

Java 程序设计练习题（1）

一、单项选择题

- 下面（ B ）是合法的 Java 标识符。
A、#_pound B、_underscore C、5Interstate D、class
- 下面（ D ）赋值语句不会出现编译警告或错误。
A、float f=1.3; B、char c="a"; C、byte b=257; D、int i=10;
- 编译一个定义了两个类和三个方法的 Java 源程序文件，总共会产生（ D ）个字节码文件。
A、5 个字节码文件，以“.java”为扩展名 B、2 个字节码文件，以“.java”为扩展名
C、5 个字节码文件，以“.class”为扩展名 D、2 个字节码文件，以“.class”为扩展名
- 有关类的说法，正确的是（ B ）。
A、类具有封装性，所以类的数据是不能被访问的
B、类具有封装性，但可以通过类的公共接口访问类中的数据
C、声明一个类时，必须使用 public 修饰符
D、每个类中必须有 main 方法，否则程序无法运行
- 执行完下面程序片段后，（ B ）的结论是正确的。

```
int a, b, c;  
a = 1;  
b = 2;  
c = (a + b > 3 ? a++ : b++);
```


A、a 的值是 2，b 的值是 3 B、a 的值是 1，b 的值是 3
C、a 的值是 1，b 的值是 2 D、c 的值是 false
- 将类的成员的访问权限设置为默认的，则该成员能被（ A ）。
A、同一包中的类访问 B、其他包中的类访问
C、所有的类访问 D、所有的类的子类访问
- 下面的方法重载，正确的是（ C ）。
A、int fun(int a, float b) { } B、float fun(int a, float b) { }
 float fun(int a, float b) { } float fun(int x, float y) { }
C、float fun(float a) { } D、float fun1(int a, float b) { }
 float fun(float a, float b) { } float fun2(int a, float b) { }
- 下面关于继承的说法，正确的是（ D ）。
A、超类的对象就是子类的对象 B、一个类可以有几个超类
C、一个类只能有一个子类 D、一个类只能有一个超类
- 下面关于多态性的说法，正确的是（ C ）。
A、一个类中不能有同名的方法

- B、子类中不能有和父类中同名的方法
- C、子类中可以有和父类中同名且参数相同的方法
- D、多态性就是方法的名字可以一样，但返回的类型必须不一样

10. 关于下面程序，(A) 的结论是正确的。

```
class J_SubClass extends J_Test { }

public class J_Test {
    J_Test(int i) {
        System.out.println(i);
    }

    public static void main(String[] args) {
        J_SubClass a = new J_SubClass();
    }
}
```

- A、不能通过编译，因为类 J_Test 没有定义无参数的构造方法
- B、不能通过编译，因为类 J_SubClass 没有定义无参数的构造方法
- C、不能通过编译，因为没有实现 J_SubClass(int i) 的构造方法
- D、可以成功通过编译

11. 关于下面程序片段，(B) 的结论是正确的。

```
String a = "Java";
String b = "Java";
String x = "Ja";
String y = "va";
String c = x + y;
```

- A、a 和 b 指向同一个实例对象，a 和 c 指向同一个实例对象
- B、a 和 b 指向同一个实例对象，a 和 c 不指向同一个实例对象
- C、a 和 b 不指向同一个实例对象，a 和 c 指向同一个实例对象
- D、a 和 b 不指向同一个实例对象，a 和 c 不指向同一个实例对象

12. 假设下面的程序代码都放在 MyClass.java 文件中，(D) 程序代码能够编译通过。

- | | |
|----------------------|--------------------------|
| A、import java.awt.*; | B、package mypackage; |
| package mypackage; | import java.awt.*; |
| class MyClass { } | public class myClass { } |
| C、int m; | D、/*This is a comment*/ |
| package mypackage; | package mypackage; |
| import java.awt.*; | import java.awt.*; |
| class MyClass { } | public class MyClass { } |

13. 当需要在文件中写入字符而不是字节时，在下面的类中最好选用（ B ）类。

- A、java.io.RandomAccessFile B、java.io.PrintWriter
C、java.io.PrintStream D、java.io.PrintOutputStream

14. 关于事件监听，正确的是（ C ）。

- A、一个事件监听器只能监听一个组件
B、一个事件监听器只能监听处理一种事件
C、一个组件可以注册多个事件监听器，一个事件监听器也可以注册到多个组件上
D、一个组件只能引发一种事件

二、填空题

1. Class 类是由 Java 编译器自动生成的，它伴随每个类。
2. 任何一个 Java 程序都默认引入一个包，这个包的名字是 java.lang。
3. 在 Java 语言中，有一个类是所有类或接口的父类，这个类的名称是 java.lang.Object。
4. 在 Java 中所实现的二维数组，实际上是由一维数组构成的 数组。
5. 在 Java 中，字符串是作为 对象 出现的。
6. 接口可以看作是仅容纳 抽象方法、常量，而又不能直接生成对象的特殊抽象类。
7. FileInputStream 是字节流，BufferedWriter 是字符流，ObjectOutputStream 是 对象流。
8. 在 Java 中，组件不能独立地显示出来，必须将组件放在一定的 容器 中才能显示。
9. Applet 是能够嵌入到 HTML 格式中的文件，并能够在浏览器中运行的 Java 类。

