

## 实验七 多线程并发实验

[返回](#)

姓名：李全欣 班级：计算机2018级4-7 成绩：93分

## 一.简答题 (共5题,100.0分)

1

一、实验目的：

1. 掌握使用Thread子类和Runnable接口创建多线程的方法。
2. 掌握线程的执行过程。

二、实验环境

(5.0分)

正确答案：

我的答案：

操作系统： windows 10 64位

JDK版本 jdk-11.0.6

编译器： eclipse-inst-win64

2

- 1（题目编号7179）、利用多线程技术编写一个模拟龟兔赛跑的程序，要求如下：（1）乌龟每次跑一个单位，兔子每次跑10个单位；（2）每个线程运行时，判断是否达到终点，如果到达终点，给出提示信息，未到终点则提示目前距离终点的距离，并判断是否领先；（3）如果兔子领先，则显示“我跑得快，睡一觉”信息，并睡一会。

(26.0分)

正确答案：

我的答案：

```
class T1 implements Runnable{
    final String RABBIT="兔子";
    final String TORTOISE="乌龟";
    volatile int rabbit;//兔子距离
    volatile int tortoise;//乌龟距离
    volatile String winner;
    @Override
    public void run() {
        // TODO Auto-generated method stub
        for(int i=0;i<=100;i++) {
            if(Thread.currentThread().getName().equals(RABBIT)&&winner==null) {
                if(10*i>=100) {
                    winner=RABBIT;
                    System.out.println(RABBIT+"到达终点--->" +winner+"is the winner!");
                    break;
                }
                rabbit=100-10*i;
                System.out.println(RABBIT+"距离终点"+rabbit);
                if(rabbit<tortoise) {
                    System.out.println("我跑得快,睡一觉");
                    try {
                        Thread.sleep(1);
                    } catch (InterruptedException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }
            }
        }
    }
}
```

```

    }
    }
    }
    if(Thread.currentThread().getName().equals(TORTOISE)&&winner==null) {
    if(i>=100) {
    winner=TORTOISE;
    System.out.println(TORTOISE+"到达终点--->" +winner+"is the winner!");
    break;
    }
    tortoise=100-i;
    System.out.println(TORTOISE+"距离终点"+tortoise);
    }
    }
    }
    }
    public class Test1 {

    public static void main(String[] args) {
    // TODO Auto-generated method stub
    T1 t1=new T1();
    Thread thread=new Thread(t1,"兔子");
    Thread thread1=new Thread(t1,"乌龟");
    thread.start();
    thread1.start();
    }
    }

```

<terminated> Test1 (1) [Java Applicat

```

乌龟距离终点100
兔子距离终点100
乌龟距离终点99
兔子距离终点90
乌龟距离终点98
我跑得快,睡一觉
乌龟距离终点97
乌龟距离终点96
乌龟距离终点95
乌龟距离终点94
乌龟距离终点93
乌龟距离终点9
乌龟距离终点8
乌龟距离终点7
乌龟距离终点6
乌龟距离终点5
乌龟距离终点4
乌龟距离终点3
乌龟距离终点2
乌龟距离终点1
乌龟到达终点--->乌龟is the winner!

```

- 3 2 (题目编号8690)、编写多线程应用程序,模拟多人过独木桥的模拟。独木桥每次只能通过一个人,每个人通过木桥的时间为5秒,随机生成10个人,同时准备过此独木桥,显示一下每次通过独木桥人的姓名。需要用到随机数。
- 注意: (1) 再开始过桥时输出: \*\*开始过桥! 过完桥后输出: \*\*已过桥!
- (2) 随机选人的时候,每个人都要选到,不能重复选。
- (27.0分)

正确答案:

我的答案:

```

package experinment07;

import java.util.Random;
class T2 implements Runnable{
    static Object object=new Object();
    @Override
    public void run() {

