

Java私塾-最专业的Java就业培训专家，因为专业，所以出色！值得你的信赖！



# 私塾在线 《软件系统功能设计实战训练》 ——跟着CC学设计系列精品教程

10101010101010101010101010101

## 本周设计作业

**n** 本周设计作业的项目背景：

编号/文号生成器——来自中国人寿企业年金系统

**n** 学习目标：

- (1) 在实战中练习设计的理念和方法
- (2) 学习如何设计API，如何应对业务的变更，体会应用功能需求的多变性，体会优秀设计的重要性的和对变化的支持
- (3) 学习缓存的应用，高并发的处理策略等
- (4) 综合应用工厂方法/简单工厂模式、桥接模式、策略模式、适配器模式、值对象模式、单例模式、观察者模式等，包含着这些模式但不限于这些模式该设计方式可应用于多种有类似功能的系统，比如：固定资产编号、文件编号、所有需要有格式号的系统……

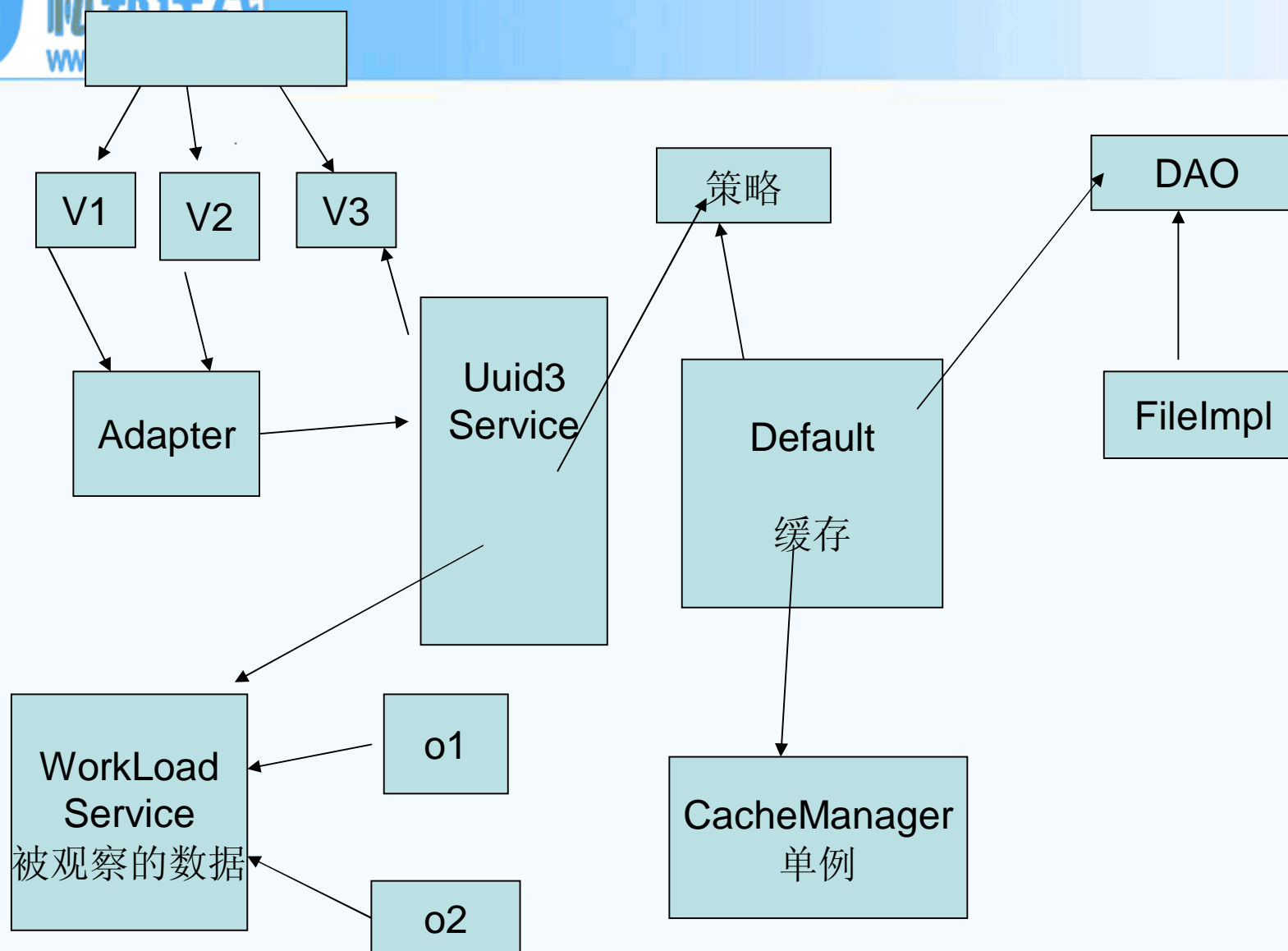
## 本周设计作业

### n 基本功能

- 1: 在保证兼容前两版的情况，实现第三版，要求能够对不同的文号生成进行缓存，以提高运行速度，提高并发的支持能力，引入缓存的使用，还有单例模式
- 2: 实现业务工作量监控，如果某个业务，每天新增的业务量达到一定的阈值，就会启动相应的业务调整，比如检查是否有恶意攻击，或者是在某些岗位增派人员进行处理等，或者是暂停收取新业务等处理，引入观察者模式。