三、判断题

- 1、Java 程序一般应当含有 main 方法，因为它是所有 Java 程序执行的入口。（ × ）
- 2、Java 语言的标识符是不区分大小写的。（ × ）
- 3、Java 语言为所有的 Java 程序自动导入包“java.lang”，因此 Java 程序可以直接用“java.lang”中的类和接口。位于类继承关系层次结构树的根部的类 Object 就是在包“java.lang”中的类。（ √ ）
- 4、有时候为了避免引起混淆，构造方法的方法名可以不与所属类名同名。（ × ）
- 5、Java 语言规定，任何一个子类的构造方法都必须调用其父类的构造方法（包括隐式调用），并且调用父类构造方法的语句必须是子类构造方法的第一条语句。（ √ ）
- 6、Java 语言对内存的释放是采用垃圾回收机制。Java 虚拟机自动判断并收集“垃圾”，但一般不会立即释放它们的存储空间。（ × ）
- 7、在编译 Java 源程序时，计算机根本不会去识别各个变量名的具体含义，因此命名规范对编写 Java 程序而言不是必要的，而且有可能会降低编写 Java 源程序的效率。（ × ）

四、程序改错题

- 1、请改正程序中的错误，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中 “/*\$1*/” 行中的语句有错误，请改正。（注意： 不改动程序的结构，不得增行或删行）

【运行结果】

Call Constructor of Base Class, 1

Call Constructor of Derived Class, 2

【程序代码】

```
class Base {
    public Base(int i) {
        System.out.println("Call Constructor of Base Class, " + i);
    }
}
class Test extends Base {
    private int b;
    public Test(int a, int b) {
        Base(a);          /* $1 */      super(a);
        System.out.println("Call Constructor of Derived Class, " + b);
    }
    public static void main(String[] args) {
        Test obj = new Test(1, 2);
    }
}
```

2、请改正程序中的错误，使它得到正确的结果。

【操作说明】

如下 java 源程序文件，该文件中 “/*\$1*/” 行中的语句有错误，请改正。（注意： 不改动程序的结构，不得增行或删行）

【运行结果】

30

【程序代码】

```
public class Test {
    private int value;
    public void Test(int n)      /* $1 */  public Test(int n)
    {
        value = n;
    }
    public int getMax(int x, int y) {
        if(x >= y && x >= value)
            return x;
        else if(y >= value)
            return y;
        else
            return value;
    }
}
```

```

    }
    public static void main(String[] args) {
        Test obj = new Test(10);
        System.out.println(obj.getMax(20, 30));
    }
}

```

五、程序填空题

1、请完善程序，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中,定义了 Bicycle 类及其子类 MountainBike 类,Bicycle 类的成员变量 model 只能被类自身的方法所访问。该文件中 “/*\$1*/” 和 “/*\$2*/” 行是需要补充的语句,请仔细阅读程序,补充相应的语句,使整个程序能够正常运行。(注意: 不改动程序的结构,不得增行或删行)

【运行结果】

China-GIANT ATX-660

【程序代码】

```

class Bicycle {
    protected String make;        //生产厂家
    /*$1*/ String model;          //型号 private
    public Bicycle(String make, String model) {
        this.make = make;
        this.model = model;
    }
    public String toString() {
        return(make + " " + model);
    }
}

class MountainBike extends Bicycle {
    public boolean hasSuspension; //悬挂
    public MountainBike(String make, String model, boolean hasSuspension) {
        /*$2*/ super(make,model);
        this.hasSuspension = hasSuspension;
    }
}

public class Test {
    public static void main(String[] args) {
        MountainBike giant = new MountainBike("China-GIANT","ATX-660",true);
        System.out.println(giant);
    }
}

```

2、请完善程序，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中，程序的功能是输入 x，按下列公式求分段函数 g(x) 的值。

$$g(x) = \begin{cases} 0.75x & x < 0 \\ \sqrt{x} & 0 \leq x \leq 20 \\ x - 6 & x > 20 \end{cases}$$

该文件中 “/*\$1*/”、“/*\$2*/” 和 “/*\$3*/” 行是需要补充的语句，请仔细阅读程序，补充相应的语句，使整个程序能够正常运行。（注意： 不改动程序的结构，不得增行或删行）

【运行结果】

请输入 x 的值:

9↵

g(9.0)=3.0

【程序代码】

```
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        /*$1*/    Scanner scanner = new Scanner(System.in);
        double x, y;
        System.out.println("请输入 x 的值: ");
        x = scanner.nextDouble();
        if(x < 0)
            y = 0.75 * x;
        else if(/*$2*/    x <= 20
            y = /*$3*/    Math.sqrt(x);
        else
            y = x - 6;
        System.out.println("g(" + x + ")=" + y);
    }
}
```

六、程序阅读题

1、以下程序代码存放在文件 Two.java 中，读程序，写出编译和运行该程序时的输出结果。

```
public class Two {
    public static void main(String[] args) {
        One one = new One("Hello");
        one.getString("Xiao ming!");
    }
}

public class One {
    private String myStr;
    public One(String str) {
        myStr = str;
    }
    public void getString(String str) {
```

```

        System.out.println(myStr + " " + str);
    }
}

```

输出结果：

编译时出现错误，一个文件中只能有一个 public 类

2、读程序，写出和程序输出格式一致的输出结果。

```

public class J_Test {
    String m_kobe = "1";
    String[] m_king = {"2"};
    public static void main(String[] args) {
        J_Test app = new J_Test();
        app.mb_operate(app.m_kobe, app.m_king);
        System.out.println(app.m_kobe+app.m_king[0]);
    }
    static void mb_operate(String kobe, String[] king) {
        kobe = new String("3");
        king[0] = new String("4");
    }
}

```

输出结果：

14

3、编译下面程序会产生哪些文件？

```

interface J_SuperClass {
    void mb_method();
}
public class J_Test {
    class J_InnerClass1 { }
    J_Test m_test = new J_Test() {
        int m_data = 1;
        public void mb_method() {
            System.out.println(m_data);
        }
    };
    void mb_method() { }
    public static void main(String[] args) {

