1. 下列关于计算机系统和Java编程语言的说法，正确的是（C）
2. 计算机是由硬件、操作系统和软件组成，操作系统是缺一不可的组成部分。
3. Java语言编写的程序源代码可以不需要编译直接在硬件上运行。
4. 在程序中书写注释不会影响程序的执行，可以多写一些详细的注释。
5. Java的集成开发环境（IDE），如Eclipse，是开发Java语言必需的软件工具。
6. 下列选项中符合Java命名规则的标识符是（D）
7. 2japro B. &Class C. const D. \_123
8. 下列在Java语言中关于数据类型和包装类的说法，正确的是（B）
9. 基本（简单）数据类型是包装类的简写形式，可以用包装类替代基本（简单）数据类型。
10. long和double都占了64位（64bit）的存储空间。
11. 默认的整数数据类型是int，默认的浮点数据类型是float。
12. 和包装类一样，基本（简单）数据类型声明的变量中也具有静态方法，用来完成进制转化等。
13. 下面赋值语句中正确的是（A）
14. double d=5.3e12; B. float f=11.1;
15. int i=0.0; D. Double oD=3;
16. 如果int x=20, y=5，则语句System.out.println(x+y +""+(x+y)+y); 的输出结果是（D）
17. 2530 B. 55 C. 2052055 D. 25255
18. 如下Java语句 double x=2.0; int y=4; x/=++y; 执行后，x的值是（C）
19. 0.5 B. 2.5 C. 0.4 D. 2.0
20. 下列关于Java语言中String和char的说法，正确的是（C）
21. String是Java定义的一种基本数据类型。
22. String是以“\0”结尾的char类型的数组char[]。
23. 使用equals()方法比较两个String是否内容一样（即字符串中的各个字符都一样）。
24. Char类型在Java语言里面存储的是ASCII码。
25. 下列关于Java中类的构造方法的描述，正确的是（B）

A． 构造方法的返回类型为void

B. 可以定义一个类而在代码中不写构造方法。

C. 在同一个类中定义的重载构造方法不可以相互调用。

D. 子类不允许调用父类的构造方法。

1. 下列关于Java类中方法的定义，正确的是（D）
2. 若代码执行到return语句，则将当前值返回，而且继续执行return语句后面的语句。
3. 只需要对使用基本数据类型定义的属性使用getter和setter，体现类的封装性。
4. 方法的返回值只能是基本数据类型。
5. 在同一个类中定义的方法，允许方法名称相同而形参列表不同，并且返回值数据类型也不同。

10. 关于Java语言中的final关键字的使用，下列说法正确的是（C）

A．在使用final关键字定义的方法里面使用final关键字定义变量。

B．使用final关键字定义变量，必须在变量定义的同时给定变量的具体数值，完成变量初始化。

C．final关键字可以用来修饰方法，表明此方法不可以被子类重写。

D．使用final关键字定义变量，必须同时使用static关键字。

11. 以下关于Object类的说法正确的是（A）

A. Java中所有的类都直接或间接继承自Object，无论是否明确的指明，无论其是否是抽象类。

B. Java中的接口(interface)也继承了Object类

C. 利用“==”比较两个对象时，Java调用继承自Object的equals方法，判断是否相等。

D． 如果类的定义中没有重新定义toString()方法，则该类创建的对象无法使用toStrig()方法。

12. 下列代码的执行结果是（B）

|  |
| --- |
| public class Test {  public static int a = 1;  public static void main(String[] args) {  int a = 10;  a++; Test.a++;  Test t=new Test();  System.out.println("a=" + a + " t.a=" + t.a);  }  } |

1. a=10 t.a=3 B. a=11 t.a=2 C. a=12 t.a=1 D. a=11 t.a=1

13. 以下关于abstract关键字的说法，正确的是（D）

A． abstract 可以与final 并列修饰同一个类。

B． abstract 类中不可以有private的成员。

C． abstract 类中必须全部是abstract方法。

D． abstract方法必须在abstract 类中。

14. 下列关于继承的描述正确的是（C）

A． 在Java中允许定义一个子类的引用，指向父类的对象。

B． 在Java中一个子类可以继承多个抽象类，在extends关键字后依次列出，用逗号隔开。

C． 在Java中继承是通过extends关键字来描述的，而且只允许继承自一个直接父类。

D． 在Java中抽象类之间不允许出现继承关系，所有的抽象类都相互独立。

15. 下列关于包（package）的描述，正确的是（D）

1. 包（package）是Java中描述操作系统对多个源代码文件组织的一种方式。
2. import语句将所对应的Java源文件拷贝到此处执行。
3. 包（package）是Eclipse组织Java项目特有的一种方式。
4. 定义在同一个包（package）内的类可以不经过import而直接相互使用。

16. 关于下面代码 int[] x=new int[25]; 描述正确的是（C）

A. x[25]存放了数据“\0”。 B. x[24] 存放了数据“\0”。

C. 若访问x[25]，程序将抛出异常。 D. x[1]访问此数组的第一个元素。

17. 如下的Java程序

|  |
| --- |
| public class Test {  public static void main(String[] args) {  System.out.println(args[0]);  }  } |

