# 单例设计模式:确保一个类只有一个对象

单例设计模式

确保一个类只有一个对象

懒汉式 double checking

1、构造器私有化,避免外部直接创建对象

2、声明一个私有的静态变量

3、创建一个对外的公共的静态方法 访问该变量，如果变量没有对象，创建该对象

首先：声明一个私有的静态变量

private static Jvm instance = null;

构造器私有化：

private Jvm(){

}

创建一个对外的公共的静态方法 访问该变量，如果变量没有对象，创建该对象

public static Jvm getInstance(long time){

// c d e -->效率 提供 已经存在对象的访问效率

if(null==instance){

// a b

synchronized(Jvm.class){

if(null==instance ){

try {

Thread.sleep(time); //延时 ，放大错误

} catch (InterruptedException e) {

e.printStackTrace();

}

instance =new Jvm();

}

}

}//a

return instance;

}