```

```

class J_InnerClass2 { }

J_SuperClass a = new J_SuperClass() {
    int m_data = 1;
    public void mb_method() {
        System.out.println(m_data);
    }
};
}
}

```

输出结果:

```

J_SuperClass.class
J_Test$1.class
J_Test$1J_InnerClass2.class
J_Test$2.class
J_Test$J_InnerClass1.class
J_Test.class

```

内部类：外部类的名字\$内部类的名字

局部内部类：外部类的名字\$数字\$内部类的名字

匿名内部类：外部类的名字\$数字

4、读程序，写出和程序输出格式一致的输出结果。

```

public class J_Test {
    public static void mb_method(StringBuffer x, StringBuffer y) {
        x.append(y);
        y = x;
    }
    public static void main(String[] args) {
        StringBuffer a = new StringBuffer("A");
        StringBuffer b = new StringBuffer("B");
        mb_method(a, b);
        System.out.println(a + "," + b);
    }
}

```

输出结果:

AB,B

5、读程序，写出和程序输出格式一致的输出结果。

```

import java.io.RandomAccessFile;
public class J_Test {
    public static void main(String[] args) throws Exception {
        RandomAccessFile f = new RandomAccessFile("a.txt", "rw");

```



```

        f.writeBoolean(true);
        f.writeBoolean(false);
        for(int i = 1; i < 10; ++i)
            f.writeInt(i);
        f.seek(6);
        System.out.println(f.readInt());
        f.close();
    }
}

```

输出结果:

2

6、读程序，从 A、B、C、D 中选择最合适的答案。

```

import java.awt.*;
import javax.swing.*;

public class CompLay extends JFrame {
    CompLay() {
        JPanel p = new JPanel();
        p.setBackground(Color.pink);
        p.add(new JButton("One"));
        p.add(new JButton("One"));
        p.add(new JButton("One"));
        add(p, BorderLayout.SOUTH);
        setLayout(new FlowLayout());
        setSize(300, 300);
        setVisible(true);
    }
    public static void main(String[] args) {
        CompLay cl = new CompLay();
    }
}

```

- (A) 程序可以通过编译并正常运行，结果在窗口底部从左到右依次排列三个按钮
- (B) 程序可以通过编译并正常运行，结果在窗口顶部从左到右依次排列三个按钮
- (C) 程序可以通过编译并正常运行，结果在窗口中不显示任何按钮
- (D) 程序可以通过编译并正常运行，结果在窗口中只显示一个按钮

输出结果 (A、B、C、D 中选择): B

七、程序设计题

【程序描述】

确定用户输入的 m 个整数中的最大数。

【程序输入】

先给出数据的组数，对于每组数据，输入有两行，第一行为用户要输入的整数个数 m ，第二行为 m

个用空格隔开的整数。

【程序输出】

每组数据输出只有一行，为用户输入的 m 个整数中的最大数。

【输入示例】（✓表示回车）

2✓
5✓
2 3 7 5 3✓
2✓
-1 -9✓

【输出示例】

7
-1
import java.util.Scanner;
public class Test {
 public static void main(String[] args) {
 int m; //存放数据的组数
 int n; //存放数据的个数
 Scanner scanner = new Scanner(System.in);
 m = scanner.nextInt();
 for(int i=0; i<m; ++i) {
 n = scanner.nextInt();
 int x; //存放输入的数据
 int maxValue=Integer.MIN_VALUE; //存放每组数据的最大值
 for(int j=0; j<n; ++j) {
 x = scanner.nextInt();
 if(x > maxValue)
 maxValue = x;
 }
 System.out.println (maxValue);
 }
 }
}

八、程序设计题*

编写程序，要求显示一个 300×100 像素的窗口，窗口中包含三个按钮和一个标签。当点击第一个按钮时，窗口标签中显示” button1 is pressed”； 当点击第二个按钮时，窗口标签中显示” button2 is pressed”； 当点击第三个按钮时，窗口标签中显示” button3 is pressed”。如下图所示。



```
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;

public class Test extends JFrame implements ActionListener {
    //建立 1 个标签，初始文本为空，居中对齐
    JLabel displayLabel = new JLabel("", JLabel.CENTER);
    public Test() {
        //建立 3 个按钮
        JButton button1 = new JButton("Button1");
        JButton button2 = new JButton("Button2");
        JButton button3 = new JButton("Button3");
        //注册点击按钮后的事件处理程序
        button1.addActionListener(this);
        button2.addActionListener(this);
        button3.addActionListener(this);
        //得到窗体的内容窗格
        Container c = getContentPane();
        //建立 1 个面板
        JPanel bottom = new JPanel();
        //往面板中添加 3 个按钮
        bottom.add(button1);
        bottom.add(button2);
        bottom.add(button3);
        //把标签添加到窗体内容窗格北面
        c.add(displayLabel, BorderLayout.NORTH);
        //把面板添加到窗体内容窗格南面
        c.add(bottom, BorderLayout.SOUTH);
        //设置窗体大小
        setSize(300, 100);
        //显示窗体
        setVisible(true);
    }
    //点击按钮后的事件处理程序
    public void actionPerformed(ActionEvent e) {
        //得到事件源（这里是按钮）
        JButton source = (JButton)e.getSource();
        //判断按下了哪个按钮,getActionCommand 返回按钮上的文本
        if(source.getActionCommand().equals("Button1"))
            //修改按钮上的文本
            displayLabel.setText("button1 is pressed");
    }
}
```

```

        else if(source.getActionCommand().equals("Button2"))
            displayLabel.setText("button2 is pressed");
        else
            displayLabel.setText("button3 is pressed");
    }
    public static void main(String[] args) {
        new Test();
    }
}