```

```
// TODO Auto-generated method stub
synchronized(object) {
System.out.println(Thread.currentThread().getName()+"开始过桥!");
try {
Thread.sleep(5000);
} catch (InterruptedException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}
System.out.println(Thread.currentThread().getName()+"已过桥!");
}
}

}

public class Test2 {

    public static void main(String[] args) {
// TODO Auto-generated method stub
T2 t=new T2();
int [] name=new int[11];
int pName=-1;
int u=0;
Random random=new Random();
L:while(u<=10) {
pName=random.nextInt(11);
for(int j=0;j<=u;j++) {
if(name[j]==pName)
continue L;
}
name[u++]=pName;
new Thread(t,Integer.toString(pName)).start();
}
}

}

}
```

Test2 (1) [Java Application] C:\Progra  
4开始过桥！

批语

? 请参看视频讲解

- 4 3 (题目编号7180)、哈尔滨火车站下面有三个火车票代售点：哈站、哈东站、哈西站，假如哈尔滨到北京的火车票总共是200张，如何用程序来实现三个售票点同时卖票的功能。注意：考虑线程同步问题，避免出现重复卖票问题。需要考虑同步问题。
- (27.0分)

正确答案：

我的答案：

```
public class T3 implements Runnable{
    public int tickets=200;
    public int i;
    public static void main(String[] args) {
// TODO Auto-generated method stub
T3 t=new T3();
new Thread(t,"哈站").start();
new Thread(t,"哈西站").start();
new Thread(t,"哈东站").start();
}

}
```

```
@Override
public void run() {
// TODO Auto-generated method stub
for(i=1;i<=200;i++) {
synchronized(this) {
```

```

if(tickets>=1) {
    System.out.print(Thread.currentThread().getName());
    System.out.println("售出第"+i+"张票"+"还剩"+(200-i)+"张票");
}
else
break;
try {
    Thread.sleep(1);
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
tickets--;
}
}
}
}
}

```

<terminated> T3 [Java Application] C:\Pr

哈站售出第119张票还剩81张票  
 哈站售出第120张票还剩80张票  
 哈站售出第121张票还剩79张票  
 哈站售出第122张票还剩78张票  
 哈站售出第123张票还剩77张票  
 哈站售出第124张票还剩76张票  
 哈站售出第125张票还剩75张票  
 哈站售出第126张票还剩74张票  
 哈站售出第127张票还剩73张票

哈东站售出第149张票还剩51张票  
 哈东站售出第150张票还剩50张票  
 哈东站售出第151张票还剩49张票  
 哈东站售出第152张票还剩48张票  
 哈东站售出第153张票还剩47张票  
 哈东站售出第154张票还剩46张票  
 哈东站售出第155张票还剩45张票  
 哈西站售出第156张票还剩44张票

Problems JavaDoc Declaration Search Console

<terminated> T3 [Java Application] C:\Program Files\Java\jdk-11.0.6\bin\javaw.exe (2020年

哈西站售出第191张票还剩9张票  
 哈西站售出第192张票还剩8张票  
 哈西站售出第193张票还剩7张票  
 哈西站售出第194张票还剩6张票  
 哈西站售出第195张票还剩5张票  
 哈西站售出第196张票还剩4张票  
 哈西站售出第197张票还剩3张票  
 哈西站售出第198张票还剩2张票  
 哈西站售出第199张票还剩1张票  
 哈西站售出第200张票还剩0张票

## 5

### 四、实验过程分析与讨论

(15.0分)

正确答案:

我的答案:

```

System.out.println(RABBIT+"距离终点"+rabbit
if(rabbit<tortoise) {
    System.out.println("我跑得快,睡一觉");
    try {
        Thread.sleep(1);
    }
}

```

比较剩余距离用小于

synchronized 关键字，代表这个方法加锁，相当于不管哪一个线程(例如线程A)，运行到这个方法时，都要检查有没有其它线程B(或者C、D等)正在用这个方法(或者该类的其他同步方法)，有的话要等正在使用synchronized方法的线程B(或者C、D)运行完这个方法后再运行此线程A，没有的话，锁定调用者，然后直接运行。

