

一 判断题

- 1 一个 JAVA 类可以有多个父类 答案：错误
- 2 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 个字节内存空间 答案：正确
- 3 一个 JAVA 文件中最多只能有一个 public 类。 答案：正确
- 4 JAVA 语言具有异常处理、指针、多线程等特性。 答案：错误
- 5 NULL 是一个 JAVA 语言关键字。 答案：错误
- 6 一个抽象类中只能包含抽象方法。 答案：错误
- 7 JAVA 中一类只能继承一个类，实现一个接口。 答案：错误
- 8 JAVA 是一跨平台的语言。 答案：正确
- 9 JAVA 命名规范中，类名首字母要求大写。 答案：正确
- 10 线程是一个进程内可以并行执行的单位。 答案：正确

二、选择题:

1. Java 语言中，byte 类型在内存中所占的位数为 (A)
(A) 8 (B) 16 (C) 32 (D) 64
2. 类声明中，指明该类不能有子类的是 (c)
(A) public (B) abstract (C) final (D) class
3. AWT 中创建一个标识有 “确定”按钮的语句是 (c)
(A) TextField b = new TextField(“确定”);(B) Checkbox b = new Checkbox(“确定”);
(C) Buttn b = new Button(“确定”);(D) Label b = new Label(“确定”);
4. 将长整型(long)数据转换为较短的整数类型(int)数据，要进行 (b)
(A) 类型的自动转换 (B) 类型的强制转换 (C) 无需转换 (D) 无法实现
5. 下列命令中，是 Java 编译命令的是： (d)
(A) java (B) javadoc (C) jar (D) javac

三、简答题

1. 简述面向对象的三个主要特征。

答案：封装、继承、多态

2. 简述接口(interface)和抽象类(abstract)的区别。

答案：{

- 1、接口不能有方法的实现 2、接口不能有成员变量

- 3、接口可以多重继承 4、一个类只能继承一个抽象类，但可实现多个接口 5、抽象类可以有一些非抽象方法

3. 简述 Applet 的生命周期？

答案：{

init() 初始化

start() 初始化后或 Applet 被重新访问时

stop() 离开浏览器时

destroy() 关闭浏览器时

}

4. 简述 final、finally、finalize 的区别？

答案：{

final 表示常量，或最终类

finally 表示异常处理时，一定会执行的代码段

finalize 表示对象销毁前要执行的代码

}

四、分析题 下列程序中由输出的显示结果是什么？

<pre>public class Abc{ private static int i = 15; public static void main(String args[]){ int b = 5 + i; int i = 5 + b; System.out.println("i="+i); System.out.println("Abc.i="+Abc.i); } } }</pre>	运行结果： i=25 Abc.i=15
<pre>public class Test{ static int x = 5; public static void methodA(){ int x = 2; System.out.println("x in methodA() =" + (++x)); } public static void methodB(){ x *= 5; System.out.println("x in methodB() =" + x); } public static void main(String args[]) { int x = 5; methodA(); methodB(); System.out.println("x in main() =" + x); } }</pre>	运行结果： x in methodA() =3 x in methodB() =25 x in main() =5
<pre>class JavaTest { public static void main(String args[]) { int numbers[]=new int[5]; try{</pre>	运行结果： 出错了 退出程序

<pre>for(int i=1;i<=5;i++) numbers[i]=i; System.out.println("执行完成"); } catch(ArrayIndexOutOfBoundsException e) { System.out.println("出错了"); } finally { System.out.println("退出程序"); } }</pre>	
<pre>class Father{ public Father(){ System.out.println("I am father"); } } class Son extends Father{ public Son(){ System.out.println("I am son"); } public void sayHello(){ System.out.println("Hello,I am son"); } } } } public class Test{ public static void main(String args[]){ Son son = new Son(); son.sayHello(); } }</pre>	<p>运行结果：</p> <p>I am father</p> <p>I am son</p> <p>Hello,I am son</p>

