

synchronized I

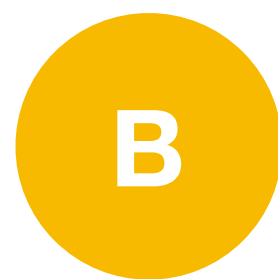
未标记同步的代码段

A

普通方法

B

同步代码块



```

public static void main(String[] args){
    Runnable runnable = new Runnable() {
        @Override
        public synchronized void run() {
            for(int i=0; i<10; i++){
                System.out.print(i);
            }
        }
    };

    new Thread(runnable, "T1").start();
    new Thread(runnable, "T2").start();
}

```

00112342534567896789

synchronized 关键字

```
public synchronized void doSomething(){  
    // 这里的代码不会有几个线程同时执行  
}
```

```
synchronized (this) {  
    // 这里的代码也不会有几个线程同时执行  
}
```

```
public class Sync {  
    public void doA() {  
        synchronized (this) {  
            // A  
        }  
    }  
  
    public synchronized void doB() {  
        // B  
    }  
  
    public synchronized static void doC() {  
        // C  
    }  
  
    public static void doD() {  
        synchronized (Sync.class) {  
            // D  
        }  
    }  
}
```

监视器

- `synchronized(object)` `object`是监视器
- 被 `synchronized` 修饰的方法
 - 静态方法：监视器是所属类的class对象
 - 非静态方法：监视器是this
- 同一个监视器内所有代码都是同步的，不论是否在同一函数、同一个类内

```
public synchronized void methodA(){  
    // A  
}  
public synchronized void methodB(){  
    // B  
}  
public void methodC(){  
    synchronized (this){  
        // C  
    }  
}
```

```
public class Sync {
```

```
    public void doA() {  
        synchronized (this) {  
            // A  
        }  
    }
```

监视器: Sync.this

```
    public synchronized void doB1() {  
        // B1  
    }
```

监视器: Sync.this

```
    public synchronized void doB2() {  
        // B2  
    }
```

监视器: Sync.this

```
    public synchronized static void doC() {  
        // C  
    }
```

监视器: Sync.class

```
    public void doD() {  
        synchronized (Sync.class) {  
            // D  
        }  
    }
```

监视器: Sync.class


```

Runnable runnable = new Runnable() {
    private List<String> list = new ArrayList<>();

    @Override
    public void run() {
        for(int i=0; i<3; i++){
            synchronized(this){
                list.add(Thread.currentThread().getName() + i);
            }
            list.add(Thread.currentThread().getName() + "-" + i);
            System.out.println(String.join(",", list));
        }
    }

    System.out.println(String.join(",", list));
}
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();
};
new Thread(runnable, "A").start();
new Thread(runnable, "B").start();
new Thread(runnable, "C").start();

```

“实践出真知”

– 写个程序验证下