

Java内存模型

- 什么是内存模型,为什么需要它
 - 平台的内存模型
 - 串行一致性
 - 重排序
 - Java内存模型简介
 - JMM为程序中所有的操作定义了一个偏序关系,称为Happens-Before
 - 程序顺序规则
 - 监视器锁规则
 - volatile变量规则
 - 线程启动规则
 - 线程结束规则
 - 中断规则
 - 终结器规则
 - 传递性
- 发布
 - 除了不可变对象以外,使用被另一个线程初始化的对象通常都是不安全的.除非对象的发布操作是在使用该对象的线程开始之前执行
 - 安全的发布
 - 提前初始化
 - 延长初始化占位类模式

```
@ThreadSafe
public class ResourceFactory {
    private static class ResourceHolder {
        public static Resource resource = new Resource();
    }

    public static Resource getResource() {
        return ResourceFactory.ResourceHolder.resource;
    }

    static class Resource {
    }
}
```
 - 双重检查锁(不安全),如要使用,把变量设置为volatile

- 初始化过程的安全性

- 初始化安全性将确保,对于被正确构造的对象,所有线程都能看到由构造函数为对象给各个final域设置的正确值,而不管采用何种方式来发布对象.
- 对于可以通过被正确构造对象中某个final域到达的任意变量(例如某个final数组中的元素,或者由一个final域引用的HashMap的内容)将同样对于其他线程是可见的.
- 初始化安全性只能保证通过final域可达的值从构造过程完成时开始的可见性.对于非final域的可达性,或者在构造过程完成后可能改变的值,必须采用同步来确保可见性