

单例设计模式

好处

1. 对于频繁使用的对象，可以省略创建对象所花费的时间，这对于那些重量级对象而言，是非常可观的一笔系统开销
2. 由于new操作的次数减少，因而对系统内存大使用频率也会降低，这将减轻GC压力，缩短GC停顿时间

为什么不直接使用全局变量来确保只有一个实例

利用单例模式可以做到在需要的时候才创建对象，这样做就避免了不必要的资源浪费

两种构建方式

3. 饿汉式：全局的单例实例在类装载时构建
 4. 懒汉式：全局的单例实例在第一次被使用时构建
- 代码

- 饿汉式(线程安全)

```
public class Singleton{
    //在静态初始化器中创建单例实例，这段代码保证了线程安全
    static Singleton uniqueInstance = new Singleton();
    private Singleton(){}
    public static Singleton getInstance(){
        return uniqueInstance;
    }
}
```

- 懒汉式(静态内部类实现线程安全)

```
public class Singleton{
    private Singleton(){}
    public static Singleton getInstance(){
        return Inner.Singleton;
    }
    private static class Inner{
        private static Singleton singleton = new Singleton();
    }
}
```