

## Практическая работа по паттерну проектирования "Команда" на C#

### Цель

Целью данной практической работы является изучение и применение паттерна проектирования "Команда" (Command) на языке программирования C#.

### Задание

1. Создать абстрактный класс Command, который будет представлять команду, выполняемую в системе. У этого класса должен быть метод Execute, который будет выполнять конкретное действие.
2. Создать несколько конкретных классов, реализующих абстрактный класс Command. Например, LightOnCommand, LightOffCommand, VolumeUpCommand, VolumeDownCommand и другие, представляющие различные действия.
3. Создать класс Invoker, который будет представлять собой "инициатор" выполнения команд. У этого класса должен быть метод SetCommand, позволяющий установить конкретную команду для выполнения, и метод ExecuteCommand, выполняющий установленную команду.
4. Создать класс Receiver, представляющий объект, над которым выполняются команды. Например, Light или AudioSystem. У этого класса должны быть методы, которые будут вызываться при выполнении команд.
5. Создать клиентский код, который будет создавать объекты команд, инвокера и получателя, а затем устанавливать команды в инвокер и выполнять их.

### Пример реализации

```
// Абстрактный класс команды
abstract class Command
{
    public abstract void Execute();
}

// Конкретная команда для включения света
```

```
class LightOnCommand : Command
{
    private readonly Light _light;

    public LightOnCommand(Light light)
    {
        _light = light;
    }

    public override void Execute()
    {
        _light.TurnOn();
    }
}

// Конкретная команда для выключения света
class LightOffCommand : Command
{
    private readonly Light _light;

    public LightOffCommand(Light light)
    {
        _light = light;
    }

    public override void Execute()
    {
        _light.TurnOff();
    }
}

// Класс инвокера
class Invoker
{
    private Command _command;

    public void SetCommand(Command command)
    {
        _command = command;
    }

    public void ExecuteCommand()
    {
        _command.Execute();
    }
}
```

```

    }
}

// Класс-получатель
class Light
{
    public void TurnOn()
    {
        Console.WriteLine("Свет включен.");
    }

    public void TurnOff()
    {
        Console.WriteLine("Свет выключен.");
    }
}

class Program
{
    static void Main(string[] args)
    {
        // Создаем объекты
        Light light = new Light();
        LightOnCommand lightOn = new LightOnCommand(light);
        LightOffCommand lightOff = new
LightOffCommand(light);

        Invoker remote = new Invoker();

        // Устанавливаем и выполняем команду включения света
        remote.SetCommand(lightOn);
        remote.ExecuteCommand();

        // Устанавливаем и выполняем команду выключения
света
        remote.SetCommand(lightOff);
        remote.ExecuteCommand();
    }
}

```

#### Задания

1. Расширьте пример, добавив новые команды и получатели (например, команды для управления аудиосистемой, телевизором и др.).

2. Измените код так, чтобы можно было отменять выполненные команды (реализовать операцию "отмены").

3. Попробуйте реализовать паттерн "Команда" с использованием делегатов и событий в C#.