第一题：git reset的作用？soft, hard, mixed的区别？add的作用？

【答案】

git reset的作用是重置到特定的一次提交，删除中间的错误提交。

如果当前分支是master，且HEAD指向最新的一次提交。

暂存区：add命令，从工作区加入暂存区，等待提交

工作区：当前正在修改的区域

git reset [--soft, --mixed, --hard] commitId

soft：将HEAD指向指定的commitId，不修改暂存区和工作区

mixed：将HEAD指向指定的commitId，并清空暂存区，但工作区的代码不变

hard：将HEAD指向指定的commitId，清空暂存区，工作区代码也回退，此时可能丢失数据

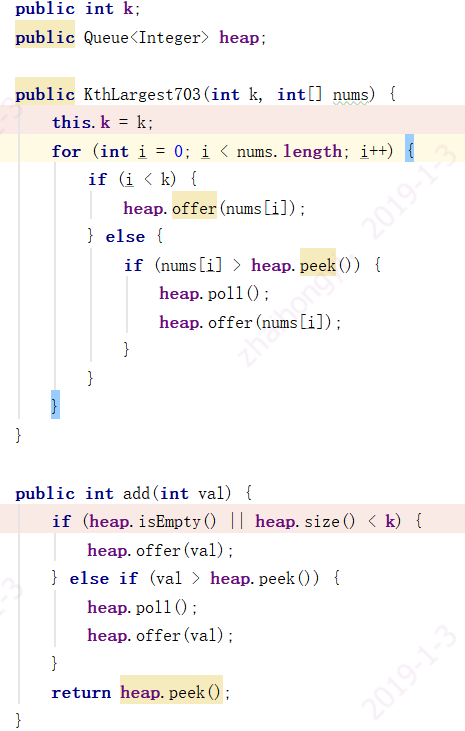
【回答】将本地工作区的代码重置到某一个版本。hard会丢失。add是加入版本控制？soft和mixed不清楚。

第二题：数据流中第K大的元素

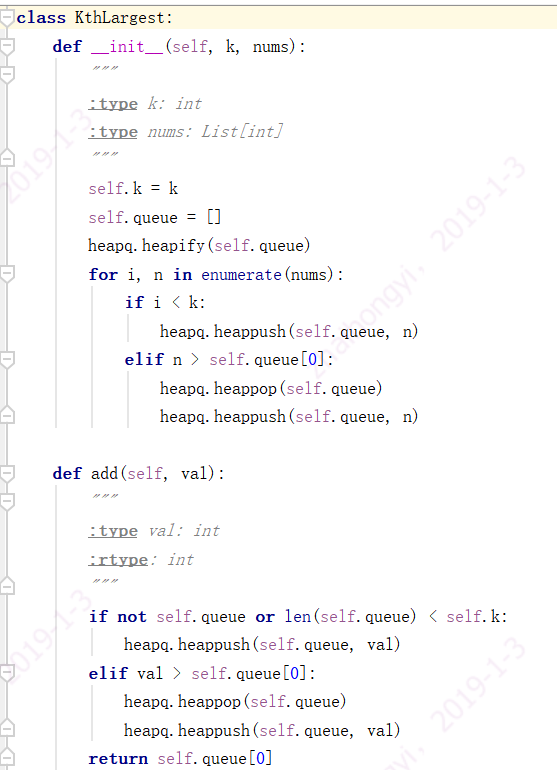
<https://leetcode-cn.com/problems/kth-largest-element-in-a-stream/description/>

【答案】

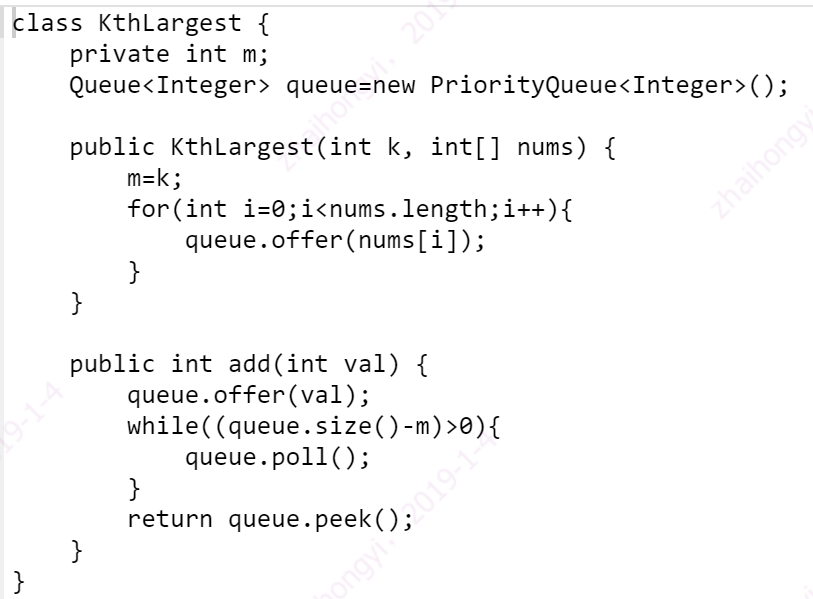
【Java】



【python】



【回答】



第三题：请说出你在工作中使用到的设计模式，好处是什么。

什么是模板模式？什么是代理模式？f什么是适配器模式？单例模式如何实现？

【答案】

模板模式：父类定义执行方法，抽象方法，不同子类实现不同。

代理模式：代理类和委托类（被代理类）具有相同的方法。代理类不做真正的业务处理，可以做一些调用前检查和调用后的统计操作。这些公共操作没必要写在委托类中，因此符合开闭原则，在不改变委托类的同时，增强功能。

代理分为静态代理和动态代理。静态代理是指定类委托类的类型，只能代理一种类型。而动态代理使用反射实现，当构造代理类时传入了任何类型的委托对象，调用代理对象的invoke方法，并传入代理对象、调用的方法和参数。可以生成对任意委托对象的代理对象。



适配器模式：即A实例的动作需要委托B实例完成，B实例通过set方法（或其他适配方法）适配到A实例中。

单例模式：饿汉模式：静态属性，加载类时构造实例；懒汉模式：使用时构造实例，用volatile + 同步 + 双重校验。

【回答】

模板模式：交易流程，相似动作，校验、落库、发往银行、更新库、返回机构，一些列动作，大部分交易一样，抽象出来，抽象类。每个业务都会集成，有些交易特殊业务流程，复写某个动作。公用的抽象方法有些是在抽象类中实现，有些是在子类中实现。

模板模式好处：减少代码冗余，可读性强，增加复用。

什么场景使用模板模式：业务流程有相似的，顺序一致。

抽象类可以实例化，只能继承，抽象类中一定要有抽象方法。

工作中用的设计模式：继承抽象类、实现接口、子类继承父类，调用父类方法，复写，实现接口实现接口所有方法，实现多个接口。抽象类一定要实现接口的全部方法。