```

九、程序设计题

编写一个计算图形面积的程序，程序应当能够计算并输出矩形、圆的面积。考虑到程序的未来扩展，设计一个图形抽象类：Shape，在此基础上派生出图形类 Rectangle 类和 Circle 类。

(1) Rectangle 类基本信息：宽度、高度。

(2) Circle 类基本信息：圆心坐标、半径。

(3) 每个图形类有多个构造方法：默认构造方法、带参数的构造方法；成员变量为 private 属性，成员方法为 public 属性。

(4) 每个图形类有计算图形面积 getArea() 方法，显示图形的基本信息 toString() 方法，以及访问器方法 set/get。

```

abstract class Shape {
    public abstract double getArea();
}

class Rectangle extends Shape {
    private double width;
    private double height;
    public Rectangle(double w, double h) {
        width = w;
        height = h;
    }
    public Rectangle() {
        this(0, 0);
    }
    public double getArea() {
        return width * height;
    }
    public double getWidth() {
        return width;
    }
    public void setWidth(double w) {
        width = w;
    }
    public double getHeight() {
        return height;
    }
}

```

```

    }
    public void setHeight(int h) {
        height = h;
    }
    public String toString() {
        return("Rectangle: width = " + width + ", height = " + height);
    }
}

```

```

class Circle extends Shape {
    private double x;
    private double y;
    private double radius;

    public Circle(double xx, double yy, double r) {
        x = xx;
        y = yy;
        radius = r;
    }
    public double getArea() {
        return Math.PI * radius * radius;
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(double r) {
        radius = r;
    }
    public double getX() {
        return x;
    }
    public void setX(double xx) {
        x = xx;
    }
    public double getY() {
        return y;
    }
    public void setY(double yy) {
        y = yy;
    }
    public void setXY(double xx, double yy) {
        x = xx;
        y = yy;
    }
    public String toString() {

```

```
        return("Circle: (" + x + ", " + y + "), radius = " + radius);  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Rectangle r = new Rectangle(10, 10);  
        Circle c = new Circle(10, 10, 50);  
  
        System.out.println(r);  
        System.out.println("area = " + r.getArea());  
  
        System.out.println(c);  
        System.out.println("area = " + c.getArea());  
    }  
}
```

Java 程序设计练习题（2）

一、单项选择题

1. 作为 Java 应用程序入口的 main 方法，其声明格式可以是（ A ）。

A、public static void main(String[] args)

B、public static int main(String[] args)

C、public void main(String[] args)

D、public int main(String[] args)

2. 在一个合法的 Java 源程序文件中定义了 3 个类，其中属性为 public 的类可能有（ B ）个。

A、0

B、1

C、2

D、3

3. 下面（ B ）单词是 Java 语言的关键字。

A、sizeof

B、abstract

C、null

D、String

4. 下面（ C ）语句不会出现编译警告或错误。

A、float f = 1.3;

B、char c = "a";

C、byte b = 25;

D、boolean d = null;

5. 下面程序片段输出的是（ D ）。

```
int i = 0, j = 9;
```

```
do {
```

```
    if(i++ > --j)
```

```
        break;
```

```
} while(i < 4);
```

```
System.out.println("i = " + i + " and j = " + j);
```

A、i = 4 and j = 4

B、i = 5 and j = 5

C、i = 5 and j = 4

D、i = 4 and j = 5

6. 对于下面（ B ）类定义，可以通过“new J_Class()”生成类 J_Class 的实例对象。

A、public class J_Class {

public J_Class(void) { }

}

B、public class J_Class { }

C、public class J_Class {

public J_Class(String s) { }

}

D、public class J_Class {

public void J_Class() { }

public J_Class(String s) { }

}

7. 要求设计一个类，它拥有一个特殊的成员域，该成员域必须能够被这个类的子类访问到，但是不能被不在同一个包内的其他类访问到。下面（ C ）可以满足上述要求。

- A、该成员域的封装属性设置为 `public`
- B、该成员域的封装属性设置为 `private`
- C、该成员域的封装属性设置为 `protected`
- D、该成员域不需要特殊的封装属性

8. 关于下面程序，（ D ）的结论是正确。

```
public class J_Test {  
    public static void main(String[] args) {  
        int[] a = new int[5];  
        boolean[] b = new boolean[5];  
        System.out.println(a[4]);  
        System.out.println(b[5]);  
    }  
}
```

- A、程序可以通过编译并正常运行，结果输出 “0false”
- B、程序可以通过编译并正常运行，结果输出 “1true”
- C、程序无法通过编译
- D、程序可以通过编译，但无法正常运行

9. 下面这段代码会产生（ C ）个 `String` 对象。

```
String s1 = "hello";  
String s2 = s1.substring(2, 3);  
String s3 = s1.toString();  
String s4 = new StringBuffer(s1).toString();
```

- A、1
- B、2
- C、3
- D、4

10. 关于下面程序，（ D ）的结论是正确的。

```
public class J_Test {  
    public static void main(String[] args) {  
        try {  
            return;  
        }  
        finally {  
            System.out.println("1");  
        }  
    }  
}
```


A、上面程序含有编译错误
B、上面程序在运行时会产生一个运行时异常
C、上面程序会正常运行，但不产生任何输出
D、上面程序输出“1”

A、如果文件“data.txt”存在，则将抛出 IOException 异常

B、如果文件“data.txt”存在，则将在文件的末尾开始添加新内容

C、如果文件“data.txt”存在，则将覆盖掉文件中已有的内容

D、如果文件“data.txt”不存在，则将抛出 IOException 异常

A、调用方法 `setLayout`

B、容器一旦生成，它的布局方式就不能改变

C、调用方法 `setLayoutManager`

D、调用方法 `updateLayout`

A、当小应用程序从运行态进入停止态时，首先要调用的成员方法是 pause()

