

一、单选题(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()。

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. **3** B. 3.5

$2 \pm = 3 / 2$

C. 4 D. 5

$= 3.$

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()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()。

A. public B. private

C. **static** D. protected

编辑源文件
编译生成字节码
解释运行

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. **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)

编辑源文件, 编译生成字节码,

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、解释运行字节码

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

Applet

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。

3

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。

true false

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,

抽象方法

abstract 方法

编辑源文件, 编译生成字节码,
解释运行字节码

最终方法.
final方法

该方法必须在抽象类之中定义。 是不能被当前类的子类重新定义的方法。

package MyPackage;

8、创建一个名为 MyPackage 的包的语句是，

该语句应该放在程序的位置为：

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 。

0 1 2 3 4 5 6

int s = 0;

120

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 重继承，但通过接口的定义可以实现 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、

```
package a01;
```

```
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 );
```

```
    }
```

```
}
```

计算 1/1,1/2,1/3.....到 1/100 的和

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();
}
}从键盘读入十个数字，存入整型数组 a 中，然后逆序输出这十个整数

```

3.

```

package a01;

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();
    }
}
}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 dd
{
    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);
    }
}

```

求两个数的最大值

写出下面程序的运行结果(每小题 10 分，共 30 分)

1.

```

4 import java.io.*;
5 public class b01
6 {
7     public static void main(String args[ ]){
8         AB s = new AB("Hello!", "I love JAVA.");
9         System.out.println(s.toString( ));
10    }
11 }
12
13 class AB {
14     String s1;
15     String s2;
16     AB( String str1 , String str2 )
17     { s1 = str1; s2 = str2; }
18     public String toString( ){
19         return s1+s2;
20     }
21 }

```

Hello! I love JAVA.

```

3 import java.io.* ;
4 public class b02
5 {
6     public static void main(String args[ ]){
7         int i , s = 0 ;
8         int a[ ] = { 10 , 20 , 30 , 40 , 50 , 60 , 70 , 80 , 90 };
9         for ( i = 0 ; i < a.length ; i ++ )
10             if ( a[i]%3 == 0 ) s += a[i] ;
11         System.out.println("s="+s);
12     }
13 }

```

s = 180

3、

```

3 import java.io.* ;
4 public class b03{
5     public static void main(String args[ ]){
6         System.out.println("a="+a+"\nb=" + b );
7     }
8 }
9 class SubClass extends SuperClass
10 {
11     int c;
12     SubClass(int aa,int bb,int cc)
13     {
14         super(aa,bb);
15         c=cc;
16     }
17 }
18
19 class SubSubClass extends SubClass
20 {
21     int a;
22     SubSubClass(int aa,int bb,int cc){
23         super(aa,bb,cc);
24         a=aa+bb+cc;
25     }
26     void show(){
27         System.out.println("a="+a+"\nb="+b+"\nc="+c);
28     }
29 }
30

```

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。

```
3 import java.io.* ;
4 public class d01
5 {
6     public static void main(String args[ ]){
7         int i , n = 10 , max = 0 , min = 0 , temp = 0;
8         try {
9             BufferedReader br = new BufferedReader(
10                 new InputStreamReader(System.in)
11             );
12             max = min = Integer.parseInt(br.readLine( ));
13         } catch ( IOException e ) { } ;
14         for ( i = 2 ; i <= n ; i ++ ) {
15             try {
16                 BufferedReader br = new BufferedReader(
17                     new InputStreamReader(System.in));
18                 temp = Integer.parseInt(br.readLine( ));
19                 if (temp > max ) max=temp;
20                 if (temp < min) min=temp;
21             } catch ( IOException e ) { } ;
22         }
23         System.out.println("max="+max+"\nmin="+min);
24     }
25 }
```

3、 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。

```

3 import java.applet.* ;
4 import java.awt.* ;
5 public class d02 extends Applet
6 {
7     Complex a,b,c ;
8     public void init() {
9         a = new Complex(1,2);
10        b = new Complex(3,4);
11        c = new Complex();
12    }
13    public void paint(Graphics g) {
14        c=a.complexAdd(b);
15        g.drawString("第一个复数: "+a.toString(),10,50);
16        g.drawString("第二个复数: "+b.toString(),10,70);
17        g.drawString("两复数之和: "+c.toString(),10,90);
18    }
19 }
20 class Complex
21 {
22     int RealPart ; // 复数的实部
23     int ImaginPart ; // 复数的虚部
24     Complex() { RealPart = 0 ; ImaginPart = 0 ; }
25     Complex(int r , int i){
26         RealPart = r ; ImaginPart = i ;
27     }
28     Complex complexAdd(Complex a){
29         Complex temp = new Complex() ; // 临时复数对象
30         temp.RealPart=RealPart+a.RealPart;
31         temp.ImaginPart=ImaginPart+a.ImaginPart;
32         return temp;
33     }
34     public String toString() {
35         return ( RealPart+" + "+ImaginPart+" i ");
36     }
37 }

```



