责任链模式

概念

- 定义：责任链模式（Chain of Responsibility Pattern）是一种行为设计模式，它允许将请求沿着处理者链传递，直到有一个处理者处理它；可以避免请求的发送者与接收者之间的紧耦合关系
- 结构：
  - Handler（处理者）：定义一个处理请求的接口，并且可以选择设置下一个处理者
  - ConcreteHandler（具体处理者）：实现处理请求的接口，如不能处理请求，则将其传递给下一个处理者
  - Client（客户端）：向链中的第一个处理者发送请求
- 优点
  - 降低耦合度：请求发送者和接收者解耦，发送者不需要知道具体哪个对象处理请求，只需将请求传递给链中的第一个处理者
  - 增强系统可扩展性：可以根据需要增加新的请求处理类，符合开闭原则
  - 职责分担：每个处理者只需处理自己负责的部分，其他部分传递给下一个处理者，符合单一职责原则
  - 灵活性高：可以动态地改变链内的成员或调整它们的顺序，可以动态地新增或删除处理者
- 缺点
  - 性能问题：如果责任链过长，或者链中的处理者处理时间过长，可能会影响性能
  - 调试困难：请求在链中传递时，可能很难跟踪和调试，特别是在链条较长且处理逻辑复杂的情况下

实例

- 抽象处理者
  - *名字
  - 下一个处理者的引用
  - 设置下一个处理者的方法
  - 处理请求的抽象方法

```
1 // 抽象处理者
2 public abstract class Handler
```

```
{
    // 处理者的名字
    public string Name { get; set; }
    // 下一个处理者的引用
    protected Handler successor;
    // 设置下一个处理者
    public void SetSuccessor(Handler successor)
    {
        this.successor = successor;
    }
    // 处理请求
    public abstract void HandleRequest(int request);
}
```

- 通用处理者：继承抽象处理者，提供设置处理范围的接口，重写处理请求的方法

```
// 通用处理者
public class RangeHandler : Handler
{
    private readonly int _min;
    private readonly int _max;

    public RangeHandler(int min, int max)
    {
        _min = min;
        _max = max;
    }

    public override void HandleRequest(int request)
    {
        if (request >= _min && request < _max)
        {
            Console.WriteLine($"{Name} handled request {request}");
        }
        else if (successor != null)
        {
            successor.HandleRequest(request);
        }
    }
}
```

- 具体处理者

```
class Program
{
    static void Main(string[] args)
    {
        // 创建具体处理者
        Handler h1 = new RangeHandler(0, 10) { Name = "Handler1" };
        Handler h2 = new RangeHandler(10, 20) { Name = "Handler2" };
        Handler h3 = new RangeHandler(20, 30) { Name = "Handler3" };
        // 设置责任链
        h1.SetSuccessor(h2);
        h2.SetSuccessor(h3);

        // 生成并处理请求
        int[] requests = { 5, 14, 22 };
        foreach (int request in requests)
        {
            h1.HandleRequest(request);
        }
    }
}
```

- 返回值

```
Handler1 handled request 5
Handler2 handled request 14
Handler3 handled request 22
```