's size changing"""
    prompt = input("Change shape's new size (5-100): ")
    try:
        Validators.validate_value(prompt, 4, 101)
        size = int(prompt)
    except ValueError as e:
        print(e)
        return
    self.ascii.change_size(size)
    print("Shape's size was changed successfully")

```

```

def move_shape(self):
    """The UI for moving the shape"""
    prompt_x = input("Move shape by (x) (-10-10): ")
    try:
        Validators.validate_value(prompt_x, -10, 10)
        x = int(prompt_x)
    except ValueError as e:
        print(e)

```

```

        return
    prompt_y = input("Move shape by (y) (-10-10): ")
    try:
        Validators.validate_value(prompt_y, -10, 10)
        y = int(prompt_y)
    except ValueError as e:
        print(e)
        return
    prompt_z = input("Move shape by (z) (-10-10): ")
    try:
        Validators.validate_value(prompt_z, -10, 10)
        z = int(prompt_z)
    except ValueError as e:
        print(e)
        return
    self.ascii.move(x, y, z)
    print("Shape was moved successfully")

def change_color(self):
    """The UI for changing the shape's color"""
    color_prompt = input("Choose the color of your shape:\n"
        "1 - Red\n"
        "2 - Green\n"
        "3 - Yellow\n"
        "4 - Blue\n"
        "5 - Magenta\n"
        "6 - Cyan\n"
        "7 - Light gray\n"
        "8 - Random\n"
        "0 - Default\n"
        "Your choice: ")
    try:
        Validators.validate_color(color_prompt, self)
    except ValueError as e:
        print(e)

```

```
return  
print("Shape's color changed successfully")
```

На рис. 1-4 зображено результат виконання програми:

```
Select shape:  
1 - Cube  
2 - Pyramid  
3 - Sphere  
Your choice: 3  
Enter shape's size (5-100): 8  
Shape was created successfully  
1 - Create a new shape  
2 - Change shape's size  
3 - Move shape  
4 - Change shape's color  
5 - Print shape  
6 - Export shape  
Your choice: 5  
   %%%@  
  %###*@  
 %#####*@  
%#####*@  
%#####*@  
%#####*@  
@*#####*@  
 @****@  
  @@@@
```

Рис. 1. Приклад генерації та виведення ASCII-арту


```

1 - Create a new shape
2 - Change shape's size
3 - Move shape
4 - Change shape's color
5 - Print shape
6 - Export shape
Your choice: 3
Move shape by (x) (-10-10): 5
Move shape by (y) (-10-10): 0
Move shape by (z) (-10-10): 0
Shape was moved successfully
1 - Create a new shape
2 - Change shape's size
3 - Move shape
4 - Change shape's color
5 - Print shape
6 - Export shape
Your choice: 5

      %%%/%
    %#####%
  %%%#%/%@#*%@
  %%%#%#####*%@*%
  %#%#####%@*@
  %###%#####*@#%
  %#%#####*@*@
  %#%#####*@*@
  %#%#####*@*@
  %#%#####*@*@
  %#%#####*@*@
  %*@%*#####*@*@
  %*@%*#####*@*@
  %*@%*#####*@*@
  @%*#@@@@*@@@
  @%*#####@
  @@@@

```

Рис. 3. Приклад зміни позиції арту

Висновок:

Виконуючи 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.