ФГАОУ ВО «Омский государственный технический университет» Кафедра «Автоматизированные системы обработки информации и управления»

### ОТЧЁТ

по дисциплине «Разработка программного кода» студента Чернуха Ильи Евгеньевича группы ИВТМ-221

на тему «Структурные шаблоны проектирования»

### Пояснительная записка

Шифр работы РГР–02068999–43–Ивтм-221 ПЗ

Направление 09.04.01 Информатика и вычислительная техника Направленность (профиль) «Автоматизированные системы обработки информации и управления»

Доцент П.Н. Надточий

Студент И.Е. Чернуха

Омск 2023

**Задание:**

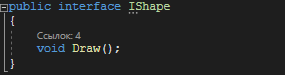
Приведите пример кода (классы\интерфейсы с комментариями) использующий какой-либо структурный шаблон проектирования и опишите задачу, которую он решает.

Данная код будет реализована с помощью языка C#. C# — объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# — объектно-ориентированный язык.

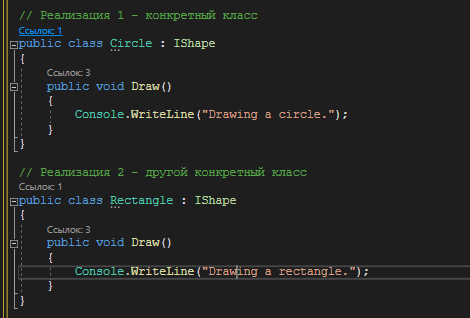
Паттерн представляет определенный способ построения программного кода для решения часто встречающихся проблем проектирования. В данном случае предполагается, что есть некоторый набор общих формализованных проблем, которые довольно часто встречаются, и паттерны предоставляют ряд принципов для решения этих проблем. На примере рассмотрим шаблон Мост.

Шаблон "Мост" (Bridge) является структурным шаблоном проектирования, который разделяет абстракцию и реализацию в отдельные иерархии классов, позволяя изменять их независимо друг от друга. Он решает задачу разделения абстракции (интерфейса) и реализации (класса), чтобы они могли изменяться независимо друг от друга.

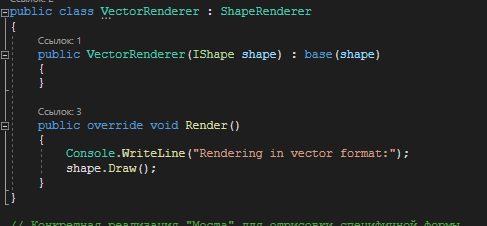
**Решение:**



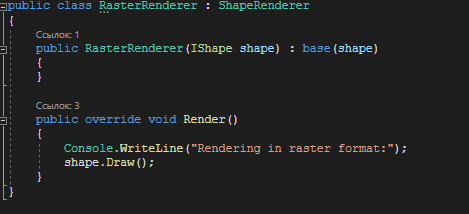
IShape: Интерфейс для операций формы (абстракция). В данном случае определен метод Draw(), который будет реализован каждым конкретным классом формы.



Circle и Rectangle: Конкретные реализации формы, реализующие интерфейс IShape.

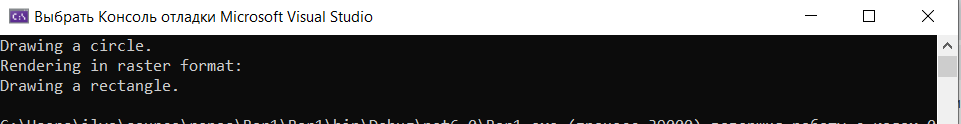


Класс VectorRenderer – конкретная реализация моста для отрисовки векторной формы.



Класс RasterRenderer – конкретная реализация моста для отрисовки растровой формы.

**Демонстрация работы:**



Создались два объекта форм и выбрали их реализацию. Отрисовали векторную и растровую форму.

**Листинг программы:**

using System;

// Абстракция - интерфейс определенных операций

public interface IShape

{

void Draw();

}

public class Circle : IShape

{

public void Draw()

{

Console.WriteLine("Drawing a circle.");

}

}

public class Rectangle : IShape

{

public void Draw()

{

Console.WriteLine("Drawing a rectangle.");

}

}

// Абстрактный класс "Мост", связывающий абстракцию и реализацию

public abstract class ShapeRenderer

{

protected IShape shape;

public ShapeRenderer(IShape shape)

{

this.shape = shape;

}

public abstract void Render();

}

// реализация "Моста" для отрисовки векторного изображения

public class VectorRenderer : ShapeRenderer

{

public VectorRenderer(IShape shape) : base(shape)

{

}

public override void Render()

{

Console.WriteLine("Rendering in vector format:");

shape.Draw();

}

}

// реализация "Моста" для отрисовки растрового изображения

public class RasterRenderer : ShapeRenderer

{

public RasterRenderer(IShape shape) : base(shape)

{

}

public override void Render()

{

Console.WriteLine("Rendering in raster format:");

shape.Draw();

}

}

public class Program

{

public static void Main(string[] args)

{

// Создаем объекты форм и выбираем их реализации

IShape circle = new Circle();

IShape rectangle = new Rectangle();

// Отрисовка векторной формы

ShapeRenderer vectorRenderer = new VectorRenderer(circle);

vectorRenderer.Render();

// Отрисовка растровой формы

ShapeRenderer rasterRenderer = new RasterRenderer(rectangle);

rasterRenderer.Render();

}

}