单例设计模式

1. 如果创建一个类，不想让外界实例化，可以创建一个构造方法且私有化，且没有其他的构造方法，这样外界就不可对该类进行实例化了。因为对于一个类，默认有1个空参数的公共的构造方法，将此覆盖，且私有化，外界就访问不了了。
2. 单例设计模式：

**首先把该类的对象创建好，什么时候需要什么时候给出****。**

public final class DriverUtils2 {

private DriverUtils2() {

}

**private static DriverUtils2 dus = new DriverUtils2();**

**public DriverUtils2 getInstance() {**

**return dus;**

**}**

}

1. 延迟加载的单例设计模式：

**什么时候需要该类的对象，什么时候就创建然后给出。**

**延迟加载的单例设计模式，需要注意同步问题。利用双重判断的方法来进行同步判断，以防产生多个实例对象。**

**public** **class** DriverUtils3 {

**private** DriverUtils3(){

}

**private** DriverUtils3 dus3 = **null**;

**public** DriverUtils3 getInstance() {

**if (dus3 == null) {**

**synchronized** (DriverUtils3.**class**) {

if(dus3 == null) {//双重检查，不可缺少，否则会造成创建多个对象。

dus3 = **new** DriverUtils3();

}

}

}

**return** dus3;

}

}