作业详情

Java语言实习6 线程(2023-)

题量: 2 满分: 100

作答时间: 2023-10-15 18:50 至 2023-10-21 00:50

一. 填空题

2



一. 填空题 (共2题)

1. (填空题)

利用线程类设计一个抢票小程序,输入当日余票数后,有4个窗口同时开始抢票,每抢到一张票就在屏幕输出抢到票的窗口编号,当票 数售完程序截止。效果如下图所示:

```
博输入今日票数:
开始售票!
窗口1拉到票了
窗口1抢到票了
窗口1扯到票了
窗口1拉到票了
第四1批到第7
窗口1拉到票了
留口3抢到票了
第口2批到票7
窗口4扯到票了
类首引用的类是____(1)_____类;假设程序名为ThreadShell,线程名为name,票数为整型的num,则定义num参数的代码为
  __(2)_____;为name和num赋值的ThreadShell类的构造方法为:
public ThreadSell(String name, int num) {
    ___(3)为name赋值_____;
    ___(4)为num赋值_____;}
截取run方法的代码图: ____(5) 注: 5-7空为结果截图_
现修改代码功能,假设用户输入学号,余票数为学号后2位,如余票≥35,则调整为35票后开始售票,main方法截图为__(6)_
结果截图为____(7)_
```

```
我的答案:
```

```
(1) Thread
```

(2) private int num

```
(3) this.name = name
(4) this.num = num
    public void run() {
        try {
            Random random = new Random();
            int randomnum = random.nextInt(100);
            Thread.sleep(randomnum);
        } catch (InterruptedException e) {
            e.printStackTrace();
        while (!isRunning) {
            synchronized (buyticket.class) {
                if (num > 0) {
                    System.out.println(name + "抢到票了");
                    num--;
                 } else {
                     isRunning = true;
                    buyticket.class.notifyAll();
                    break;s
            }
        }
```

作业详情

```
public static void main(string[] args) {
       Scanner scanner = new Scanner(System.in);
System.out.print("请输入学号: ");
       String studentId = scanner.next();
      int num = Integer.parseInt(studentId.substring(studentId.length() - 2));
int adjustedNum = Math.max(num, 35);
                                                                                                         一. 填空题
       // 创建并启动线程
                                                                                                                 2
       Thread t1 = new Thread(new buyticket(adjustedNum, "窗口1"));
t1.start();
       Thread t2 = new Thread(new buyticket(adjustedNum, "窗口2"));
       t2.start();
       Thread t3 = new Thread(new buyticket(adjustedNum, "窗口3"));
       t3.start();
       Thread t4 = new Thread(new buyticket(adjustedNum, "\boxtimes \square 4"));
       t4.start();
    ₩ COHSOIE ↔
   <terminated> BuyTicketWithSemaphor
   请输入学号: 10909202154
   窗口1抢到票了
   窗口4抢到票了
   窗口2抢到票了
   窗口2抢到票了
   窗口3抢到票了
   窗口3抢到票了
   窗口3抢到票了
   窗口3抢到票了
   窗口3抢到票了
   窗口3抢到票了
(7) 窗口3抢到票了
正确答案:
(1) Thread; Thread类
(2) static int num; static int num = 0;
(3) super(name):
(4) this.number = number;
  public void run() {
   while(number-->0)
         System.out.println(getName()+"拉到票7");
(5) }
(6)
```

作业详情

```
num = Integer.parseInt((no.substring(no.length()-2, no.length())));
   num = num >= 35?35:num;
   Thread1 t1 = new Thread1("\boxtimes \square 1", num);
                                                                                       一. 填空题
   Thread1 t2 = new Thread1("\boxtimes \square 2", num);
  Thread1 t3 = new Thread1("\overline{a} \square 3", num);
   Thread1 t4 = new Thread1(" <math>\boxtimes \square 4", num);
                                                                                              2
   t1.start();
   t2.start();
   t3.start();
   t4.start();
     请输入学号:
     2099887230
     窗口1抢到票了
     窗口3抢到票了
     窗口2抢到票了
  (7) 窗口4抢到票了
```

