

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

## 6. Prototype 原型（创建型模式）

李建忠

[jianzhong.lee@gmail.com](mailto:jianzhong.lee@gmail.com)

设计模式论坛

<http://forum.sucent.com>

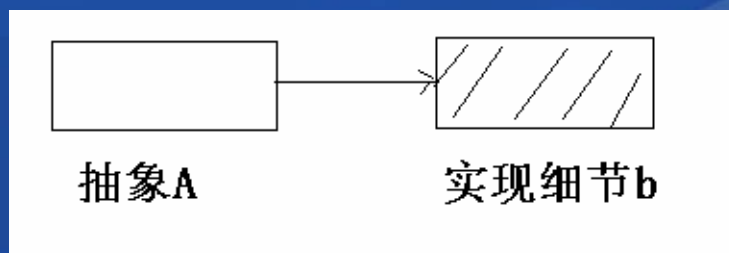
上海祝成科技 高级培训讲师

[www.sucent.com](http://www.sucent.com)

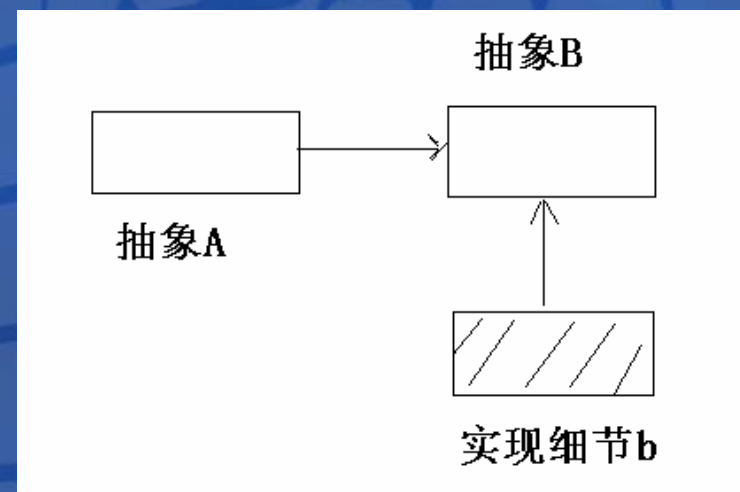
# 依赖关系的倒置

抽象不应该依赖于实现细节，实现细节应该依赖于抽象。

— 抽象A直接依赖于实现细节b



— 抽象A依赖于抽象B，实现细节b依赖于抽象B



# 动机 (Motivation)

在软件系统中，经常面临着“某些结构复杂的对象”的创建工作；由于需求的变化，这些对象经常面临着剧烈的变化，但是它们却拥有比较稳定一致的接口。

如何应对这种变化？如何向“客户程序（使用这些对象的程序）”隔离出“这些易变对象”，从而使得“依赖这些易变对象的客户程序”不随着需求改变而改变？

# 意图 (Intent)

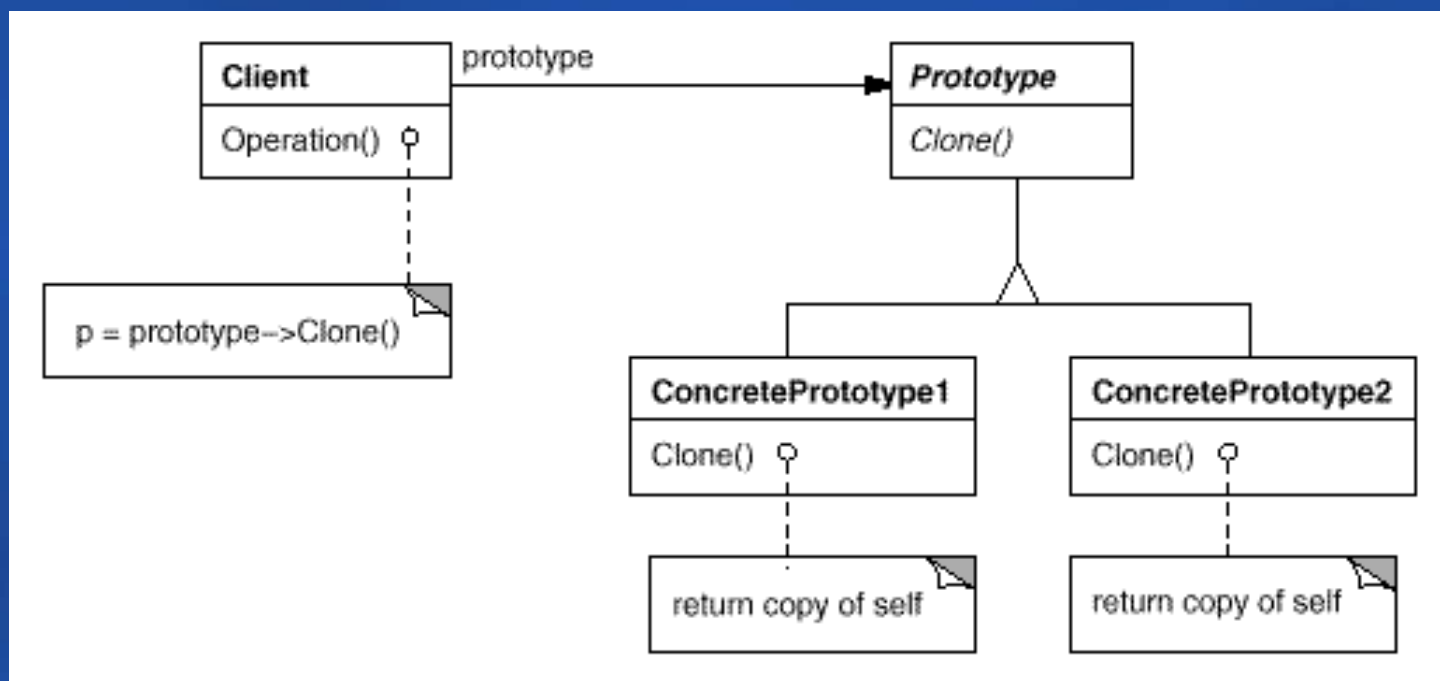
您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

使用原型实例指定创建对象的种类，然后通过拷贝这些原型来创建新的对象。

——《设计模式》GoF

# 结构 (Structure)





# 例说Prototype应用

您的潜力，我们的动力

**Microsoft**<sup>®</sup>  
微软(中国)有限公司