### n 作业要求

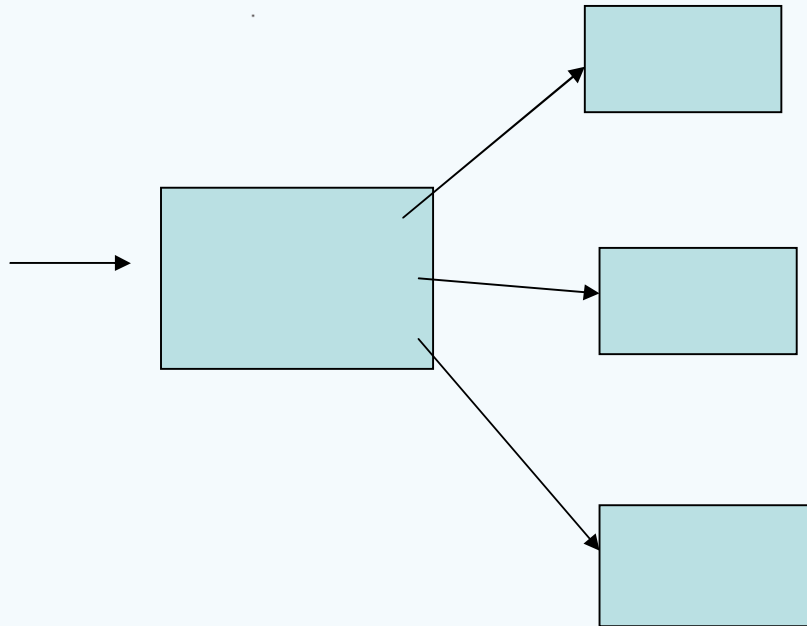
- 1: 在老师给定的概要代码基础上，实现上述基本要求的功能设计
- 2: 重点在接口和功能实现的设计上，无须关注具体实现
- 3: 对于每个api，可以适当写点样例代码，能够调用运行更佳
- 4: 考虑合理的结构，职责的划分，以及设计模式的合理使用



做最好的在线学习社区

网 址: <http://sishuok.com>  
咨询QQ: 2371651507





**做最好的在线学习社区**

网 址：<http://sishuok.com>

咨询QQ：2371651507

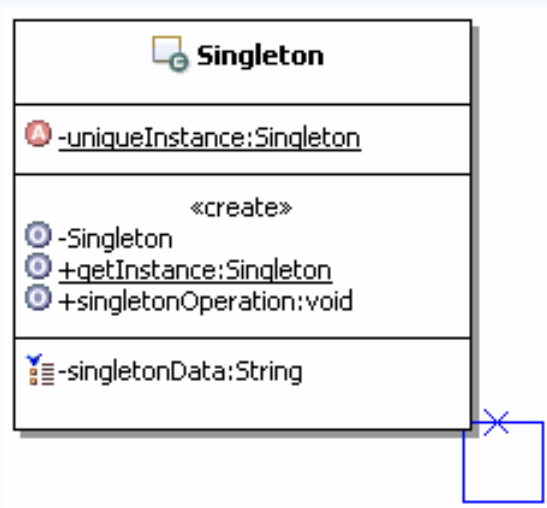
私塾在线<http://sishuok.com?frombook> 独家提供配套教学视频，更有大量免费视频独家大放送

## 初识单例模式

### n 定义

保证一个类仅有一个实例，并提供一个访问它的全局访问点。

### n 结构和说明



Singleton:

负责创建Singleton类自己的唯一实例，并提供一个getInstance的方法，让外部来访问这个类的唯一实例。

做最好的在线学习社区

网 址: <http://sishuok.com>

咨询QQ: 2371651507



## 单例模式的知识要点

### n 单例模式的知识要点

- 1: 单例模式是用来保证一个类在运行期间只会被创建一个类实例的
- 2: 单例模式还提供了一个全局唯一访问这个类实例的访问点，通常就是那个getInstance方法
- 3: 单例模式只关心类实例的创建问题，并不关心具体的业务功能。
- 4: 单例模式是一个虚拟机范围内单例，不适用于集群等环境
- 5: 单例模式的实现有很多种，除了常见的懒汉式和饿汉式外，还有利用缓存来实现、双重检查加锁的实现、Lazy initialization holder class模式、以及枚举的实现方式等
- 6: 使用单例模式的时候要注意它的线程安全性
- 7: 单例模式体现了延迟加载、缓存等常见的设计思想
- 8: 单例模式是可以很容易的扩展到多实例控制的



