您的潜力，我们的动力
Microsoft
微软（中国）有限公司

C#面向对象设计模式纵横谈

11. Façade外观（结构型模式）

李建忠
jianzhong.lee@gmail.com

设计模式论坛:
forum.softcompass.com

上海祝成科技 高级培训讲师
www.softcompass.com

msdn
MSDN Webcasts

系统的复杂度

假设我们需要开发一个坦克模拟系统用于模拟坦克车在各种作战环境中的行为，其中坦克系统由引擎、控制器、车轮、车身等各子系统构成。

```
public class Wheel
{
      public void WAction1() {....}
      public void WAction2() {....}
}
```

```
public class Engine
{
      public void EAction1() {....}
      public void EAction2() {....}
}
```

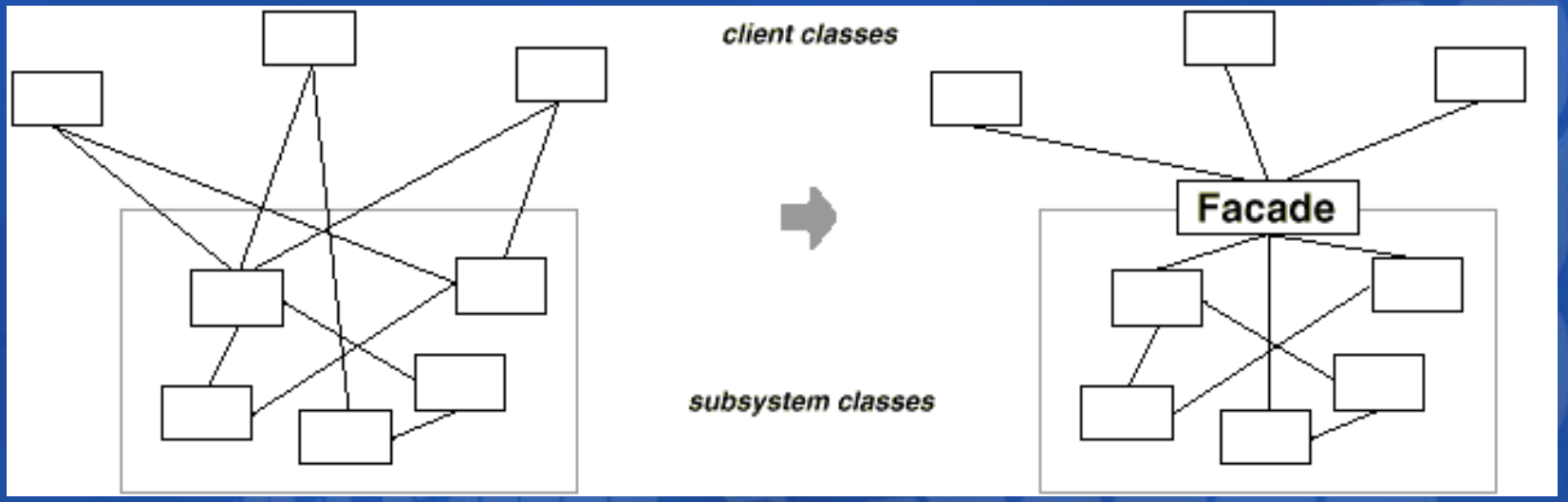
```
public class Bodywork
{
      public void BAction1() {....}
      public void BAction2() {....}
}
```

```
public class Controller
{
      public void CAction1() {....}
      public void CAction2() {....}
}
```

