一、概况：设计模式是一套被反复使用、多数人知晓、晋国分类编目的、代码设计经验总结

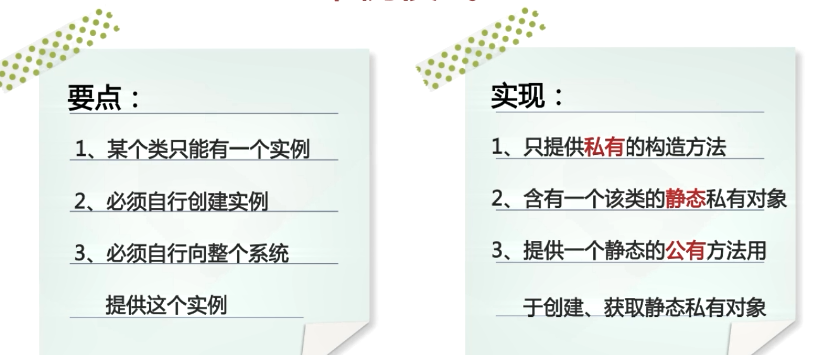


设计模式是基于场景的解决方案

二、单例模式

1、目的：使得一个对象成为该类系统中的唯一实例

2、定义：一个类有且仅有一个实例，并且自行实例化像整个系统提供



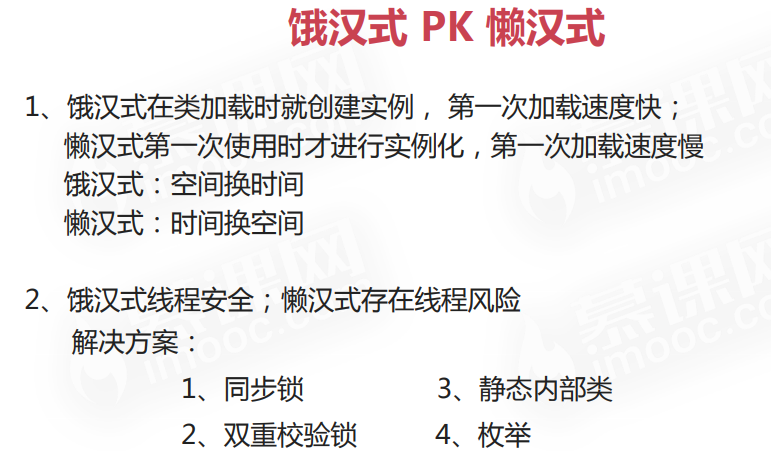
3、实现方案：饿汉式、懒汉式

a）饿汉式：

package com.singleton;  
//饿汉式：创建对象实例的时候直接初始化，空间换时间  
public class SingletOnOne {  
 //1、创建类中私有构造  
 private SingletOnOne(){  
  
 }  
 //2、创建该类型的私有静态实例  
 private static SingletOnOne instance = new SingletOnOne();  
 //3、创建共有静态方法返回静态实例对象  
 public static SingletOnOne getInstance(){  
 return instance;  
 }  
}

b）懒汉式：类内实例对象创建时并不直接初始化，知道第一次道用get方法时，才完成初始化操作，时间换空间

package com.singleton;  
//懒汉式  
public class SingletOnTwo {  
 //1、创建私有构造方法  
 private SingletOnTwo(){  
  
 }  
 //2、创建静态的该类实例对象  
 private static SingletOnTwo instance = null;  
 //3、创建开放的静态方法提供实例对象  
 public static SingletOnTwo getInstance(){  
 if(instance == null){  
 instance = new SingletOnTwo();  
 }  
 return instance;  
 }  
}



4、优点：

内存中只有一个对象，节省空间

避免平凡销毁创建对象，提高性能

避免对共享资源多重占用

5、缺点

扩展比较困难

如果实例化后的对象长期不利用，系统将默认为垃圾进行回收，造成对象状态丢失

6、使用场景

创建对象时占用资源过多，但同上又需要用到该类对象

对系统内资源要求统一读写，如读写配置信息

当多个实例存在可能引起程序逻辑错误，如号码生成器