Java 程序设计 LAB10

```
/**
* 学号:19373073
* 姓名:何潇龙
* JDK版本:jdk1.8.0_91
* 代码文件编码方式:UTF-8
* IDE:IDEA
*/
```

1. 简述程序，进程，线程的概念 简答

> 程序（Program）：程序的为了让计算机执行某些操作或解决某个问题而编写一系列有序的指令集合。
>
> 进程（Process）：进程是计算机中的程序关于某数据集合上的一次运行活动，是系统进行资源分配和调度的基本单位，是操作系统结构的基础。
>
> 线程（Thread）：是操作系统进程中能够独立执行的实体（控制流），是处理器调度和分派的基本单位。 可以看成是进程内的多条执行路径。

2. 产生死锁的四个条件是什么? 简答

> 互斥条件：指进程对所分配到的资源进行排它性使用，即在一段时间内某资源只由一个进程占用。如果此时还有其它进程请求资源，则请求者只能等待，直至占有资源的进程用毕释放。
>
> 请求与保持条件：进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源 已被其他进程占有，此时请求进程被阻塞，但对自己已获得的资源保持不放。
>
> 不可剥夺条件：进程所获得的资源在未使用完毕之前，不能被其他进程强行夺走，即只能 由获得该资源的进程自己来释放（只能是主动释放)。
>
> 循环等待条件：指在发生死锁时，必然存在一个进程——资源的环形链，即进程集合{P0，P1，P2，…，Pn}中的P0正在等待一个P1占用的资源；P1正在等待P2占用的资源，……，Pn正在等待已被P0占用的资源。

3. 创建线程的两种方式分别是什么？各有什么优缺点 简答

> 第一种方式：使用Runnable接口创建线程
>
> 第二种方式：直接继承Thread类创建对象
>
> 使用Runnable接口创建线程
>
> 1.可以将CPU，代码和数据分开，形成清晰的模型
>
> 2.线程体run()方法所在的类可以从其它类中继承一些有用的属性和方法
>
> 3.有利于保持程序的设计风格一致

> 直接继承Thread类创建对象
>
> 1.Thread子类无法再从其它类继承（java语言单继承）。
>
> 2.编写简单，run()方法的当前对象就是线程对象，可直接操作。

4. 判断题

(1) 进程是线程 `Thread` 内部的一个执行单元，它是程序中一个单一顺序控制流程。

> 错误

(2) 一个进程可以包括多个线程。两者的一个主要区别是：线程是资源分配的单位，而进程是CPU调度和执行的单位。

> 错误

(3) 线程可以用 `yield` 使低优先级的线程运行。

> 错误

(4) 当一个线程进入一个对象的一个 `synchronized` 方法后，其它线程可以再进入该对象的其它同步方法执行。

> 错误

(5) `notify` 是唤醒所在对象 `wait pool` 中的第一个线程。

> 错误

5. 程序输出简答

```
public class Main {
    public static void main(String[] args) {
        SyncThread syncThread = new SyncThread();
        Thread thread1 = new Thread(syncThread, "SyncThread1");
        Thread thread2 = new Thread(syncThread, "SyncThread2");
        thread1.start();
        thread2.start();
    }
}
class SyncThread implements Runnable {
    private static int count;
    public SyncThread() {
        count = 0;
    }
    public synchronized void run() {
        for (int i = 0; i < 5; i++) {
            try {
```

```
                System.out.println(Thread.currentThread().getName() + ":" +
(count++));
                Thread.sleep(100);//【1】
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
    public int getCount() {
        return count;
    }
}
```

(1) 请写出上述程序的输出

> SyncThread1:0
> SyncThread1:1
> SyncThread1:2
> SyncThread1:3
> SyncThread1:4
> SyncThread2:5
> SyncThread2:6
> SyncThread2:7
> SyncThread2:8
> SyncThread2:9

(2) 用 `synchronized` 修饰 `run()` 的作用是什么?

> 多线程访问时，采用了加锁机制，当一个线程访问该类的某个数据时，进行保护，其他线程不能进行访问直到该线程读取完，其他线程才可使用。当两个并发线程访问同一个对象中的synchronized代码块时，在同一时刻只能有一个线程得到执行，另一个线程受阻塞，必须等待当前线程执行完这个代码块以后才能执行该代码块。两个线程是互斥的，因为在执行synchronized代码块时会锁定当前的对象，只有执行完该代码块才能释放该对象锁，下一个线程才能执行并锁定该对象。

(3) 标号【1】处 `sleep` 的作用是什么？如果改为 `wait(100);` 输出会发生改变吗，为什么?

