# JAVA编程进阶上机报告

****

**学 院 智能与计算学部**

**专 业 软件工程**

**班 级 3 班**

**学 号 3018216157**

**姓 名 郑开**

1. **实验要求**

Singleton pattern. In software engineering, the singleton pattern is a software design pattern that restricts the instantiation of a class to one object. This is useful when exactly one object is needed to coordinate actions across the system. The concept is sometimes generalized to systems that operate more efficiently when only one object exists, or that restrict the instantiation to a certain number of objects.

The singleton design pattern describes how to solve the above problems:

(1)Hide the constructor of the class.

(2)Define a public static operation that returns the sole instance of the class.

Please implement the singleton pattern in Java by yourself.

1. **源代码**

public class Singleton {   
 private static Singleton s = null;   
 private Singleton(){   
 }   
   
 public static synchronized Singleton getInstance() {   
 if(s == null){   
 s = new Singleton();   
 return s;   
 }   
 return s;   
 }   
}

单例模式要求一个类只有一个实例。所以构造函数必须私有。因为构造函数私有，所以只能在类内创建实例，并对外提供静态函数返回改实例。

1. **实验结果**

实现了单例模式。