B、当关闭正处于运行态的小应用程序时，首先要调用的成员方法是 destory()

C、当小应用程序从停止态进入运行态时，首先要调用的成员方法是 paint()

D、当关闭正处于停止态的小应用程序时，首先要调用的成员方法是 destory()

A、类 `java.sql.DriverManager` 的 `getDriver` 方法

B、类 `java.sql.DriverManager` 的 `getDrivers` 方法

C、`java.sql.Driver` 的方法 `connect`

D、类 `java.lang.Class` 的 `forName` 方法

1. Java 语言通过 垃圾回收 机制简化了程序的内存管理。
2. Java 标识符是由 字母、数字、下划线、美元符号 组成的字符序列。
3. 面向对象的计算机语言一般应具有 3 个基本特性，分别是 封装、继承、多态。
4. Java 中的 java.lang.Object 类是最顶层的类，所有其他的类都是它直接或间接的子类。
5. 在 Java 的基本数据类型中，char 类型采用 Unicode 编码方案，这样，无论是中文字符还是英文字符，都是占 2 字节内存空间。
6. 类的静态成员属于这个类的 所有对象。
7. 一个类要具有可序列化的特性一般就必须实现接口 java.io.Serializable。
8. 类 String 本身负责维护一个字符串池。该字符串池存放 字符串常量 所指向的字符串示例，以及调用过类 String 成员方法 intern 后的字符串示例。

三、判断题

- 1、父类的构造方法不可以被子类继承。(☒)
- 2、在接口中定义的方法都只能是没有方法体的抽象方法。(☒)
- 3、Java 程序分成应用程序和小应用程序两类，所以一个 Java 程序不可能既是应用程序，又是小应用程序。(☐)
- 4、javadoc 是一种从文档注释生成 HTML 帮助文件的工具。它可以处理在 java 源程序中介于 “/*” 和 “*/” 之间的注释，并生成相应的程序文档。(☐)
- 5、采用事件适配器比采用事件监听器的执行效率要高，因为采用事件适配器可以只处理所需处理的事件。(☐)
- 6、每个 java.sql.Statement 实例对象只能对应一个 java.sql.ResultSet 实例对象，所以如果执行 SQL 语句返回多个结果，一般需要分多次取得返回结果。(☒)

四、程序改错题

- 1、请改正程序中的错误，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中 “/*\$1*/” 行中的语句有错误，请改正。(注意： 不改动程序的结构，不得增行或删除行)

【运行结果】

Base function

Derived function

【程序代码】

```
class Base {
    public void fun() {
        System.out.println("Base function");
    }
}

public class Test extends Base {
    public void fun() {
        System.out.println("Derived function");
    }

    public static void main(String[] args) {
        Base b = new Base();
        Test dp, d = new Test();
        dp = b;    /*$1*/    dp = d;
        b.fun( );
        dp.fun( );
    }
}
```

2、请改正程序中的错误，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中 “/*\$1*/” 行中的语句有错误，请改正。（注意： 不改动程序的结构，不得增行或删除行）

【运行结果】

a=10

【程序代码】

```
class One {
    private int a;
    public void setOne(int x) { a=x; }
    public void showOne(){ System.out.println("a=" + a); }
}
public class Test extends One {
    public void setTwo(int x) { setOne(x); }
    public static void main(String[] args) {
        Test obj = new Test();
        obj.setTwo(10);
        System.out.println("a=" + obj.a);          /*$1*/    obj.showOne();
    }
}
```

五、程序填空题

1、请完善程序，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中，程序的功能是在数组中查找键盘键入的整数。如果找到则输出该整数在数组中的序号，否则输出 “not found”。例如，运行程序，输入 99，程序输出：99 在数组中的序号是 5；输入 59，程序输出：not found。该文件中 “/*\$1*/”、“/*\$2*/” 和 “/*\$3*/” 行是需要补充的语句，请仔细阅读程序，补充相应的语句，使整个程序能够正常运行。（注意： 不改动程序的结构，不得增行或删除行）

【程序代码】

```
import java.io.*;
import java.util.Scanner;
public class J_Test {
    public static void main(String[] args) throws IOException {
        int _____ /*$1*/ _____ = {83,75,92,67,55,99,78,61};    a[]
        Scanner in = new Scanner(System.in);
        int x = Integer.parseInt(in.readLine());
        int index = -1;
        for(int i = 0; _____ /*$2*/ _____; ++i) {            i < a.length
            if(a[i] == x) {
                index = i;
                break;
            }
        }
    }
}
```

```

    }
}
if(_____/*$3*/_____)    index == -1
    System.out.println("not found");
else
    System.out.println(x + "在数组中的序号是" + index);
}
}

```

2、请完善程序，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中，程序的功能是判断回文（字符串顺读和倒读都一样，则是回文。如“abba”是回文）。方法 palindrome 判断字符串 s 是否是回文，若是回文，返回 true，否则返回 false。该文件中“/*\$1*/”、“/*\$2*/”和“/*\$3*/”行是需要补充的语句，请仔细阅读程序，补充相应的语句，使整个程序能够正常运行。（注意：不改动程序的结构，不得增行或删行）