五、判错题：

<pre>abstract class Father{ abstract public void methodA (); public void methodB(){ System.out.println("Father.methodB"); } } class Son extends Father{ public void methodB(){ System.out.println("methodB"); } } public static void main(String[] args){ Son son = new Son(); son.methodB(); } }</pre>	<p>错误分析：</p> <p>子类未实现父类的抽象方法</p>
<pre>class JavaTest { private String name = "world" ; public static void main(String args[]) { name = " the third " + 110 +name; System.out.println("Hello "+name); } } }</pre>	<p>错误分析：</p> <p>静态方法不能直接访问实例变量</p>
<pre>class JavaTest { public static void main(String args[]) { Double d[] = new Double[10]; int i; for(i=0;i<10;i++) { System.out.println(d[i].doubleValue()); } } } }</pre>	<p>错误分析：</p> <p>空指针错误</p> <p>数组中的对象没有初始化</p>
<pre>class JavaTest { public double getSqrt(float num) { return Math.sqrt(num); } public static void main(String args[]) { JavaTest t=new JavaTest(); double num = 100;</pre>	<p>错误分析：</p> <p>类型不匹配</p> <p>double 型不能自动转为float 型</p>

<pre>int numbers[]=new int[5]; try{ for(int i=0;i<5;i++) numbers[i]=i; System.out.println("执行完成"); } catch(ArrayIndexOutOfBoundsException e) { System.out.println("出错了"); } finally { System.out.println("退出程序"); } }</pre>	执行完成 退出程序
<pre>public class Test{ static int x = 5; public static void methodA(){ x = x+5; System.out.println("x in methodA() =" + x); } public static void main(String args[]) { int x = 5; methodA(); System.out.println("x in main() =" + x); } }</pre>	运行结果： x in methodA() =10 x in main() =5
<pre>class Father{ public Father(){ System.out.println("Father"); } public void sayHello(){ System.out.println("Hello World"); } } class Son extends Father{ public Son(){ System.out.println("Son"); } }</pre>	运行结果： Father Son Hello World

<pre> } public class Test{ public static void main(String args[]){ Son son = new Son(); son.sayHello(); } }</pre>	
五、判错题：	
<pre>interface abstract Study{ public void study(); } class Student implements Study{ }</pre>	错误分析： interface 不能用 abstract 修饰
<pre>class JavaTest { public static void main(String args[]) { Double d; int i = 5; i = i+5+d.intValue(); System.out.println(i); } }</pre>	错误分析： 变量D未初始化
<pre>class JavaTest { private String name = "world"; public static void main(String args[]) { System.out.println("Hello "+name); } }</pre>	错误分析： main 函数中，不能对非静态字段进行静态引用
<pre>class JavaTest { public static void main(String args[]) { String name = "Apple"; System.out.println("Hello"+name); String name = "Dog"; System.out.println("Hello"+name); } }</pre>	错误分析： 重复对局部变量赋值

六、编程题

1. 写一个函数，在指定数组中查找某个数是否存在，如果找到返回 true，找不到返回 false，函数定义如下，请给出其实现。

```
boolean findNumber(int a[],int num);

//a[]是数组，num 是要查找的数           (注：用 a.length 可以得到数组的长度)
```

```
boolean findNumber(int a[], int num) {

    int length = a.length;

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

        if (a[i] == num)    {

            return true;

        }

    }

    return false;

}
```

JAVA(三)

一 判断题

- 1 JAVA 中一个类只能实现一个接口，但可以继承多个类。 y
- 2 JAVA 程序中一个汉字和一个英文字母占的字节数相同 x
- 3 一个抽象类中只能包含抽象方法。 x
- 4 最终类(final)不能派生子类，最终方法不能被覆盖。 y
- 5 Java 源文件中的类名必须要与文件名相同。 y

二、选择题

- 1. 下面哪一个表示十六进制整数？ A
(A)0XA6 (B) 1234L (C) -840 (D) 0144
- 2.下列有关一个 Java 文件的叙述，正确的是？ D
a.可以有 2 个以上 package 语句 b.可以有 2 个以上 import 语句 c.可以有 2 个以上 public 类 (D) 只能有 1 个类定义
- 3. 下列语句会产生编译错误的是？ D
(A) float F = 1024.0F;(B) double D = 1024.0;(C) byte B = 1024;(D) char C = 1024;
- 4. 按照 Java 的标识符命名规则，下列哪个标识符作为类的名字合适？ B
(A) Helloworld (B) HelloWorld (C) helloworld (D) helloWorld
- 5. 以下选项中哪一个不是 Java 的特点： A A 自动垃圾回收 B 跨平台 (C) 面向对象 (D) 多重继承

三、简答题（每题 4 分，共 20 分）

1.写一个 HelloWorld 的程序

```
answer:

public class HelloWorld {

public static void main(String[] args){
```

```
System.out.println("Hello World!");

}

}
```

