

《JAVA 语言程序设计》期末考试模拟试题

(适用对象： 06 级计算机类专业)2008 年 6 月

一、单项选择题 (每小题 2 分，共 10 分)

1、编译 Java Application 源程序文件将产生相应的字节码文件，这些字节码文件的扩展名为 (B)。

A. .java

B. .class

C. .html

D. .exe

2、设 $x = 1, y = 2, z = 3$ ，则表达式 $y + = z - - / + + x$ 的值是 (A)。

A. 3

B. 3.5

C. 4

D. 5

3、在 Java Applet 程序用户自定义的 Applet 子类中，一般需要重载父类的 (D)方法来完成一些画图操作。

A. start()

B. stop()

C. init()

D. paint()

4、不允许作为类及类成员的访问控制符的是 (C)。

A. public

B. private

C. static

D. protected

5、为 AB 类的一个无形式参数无返回值的方法 method 书写方法头，使得使用类名 AB 作为前缀就可以调用它，该方法头的形式为 (A)。

A. static void method()

B. public void method()

C. final void method()

D. abstract void method()

二、填空题 (每空格 1 分，共 20 分)

1、开发与运行 Java 程序需要经过的三个主要步骤为 编辑源程序、编译生成字节码和解释运行字节码。

2、如果一个 Java Applet 源程序文件只定义有一个类，该类的类名为 MyApplet，则类 MyApplet 必须是 Applet 类的子类并且存储该源程序文件的文件名为 MyApplet。

3、如果一个 Java Applet 程序文件中定义有 3 个类，则使用 Sun 公司的 JDK 编译器 javac.exe 编译该源程序文件将产生 3 个文件名与类名相同而扩展名为 .class 的字节码文件。

4、在 Java 的基本数据类型中，char 型采用 Unicode 编码方案，每个 Unicode 码占用 2 字节内存空间，这样，无论是中文字符还是英文字符，都是占用 2 字节内存空间。

5、设 $x = 2$ ，则表达式 $(x++) / 3$ 的值是 0。

6、若 $x = 5, y = 10$ ，则 $x < y$ 和 $x \geq y$ 的逻辑值分别为 true 和 false。

7、抽象 (abstract) 方法 方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。最终 (final) 方法 方法是不能被当前类的子类重新定义的方法。

8、创建一个名为 MyPackage 的包的语句是 package MyPackag，该语句应该放在程序的位置为：应该在程序第一句。

9、设有数组定义：int MyIntArray[] = { 10, 20, 30, 40, 50, 60, 70}; 则执行以下几个

语句后的输出结果是 120。

```
int s = 0;
for (int i = 0; i < MyIntArray.length; i++)
    if (i % 2 == 1) s += MyIntArray[i];
System.out.println(s);
```

10、在 Java 程序中，通过类的定义只能实现 单 重继承，但通过接口的定义可以实现 多 重继承关系。

三、写出下列程序完成的功能。（每小题 5 分，共 20 分）

```
1、 public class Sum
{ public static void main( String args[] )
{ double sum = 0.0;
  for (int i = 1; i <= 100; i++)
    sum += 1.0/(double) i;
  System.out.println( "sum="+sum );
}
} 答：计算 1/1+1/2+1/3+...+1/100 的值
```

```
2、 import java.io.*;
public class Reverse
{ public static void main(String args[] )
{ int i, n = 10;
  int a[] = new int[10];
  for (i = 0; i < n; i++)
  try {
    BufferedReader br = new BufferedReader(
      new InputStreamReader(System.in));
    a[i] = Integer.parseInt(br.readLine()); // 输入一个整数
  } catch (IOException e) {} ;
  for (i = n - 1; i >= 0; i--)
    System.out.print(a[i]+" ");
  System.out.println();
}
} 答：从标准输入（键盘）读入 10 个整数存入整型数组 a 中，然后逆序输出这 10 个整数
```

```
3、 import java.awt.*;
public class abc
{ public static void main(String args[])
{ new FrameOut();
}
class FrameOut extends Frame // Frame 为系统定
{ Button btn; // 义的窗框类
  FrameOut()
}
```

```

        {
            super(" 按钮 ");
            btn = new Button(" 按下我 ");
            setLayout(new FlowLayout( ));
            add(btn);
            setSize(300,200);
            show( );
        }

```

}答：创建一个标题为 "按钮 "的窗框，窗框中显示有 "按下我 "字样的按钮。

```

4、 import java.io.*;
public class abc
{
    public static void main(String args[])
    {
        SubClass sb = new SubClass( );
        System.out.println(sb.max( ));
    }
}

class SuperClass
{
    int a = 10 , b = 20 ;
}

class SubClass extends SuperClass
{
    int max( ){ return ((a>b)?a:b); }
}

```

}答：求两个数的最大值。

四、写出下面程序的运行结果 （每小题 10 分，共 30 分）

```

1、 import java.io.*;
public class abc
{
    public static void main(String args[ ])
    {
        AB s = new AB("Hello!", "I love JA VA.");
        System.out.println(s.toString( ));
    }
}

class AB {
    String s1;
    String s2;
    AB( String str1 , String str2 )
    {
        s1 = str1; s2 = str2;
    }
    public String toString( )
    {
        return s1+s2;
    }
}

```

}答：1、 Hello! I love JAVA.

```

2、 import java.io.* ;
public class abc
{
    public static void main(String args[ ])
    {
        int i , s = 0 ;
        int a[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 };
    }
}

```

```

        for ( i = 0 ; i < a.length ; i ++ )
            if ( a[i]%3 == 0 )    s += a[i] ;
        System.out.println("s="+s);
    }
}答： s = 180