2. (填空题)

```
作业详情
  public void run() {
     for(;;) {
try {
}catch(InterruptedException e) {
            ;}
}
}
public static void main(String[] args) {
public static void main(String[] args) {
proper out = new Output();
OutPout i= 'N';i<='T';i++) {
for(int i= new CharThread((char)i, out);
t = new CharThread((char)i, out);
t .start();
       ) _ new Parall();
         *. Set priority(Thread .MAX_PRIORITY);
          t.setDaemon(true);
          t.start();
       }
   }
class Output {
int len=0;
      public synchronized void out(char c) {
         }
class CharThread extends Thread{
    char me;
Output out;
    public CharThread(char ch,Output o) {
       me = ch;
out = o;
   public void run() {
       for(
           out.out(me);
   }
本题考核多线程。将类名定义为姓名首字母_Thread,图片中有4处填空和1处需修改的代码,功能为休眠10毫秒,循环80次,每15次
循环换行。
根据代码顺序,第一处填空___(1)_____,第二处填空____(2)______,第三处为需要修改的代码____(3)_____,第四处填空
       _(4)_____,第五处填空__(5)___
将代码功能修改为从A至G的字符输出,回答以下问题:
该程序创建了_(6)__个字符的输出,按顺序写出所有字符(无需空格或引号)___(7)___,每个线程循环___(8)___次,结果有
__ (9) ____行,行数计算公式为__ (10) _____,Output对象借助synchronized关键字实现了___ (11)_
                                                                                            ____,线程启动进入死循
环,该线程在main()方法中被终止,原理为类线程被设置为_(12)____线程,当字符输出完毕,线程终止。
输出结果截图为___(13)_
   我的答案:
   (1) cxy_Thread
   (2) Thread.sleep(10);
   (3) t=new cxy_Thread()
   (4) (++len \% 15 == 0){
   (5) (int i = 0; i < 80; i++){
   (6) 560
```

一. 填空题

2

作业详情

AAAAAAAAAAAAA AAAAAAAAAAAA AAAAAAAAAAAA 一. 填空题 AAAAAAAAAAAA AAAAABBBBBBBBB 2 BBBBBBBBBBBBBBB BBBBBBBBBBBBBB BBBBBBBBBBBBBBB BBBBBBBBBBBBBB BBBBBBBBBBCCCCC CCCCCCCCCCCCC CCCCCCCCCCCCC CCCCCCCCCCCCC CCCCCCCCCCCCC CCCCCCCCCCCCC DDDDDDDDDDDDDD DDDDDDDDDDDDD DDDDDDDDDDDDDD DDDDDDDDDDDDD DDDDDDDDDDDDDD DDDDDEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEFFFFF FFFFFFFFFFFF FFFFFFFFFFF FFFFFFFFFFF FFFFFFFFFFFF FFFFFFFFFFFF GGGGGGGGGGGG GGGGGGGGGGGGG GGGGGGGGGGGG GGGGGGGGGGGGG GGGGGGGGGGGG GGGGG (8) 80 (9) 37 (10) + + len % 15 = = 0(11)1.一次只有一个线程可以进入out方法的代码块,其他线程必须等待当前线程退出该方法后才能进 入。这确保了线程之间不会互相干扰。 2.由于只有一个线程可以进入out方法,因此输出操作是按顺序执行的,不会出现混乱的输出。 3.使用synchronized确保了对共享数据(如 len变量)的访问是线程安全的,不会发生数据不一致的 情况。 (12) 守护线程

作业详情

AAAAAAAAAAAA αααααααααααα AAAAAAAAAAAA AAAAABBBBBBBBBB BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBBB BBBBBBBBBBBBBBB BBBBBBBBBBCCCCC cccccccccccc CCCCCCCCCCCCC cccccccccccc cccccccccccc cccccccccccc DDDDDDDDDDDDDD ממממממממממממ DDDDDDDDDDDDDD DDDDDDDDDDDDDD DDDDDDDDDDDDDD DDDDDEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEEEE EEEEEEEEEFFFFF FFFFFFFFFFFFF FFFFFFFFFFFFFFF FFFFFFFFFFFF FFFFFFFFFFFFFF FFFFFFFFFFFFF GGGGGGGGGGGG GGGGGGGGGGGG GGGGGGGGGGGG GGGGGGGGGGGG GGGGGGGGGGGG GGGGG (13) (1) 类首为首字母_Thread extends Thread; (3) t=new Parall()改为t=new 类名();

正确答案:

- (2) sleep(10);
- (4) if (++count%15==0)
- (5) for(int i=0; i<80; i++)
- (6) 7
- (7) ABCDEFG
- (8) 80
- (9) 38
- (10) INT(7*80/15)+1
- (11) 保证对System.out的顺序访问
- (12) 守护线程t.setDaemon(true)

一. 填空题



作业详情

