

Java设计模式之Singleton模式计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/646/2021_2022_Java_E8_AE_BE_E8_AE_A1_c97_646100.htm

Java Singleton 模式用来保证在运行的应用程序中，一个Class只是实例化一次，也就是只有一个相应的对象存在。在 web 程序中我们会用一个核心的分配功能的Servlet程序，在这里我们就可以运用这种设计模式了。

一般Singleton模式通常有几种形式: 第一种形式: 定义一个类，它的构造函数为private的，它有一个static的private的该类变量，在类初始化时实例化，通过一个public的getInstance方法获取对它的引用,继而调用其中的方法。

```
public class Singleton { private Singleton(){} //在自己内部定义自己一个实例，是不是很奇怪？ //注意这是private 只供内部调用 private static Singleton instance = new Singleton(). //这里提供了一个供外部访问本class的静态方法，可以直接访问 public static Singleton getInstance() { return instance. } }
```

```
第二种形式: public class Singleton { private static Singleton instance = null. public static synchronized Singleton getInstance() { //这个方法比上面有所改进，不用每次都进行生成对象，只是第一次 //使用时生成实例，提高了效率！ if (instance==null) instance = new Singleton(). return instance. } }
```

例子：/** * 数据库连接管理类 100Test 下载

频道开通，各类考试题目直接下载。详细请访问
www.100test.com