**Java程序设计LAB10**

**实验目标：**

1. 理解程序、进程和多线程概念及特点
2. 理解线程的状态、生命周期及调度策略
3. 理解并掌握线程类(Thread)和Runnable接口，并能够进行相关应用的程序设计，实现多线程编程

**实验题目：**

1.【简答题】简述程序，进程，线程的概念

2.【简答题】产生死锁的四个条件是什么？

3.【简单题】创建线程的两种方式分别是什么？各有什么优缺点。

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。

(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。

(3)线程可以用yield使低优先级的线程运行。

(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。

(5) notify是唤醒所在对象wait pool中的第一个线程。

5.【程序输出题】

|  |
| --- |
| **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

(2)用synchronized修饰run()的作用是什么？

(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？

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

(2)简述上述程序的功能

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

(2)详细说明上述程序的功能**(三线程交替打印ABC)**

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

8.【编程题】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：

12A 34B 56C 78D 910E 1112F 1314G 1516H 1718I 1920J 2122K 2324L 2526M 2728N 2930O 3132P 3334Q 3536R 3738S 3940T 4142U 4344V 4546W 4748X 4950Y 5152Z

hint：

(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

9.【编程题】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。

注：开放性问题，可参阅各种资料