2. 试解释 Java 关键字 final 与 finally 的意义及用法。

Answer:

final 用于声明属性，方法和类，分别表示属性不可交变，方法不可覆盖，类不可继承。

finally 是异常处理语句结构的一部分，表示总是执行。

finalize 是 Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，供垃圾收集时的其他资源回收，例如关闭文件等。

3.举例说明 java 异常处理的一般程序结构？

Answer:

```
try {

someReallyExceptionalMethod();

} catch (NullPointerException n) { // a subclass of RuntimeException

...

} catch (RuntimeException r) { // a subclass of Exception

...

} catch (IOException i) { // a subclass of Exception

...

} catch (MyFirstException m) { // our subclass of Exception

...

} catch (Exception e) { // a subclass of Throwable

...

} catch (Throwable t) {

... // Errors, plus anything not caught above are caught here

} finally {...}
```

3.Java 中异常和错误类都是从 Throwable 类中继承下来的，即 Throwable 类有两个子类：Error 类及 Exception 类，Error 类的实例是 Java 运行环境的内部错误，这些错误都是非常少及非常致命的，我们不能或者很少能对这些错误作出处理，或者捕获这些错误。

进程和线程的区别是什么？

Answer:

(1)进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位。

(2)线程是进程的一个实体,是 CPU 调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源。

(3)一个线程可以创建和撤销另一个线程，同一个进程中的多个线程之间可以并发执行。

四、分析题：下列程序中由输出的显示结果是什么？

<pre>public class abc{ public static void main(String args[]) { String s1 = "Hello!"; String s2 = new String("World!"); System.out.println(s1.concat(s2)); } }</pre>	<div>运行结果：</div> <div> </div> <div>Hello!World!</div>
<pre>public class Test{ static int x = 5; public static void methodA(){ int x = 2; ++x; System.out.println("x in methodA() =" + x); } public static void methodB(){ x *= 5; System.out.println("x in methodB() =" + x); } public static void main(String args[]) { int x = 5; methodA(); methodB(); System.out.println("x in main() =" + x); } }</pre>	<div>运行结果：</div> <div> </div> <div>x in methodA() =3</div> <div>x in methodB() =25</div> <div>x in main() =5</div>
<pre>class JavaTest { public static void main(String args[]) { int numbers[]=new int[10]; try{ for(int i=1;i<=10;i++) numbers[i]=i; } catch(ArrayIndexOutOfBoundsException e) { System.out.println("下标越界"); } finally { System.out.println("退出函数。"); } } }</pre>	<div>运行结果：</div> <div> </div> <div>下标越界</div> <div>退出函数。</div> <div>退出 main()</div>

<pre> } System.out.println("退出 main()"); } }</pre>	
<pre>class Father{ void printMe(){ System.out.println("father"); } } class Son extends Father{ void printMe(){ System.out.println("son"); } void printAll(){ super.printMe(); this.printMe(); printMe(); } } public class Test{ public static void main(String args[]){ Son son = new Son(); son.printAll(); } }</pre>	<div>运行结果：</div> <div> </div> <div>father</div> <div>son</div> <div>son</div>
<div>五、判断题：（每题 5 分，共 20 分）</div>	
<pre>class Father{ void methodA (){ System.out.println("methodA"); } } class Mother{ void methodB(){ System.out.println("methodB"); } } class Son extends Father,Mother void methodC(){</pre>	<div>错误分析：</div> <div> </div> <div>extends 不能继承多个类</div>

<pre>System.out.println("methodC"); } }</pre>	
<pre>class JavaTest { int i; public static void main(String args[]) { i = 5; i = i*i*i; System.out.println("5 的立方为"+i); } }</pre>	错误分析： static 函数不能调用非static 的变量
<pre>class JavaTest { public static void main(String args[]) { Integer b[] = new Integer[100]; int i; for(i=0;i<100;i++) { System.out.println(b[i].intValue()+1); } } }</pre>	错误分析： 数组里的对象没有初始化
<pre>abstract class Father{ public void printMsg(String msg) { System.out.println("father:"+msg); } public abstract void printMsg2(); } public class Child extends Father{ public void printMsg(String msg) { System.out.println("child:"+msg); } public static void main(String[] args) { Child ch = new Child(); Father fa = (Father)ch; fa.PrintMsg(); } }</pre>	错误分析： 子类 Child 未重写父类函数 pringMsg2()

