

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

## 12. Flyweight享元（结构型模式）

李建忠

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

设计模式论坛:

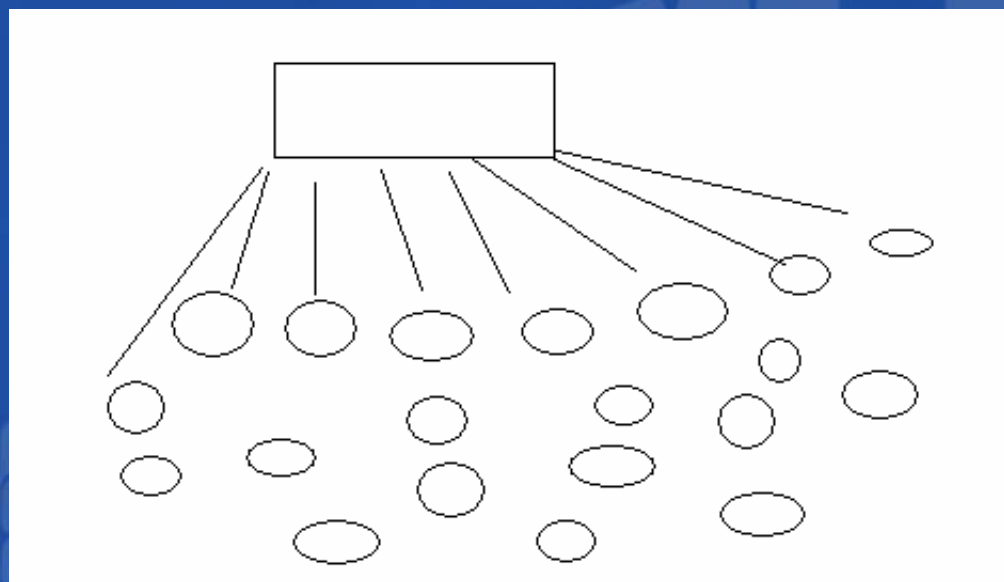
[forum.softcompass.com](http://forum.softcompass.com)

上海祝成科技 高级培训讲师

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

# 面向对象的代价

面向对象很好地解决了系统抽象性的问题，同时在大多数情况下，也不会损及系统的性能。但是，在某些特殊的应用中，由于对象的数量太大，采用面向对象会给系统带来难以承受的内存开销。比如图形应用中的图元等对象、字处理应用中的字符对象等。



