

# 设计模式:必须知道的设计模式

疯狂代码 <http://CrazyCoder.cn/> [j:http://CrazyCoder.cn/Programming/Article49829.html](http://CrazyCoder.cn/Programming/Article49829.html)

## 引言

设计模式是面向对象思想集大成GOF在其经典著作中整理总结了23种设计模式又可分为:创建型、结构型和行为型3个大类对于软件Software设计者来说般过程就是在熟练掌握语言背景基础上了解类库大致框架和常用和接口等然后多再在百般锤炼中提高对软件Software设计思想认识

软件Software设计者要清楚自己定位和方向味沉溺于技术细节思路是制约个人技术走向成熟毒药因此学习软件Software设计了解软件Software工程是每个开发人员必修课笔者在此不想详细描述各个设计模式细节我想google和baidu上资料已经多如牛毛了而且争取学习思路方法也不是了解所有设计模式就可以无敌于天下我所强调学习思路方法就是在熟练掌握基本要素基础上了解大致框架这条不仅是学习类库思路方法对设计模式来说是可行同时切记是在平时积累中不断体会和实战因此本文目就是将23种模式中必须掌握几个最关键、最常用设计模式做以整理总结和简述

## 1 Factory Pattern

上榜理由:将中创建对象操作单独出来处理大大提高了系统扩展柔性接口抽象化处理给相互依赖对象创建提供了最好抽象模式

## 2 Facade Pattern

上榜理由:将表现层和逻辑层隔离封装底层复杂处理为用户提供简单接口这样例子随处可见门面模式很多时候更是种系统架构设计在我所做项目中就实现了门面模式接口为复杂系统解耦提供了最好解决方案

## 3 Command Pattern

上榜理由:将请求封装为对象从而将命令执行和责任分开通常在队列中等待命令这和现实多么相似呀如果你喜欢发号施令请考虑你ICommond吧

## 4 Strategy Pattern

上榜理由:策略模式将易于变化部分封装为接口通常Strategy 封装些运算法则使的能互换Bruce Zhang在他博客中提到策略模式其实是种“面向接口”编程思路方法真是恰如其分

## 5 Iterator Pattern

上榜理由:相信任何系统中都会用到、集合、链表、队列这样类型吧那么你就不得不关心迭代模式来龙去脉在遍历算法中迭代模式提供了遍历顺序访问容器GOF给出定义为:提供种思路方法访问个容器(container)对象中各个元素而又不需暴露该对象内部细节.NET中就是使用了迭代器来创建用于foreach集合

## 6 Adapter Pattern

上榜理由:在原类型不做任何改变情况下扩展了新接口灵活且多样适配切旧俗这种打破旧框框适配新格局思想是面向对象精髓以继承方式实现类Adapter模式和以聚合方式实现对象Adapter模式各有千秋各取所长看来把它叫做包装器点也不为过

## 7 Observer Pattern

上榜理由:定义对象间种对多依赖关系,当个对象状态发生改变时,所有依赖于它对象都得到通知并被自动更新观察者和被观察者分开为模块划分提供了清晰界限在.NET中使用委托和事件可以更好实现观察者模式事件注册和撤销不就对应着观察者对其对象观察吗?

## 8 Bridge Pattern

上榜理由:把实现和逻辑分开对于我们深刻理解面向对象聚合复用思想甚有助益

## 9 Singleton Pattern

上榜理由:改善全局变量和命名空间冲突可以说是种改良了全局变量这种个类只有个例子且提供个访问全局点方式更加灵活保证了例子创建和访问约束.NET Frameeork已经封装了Singleton类我们拿来即可

## 整理总结

仁者见仁以上只是笔者家的言更重要真知灼见皆来源于实战设计思想和模式应用也来源于不断学习和反复我也将如既往此文只是开端未来才是不断探索

2009-1-9 15:48:03

疯狂代码 <http://CrazyCoder.cn/>