【程序代码】

```

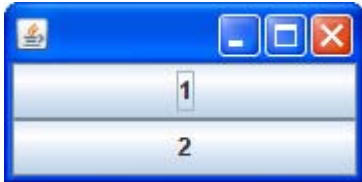
import java.util.Scanner;
public class J_Test {
    /*$1*/    static boolean palindrome(String s)
    {
        int len = s.length();
        int h = 0, t = len - 1;
        while(h <= t) {
            if(s.charAt(h) != s.charAt(t))
                break;
            h++;
            t--;
        }
        if(/*$2*/    h>t
            return true;
        else
            return false;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("请输入字符串: ");
        String s = /*$3*/    scanner.next();
        if(palindrome(s))
            System.out.println(s + " 是回文");
        else
            System.out.println(s + " 不是回文");
    }
}

```

3、请完善程序，使它得到正确的结果。

【操作说明】

如下 java 源程序文件中，程序的功能是在图形界面上显示两个按钮，当用户按下一个活动按钮时，该按钮将被置成灰色。当用户按下一个活动按钮时，另一个按钮将处于激活状态。该文件中 “/*\$1*/” 和 “/*\$2*/” 行是需要补充的语句，请仔细阅读程序，补充相应的语句，使整个程序能够正常运行。（注意：不改动程序的结构，不得增行或删行）



【程序代码】

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class J_Test extends JFrame {
    JButton m_button[] = new JButton[2];

    public J_Test() {
        Container c = getContentPane();
        m_button[0] = new JButton("1");
        m_button[0].addActionListener(a);
        c.add(m_button[0], BorderLayout.NORTH);
        m_button[1] = new JButton("2");
        m_button[1].addActionListener(a);
        c.add(m_button[1], BorderLayout.SOUTH);
    }

    /*$2*/ ActionListener a = new ActionListener()
    {
        public void actionPerformed(ActionEvent e) {
            if(e.getSource() == m_button[0]) {
                m_button[0].setEnabled(false);
                m_button[1].setEnabled(true);
            }
            else {
                m_button[0].setEnabled(true);
                m_button[1].setEnabled(false);
            }
        }
    };

    public static void main(String[] args) {
        J_Test app = new J_Test();
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        app.setSize(180, 90);
        /*$1*/ app.setVisible(true);
    }
}
```

```
}  
}
```

六、程序阅读题

1、读程序，写出和程序输出格式一致的输出结果。

```
public class J_Test {  
    public static void main(String[] args) {  
        int sum = 0;  
        outer:  
        for(int i = 1; i < 10; ++i) {  
            inner:  
            for(int j = 1; j < 3; ++j) {  
                sum += j;  
                if(i + j > 6)  
                    break inner;  
            }  
        }  
        System.out.println("sum = " + sum);  
    }  
}
```

输出结果：

sum = 19

2、读程序，写出和程序输出格式一致的输出结果。

```
class J_SuperClass {  
    void mb_method() {  
        System.out.println("J_SuperClass::mb_method");  
    }  
    static void mb_methodStatic() {  
        System.out.println("J_SuperClass::mb_methodStatic");  
    }  
}  
public class J_Test extends J_SuperClass {  
    void mb_method() {  
        System.out.println("J_Test::mb_method");  
    }  
    static void mb_methodStatic() {  
        System.out.println("J_Test::mb_methodStatic");  
    }  
    public static void main(String[] args) {  
        J_SuperClass a = new J_Test();  
        a.mb_method();  
        a.mb_methodStatic();  
        J_Test b = new J_Test();  
        b.mb_method();  
    }  
}
```

```

        b.mb_methodStatic();
    }
}

```

输出结果:

```

J_Test::mb_method
J_SuperClass::mb_methodStatic
J_Test::mb_method
J_Test::mb_methodStatic

```

3、读程序，写出和程序输出格式一致的输出结果。

```

class ValHold {
    public int i = 10;
}

public class ObParm {
    public static void main(String[] args) {
        ObParm o = new ObParm();
        o.amethod();
    }
    public void amethod() {
        int i = 99;
        ValHold v = new ValHold();
        v.i = 30;
        another(v, i);
        System.out.println(v.i + " " + i);
    }
    public void another(ValHold v, int i) {
        i = 0;
        v.i = 20;
        ValHold vh = new ValHold();
        v = vh;
        System.out.println(v.i + " " + i);
    }
}

```

输出结果:

```

10 0
20 99

```

4、读程序，写出和程序输出格式一致的输出结果。

```

class J_SuperClass {
    int m_data;
    J_SuperClass() {
        mb_add(1);
    }
    public void mb_add(int i) {
        m_data += i;
    }
}

```

```

    }
    public void mb_print() {
        System.out.println(m_data);
    }
}
class J_SubClass extends J_SuperClass {
    J_SubClass() {
        mb_add(2);
    }
    public void mb_add(int i) {
        m_data += i * 2;
    }
}
public class J_Test {
    public static void mb_method(J_SuperClass a) {
        a.mb_add(6);
        a.mb_print();
    }
    public static void main(String[] args) {
        mb_method(new J_SubClass());
    }
}

```

输出结果:

18

5、读程序，写出和程序输出格式一致的输出结果。

```

public class J_Test {
    public static void mb_method(int i) {
        try {
            if(i == 1)
                throw new Exception();
            System.out.print("1");
        }
        catch(Exception ex) {
            System.out.print("2");
            return;
        }
        finally {
            System.out.print("3");
        }
        System.out.print("4");
    }
    public static void main(String[] args) {
        mb_method(0);
        mb_method(1);
    }
}

```



