# 技术文章

## 说明

1. 文字语言随意化，口语风格，易理解，程序猿亲切的语言，必须融入专业知识。
2. 能过紧跟时事热点最好，其中融入专业名词。
3. 填入参考或者相关的文献资料
4. 有啥说啥，不必为了写文章而矫情
5. 鼓励学习，进步，快乐和幸福。
6. 评论中专业性或有用的则引入到正文中。

单独申请微博账号？

## 程序猿的吐槽一——序言

介绍

账号

说明（开始原因）

免责声明

希望

以前上班的地方不能上网，所以这么多年白天上班的时候几乎没有上过网。又加之自己喜欢看书，所以书倒是看了很多。憋了这么久之后，现在终于可以在有互联网的情况下一吐为快。吐槽不分对错，不分内容，不分对象，吐槽仅仅是为了吐槽而已。可以较真也可以不较真。真可谓是“假作真时真亦假”。

既然是程序猿吐槽，那么就合成了一个用来吐槽的名字。使用了我最崇拜的两个人的名字合成了一个Martin Stallman（Martin Flower和Richard Stallman）。

正如开头所说的，吐槽不分对错，没有理由。都是随心所想而已。当然吐槽也是有引用的，我都会尽量标明。这算我的免责声明吧，如有雷同，纯属巧合。

因为个人关注方向的不同，所以吐槽中的背景或者依赖的一些知识或者侧重点就会不同。我个人的方向有必要说一下，免得大家的侧重点或者关注点不同而产生沟通的深沟。个人喜欢或者关注面向对象，重构，模式，组件化，交互理论。除掉计算机技术方面，还喜欢看经济学，历史，心理学，文学方面的东西。当然空闲时间喜欢羽毛球还有台球，书法也是我的最爱，毛笔和硬笔书法都是。

最后想要吐槽的可以一起来吐槽。大家吐槽才是真的吐槽。

## 程序猿的吐槽二

## 程序猿的吐槽三

## 程序猿的吐槽四

## 面向对象模式（OOP-Pattern）

**动态加载**（set关联对象），继承变成组合。

桥接模式核心思想。

状态模式

命令模式(针对指定命令使用专一的调用者Invoker,Invoker和命令是同生死的(这里是聚合,应该不是同生死,即调用者可以调用不同命令),代码实现中有瑕疵,使用了构造方法)

备忘录模式——CareTaker类对拥有的Memento进行动态的加载。前者管理后者。

**代理控制**（当前类方法的核心使用关联对象的方法，即当前方法核心业务处理实际是一个代理方法，用于解耦或者分离控制）

代理模式，

命令模式。

装饰模式

外观模式（关联对象方法的组合，一组方法）

适配器模式

策略模式

状态模式

命令模式(调用者的Action执行命令的Excute),命令执行的核心(Excute)是命令接收的执行(receiver.Action(), )

中介者模式——使用中介者类将同事类之间的关系分立开来，在中介者类中进行控制，如果执行则在中介者使用同事类方法，也是一种代理。

**同生同死**（关联对象和当前对象同样生命周期，构造方法中设置关联对象）

策略模式()

命令模式(每一个命令的生存都依托一个命令接受者,同生同死)

中介者模式（Colleague类出生的时候就需要拥有一个Mediator对象）

**对象缓存**（缓存和延迟加载，关联的对象不立即创建，需要时创建，也不立即销毁，用于缓存）

单例模式

享元模式

**分离无关操作**（将容易变化的部分分离出来）

访问者模式，（将Element结果中的访问操作分离出来，形成Visitor类）