**西安电子科技大学**

**考试时间 120 分钟**

**试 题**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **题号** | **一** | **二** | **三** | **四** | **总分** |
| **分数** |  |  |  |  |  |

**1.考试形式：闭卷□ 开卷□ ；2.本试卷共三大题，满分100分；**

**3.考试日期： 年 月 日；(答题内容请写在装订线外)**

**一、简答题（每题5分，共20分）**

**1. 简述方法重载与方法重写的区别。**

**2. 简述如何保证向下转型（downcasting）的正确性。**

**3. 简述抽象类与接口的区别。（6分）**

**4. 画出线程生命周期图，并简述wait阻塞和sleep阻塞的区别。**

**二、读程题（每空2分，共40分）**

**给出下列Java程序代码片段的运行结果**

**1.**

|  |
| --- |
| String str1 = "first string";  String str2 = new String("second string");  String str3 = “third string”;  str3 = str2;  str2 = str1;  System.out.println(str1);  System.out.println(str2);  System.out.println(str3); |
| 程序输出结果： |

**2.**

|  |
| --- |
| class Employee{  public int id;  public static int serialNum = 1;  Employee(){ id=serialNum ++; }  }  public class EmployeeTest{  public static void main(String[] args){  Employee e1=new Employee();  Employee e2=new Employee();  System.out.println("e1.serialNum:"+e1.serialNum);  System.out.println("e1.id:"+e1.id);  System.out.println("e2.serialNum:"+e2.serialNum);  System.out.println("e2.id:"+e2.id);  }  } |
| 程序输出结果： |

**3.**

|  |
| --- |
| class Fu {  static int fustatic = 10;  int fus = 100;  static {  System.out.println("static block of Fu: " + Zi. zistatic);  }  {  System.out.println("constructor block of Fu: " +Fus);  }  Fu() { System.out.println("Fu(): " + fustatic); }  }  class Zi extends Fu {  static int zistatic = 20;  int zis = 50;  static {  System.out.println("static block of Zi: " + Zi. zistatic);  }  {  System.out.println("constructor block of Zi: " + zis);  }  Zi() { System.out.println("Zi(): " + zistatic); }  }  public class TestDemo {  public static void main(String[] args) {  new Zi();  }  } |
| 程序输出结果： |

**4.**

|  |
| --- |
| public class Test extends Base{  private int i=2;  public void printI(){  System.out.println(this.i);  System.out.println(this.getI());  }  public static void main(String[] args){  Base base=new Test();  base.increase();  base.printI();    if(base instanceof Test){  Test sub=(Test)base;  sub.increase();  sub.printI();  }  }  }  class Base{  private int i=100;  public void printI(){  System.out.println(this.i);  System.out.println("do nothing");  }  public final void increase(){  this.i++;  }  public int getI(){  return this.i;  }  } |
| 程序输出结果：（每空2分） |

**5. 假设下列程序中s1语句会抛出EType2异常**

|  |
| --- |
| try{  System.out.println("before Exception. ");  s1;  System.out.println("after Exception.");  } catch (Exception e){  System.err.println("In Exception catch process.");  } catch(EType2 e){  System.out.println("In Etype2 catch process.");  } finally{  System.out.println("In finally process.");  } |
| 程序输出结果： |

**三、改错题（共10分）**

**键盘录入多个数据，以0结束，要求在控制台输出这多个数据中的最大值.**

**下面Java程序中存在五处编译错误，指出错误位置，说明错误原因并更正。**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | import java.util.\*;  public class ArrayListDemo {  public static void main(String[] args) {  // 创建键盘录入数据对象  Scanner sc = new Scanner(System.out);  // 键盘录入多个数据,我们不知道多少个，所以用集合存储  ArrayList<Object> array = new ArrayList<int>();  // 以0结束。这个简单，只要键盘录入的数据是0，我就  // 跳出循环，不继续处理了  while (true) {  System.out.println("请输入数据：");  int number = sc.nextInt();  if (number != 0) {  array.add(number);  } else {  break;  }  }  // 把集合转成数组  Integer[] i = new int[array.length()];  array.toArray(i);  // 对数组排序  Arrays.sort(i);  // 获取该数组中的最大索引的值  System.out.println("数组最大值是:" + i[i.length()]);  }  } |

**四、编程题（共30分）**

**1. 按照以下要求编写程序。（10分）**

**需求：编程实现一个类，该类包含一个静态方法，该方法实现将源字符数组中的数据按指定位置及长度复制到目标数组中。**

**（1）该静态方法原型为：**

**public static void arrayCopy(char[] source, char[] dest,**

**int s\_index, int d\_index, int length)**

**// source 源字符数组引用**

**// dest 目标字符数组引用**

**// s\_index 源字符数组复制起始位置下标**

**// d\_index 目标字符数组复制起始位置下标**

**// length 复制长度**

**（2）定义自定义异常类NotEnoughSourceException，当源字符数组内容不够完成复制任务（s\_index大于等于源字符数组长度或s\_index+length大于源字符数组长度）时，抛出该异常。**

**（3）当目标字符数组不够容纳待复制的字符串（d\_index+length大于目标字符数组长度）时，仅复制目标数组能容纳下的字符串部分，多出的字符串忽略。**

**2. 按照以下要求编写程序。（20分）**

**需求：实现一个三个窗口同时销售火车票的程序设计，要求三个窗口累计销售300张结束运行，销售每张票的同时输出该票的售出窗口和票号，每张票不能被重复销售。**

**设计说明：**

**该程序设计一个火车票销售类、一个测试类。**

**1、火车票销售类**

**通过该类启动一个线程实现一个窗口销售火车票，并输出该窗口名称和票号**

**该类的构成**

**成员变量：表示票的数量**

**成员方法：销售一张车票数量减少一张，将销售的窗口名称和票号输出。**

**售票方法是通过线程方法（run）实现多线程功能的。 该方法需要加同步处理（同步方法或同步块）。**

**该类可通过实现Runnable接口来重写run()方法，也可通过继承Thread类重写run()方法。**

**2、测试类**

**main方法的内容：**

**启动三个线程分别表示三个售票窗口。**