若采用命令行“java Test one two three”调用，则程序输出的结果为：B

1. Test B. one C. two D. java

18. 对于文件的描述正确的是（D）

A. 文本文件是以“.txt”为后缀名的文件，其他后缀名的文件是二进制文件。

B. File类是Java中对文件进行读写操作的基本类。

C. 无论文本文件还是二进制文件，读到文件末尾都会抛出EOFException异常。

D. Java中对于文本文件和二进制文件，都可以当作二进制文件进行操作。

19. 对于Java中异常的描述正确的是（D）

A． 用throws定义了方法可能抛出的异常，那么调用此方法时一定会抛出此异常。

B． 如果try块中没有抛出异常，finally块中的语句将不会被执行。

C． 抛出异常意味着程序发生运行时错误，需要调试修改。

D． Java中所有的可不检测（unchecked）异常都来自RuntimeException类或其子类。

20. 对于如下代码段

|  |
| --- |
| class A{public A foo(){return this;}}  class B extends A{public A foo(){return this;}}  class C extends B  {    } |

可以放入到横线位置，使程序正确编译运行，而且不产生错误的选项是（C）

A. public void foo(){} B. public int foo(){return 1;}

C. public A foo(B b){return b;} D. public A foo(){return A;}

**二、读程序题（四选三，共计24分）**

1.

public class ReadOne {

public static void main(String[] args) {

int count = 0;

for (int index = 0; index <= 20; index++) {

switch (index++ % 4) {

case 0:

count = ++index;

case 1:

count += index;

break;

default:

System.out.println(count);

}

}

}

}

2.

public class ReadTwo {

public static final int N=4;

public static void main(String[] args) {

int[][] m = new int[N][N];

Calc(m, N);

Print(m);

}

public static void Calc(int[][] m, int num) {

int k = N;

for (int i = 0; i <= num / 2; i++) {

for (int j = i; j < num - i; j++)

m[i][j] = k++;

for (int j = i + 1; j < num - i; j++)

m[j][num - i - 1] = k++;

for (int j = num - i - 2; j >= i; j--)

m[num - i - 1][j] = k++;

for (int j = num - i - 2; j > i; j--)

m[j][i] = k++;

}

}

public static void Print(int[][] m) {

for (int i = 0; i < m.length; i++) {

for (int j = 0; j < m[i].length; j++)

System.out.print(m[i][j]+"\t");

System.out.println();

}

}

}

3. public class ReadThree {

static public void main(String[] args) {

ExceptionScope demo = new ExceptionScope();

demo.level1();

}

}

class ExceptionScope {

public void level1() {

System.out.println("Level 1 beginning.");

try {

level2();

} catch (Exception problem) {

System.out.println("Exception Unknown");

}

System.out.println("Level 1 ending.");

}

public void level2() {

System.out.println("Level 2 beginning.");

String[] codes = { "A370", "A37A",""};

char zone;

int number;

for (int i = 0; i <= codes.length; i++) {

try {

zone = codes[i].charAt(0);

number = Integer.parseInt(codes[i].substring(1, 4));

System.out.println("zone: " + zone + "\tnumber: " + number);

} catch (StringIndexOutOfBoundsException exception) {

System.out.println("Improper code length ");

} catch (NumberFormatException exception) {

System.out.println("Improper numeric value");

} finally {

System.out.println("processing" + i);

}

}

System.out.println("Level 2 ending.");

}

}

4. public class ReadFour {

public static void main(String[] args) {

String s = "1234";

List(s.toCharArray(), s.length(), 0);

}

private static void swap(char[] a, int x, int y) {

char temp = a[x];

a[x] = a[y];

a[y] = temp;

}

public static void List(char[] array, int n, int k) {

if (n == k) {

if (array[2] != '4') {

String str = new String(array);

//indexOf(String str)方法返回包含str字串的第一次出现的位置，  
//如果返回值为负数，表示没有找到这样的匹配

if (str.indexOf("23") < 0 && str.indexOf("32") < 0)

System.out.println(str);

}

} else {

for (int i = k; i < n; i++) {

swap(array, k, i);

List(array, n, k + 1);

swap(array, i, k);

}

}

}

}

**三、写程序题（共计36分）。**

1. 数字625是一个奇特的整数，因为625的平方是390625，最后三位数字还是625。请书写一个程序， 找出所有**三位正整数**中具有这样特征的数。（本题6分）

public class WriteOne {

public static void main(String[] args) {

//请在此处书写合适的代码完成此程序

|  |
| --- |
|  |

}

}

1. 下列程序通过数组s []，完成对两个形状面积的计算，请在给定的位置书写合适的代码，完成此程序。

abstract class Shape{

private String name;

public Shape(String \_name) {

name = \_name;

}

public String getName() {

return name;

}

//请在此处完成对抽象类中相关方法的定义（本题2分）

|  |
| --- |
|  |

}

class Circle extends Shape {

private double rad;