> 使线程隔100毫秒再执行。会。sleep 是线程类（Thread）的方法，导致此线程暂停执行指定时间，给执行机会给其他线程，但是监控状态依然保持，到时后会自动恢复。调用sleep 不会释放对象锁。
> wait 是Object 类的方法，对此对象调用wait 方法导致本线程放弃对象锁，进入等待此对象的等待锁定池，只有针对此对象发出notify 方法（或notifyAll）后本线程才进入对象锁定池准备获得对象锁进入运行状态。sleep方法没有释放锁，而wait方法释放了锁，使得其他线程可以使用同步控制块或者方法。

6. 程序补全题

```
public class Ticket {
    public static void main(String[] args) {
        // 创建线程类对象
        TicketSalSystem st = new TicketSalSystem();
```

```
        // 启动6次线程
        for (char i = 'A'; i <= 'F'; i++)
        {
            /*
             Thread类的构造方法如下：
             public Thread(Runnable run,String name)
             在创建线程的同时，创建线程名称
             */
            new Thread(【1】, "售票口" + i).start();
        }
    }
}
class TicketSalSystem implements Runnable
{
    // 定义变量---票数/票号
    public int ticket = 100;
    public int count = 0;
    // 重写run()方法
    @Override
    public void run()
    {
        // 定义while循环， 循环售票
        while (【2】)
        {
            // 根据题的要求，实现同步，此时定义同步代码块
            synchronized (【3】)// 传入对象
            {
                // 判断是否还有票，如果大于零说明还有票可卖
                if (ticket > 0)
                {
                    try
                    {
                        【4】// 线程休眠0.5秒
                    } catch (InterruptedException e)
                    {
                        e.printStackTrace();
                    }
                    count++; // 票号++
                    ticket--;// 循环售票，卖一张少一张
                    // 输出当前的售票窗口和票号
                    System.out.println(Thread.currentThread().getName()
                            + "\t当前票号：" + count);
                }
            }
        }
    }
}
```

(1) 补全标号处的代码

> 补全代码，并将完整代码文件放在文件夹oo/06下

(2) 简述上述程序的功能

> A到F的六个窗口同时售卖100张票，票的票数上被加了同步锁，意味着这100张票是被这六个窗口共同售卖的，大家共享一样的票数。

7. 程序补全题

```
public class ThreadPrint {
    public static void main(String[] args) throws InterruptedException{
        Object a=new Object();
        Object b=new Object();
        Object c=new Object();
        Thread8 threadA=new Thread8("A",c,a);
        Thread8 threadB=new Thread8("B",a,b);
        Thread8 threadC=new Thread8("C",b,c);
        new Thread(threadA).start();
        Thread.sleep(100);
        new Thread(threadB).start();
        Thread.sleep(100);
        new Thread(threadC).start();
        Thread.sleep(100);
    }
}
class Thread8 implements Runnable{
    private String name;
    private Object prev;
    private Object self;
    public Thread8(String name,Object prev,Object self){
        this.name=name;
        this.prev=prev;
        this.self=self;
    }
    @Override
    public void run(){
        int count=10;
        while(count>0){
            synchronized (prev){
                synchronized (self){
                    System.out.print(name);
                    count--;
                    【1】
                }
                try{
                    if(count==0)
                        【2】
                    else
                        【3】
                }catch (InterruptedException e){
                    e.printStackTrace();
                }
            }
        }
    }
}
```

(1) 补全标号处的代码

> 补全代码，并将完整代码文件放在文件夹oo/07下

(2) 详细说明上述程序的功能

> 交替输出ABC，输出10遍

(3) 主函数 `main` 中的 `Thread.sleep(100)` 语句不能省略，请简述原因。

> 如果省略的话可能ABC三个的执行顺序就不确定，三个线程一起给对象上锁，可能会出现死锁

(4) 主函数 `main` 中的 `Thread.sleep(100)` 语句全部去掉后程序可能出现死锁吗？试举例说明。

> 可能。比如B先执行，他给a上了锁，接下来A执行，因为a上了锁，所以A无法notify a，而是给c上锁，接下来C也无法notify c，给b上锁。a,b,c,全部被锁上了，三个进程只能全部等待。

8. 创建两个线程，其中一个输出1-52，另外一个输出A-Z。输出格式要求： 编程

```
12A 34B 56C 78D 91OE 1112F 1314G 1516H 1718I 192OJ 2122K 2324L 2526M 2728N 293OO
3132P 3334Q 3536R 3738S 394OT 4142U 4344V 4546W 4748X 495OY 5152Z
```

注意：

> (1) 可以参考T7
> (2) 12A 34B..... 看清楚A和3之间是有一个空格的，且输出以Z结尾，即结尾不能有空格
> (3) 请确保你的程序能够正常结束，不要在输出Z之后一直阻塞下去
> (4) 期末考试可能有类似的题目

> 补全代码，并将完整代码文件放在文件夹oo/08下

9. 试使用Object原生的wait()和notify()对生产者消费者问题(一个生产者，一个消费者，buffer容量有限)进行同步。 编程

注意：

> 开放性问题，可参阅各种资料

> 补全代码，并将完整代码文件放在文件夹oo/09下