## 动机 (Motivation)

采用纯粹对象方案的问题在于大量细粒度的对象会很快充斥在系统中，从而带来很高的运行时代价——主要指内存需求方面的代价。

如何在避免大量细粒度对象问题的同时，让外部客户程序仍然能够透明地使用面向对象的方式来进行操作？

# 意图 (Intent)

您的潜力，我们的动力

**Microsoft**  
微软(中国)有限公司

运用共享技术有效地支持大量细粒度的对象。

——《设计模式》GoF

# 例说Flyweight应用

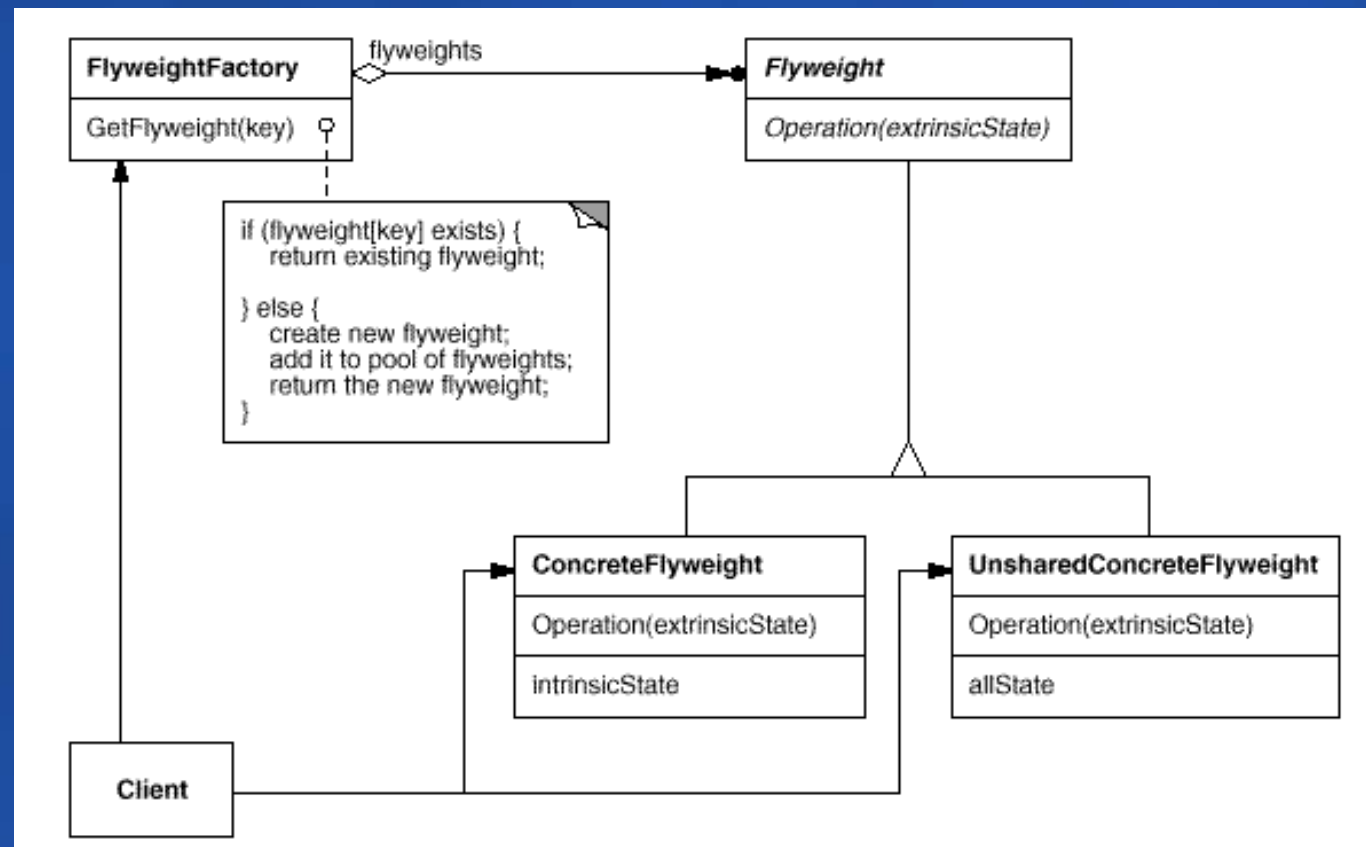
您的潜力，我们的动力

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

Codes in VS.NET



# 结构 (Structure)



# Flyweight模式的几个要点

- 面向对象很好地解决了抽象性的问题，但是作为一个运行在机器中的程序实体，我们需要考虑对象的代价问题。  
Flyweight设计模式主要解决面向对象的代价问题，一般不触及面向对象的抽象性问题。
- Flyweight采用对象共享的做法来降低系统中对象的个数，从而降低细粒度对象给系统带来的内存压力。在具体实现方面，要注意对象状态的处理。
- 对象的数量太大从而导致对象内存开销加大——什么样的数量才算大？这需要我们仔细的根据具体应用情况进行评估，而不能凭空臆断。

您的潜力，我们的动力

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

# .NET架构中的Flyweight应用

**Codes in VS.NET**



# 推荐资源


- 《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

## 获取更多MSDN资源

- **MSDN中文网站**  
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**  
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**  
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**  
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.msp>



# 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**