六、编程题（每题 10 分，共 20 分）

1. 排序：整数数组 a[]中有若干个整数，请写程序将数组中的整数从小到大排列。

void sortArray(int a[],int len) //a[]为待排序的数组，len 为数组元素个数

```
void sortArray(int[] a, int len) {

    int i, j, k;

    for (i = 0; i < len - 1; i++) {

        for (j = i; j < len; j++) {

            if (a[i] > a[j]) {

                k = a[i];

                a[i] = a[j];

                a[j] = k;

            }

        }

    }

    for (i = 0; i < len; i++) {

        System.out.println("NO" + i + ": " + a[i]);

    }

}
```

Java(四)

一 判断题

- 1 Java 语方中，char 型采用 Unicode 编码方案，字母与汉字都占用 16 位内存空间 y
- 2 JAVA 命名习惯中，变量名和方法名首字母小写。 y
- 3 true、null、false 都是 JAVA 语言的关键字。 x
- 4 JAVA 象 C 语言一样，具有指针操作。 x
- 5 线程是一个进程内可以并行执行的单位。 y

二、选择题

- 1. 下列语句哪一个正确 (b)
 - A. Java 程序经编译后会产生 machine code B. Java 程序经编译后会产生 byte code
 - C. Java 程序经编译后会产生 DLL D. 以上都不正确
- 2. 下列哪一种叙述是正确的 (c)
 - A. abstract 修饰符可修饰字段、方法和类 B. 抽象方法的 body 部分必须用一对大括号 {} 包住
 - C. 声明抽象方法，大括号可有可无 D. 声明抽象方法不可写出大括号
- 3. 下列哪种说法是正确的 (d)
 - A. 实例方法可直接调用超类的实例方法 B. 实例方法可直接调用超类的类方法
 - C. 实例方法可直接调用其他类的实例方法 D. 实例方法可直接调用本类的类方法
- 4. 下列运算符合法的是 (a)
a.&& b.<> c.if d.:=
- 5. 执行如下程序代码

```
a=0;c=0;

do{
```

--c;

a=a-1;

}while(a>0);

后，C 的值是（c）

A. 0B. 1 C. -1 D. 死循环

三、简答题

1. 请说明 String 与 StringBuffer 类的主要区别

answer:

StringBuffer 对象的内容可以修改；而 String 对象一旦产生后就不可以被修改，重新赋值其实是两个对象。

2 什么是方法重载以及方法重载的好处？

Answer:

定义：一个类中含两个以上同名的方法，他们之间就构成重载

特点：

1、必须是两个以上同名的方法

2、方法之间的参数组合必须不同（参数的数目不同或者参数的类型不同）

3、方法的返回值不能作为判断方法之间是否构成重载的依据

方法重载的主要好处就是，不用为了对不同的参数类型或参数个数，而写多个函数。多个函数用同一个名字，但参数表，即参数的个数或(和)数据类型可以不同，调用的时候，虽然方法名字相同，但根据参数表可以自动调用对应的函数

3 请解释 Java 语言的跨平台特性。

Answer:

Java 的跨平台特性也被称为可移植性、平台无关性，或者一次编写处处运行。他的意思就是如果用 Java 语言编写一个应用，那么就可以在不同平台上运行，而不需要为不同平台单独运行开发。之所以能实现跨平台的特性。主要得益于 Java 虚拟机（JVM），JVM 解释器在运行 Java 应用时根据当前平台进行解释，解释成符合当前平台规范的机器码，所以可以实现同样的应用在不同平台上都能运行。

4 请说明一个 JAVA 类中主要包含哪几个元素？并说明每种元素的作用。

Answer:

无论简单还是复杂的 JAVA 应用，都是由若干个类组成，所以类是 JAVA 应用的组成单位。了解一个类中包含的主要元素能够对类有一个清晰的认识。一个类中往往会有五种元素，即属性、方法、构造方法、块以及内部类、其实块和内部类比较少见。 参考答案：JAVA 类中主要包含属性、方法、构造方法、块以及内部类。属性用来定义对象的数据；方法用来定义对象的行为；构造方法可以用来创建对象；块能够用来在类加载时执行操作或者在每次实例化前执行通用操作；内部类作为类的一个成员存在，能够访问外部类的属性和方法

四、分析题（每题 5 分，共 20 分）：

<pre>public class abc{ public static void main(String args[]){ String s1 = "Hello!"; String s2 = new String("World!"); System.out.println(s1); } }</pre>	<p>运行结果：</p> <p> </p> <p>Hello!</p> <p>World!</p>
--	---