```
}  
}
```

输出结果:

13423

6、读程序，从 A、B、C、D 中选择最合适的答案。

```
import java.awt.*;  
import javax.swing.*;  
public class J_Test extends JFrame {  
    public static void main(String[] args) {  
        J_Test app = new J_Test();  
        app.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        Container ct = app.getContentPane();  
        ct.setLayout(new GridLayout(1, 2));  
        JButton[] b = { new JButton("one"),new JButton("Two"),new JButton("Three") };  
        ct.add(b[0]);  
        ct.add(b[1]);  
        ct.add(b[2]);  
        app.setSize(220, 70);  
        app.setVisible(true);  
    }  
}
```

- (A) 程序可以通过编译并正常运行，结果在程序界面上从左到右依次排列三个按钮
- (B) 程序可以通过编译并正常运行，结果在程序界面上三个按钮分成两行排列
- (C) 程序可以通过编译并正常运行，结果在程序界面上只出现两个按钮
- (D) 程序无法通过编译

输出结果 (A、B、C、D 中选择): A

七、程序设计题

【程序描述】

接收用户输入的 m 个正整数，返回 m 个正整数数字逆序后的结果值。例如，输入一个整数 7631，输出 1367。

【程序输入】

第一行一个正整数 m ，表示有 m 组测试数据；以下 m 行，每行一个正整数 n 。

【程序输出】

共 m 行，对于每组输入数据输出一行，即数字逆序后的结果值。

【输入示例】(✓表示回车)

3✓

7631✓

101✓

51 ✓

【输出示例】

1367

101

15

```
import java.util.Scanner;
```

```
public class Test {
```

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

```
        int m;        //存放数据的组数
```

```
        int n;        //存放输入的数据
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        m = scanner.nextInt();
```

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

```
            n = scanner.nextInt();
```

```
            int combination = 0;
```

```
            do {
```

```
                combination = combination * 10 +(n % 10);
```

```
                n /= 10;
```

```
            } while(n > 0);
```

```
            System.out.println(combination);
```

```
        }
```

```
    }
```

```
}
```

Java 程序设计练习题（3）

一、单项选择题

- 下面选项中，（ B ）是 Java 关键字。
A、then B、continue C、java D、PUBLIC
- 下面语句中，正确的是（ B ）。
A、boolean b="true"; B、double x=2.5f; C、char c="A"; D、float y=0.8d;
- 设有定义“int k=3;”，语法正确且值为 true 的表达式是（ D ）。
A、k=3; B、k++>3;
C、k--==3&& k++==3; D、k++==3||++k>3;
- 设有定义：String s="World";，下列语句错误的是（ D ）。
A、int m=s.indexOf('r'); B、char c=s.charAt(0);
C、int n=s.length(); D、String str=s.append('2');
- 假设在 Java 源程序文件“MyClass.java”中只含有一个类，而且这个类必须能够被位于一个庞大的软件系统中的所有 Java 类访问到，那么下面（ C ）声明有可能是符合要求的类声明。
A、private class MyClass extends Object
B、public class myclass extends Object
C、public class MyClass
D、class MyClass extends Object
- 在 Java 中，用 package 语句说明一个包时，该包的层次结构必须是（ A ）。
A、与文件目录的层次相同 B、与文件的结构相同
C、与文件类型相同 D、与文件大小相同
- 下面关于数组的说法，错误的是（ B ）。
A、数组是最简单的复合数据类型，是一系列数据的集合
B、声明数组时，必须分配内存
C、数组的元素可以是值（基本数据类型）、对象或其他数组
D、一个数组中的所有值都必须是相同的类型
- 下面关于方法的说法，错误的是（ C ）。
A、Java 中的方法参数传递时传值调用，而不是地址调用
B、方法体是对方法的实现，包括变量声明和 Java 的合法语句
C、如果程序定义了一个或多个构造方法，在创建对象时，也可以用系统自动生成空的构造方法
D、类的私有方法不能被其子类直接访问
- 下面关于内部类的说法，错误的是（ A ）。
A、内部类不能有自己的成员方法和成员变量
B、内部类可用 abstract 修饰定义为抽象类，也可以用 private 或 protected 定义

- C、内部类可作为其他类的成员，而且可访问它所在类的成员
D、除 static 内部类外，不能在类内声明 static 成员
10. 在字节流中，可以使用文件名作为参数的类有（ C ）。
- A、DataInputStream B、BufferedReader
C、FileInputStream D、FileReader
11. 语句：JTextField=new JTextField(8);，语句中的 8 表示文本框的（ A ）。
- A、列宽 B、所显示的字符串 C、行高 D、编号
12. 文档标签@exception 在文档注释中的格式一般是（ A ）。
- A、@exception 类名 描述 B、@exception 异常描述
C、@exception 异常变量名 描述 D、不存在文档标签@exception

二、填空题

1. Java 虚拟机运行 Java 程序的基本步骤是：首先从后缀为 class 的文件加载代码到内存中，接着在内存中检测代码的合法性和安全性，然后解释执行合法和安全的代码。
2. 在 Java 语言中，字符串直接量是用双引号括起来的字符序列，字符串不是字符数组，而是类 String 的实例对象。
3. 类 System 的三个成员域 in、out、err 分别指向标准输入流、标准输出流和标准错误输出流。
4. 有时运行一个 Java 小应用程序需要多个文件，如“.class”文件、声音文件、图像文件等。为了提高效率，可以考虑将运行小应用程序所需要的各种文件做成一个 jar 文件。
5. 假设用 Java 语言为某大学（网址为“hznu.edu.cn”）编写了图像处理包（image），那么按照 Sun 公司的建议，这个包的名称最好为 cn.edu.hznu.image。
6. 在编写和运行数据库程序之前需要建立起 Java 数据库程序的开发环境，这至少需要安装 JDK、数据库和 JDBC 驱动程序。
7. 传递给实现了 java.awt.event.MouseMotionListener 接口的类中 mouseDragged 方法的事件对象是 MouseEvent 类。

三、判断题

- 1、假设有一个 Java 源程序文件，它只定义了一个具有 public 属性的类 Hello，那么编译该文件的命令是“javac Hello”。（ × ）
- 2、switch 语句中可以没有 default 子句。（ √ ）
- 3、常量 87.363 的数据类型是 float。（ × ）
- 4、构造方法一般不允许有任何返回值，因此需要在构造方法返回类型处标注为 void。（ × ）
- 5、如果在定义一个类的时候没有用到关键字 extends，则这个类没有直接父类。（ × ）
- 6、抽象方法不能含有方法体，并且必须在抽象类中。（ × ）
- 7、如果类的访问控制符是 public，则类中成员的访问控制属性也必须是 public。（ × ）
- 8、假设文件“a.txt”的长度为 100 字节，那么当正常运行语句“OutputStream f = new

- FileOutputStream(new File("a.txt"));" 之后, 文件 "a.txt" 的长度变为 0 字节。(√)
- 9、对于数组 `int[][] t = {{1, 2, 3}, {4, 5, 6}}` 来说, `t.length` 等于 3, `t[0].length` 等于 2。(×)

四、程序阅读题

1、读程序, 写出编译和运行该程序时的输出结果。

