

作业 4 说明

0. 实现一个名为 **EmuExe** 的类，类中定义一个静态方法 `static void execute()`。其实现可以为

```
try {
    Thread.sleep((int)(Math.random()*1000));
} catch (Exception e) {
}
```

或者你可以自己定义等待的方式。

1. 实现一个名为 **ThreadSync** 的类，类中定义 `main` 方法。
2. 该方法执行时，将创建两个线程(请分别命名线程的类为 **ThreadA** 和 **ThreadB**，以下称为线程 1 和线程 2)。并执行这两个线程。
3. 两个线程的 `run` 方法，各包含一个循环（循环 100 次）。
4. 线程 1 的循环，调用 `incA` 方法，该方法每次都自增一个 **ThreadA** 的整形变量（请定义为 `public static volatile` 类型，并命名为 **A**）的值，变量 **A** 的初始值为 0，每次循环皆自增 1，也即 100 次循环后，**A** 的值为 100。`incA()` 方法调用 `EmuExe.execute()` 模拟执行的时间，也即 `run` 方法的实现类似于

```
for(int i = 0; i < 100; i++) {
    //可以加入同步代码
    incA();
    //可以加入同步代码
}
```

`incA` 方法的实现必须为:

```
A++;
EmuExe.execute();
```

可以在 `run` 方法的 `incA()` 调用前后加入自己的同步代码。

5. 线程 2 的循环，调用 `decB()` 方法，该方法每次都自增一个 **ThreadB** 的成员整形变量（请定义为 `public static volatile` 类型，并命名为 **B**）的值，变量 **B** 的初始值为 100，每次循环皆自减 1，也即 100 次循环后，**B** 的值为 0。每次循环，用调用 `EmuExe.execute()` 模拟执行的时间，也即 `run` 方法的实现类似于

```
for(int i = 0; i < 100; i++) {
    //可以加入同步代码
    decB();
    //可以加入同步代码
}
```

`decB` 方法的实现必须为:

```
B--;
EmuExe.execute();
```

可以在 `run` 方法的 `decB()` 调用前后加入自己的同步代码

6. 要求 1: 线程 1 的 `for` 循环的第 *i* 次迭代循环的 `incA` 方法与线程 2 的第 *i* 次迭代循环的 `decB` 方法，必须都在线程 1 的 `for` 循环的第 *i-1* 次迭代循环与线程 2 的第 *i-1* 次迭代循环之后执行（for each *i* = 2, 3, 4, ..., 100）。
7. 要求 2: 线程 1 和线程 2 的 `for` 循环第 *i* 次循环的 `incA()` 方法与 `decB()` 方法，

不能强行同步，必须保证这两个方法可以并行执行。也即线程 1for 循环的第 i 次迭代循环的 incA() 执行期间(interval)和线程 2for 循环的第 i 次迭代循环的 decB() 执行期间可以有重合(overlapping)。[提示：仅在标注可以加入同步代码的位置加同步相关代码，不要在 incA 和 decB 中加同步代码。]

8. 要求 3: (在满足上述两个要求前提下)，在上述循环刚刚开始的时候，可判断 A+B 的值，也即

```
for(int i = 0; i < 100; i++) {  
    //同步代码，等待另外一个线程的 i-1 次循环结束  
    if(ThreadA.A + ThreadB.B != 100) {  
        System.out.println("Error!");  
    } //此时的两个线程的 A 和 B 的值，必须等于上一次循环结束时 A  
    //和 B 的值，也即他们的和必须是 100  
    //同步代码，允许另外一个线程调用 decB() 或者 incA()  
    decB(); //对于 ThreadB，或者 incA() 对于 ThreadA  
}
```

9. 注意，TA 在评阅代码的时候会用 TA 自己实现的 EmuExe.execute() 替代你实现的该方法，其等待时间会是 TA 设定的其他的值，如使线程 1 等待 1ms，线程 2 等待 1000ms，也可能在 EmuExe.execute() 检查 A 和 B 的值以检查 incA() 和 decB() 是否可以同时运行。你的程序必须在各种设定下，都满足上述同步要求。(上述 1-9 占 80% 分数)
10. 实现 ThreadSyncTen 类，实现 10 个线程的上述同步机制（五个线程 ABCDE 分别调用 incA,..., incE，变量 A-E 的初始值为 0，五个线程 FGHIJ 分别调用 decF,...,decJ，变量 F-J 的初始值为 100，每个循环开始的时候，A+B+...+J=500）。(占 20% 分数)

测试代码：

类 EmuExe:

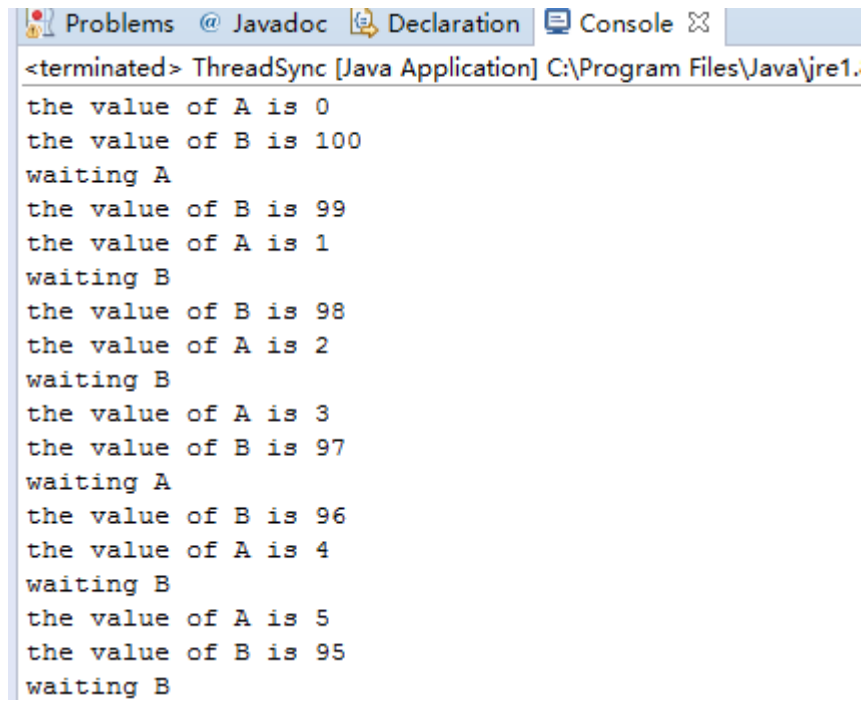
```
public class EmuExe {  
    static void execute() {  
        try {  
            Thread.sleep((int) (Math.random()*1000));  
        } catch (Exception e) {}  
    }  
}
```

测试类 ThreadSync:

```
public class ThreadSync {  
    public static void main(String [] args) {  
        ThreadA thread1 = new ThreadA();  
        ThreadB thread2 = new ThreadB();  
        thread1.start();  
        thread2.start();  
    }  
}
```

```
}  
}
```

运行输出结果（1-9 两个线程的 case）可参照下图：



```
Problems @ Javadoc Declaration Console  
<terminated> ThreadSync [Java Application] C:\Program Files\Java\jre1.  
the value of A is 0  
the value of B is 100  
waiting A  
the value of B is 99  
the value of A is 1  
waiting B  
the value of B is 98  
the value of A is 2  
waiting B  
the value of A is 3  
the value of B is 97  
waiting A  
the value of B is 96  
the value of A is 4  
waiting B  
the value of A is 5  
the value of B is 95  
waiting B
```

11. 提交姓名.zip，其中包含：

- a) ThreadSync.java
- b) ThreadSyncTen.java