<pre> System.out.println(s2); } }</pre>	
<pre>public class Abc{ public static void main(String args[]){ int b = 5; for(int i=0;i<2;i++){ b = b + i; } System.out.println(“b=“+b); } }</pre>	<p>运行结果：</p> <p> </p> <p>b=6</p>
<pre>public class Test{ static int x = 5; public static void methodA(){ int x = 2; ++x; System.out.println("x in methodA() =" + x); } public static void methodB(){ x *= 5; System.out.println("x in methodB() =" + x); } public static void main(String args[]) { int x = 5; methodA(); methodB(); System.out.println("x in main() =" + x); } }</pre>	<p>运行结果：</p> <p> </p> <p>x in methodA() =3</p> <p>x in methodB() =25</p> <p>x in main() =5</p>

<pre>class JavaTest { public static void main(String args[]) { int numbers[]=new int[10]; try{ for(int i=1;i<=10;i++) numbers[i]=i; } catch(ArrayIndexOutOfBoundsException e) { System.out.println("下标越界"); } finally { System.out.println("退出函数。"); } System.out.println("退出 main()"); } }</pre>	运行结果： 下标越界 退出函数。 退出 main()
--	--

五、改错题：（每题 5 分，共 20 分）

<pre>class Father{ void methodA (){ System.out.println("methodA"); } } class Mother{ void methodB(){ System.out.println("methodB"); } } class Son extends Father,Mother { void methodC(){ System.out.println("methodC"); } }</pre>	错误分析： 类 Son 不能继承多个类
<pre>class JavaTest { int i; public static void main(String args[]) { i = 5; i = i*i*i;</pre>	错误分析： 静态函数不能调用非静态变量

<pre> System.out.println("5 的立方为"+i); } }</pre>	
<pre>abstract Student{ abstract public void study(); } class LittleStudent extends Student{ }</pre>	错误分析： 类 LittleStudent 未重载抽象类 Student 中方法
<pre>class JavaTest { public double getSqrt(float num) { return Math.sqrt(num); } public static void main(String args[]) { JavaTest t=new JavaTest(); double num = 100; double ret=t.getSqrt(num); System.out.println("100 的平方根为"+ret); } }</pre>	错误分析： 方法 getSqrt 不支持 double 类型的参数

六. 编程题

使用 Appication(AWT 或 SWING)的方式实现一个图形界面，里面有一个文本框、一个按钮，当点击按钮后，将文本框的内容在控制台打印输出。

答案：{

```
import java.awt.*;  
  
import java.awt.event.*;  
  
public class TestClass extends Frame{  
  
    TextField tf = new TextField();  
  
    Button bn = new Button(“OK”);  
  
public TestClass() {  
  
    super();  
  
    this.setLayout(new FlowLayout());  
  
    this.add(tf);  
  
    this.add(bn);  
  
    bn.addActionListener(new ActionListener() {  
  
        @Override  
  
        public void actionPerformed(ActionEvent e) {  
  
            // TODO Auto-generated method stub  
  
            System.out.println(tf.getText());  
  
        }  
  
    }  
  
}
```

```
public static void main(String[] args){
    TestClass testClass = new TestClass();
    testClass.setSize(300,200);
    testClass.setVisible(true);
}
}
```

一 判断题

- 1 Java 源程序编译后直接生成本地机器码。 x
- 2 Java 程序中不区分大小写字母 x
3. 说明或声明数组时不分配内存大小,创建数组时分配内存大小 x
- 4 类是一种类型,也是对象的模板。 y
- 5 Java 语言会自动回收内存中的垃圾。 y

二、选择题

1. 下列标识符（名字）命名原则中，正确的是（ c ）。

A. 类名的首字母小写 B. 接口名的首字母小写

C. 常量全部大写 D. 变量名和方法名的首字母大写

2. 下面关于 Java 语言特点的描述中，错误的是（ A ）。

A. Java 是纯面向对象编程语言，支持单继承和多继承。

B. Java 支持分布式的网络应用，可透明地访问网络上的其他对象。

C. Java 支持多线程编程。 D. Java 程序与平台无关、可移植性好。

3. 编译 Java 程序后生成的面向 JVM 的字节码文件的扩展名是 (B)。

A. .java B. .class C. .obj D. .exe

4. 哪个关键字可以抛出异常? (B)

