

# 中介者模式

## 概念

- 定义：中介者模式是一种行为型设计模式，主要用于降低多个对象和类之间的通信复杂性；中介者模式定义了一个**中介对象**来封装一系列对象之间的交互；使各对象之间不需要显式地相互引用，从而使其耦合松散，且可以独立地改变它们之间的交互
- 作用：解决对象间复杂的一对多关联问题，避免对象之间的高度耦合，简化系统结构
- 实现方式：
  - 定义中介者接口：规定中介者必须实现的接口
  - 创建具体中介者：实现中介者接口，包含协调各同事对象交互的逻辑
  - 定义同事类：各个对象不需要显式地相互引用，而是通过中介者来进行交互
- 优点：
  - 降低复杂度：将多个对象间的一对多关系转换为一对一关系
  - 解耦：对象之间不再直接引用，通过中介者进行交互
  - 符合迪米特原则：对象只需知道中介者，不需要知道其他对象
- 缺点：
  - 中介者复杂性：中介者可能会变得庞大和复杂，难以维护

## 实例

中介者模式包含以下几个主要角色：

- 中介者（IMediator）：定义了一个接口用于与各个同事对象通信，并管理各个同事对象之间的关系；通常包括一个或多个事件处理方法，用于处理各种交互事件

```
1 public interface IMediator
2 {
3     // 发送消息
4     void SendMessage(string message, Colleague colleague);
5     // 注册同事
6     void RegisterColleague(Colleague colleague);
7 }
```

- 具体中介者（Concrete Mediator）：实现了中介者接口，负责实现各个同事对象之间的通信逻辑；维护一个对各个同事对象的引用，并协调它们的交互

```
1 // 具体中介者
2 public class ConcreteMediator : IMediator
3 {
4     private List<Colleague> colleagues = new List<Colleague>();
5
6     public void RegisterColleague(Colleague colleague)
7     {
8         if (!colleagues.Contains(colleague))
9         {
10             colleagues.Add(colleague);
11             colleague.SetMediator(this);
12         }
13     }
14
15     public void RemoveColleague(Colleague colleague)
16     {
17         if (colleagues.Contains(colleague))
18         {
19             colleagues.Remove(colleague);
20         }
21     }
22
23     // 发送消息
24     public void SendMessage(string message, Colleague colleague)
25     {
26         foreach (var c in colleagues)
27         {
28             if (c != colleague)
29             {
30                 c.ReceiveMessage(message);
31             }
32         }
33     }
34 }
```

- 同事对象（Colleague）：定义了一个接口或抽象类，用于与中介者进行通信；通常包括一个发送消息的方法，以及一个接收消息的方法

```
1 public abstract class Colleague
2 {
3     protected string name;
```

```

4     protected IMediator mediator;
5     public Colleague(IMediator mediator)
6     {
7         this.mediator = mediator;
8     }
9
10    public void SetMediator(IMediator mediator)
11    {
12        this.mediator = mediator;
13    }
14
15    // 消息接收方法
16    public abstract void ReceiveMessage(string message);
17    // 消息发送方法
18    public abstract void SendMessage(string message);
19 }

```

- 具体同事对象（Concrete Colleague）：实现了同事对象接口，是真正参与到交互中的对象。它会将自已的消息发送给中介者，由中介者转发给其他同事对象

```

1 public class ConcreteColleague : Colleague
2 {
3     public ConcreteColleague(IMediator mediator) : base(mediator) { }
4
5     // 消息接受方法的具体实现
6     public override void ReceiveMessage(string message)
7     {
8         Console.WriteLine(name + "received: " + message);
9     }
10
11    // 消息发送方法的具体实现
12    public override void SendMessage(string message)
13    {
14        Console.WriteLine(name + "sends: " + message);
15        mediator.SendMessage(message, this);
16    }
17 }
18

```