

多线程作业

题目

通过继承Thread或者实现Runnable的方式完成车票销售功能。

假设有N张车票需要销售，通过M个窗口（每个窗口代表一个线程）销售，保证车票销售的正常：即车票顺利卖完；不超卖；不卖重复票。

示例

如下示例代码：

示例代码仅供参考，不保证正确性。

继承Thread方式

```
public class SaleTickets1 extends Thread {
    private int ticket = 15; // 15张代售票

    @Override
    public void run() {
        while (true) {
            if (ticket > 0) { // 还有票，继续销售
                sale();
            } else {
                break;
            }
        }
    }

    public static void main(String[] args) {
        // 开放窗口的数量
        int windowsNumber = 5;

        // 创建窗口线程
        SaleTickets1[] windows = new SaleTickets1[windowsNumber];
        for (int i = 0; i < windowsNumber; i++) {
            windows[i] = new SaleTickets1();
            windows[i].setName("窗口-" + i);
        }

        // 启动线程
        for (int i = 0; i < windowsNumber; i++) {
```

```

        windows[i].start();
    }
}

public void sale() {
    try {
        // 模拟出票，会有一定延时
        Thread.sleep((long) (Math.random() * 100));
    } catch (Exception e) {
        e.printStackTrace();
    }
    System.out.println(Thread.currentThread().getName() + "卖出票
号" + ticket--);
}
}

```

实现Runnable

```

public class SaleTickets2 implements Runnable {
    private int ticket = 15; // 15张代售票

    @Override
    public void run() {
        while (true) {
            if (ticket > 0) { // 还有票，继续销售
                sale();
            } else {
                break;
            }
        }
    }

    public static void main(String[] args) {
        // 开放窗口的数量
        int windowsNumber = 5;

        // 创建窗口线程
        SaleTickets2 sale = new SaleTickets2();
        Thread[] windows = new Thread[windowsNumber];
        for (int i = 0; i < windowsNumber; i++) {
            windows[i] = new Thread(sale);
            windows[i].setName("窗口-" + i);
        }

        // 启动线程
        for (int i = 0; i < windowsNumber; i++) {
            windows[i].start();
        }
    }

    public void sale() {
        try {
            // 模拟出票，会有一定延时

```

```
        Thread.sleep((long) (Math.random() * 100));
    } catch (Exception e) {
        e.printStackTrace();
    }
    System.out.println(Thread.currentThread().getName() + "卖出票
号" + ticket--);
}
}
```

要求

- 运用尽可能多的方式实现此功能。
- 要求完成作业的过程中注意保留所有可运行代码（包括正确的、不正确的），并完成记录，如：

正确实现方式1:

xxxxxxx

正确实现方式2:

xxxxxxx

.....

正确实现方式N:

xxxxxxx

问题代码1:

xxxxx

运行结果:

xxxxxxxxx

问题原因:

xx

.....

问题代码N:

xxxxx

运行结果:

xxxxxxx

问题原因：

xx