如何使用这样的系统

A方案

B方案



动机（Motivation）

上述A方案的问题在于组件的客户和组件中各种复杂的子系统有了过多的耦合，随着外部客户程序和各子系统的演化，这种过多的耦合面临很多变化的挑战。

如何简化外部客户程序和系统间的交互接口？如何将外部客户程序的演化和内部子系统的变化之间的依赖相互解耦？

意图（Intent）

为子系统中的一组接口提供一个一致的界面，Façade模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

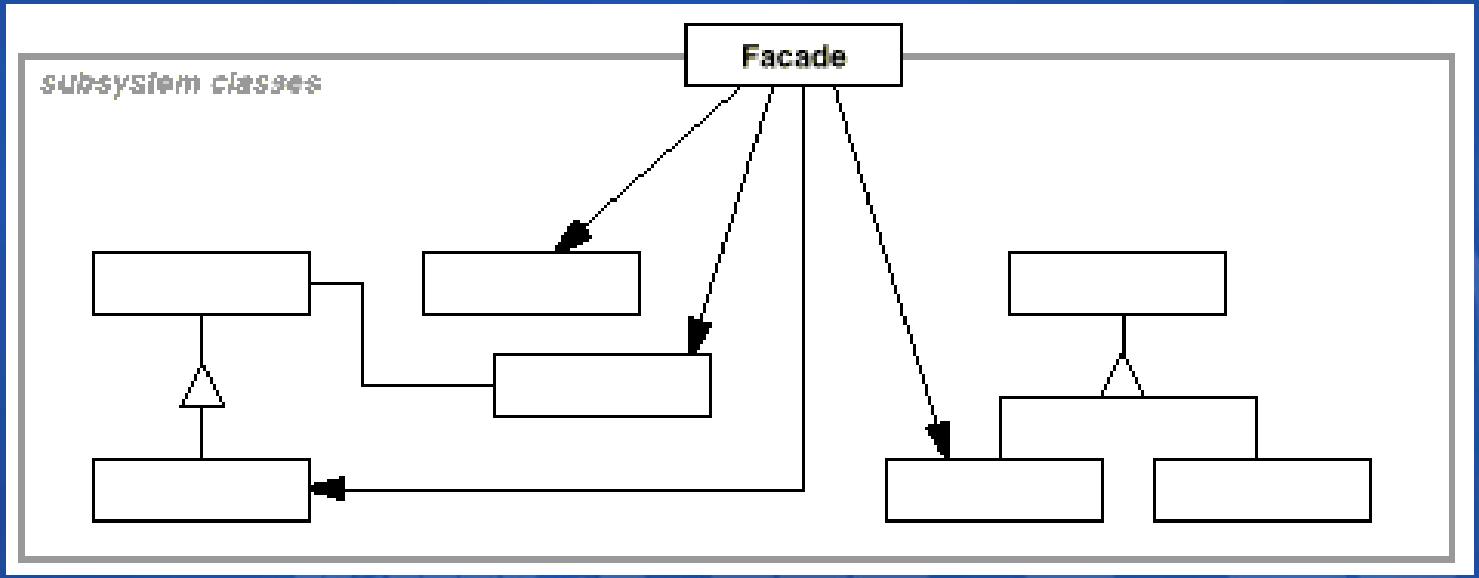
——《设计模式》GoF

例说Façade应用

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

Codes in VS.NET

结构（Structure）



Facade模式的几个要点

- 从客户程序的角度来看， Facade模式不仅简化了整个组件系统的接口，同时对于组件内部与外部客户程序来说，从某种程度上也达到了一种“解耦”的效果——内部子系统的任何变化不会影响到Façade接口的变化。
- Façade设计模式更注重从架构的层次去看整个系统，而不是单个类的层次。Façade很多时候更是一种架构设计模式。
- 注意区分Façade模式、Adapter模式、Bridge模式与Decorator模式。Façade模式注重简化接口，Adapter模式注重转换接口，Bridge模式注重分离接口（抽象）与其实现，Decorator模式注重稳定接口的前提下为对象扩展功能。

.NET架构中的Facade应用

您的潜力，我们的动力
Microsoft
微软（中国）有限公司

Codes in VS.NET

推荐资源

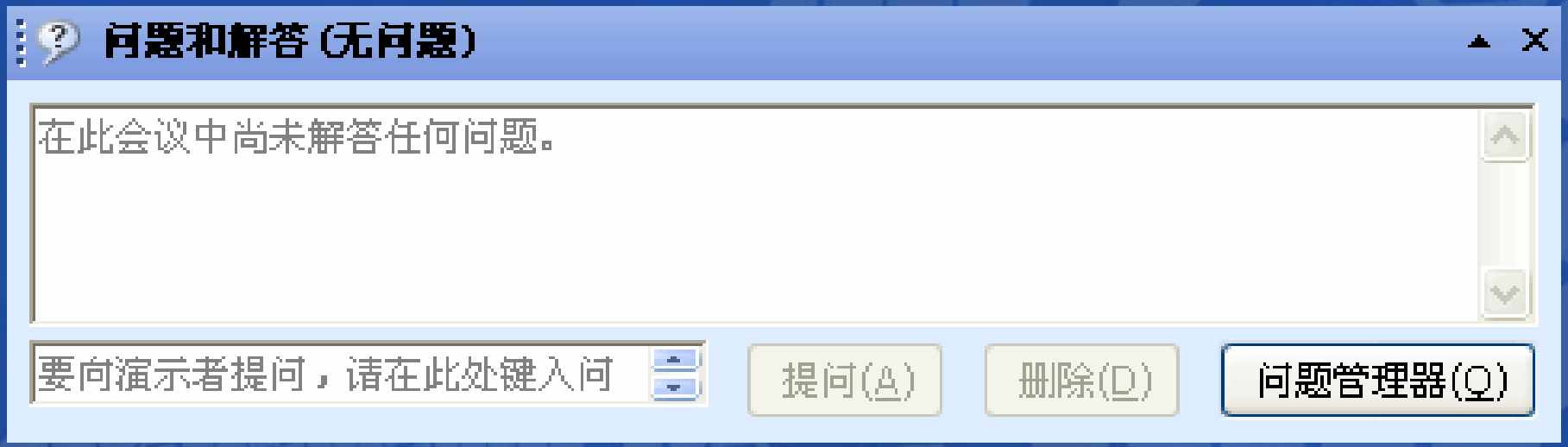
您的潜力，我们的动力
Microsoft
微软（中国）有限公司

- 《设计模式：可复用面向对象软件的基础》GoF
- 《面向对象分析与设计》Grady Booch
- 《敏捷软件开发：原则、模式与实践》Robert C. Martin
- 《重构：改善既有代码的设计》 Martin Fowler
- 《Refactoring to Patterns》Joshua Kerievsky

您的潜力，我们的动力
Microsoft
微软(中国)有限公司

Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。




您的潜力，我们的动力
Microsoft
微软（中国）有限公司
Microsoft
msdn
MSDN Webcasts