```
class J_SuperClass { }  
  
class J_SubClass extends J_SuperClass { }  
  
public class J_Test {  
    public static void main(String[] args) {  
        J_SuperClass a = new J_SuperClass();  
        J_SubClass b = new J_SubClass();  
        b = a;  
    }  
}
```

输出结果:

编译时出现错误, 不兼容的类型 (b = a;)

2、读程序, 写出和程序输出格式一致的输出结果。

```
public class J_Test {  
    public static void main(String[] args) {  
        int sum = 0;  
        outer:  
        for(int i = 1; i < 10; ++i) {  
            inner:  
            for(int j = 1; j < 3; ++j) {  
                sum += j;  
                if(i + j > 6)  
                    continue outer;  
            }  
        }  
        System.out.println("sum = " + sum);  
    }  
}
```

输出结果:

sum = 19

3、运行下面的程序, 并输入字符'0'。读程序, 写出和程序输出格式一致的输出结果。

```

import java.io.*;

class J_Exception extends Exception { }

public class J_Test {
    public static void main(String[] args) {
        try {
            int i = System.in.read();
            if(i == '0')
                throw new J_Exception();
            System.out.print("1");
        }
        catch(IOException ex) {
            System.out.print("2");
        }
        catch(J_Exception ex) {
            System.out.print("3");
        }
        finally {
            System.out.println("4");
        }
    }
}

```

输出结果：

34

4、假设文本文件“test.txt”存在。读程序，写出编译和运行该程序时的输出结果。

```

import java.io.*;

public class J_Test {
    public static void main(String[] args) {
        try {
            int ch;
            FileReader f = new FileReader("test.txt");
            while((ch = f.read()) != -1)
                System.out.print((char)ch);
            f.close();
        }
        catch(IOException ex) {

```

```

        System.out.println(ex);
    }
    System.out.println();
}
}

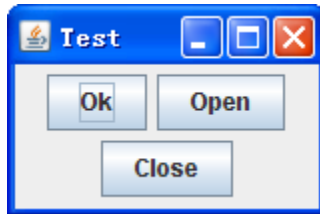
```

输出结果:

将文件 “test.txt” 的内容显示在控制台窗口中

五、程序填空题

1、下面程序的运行结果如下图。单击 “Close” 按钮关闭窗口。



请阅读程序，在（1）和（2）处补充相应的内容，使整个程序能够正常运行。

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class J_Test {
    private JFrame f;
    private JButton button1, button2, button3;
    public void go() {
        f = new JFrame("Test");
        f.setLayout(new FlowLayout());
        button1 = new JButton("Ok");
        button2 = new JButton("Open");
        button3 = new JButton("Close");
        f.add(button1);
        f.add(button2);
        f.add(button3);
        _____
        f.setSize(160, 106);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        J_Test t = new J_Test();
        t.go();
    }
}
class Monitor implements ActionListener {

```

```

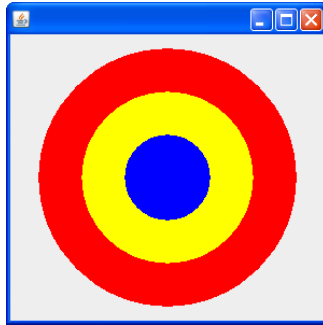
        (2) _____ {
            System.exit(0);
        }
    }
}
(1) button3.addActionListener(new Monitor());

(2) public void actionPerformed(ActionEvent e)

```

六、程序设计题

编写程序，要求在 300×300 像素的一个窗口中间显示 3 个同心圆，半径分别为 120 像素，80 像素，40 像素，填充颜色分别为红(最大的圆)，黄(中间的圆)，蓝(最小的圆)。如下图所示。



```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Test extends JFrame {
    public Test() {
        //得到窗体的内容窗格
        Container c = getContentPane();
        //往内容窗格中添加面板
        c.add(new DrawPanel());
        //设置窗体大小
        setSize(300, 300);
        //设置点击窗体右上角关闭按钮时的动作
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //显示窗体
        setVisible(true);
    }

    public static void main(String[] args) {
        new Test();
    }
}

class DrawPanel extends JPanel {
    //在面板中画图，paintComponent 由系统自动调用
    public void paintComponent(Graphics g) {

```



```
//设置颜色
g.setColor(Color.red);
//画实心圆
g.fillOval((getWidth()-240)/2, (getHeight()-240)/2, 240, 240);
g.setColor(Color.yellow);
g.fillOval((getWidth()-160)/2, (getHeight()-160)/2, 160, 160);
g.setColor(Color.blue);
g.fillOval((getWidth()-80)/2, (getHeight()-80)/2, 80, 80);
}
}
```