A. transient B. throw C. finally D. Catch

5. Swing 组件必须添加到 Swing 顶层容器相关的 (B)。

A. 分隔板上 B. 内容面板上 C. 选项板上 D. 复选框内

三、简答题

1.请说明构造方法的作用和特征

构造方法的作用是用来创建对象，使用 **new** 关键字调用。构造方法的名字必须与类的名字相同，并且大小写敏感，同时构造方法不能声明返回值类型，可以使用任意一种访问修饰符，但是不能使用其他修饰符进行修饰，如 **static**、**final**、**abstract** 等修饰符都可以修饰构造方法

2. 请列举 Java 语言中的八种基本数据类型，并说明每种数据类型的含义及长度。

Answer:

Java 语言的八种基本数据类型有：byte 字节型，8 位长度；short 短整型，16 位长度；int 整型，32 位长度；long

boolean 型，表示逻辑值，有 true 和 false 两个值，分别表示真和假

3. String 类是一个“不可变类”，请解释“不可变类”的含义。

Answer:

所谓的不可变类，就是当字符串初始化后，就不能够被改变。

4 请列举 JAVA 语言的至少三种特性

1) 面向对象 2) 可移植性 3) 安全性 4) 并发机制 5) 支持可视化图形界面

四、分析题 下列程序中在控制台的输出结果是什么?

<pre><code>public class Exercises { String str = new String("Hi !"); char[] ch = { 'L', 'i', 'k', 'e' }; public static void main(String args[]) { Exercises ex = new Exercises5_1(); ex.change(ex.str, ex.ch); System.out.print(ex.str + " "); System.out.print(ex.ch); } public void change(String str, char ch[]) { str = "How are you"; ch[1] = 'u'; } }</code></pre>	<p>运行结果:</p> <p>Hi ! Luke</p>
<pre><code>public class Test { public static void main(String[] args) { int n = 1, m, j, i; for (i = 3; i <= 30; i += 2) { m = (int) Math.sqrt((double) i); for (j = 2; j <= m; j++) if ((i % j) == 0) break; if (j >= m + 1) { System.out.print(i + " "); if (n % 5 == 0) System.out.print("\n"); n++; } } } }</code></pre>	<p>运行结果:</p> <p>3 5 7 11 13</p> <p>17 19 23 29</p>

<pre> } public class Exercises6_1 extends TT{ public void main(String args[]){ Exercises6_1 t = new Exercises6_1("Tom"); } public Exercises6_1(String s){ super(s); System.out.println("How do you do?"); } public Exercises6_1(){ this("I am Tom"); } } class TT{ public TT(){ System.out.println("What a pleasure!"); } public TT(String s){ this(); System.out.println("I am "+s); } } </pre>	运行结果： What a pleasure! I am Tom How do you do?
<pre> class Aclass { void go() { System.out.println("Aclass"); } } public class Bclass extends Aclass { void go() { System.out.println("Bclass"); } public static void main(String args[]) { Aclass a = new Aclass(); Aclass a1 = new Bclass(); a.go(); a1.go(); } } </pre>	运行结果： Aclass Bclass

五、改错题

<pre> int x = 1; while (x <= 10) { i++; }</pre>	错误说明： 变量 i 未定义
--	------------------------------

<pre> class JavaTest { private String name = “world”; public static void main(String args[]) { System.out.println("Hello "+name); } } </pre>	错误说明： 静态函数不能引用非静态成员变量
<pre> interface Study{ public void study(); } class StudyImpl implements Study{ public void sayHello(){ System.out.println("Hello"); } } </pre>	错误说明： 类 StudyImpl 未实现接口 Study 的方法 study()
<pre> class JavaTest { public static void main(String args[]) { String name = "Apple"; System.out.println("Hello" + name); String name = "Dog"; System.out.println("Hello" + name); } } </pre>	错误分析： 变量 name 重复定义并赋值

六、编程题 编写一个 Java 程序在屏幕上输出 1! +2! +3! +……+10! 的和。

```

public class Test {

    public static void main(String args[]) {

        int N = 10;

        int sum = 0;

        for (int i = 1; i <= N; i++) {

            int t = 1;

            for (int j = 1; j <= i; j++) {

                t *= j;

            }

            // System.out.println("--- t: " + t);

            sum += t;

        }

        System.out.println("sum: " + sum);

    }

}

```