## Codes in VS.NET

# Prototype模式的几个要点

- **Prototype**模式同样用于隔离类对象的使用者和具体类型（易变类）之间的耦合关系，它同样要求这些“易变类”拥有“稳定的接口”。
- **Prototype**模式对于“如何创建易变类的实体对象”采用“原型克隆”的方法来做，它使得我们可以非常灵活地动态创建“拥有某些稳定接口”的新对象——所需工作仅仅是注册一个新类的对象（即原型），然后在任何需要的地方不断地Clone。
- **Prototype**模式中的Clone方法可以利用.NET中的Object类的MemberwiseClone()方法或者序列化来实现深拷贝。

# 有关创建性模式的讨论

- Singleton模式解决的是实体对象个数的问题。除了Singleton之外，其他创建型模式解决的都是new所带来的耦合关系。
- Factory Method, Abstract Factory, Builder都需要一个额外的工厂类来负责实例化“易变对象”，而Prototype则是通过原型（一个特殊的工厂类）来克隆“易变对象”。
- 如果遇到“易变类”，起初的设计通常从Factory Method开始，当遇到更多的复杂变化时，再考虑重构为其他三种工厂模式（Abstract Factory, Builder, Prototype）。




# 推荐参考书

- 《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



# 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。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

**Microsoft®**  
微软(中国)有限公司

**Microsoft®**

msdn  


**MSDN Webcasts**