Circle(double \_rad) {

//请在此书写代码，将继承自父类的name属性赋值为“Circle” （本题2分）

|  |
| --- |
|  |

rad = \_rad;

}

public double getArea() {

return rad \* rad \* Math.PI;

}

}

class Rectangle extends Shape {

private double width;

private double length;

Rectangle(double width, double length) {

//请在此完成构造方法，并且将继承自父类的name属性赋值为“Rectangle”（本题4分）

|  |
| --- |
|  |

}

//从方法的定义开始，请在此完整的完成求矩形面积的方法 （本题4分）

|  |
| --- |
|  |

}

public class ShapeDemo {

public static void main(String[] args) {

//请在此完成对数组s []的定义（本题2分）

|  |
| --- |
|  |

s [0] = new Rectangle(1.5, 2.5);

s [1] = new Circle(1.5);

for (int i = 0; i < 2; i++)

System.out.println(s [i].getName()+"'Area is+"Area is+" + s [i].getArea());

}

}

1. 根据链表节点ListNode的描述，完成链表类LinkedList中的若干方法。

class ListNode {

public int data;

public ListNode link;

public ListNode(int newData, ListNode linkValue) {

data = newData;

link = linkValue;

}

}

public class List {

private ListNode head;

// 以下举例中，假设已经有一个链表，按序为 5-->3-->2-->4-->1

public void addANodeToStart(int newData){

//在链表开始的位置插入data数值为newData的节点（本题3分）

//例如调用addANodeToStart(6)，将形成链表6-->5-->3-->2-->4-->1

|  |
| --- |
|  |

}

public int length() {

//返回整个链表中节点的个数，比如本例中调用length()，得到的返回值为5（本题3分）

|  |
| --- |
|  |

}

public void InsertBehind(int vdata, int newdata) {

// 寻找一个链表的节点中data数值等于vdata的节点，并在此节点后插入一个新的节点，其data的数值为newdata（本题5分）

//例如调用InsertBehind(3,10)，将形成链表5-->3-->10-->2-->4-->1

|  |
| --- |
|  |

}

public void RemoveLessThan(int max) {

// 删除链表的所有节点中data数值小于max的节点（本题5分）

// 例如调用RemoveLessThan(3)，将形成链表5-->3-->4

|  |
| --- |
|  |

## 2012年Java试题答案（2012-1-12）

1. 选择题

CDBAD CCBDC ABDCD CBDDC

1. 读程序题（4选3）
   1. 8分

4

10

20

34

52

* 1. 8分

4 5 6 7

15 16 17 8

14 19 18 9

13 12 11 10

（1）

8分

Level 1 beginning.

Level 2 beginning.

zone: A number: 370

processing0

Improper numeric value

processing1

Improper code length

processing2

processing3

Exception Unknown

Level 1 ending.

* 1. 8分

2134

2431

2413

3124

3412

3421

4213

4312

1. 写程序题（）
   1. 6分

|  |
| --- |
| public class WriteOne {  public static void main(String[] args) {  int i, mulRs, last3;  for (i = 100; i <= 999; i++) {  mulRs = (int) Math.pow(i, 2);  last3 = mulRs % 1000;  if (last3 == i)  System.out.println(i);  }  }  } |

* 1. 14分

//请在此处完成对抽象类中相关方法的定义

|  |
| --- |
| public abstract double getArea(); |

//请在此书写代码，将继承自父类的name属性赋值为“Circle”

|  |
| --- |
| super("Circle"); |

//请在此完成构造方法，并且将继承自父类的name属性赋值为“Rectangle”

|  |
| --- |
| super("Rectangle");  this.width = width;  this.length = length; |

//从方法的定义开始，请在此完整的完成求矩形面积的方法

|  |
| --- |
| public double getArea() {  return width \* length;  } |

//请在此完成对数组s []的定义

|  |
| --- |
| Shape[] s = new Shape[2]; |

* 1. 16分

//在链表开始的位置插入data数值为newData的节点（

|  |
| --- |
| head = new ListNode(addData, head); |

//返回整个链表中节点的个数

|  |
| --- |
| int count = 0;  ListNode position = head;  while (position != null) {  count++;  position = position.link;  }  return count; |

// 寻找一个链表的节点中data数值等于vdata的节点，并在此节点后插入一个新的节点，其data的数值为newdata

|  |
| --- |
| ListNode position = head;  int dataAtPosition;  while (position != null) {  dataAtPosition = position.data;  if (dataAtPosition== newdata) {  position.link=new ListNode(newData, position.link);  break;  }  position = position.link;  }  if(position == null)  System.out.println("No target is found, nothing insert."); |

// 删除链表的所有节点中data数值小于max的节点（

|  |
| --- |
| if (head != null) {  while ((head.data < max) && (head.link != null))  head = head.link;  ListNode position = head;  if ((position.link == null) && (head.data < max))  {  head = null;  return;  }  while (position.link != null) {  if (position.link.data < max)  position.link = position.link.link;  else  position = position.link;  }  } else  System.out.println("Empty List"); |