

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

25. 设计模式总结

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

创建型模式

- 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）。

结构型模式

- **Adapter**模式注重转换接口，将不吻合的接口适配对接
- **Bridge**模式注重分离接口与其实现，支持多维度变化
- **Composite**模式注重统一接口，将“一对多”的关系转化为“一对一”的关系
- **Decorator**模式注重稳定接口，在此前提下为对象扩展功能
- **Façade**模式注重简化接口，简化组件系统与外部客户程序的依赖关系
- **Flyweight** 模式注重保留接口，在内部使用共享技术对对象存储进行优化
- **Proxy** 模式注重假借接口，增加间接层来实现灵活控制

行为型模式 (1)

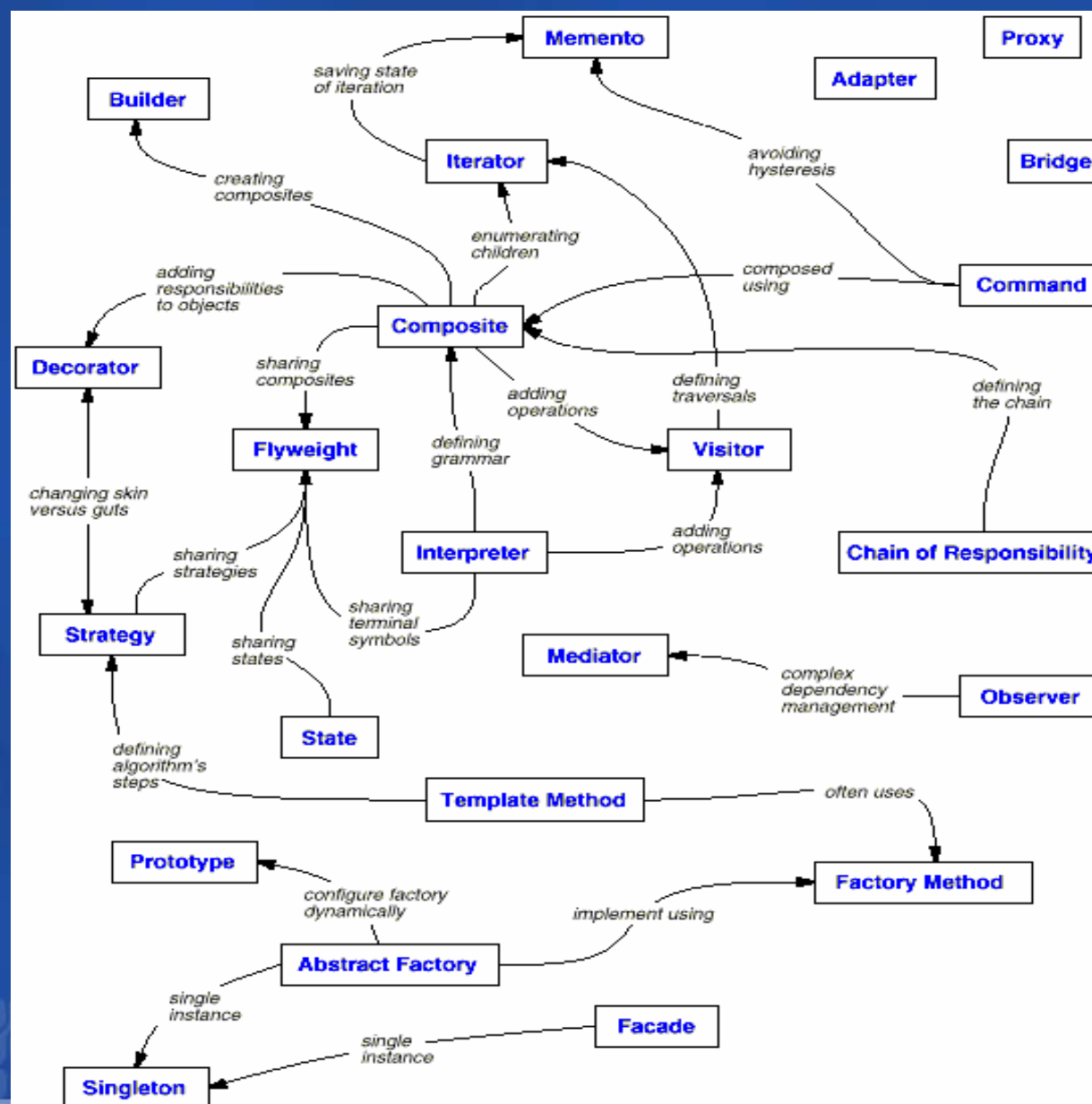
- **Template Method**模式封装算法结构，支持算法子步骤变化
- **Strategy**模式注重封装算法，支持算法的变化
- **State**模式注重封装与状态相关的行为，支持状态的变化
- **Memento**模式注重封装对象状态变化，支持状态保存/恢复
- **Mediator**模式注重封装对象间的交互，支持对象交互的变化

行为型模式（2）

- Chain Of Responsibility模式注重封装对象责任，支持责任的变化
- Command模式注重将请求封装为对象，支持请求的变化
- Iterator 模式注重封装集合对象内部结构，支持集合的变化
- Interpreter模式注重封装特定领域变化，支持领域问题的频繁变化
- Observer模式注重封装对象通知，支持通信对象的变化
- Visitor模式注重封装对象操作变化，支持在运行时为类层次结构动态添加新的操作。

您的潜力, 我们的动力

Microsoft®
微软(中国)有限公司



例说模式综合应用

您的潜力，我们的动力

Microsoft[®]
微软(中国)有限公司

Codes in VS.NET

设计模式应用总结

- 设计模式建立在对系统变化点的基础上进行，哪里有变化点，哪里应用设计模式。
- 设计模式应该以演化的方式来获得，系统的变化点往往是经过不断演化才能准确定位。
- 不能为了模式而模式，设计模式是一种软件设计的软力量，而非规范标准。不应夸大设计模式的作用。

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

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts