

```

package com.sxt.syn;

/**
 * 线程安全：在并发时保证数据的正确性，效率尽可能高
 * synchronized
 * 1. 同步方法
 * 2. 同步块
 * @author DELL
 *
 */

public class SynBlockTest8 {
    public static void main(String[] args) {
        SynWeb12306 web=new SynWeb12306();
        new Thread(web,"码畜").start();
        new Thread(web,"码农").start();
        new Thread(web,"码蝗").start();
    }
}

class SynWeb12306 implements Runnable{
    //票数
    private int ticketNums=10;
    private boolean flag=true;
    @Override
    public void run() {
        try {
            Thread.sleep(100);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        while(flag) {
            test1();
        }
    }

    //线程不安全 ticketNums对象在变

```

```

public void test1() {
    synchronized((Integer)ticketNums) {
        if(ticketNums<0) {
            flag=false;
            return;
        }
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.err.println(Thread.currentThread().getName()+"--
>" +ticketNums--);
    }
}

```

//同步块，线程安全，但范围太大-->效率低下

```

public void test2() {
    synchronized(this) {
        if(ticketNums<=0) {
            flag=false;
            return;
        }
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.err.println(Thread.currentThread().getName()+"--
>" +ticketNums--);
    }
}

```

//同步方法

```

public synchronized void test3() {

```

```

        if(ticketNums<=0) {
            flag=false;
            return;
        }
        try {
            Thread.sleep(200);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.err.println(Thread.currentThread().getName()+"--
>" +ticketNums--);
    }

```

//同步块，线程不安全，范围太小锁不住

```

public void test4() {
    synchronized(this) {
        if(ticketNums<=0) {
            flag=false;
            return;
        }
    }

    try {
        Thread.sleep(200);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    System.err.println(Thread.currentThread().getName()+"--
>" +ticketNums--);
}

```

//线程安全:尽可能锁定合理的范围(不是指代码，指数据的完整性)

//double checking

```

public void test5() {

```

```

        if(ticketNums<=0) { //考虑的是没有票的情况
            flag=false;
            return;
        }
        synchronized(this) {
            if(ticketNums<=0) { //考虑最后的1张票
                flag=false;
                return;
            }
            try {
                Thread.sleep(200);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.err.println(Thread.currentThread().getName()+"--
>" +ticketNums--);
        }
    }
}

```