```

```

3、 import    java.io.* ;
    public class abc
    {
        public static void    main(String args[ ])
        )
        {    System.out.println("a="+a+"\nb="+b);    }
    }
    class SubClass    extends    SuperClass
    {    int c;
        SubClass(int aa,int bb,int cc)
        {    super(aa,bb);
            c=cc;
        }
    }
    class SubSubClass    extends    SubClass
    {    int a;
        SubSubClass(int aa,int bb,int cc)
        {    super(aa,bb,cc);
            a=aa+bb+cc;
        }
        void show()
        {    System.out.println("a="+a+"\nb="+b+"\nc="+c);    }
    }
}答： a=60
      b=20
      c=30

```

五、使用 Java 语言编写程序。（每小题 10 分，共 20 分）

1、编写一个字符界面的 Java Application 程序，接受用户输入的 10 个整数，并输出这 10 个整数的最大值和最小值。

```

答： import    java.io.* ;
    public class abc
    {
        public static void    main(String args[ ])
        {    int i , n = 10 , max = 0 , min = 0 , temp = 0;
            try {
                BufferedReader br = new BufferedReader(
                    new InputStreamReader(System.in));
                max = min = Integer.parseInt(br.readLine( ));
            }
        }
    }

```

```

        } catch ( IOException    e ) { } ;
    for  ( i = 2 ; i <= n ; i ++ )    {
        try {
            BufferedReader    br = new BufferedReader(
                                new    InputStreamReader(System.in));
            temp = Integer.parseInt(br.readLine( ));
            if  (temp > max ) max=temp;
            if  (temp < min) min=temp;
        } catch ( IOException    e ) { } ;
    }
    System.out.println("max="+max+"\nmin="+min);
}
}
}

```

2、编写一个完整的 Java Applet 程序使用复数类 Complex 验证两个复数 1+2i 和 3+4i 相加产生一个新的复数 4+6i 。

复数类 Complex 必须满足如下要求：

(1) 复数类 Complex 的属性有：

RealPart： int 型，代表复数的实数部分

ImaginPart： int 型，代表复数的虚数部分

(2) 复数类 Complex 的方法有：

Complex()：构造函数，将复数的实部和虚部都置 0

Complex(int r , int i)：构造函数，形参 r 为实部的初值， i 为虚部的初值。

Complex complexAdd(Complex a)：将当前复数对象与形参复数对象相加，所得的结果仍是一个复数值，返回给此方法的调用者。

String ToString()：把当前复数对象的实部、虚部组合成 a+bi 的字符串形式，其中 a 和 b 分别为实部和虚部的数据。

答：import java.applet.*；

```

import java.awt.*；
public class abc extends Applet
{
    Complex a,b,c；
    public void init( )
    {
        a = new Complex(1,2);
        b = new Complex(3,4);
        c = new Complex();
    }

    public void paint(Graphics g)
    {
        c=a.complexAdd(b);
        g.drawString(" 第一个复数： "+a.toString(),10,50);
        g.drawString(" 第二个复数： "+b.toString(),10,70);
    }
}

```

```

        g.drawString(" 两复数之和：  "+c.toString(),10,90);
    }
}

class Complex
{
    int RealPart ;    // 复数的实部
    int ImaginPart ;  // 复数的虚部
    Complex() { RealPart = 0 ; ImaginPart = 0 ; }
    Complex(int r , int i)
    { RealPart = r ;    ImaginPart = i ;    }
    Complex complexAdd(Complex a)

    {
        Complex temp = new Complex( ) ; // 临时复数对象
        temp.RealPart=RealPart+a.RealPart;
        temp.ImaginPart=ImaginPart+a.ImaginPart;
        return temp;
    }
    public String toString( )
    { return ( RealPart+" + "+ImaginPart+" i ");    }
}

```

Java 线程 程序题

```

class sum implements Runnable {
    int sum = 0;
    int i;

    public void run () {
        for(i=1;i<=100;i++) {
            sum+=i;
        }
        System.out.println(" 从 1 加到 100 的结果为 "+sum);
    }
}

```

```

class sumpro {
    public static void main(String args[]) {
        sum sum1 = new sum();
        Thread t=new Thread(sum1);
        t.start();
    }
}

```

异常

```
1.import java.io.*;
```

```

class A{
    void m() throws RuntimeException{}
}
class B extends A{
    void m() throws IOException{}
}
2.import java.io.*;
class A{
    void m() throws RuntimeException{}
}
class B extends A{
    void m() throws IOException{}
}
3.public class e8{
    public static void main(String args[]){
        e8 t=new e8();
        t.first();
        System.out.println( " Hi");
    }
    public void first(){second();}
    public void second() throws Exception{
        int x[]=new int[2];
        x[3]=2;
    }
}
4.public class e10{
    public static void main(String args[]) throws Exception{
        e10 t=new e10();
        t.first();
        System.out.println( " Hi");
    }
    public void first() throws Exception{second();}
    public void second() throws Exception{
        int x[]=new int[2];
        x[3]=2;
    }
}
5 使用 super 调用父类方法
class Fish extends Animal{
    public Fish(){super(0);}
    public void eat(){
        System.out.println(" 鱼吃小鱼虫 ");
    }
    public void walk(){

```

```
        super.walk();
        System.out.println(" 鱼没有腿不会走路  ");
    }
}
```

6.接口类的实现

```
class Cat extends Animal implements Pet{
    String name;
    public Cat(String n){
        super(4);
        name=n;
    }
    public Cat(){this("");}
    public String getName(){return name;}
    public void setName(String n){name=n;}
    public void play(){
        System.out.println(" 猫玩耍 ");
    }
    public void eat(){
        System.out.println(" 猫吃猫粮 ");
    }
}
```