**JAVA编程进阶上机报告**

****

Homework2 Singleton

**学 院\_\_智能与计算学部 \_\_\_**

**专 业\_\_软件工程 \_\_\_**

**姓 名\_\_库尔班·阿布都看力木\_\_**

**学 号\_\_3018218059\_\_\_\_\_\_\_\_\_\_\_\_**

**年 级\_\_软件5班 \_\_\_\_\_\_\_\_\_\_\_\_**

**班 级\_\_2018级 \_\_\_\_\_\_\_\_\_\_\_**

# 作业要求

单重模式。在软件工程中，单例模式是一种软件设计模式，它将类的实例化限制在一个对象上。当需要一个对象来协调整个系统中的操作时，这非常有用。这个概念有时被泛化为当只有一个对象存在时更有效地操作的系统，或者将实例化限制为特定数量的对象的系统。

单例设计模式描述了如何解决上述问题：

（1） 隐藏类的构造函数。

（2） 定义返回类的唯一实例的公共静态操作。

请自己在Java中实现singleton模式。

# 源代码

**public** **class** SingLeton {

// 静态的instance对象，保证全局唯一性

**private** **static** SingLeton *instance* = **null**;

// 私有的构造函数，防止外部用new关键字创建实例对象

**private** SingLeton() {

}

// 对外的公共静态实例方法，从类级别直接可以调用此方法

**public** **static** SingLeton getInstance() {

// 通过判断instance是否为null，决定是否创建对象

**if** (*instance* == **null**) {

*instance* = **new** SingLeton();

}

**return** *instance*;

}

}