## 思考单例模式

### n 单例模式的本质

单例模式的本质是：控制实例数目

### n 何时选用单例模式

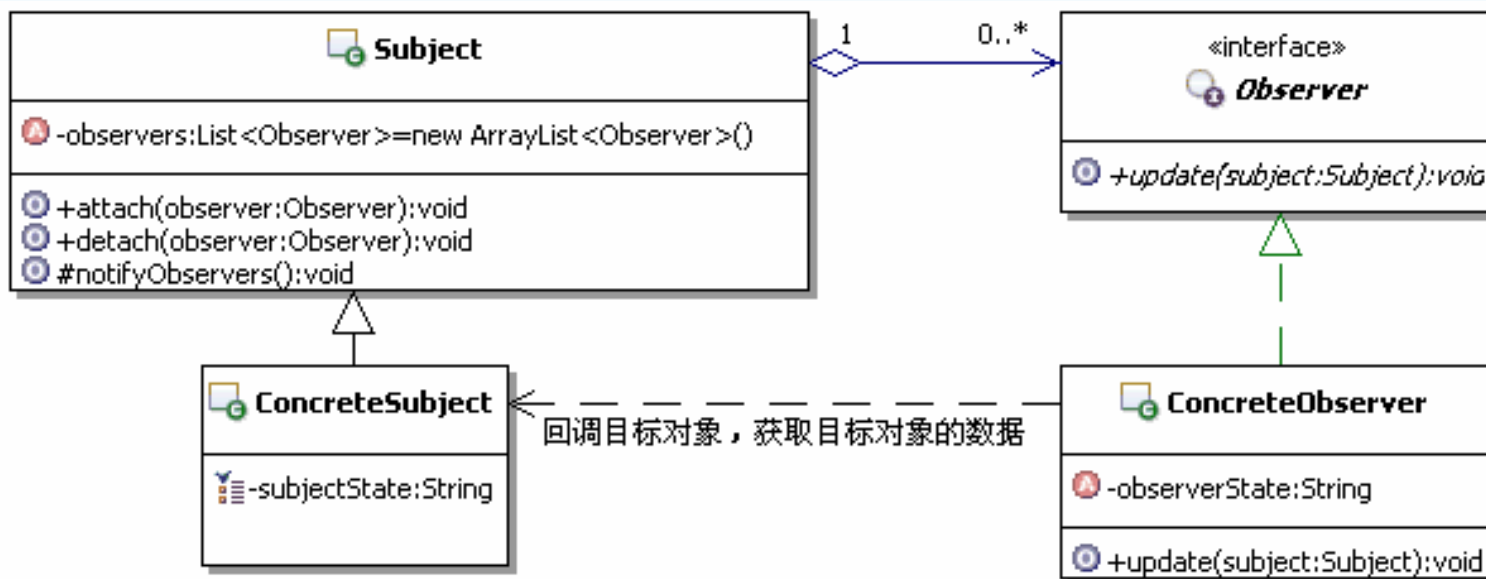
当需要控制一个类的实例只能有一个，而且客户只能从一个全局访问点访问它时，可以选用单例模式，这些功能恰好是单例模式要解决的问题

## 初识观察者模式

### n 定义

定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并被自动更新。

### n 结构和说明



## 观察者模式的知识要点

### n 观察者模式的知识要点

- 1: 观察者模式中的目标和观察者之间是典型的一对多的关系，但是也可以只有你一个观察者，也就是可以变相实现了目标和观察者之间一对一的关系
- 2: 一个观察者可以观察多个目标，通常在这种情况下，观察者应该为不同的观察者目标，定义不同的回调方法
- 3: 在观察者模式中，观察者和目标是单向依赖的，只有观察者依赖于目标，而目标是不会依赖于观察者的
- 4: 观察者模式的目标对象在实现的时候，通常需要维护观察者的注册信息
- 5: 目标在通知观察者的时候，一般要确保目标的状态更新已经结束，否则会导致观察者和目标对象的状态不一致
- 6: 如果出现目标和观察者相互观察的情况，一定要特别小心处理，因为可能会出现死循环的情况
- 7: 观察者模式在具体实现中，又分为推模型和拉模型，可以同时实现这两种模型
- 8: 可以在程序运行期间，通过动态的控制观察者，来变相的实现添加和删除某些功能处理，这些功能就是观察者在update的时候执行的功能

## 思考观察者模式

### n 观察者模式的本质

观察者模式的本质是：**触发联动**

### n 何时选用观察者模式

- 1: 当一个抽象模型有两个方面，其中一个方面的操作依赖于另一个方面的状态变化，那么就可以选用观察者模式。
- 2: 如果在更改一个对象的时候，需要同时连带改变其它的对象，而且不知道究竟应该有多少对象需要被连带改变，这种情况可以选用观察者模式，被更改的那一个对象很明显就相当于是目标对象，而需要连带修改的多个其它对象，就作为多个观察者对象了。
- 3: 当一个对象必须通知其它的对象，但是你又希望这个对象和其它被它通知的对象是松散耦合的，也就是说这个对象其实不想知道具体被通知的对象，这种情况可以选用观察者模式，这个对象就相当于是目标对象，而被它通知的对象就是观察者对象了。