单例模式

# 懒汉式

//懒汉式单例模式：安全且高效

**public** **class** Singleton {

**private** **static** Singleton *instance*;//注意必须是静态的

**private** **static** **byte**[] *lock* = **new** **byte**[0];

**public** **static** Singleton getInstance(){

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

// synchronized(Singleton.class){

**synchronized**(*lock*){

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

*instance* = **new** Singleton();

}

}

}

**return** *instance*;

}

}

# 饿汉式：不存在并发安全问题

//饿汉式单例模式

**public** **class** Singleton {

**private** **static** Singleton *instance* = **new** Singleton();

**public** **static** Singleton getInstance(){

**return** *instance*;

}

}

# 如何选择饿汉式还是懒汉式？

下面对单件模式的懒汉式与饿汉式进行简单介绍：

1、饿汉式：在**程序启动或单件模式类被加载**的时候，单件模式实例就已经被创建。

2、懒汉式：当程序**第一次访问单件模式实例**时才进行创建。

如何选择：如果单件模式实例在系统中**经常会被用到**，饿汉式是一个不错的选择。

反之如果单件模式在系统中会**很少用到或者几乎不会用到**，那么懒汉式是一个不错的选择。