

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

## 4.Builder 生成器（创建型模式）

李建忠

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

上海祝成科技高级培训讲师

# Builder模式的缘起

- 假设创建游戏中的一个房屋House设施，该房屋的构建由几个部分组成，且各个部分要富于变化。
- 如果使用最直观的设计方法，每一个房屋部分的变化，都将导致房屋构建的重新修正.....

# 动机 (Motivation)

在软件系统中，有时候面临着“一个复杂对象”的创建工作，其通常由各个部分的子对象用一定的算法构成；由于需求的变化，这个复杂对象的各个部分经常面临着剧烈的变化，但是将它们组合在一起的算法却相对稳定。

如何应对这种变化？如何提供一种“封装机制”来隔离出“复杂对象的各个部分”的变化，从而保持系统中的“稳定构建算法”不随着需求改变而改变？

# 意图 (Intent)

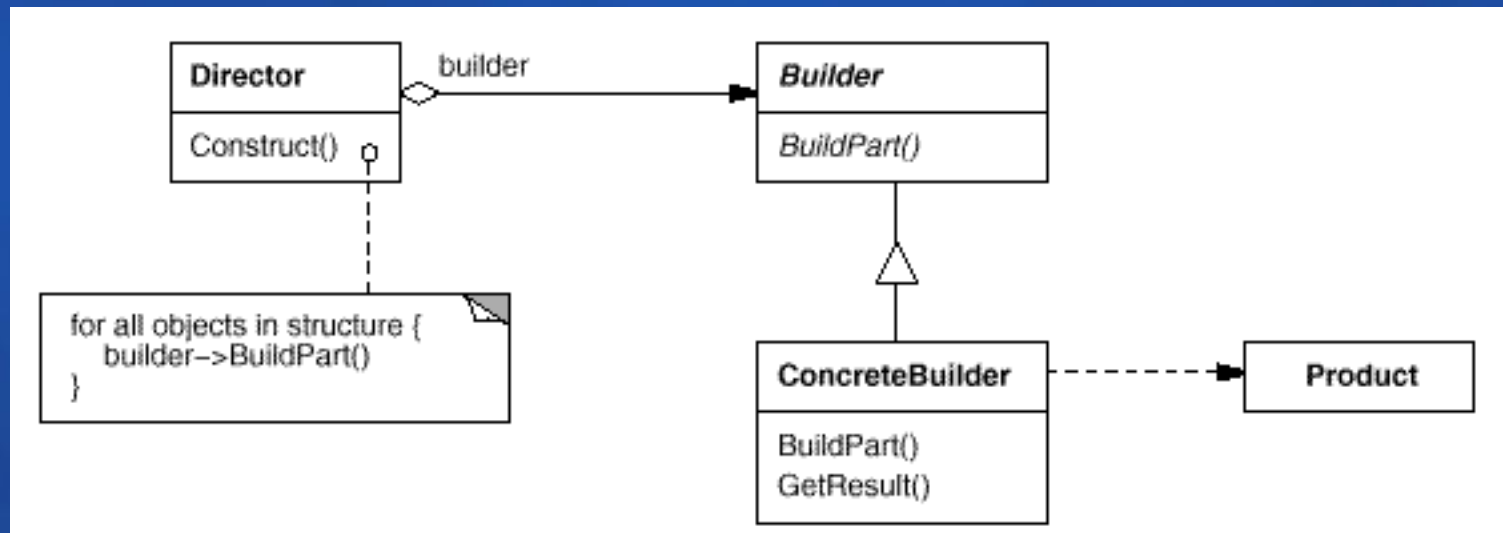
您的潜力, 我们的动力

**Microsoft**  
微软(中国)有限公司

将一个复杂对象的构建与其表示相分离, 使得同样的构建过程可以创建不同的表示。

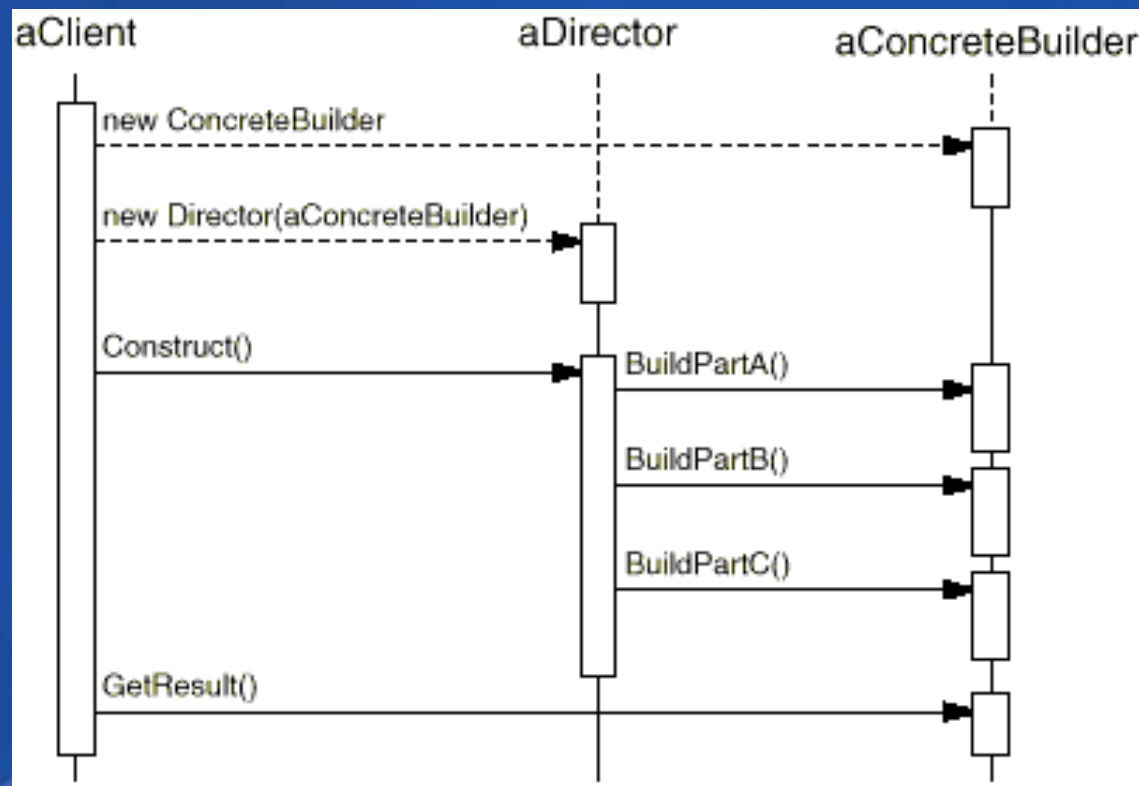
——《设计模式》GoF

# 结构 (Structure)





# 协作 (Collaborations)



# 游戏框架中的Builder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**

# Builder 模式的几个要点

- Builder 模式主要用于“分步骤构建一个复杂的对象”。在这其中“分步骤”是一个稳定的算法，而复杂对象的各个部分则经常变化。
- 变化点在哪里，封装哪里——Builder模式主要在于应对“复杂对象各个部分”的频繁需求变动。其缺点在于难以应对“分步骤构建算法”的需求变动。
- Abstract Factory模式解决“系列对象”的需求变化，Builder模式解决“对象部分”的需求变化。Builder模式通常和Composite模式组合使用。



您的潜力，我们的动力

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

# .NET框架中的Builder应用


**Codes in VS.NET**

## 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**