

单例模式

## 模式定义

确保类只有一个实例, 并提供一个全局访问点

## 问题描述

如何处理getInstance时的线程安全的问题

## Code

SinglePatternFirst.java

```
package Single;

public class SinglePatternFirst {
    private static SinglePatternFirst singlePatternFirst = null;

    private SinglePatternFirst() {
    }

    public synchronized static SinglePatternFirst getInstance() {
        if (singlePatternFirst != null) singlePatternFirst = new SinglePatternFirst();
        return singlePatternFirst;
    }
}
```

SinglePatternSecond.java

```
package Single;

public class SinglePatternSecond {
    private static SinglePatternSecond singlePatternSecond = new SinglePatternSecond();

    public static SinglePatternSecond getInstance() {
        return singlePatternSecond;
    }
}
```

SinglePatternThrid.java

```
package Single;

public class SinglePatternThrid {
    private SinglePatternThrid() {
    }

    private static SinglePatternThrid singlePatternThrid = null;

    public static SinglePatternThrid getInstance() {
        if (singlePatternThrid == null) {
            synchronized (SinglePatternThrid.class) {
                singlePatternThrid = new SinglePatternThrid();
            }
        }
        return singlePatternThrid;
    }
}
```

那么问题来了...怎么模拟多线程