

班级：\_\_\_\_\_姓名：\_\_\_\_\_学号：\_\_\_\_\_

## 《JAVA 程序设计》试题 1

### 一、 单项选择填空（每小题 2 分，共 50 分）

- 下列有关 Java 语言的叙述中，正确的是 (B)  
A、Java 是不区分大小写的 B、源文件名与 public 权限的类名必须相同  
C、源文件名其扩展名为.jar D、源文件中 public 类的数目不限
- 在 Java 中，负责对字节代码解释执行的是 (B)。字节就是 class 文件  
A、垃圾回收器 B、虚拟机 C、编译器 D、多线程机制
- 在 Java 程序设计中，由 Java 编译器默认导入的包是 (D)  
A、java.applet B、java.awt C、java.util 输入语言 D、java.lang
- 编译 Java application 源程序文件将产生相应的字节码文件，这些字节文件的扩展名为 (B)  
A、.java B、.class C、.html D、.exe
- 在 Java 中，用 package 语句定义一个包时，该包的层次结构即是 (b)  
A、与文件的结构相同 B、与文件目录的层次相同  
C、与文件类型相同 D、与文件大小相同
- 有时在程序设计中采用分治算法而选择递归，“递归”的基本思想是 (c)  
A、“让别人反复调用自己” B、“自己反复调用别人”  
C、“自己反复调用自己” D、以上说法都不对
- main 方法是 Java Application 程序执行的入口，含有 main 方法的类是主类，且 main 方法的定义是唯一的，以下哪项是正确的 (b)  
A、public static void main () B、public static void main(String args[])  
C、public static int main(String arg[])  
D、public void main(String arg[])
- 按照 Java 的标识符的行业命名规范，下列分别表示一个类和一个方法的标识符比较合适的是 (b)  
A、Helloworld,GetSwimmingScore B、HelloWorld,getSwimmingScore  
C、helloworld,getswimmingscore D、helloWorld,Getswimmingscore
- 以下程序的运行结果为 (d)  

```
public class IfTest{
    public static void main(String args[]){
        int x=3;
        {
            int y=1;
        }
        if(x==y)
            System.out.println(“equal”);
        else
            System.out.println(“Not Equal”);
    }
}
```

  
A、Not equal B、Equal C、无输出 D、编译出错，找不到 y

10. 下列说法不正确的是 (c)
- A、IOException 必须被捕获或抛出
  - B、java 语言不自动初始化变量的值
  - C、java 语言不允许同时继承一个类并实现一个接口
  - D、java 语言会自动回收内存中的垃圾
11. 在 Java 中，所有类的根类是 (a)
- A、java.lang.Object
  - B、java.lang.Class
  - C、java.applet.Applet
  - D、java.swing.JFrame
12. 下列方法中可以用来创建一个新线程的是 (b)
- A、实现 java.lang.Runnable 接口并重写 start()方法
  - B、实现 java.lang.Runnable 接口并重写 run()方法
  - C、实现 java.lang.Thread 接口并重写 run()方法
  - D、实现 java.lang.Thread 类并重写 start()方法
13. 启动 Applet 小程序时，即调用类似于应用程序 main 方法的入口方法(b)
- A、stop()
  - B、init()
  - C、start()
  - D、destroy()
14. Java 的字符类型采用的是 Unicode 编码方案，每个 Unicode 码占用\_\_\_\_\_个比特位。( b )
- A、8
  - B、16
  - C、32
  - D、64
15. 下面哪个选项是 Java 语言中的关键字？ (c)
- A. toString
  - B. Class
  - C. instanceof
  - D. Int
16. 以下哪个选项是字符串对象的长度？ (d)
- A. size
  - B. length
  - C. max\_value
  - D. length()
17. 下面程序段的执行结果是什么？ (b)
- ```
public class Exp{
    public static void main(String[] args){
        try{ int f=12;return;}
        catch(Exception e){ System.out.println(e.getMessage());}
        finally{System.out.println( "Finally!" );}
    }
}
```
- A、程序正常运行，但不输出任何结果
  - B、程序正常运行，并输出 Finally
  - C、编译能通过，但运行时会出现例外
  - D、因为 catch 语句块不对应，所以不能通过编译
18. class MethodOverload{
- ```
    public void setValue(int a, int b, float c){}
}
```
- 下列哪个方法定义是 setValue 的重载方法： (c)
- A. public void setValue (int a, int c, float b){}
  - B. public void setVar(int a, int b){}
  - C. public int setValue(int a, float c, int b){return a;}
  - D. public float setValue(int a, int c, float b){return a;}
19. 下列叙述中哪个是不正确的？ (d)
- A、final 类不可以有子类
  - B、abstract 类不可以有 abstract 的静态方法
  - C、不可以同时用 final 和 abstract 修饰一个方法
  - D、abstract 类中可以有非 abstract 方法，但该方法不可以用 final 修饰
- 接口中只能有常量和抽象方法；抽象类中可以有抽象方法和非抽象方法；

20. 阅读下列代码，

```
public class Person{  
    int[] arr[]=new int[10] [];  
    public static void main(String args[]){  
        arr[1]=new int[10];System.out.println(arr[1][0]);  
    }  
}
```

根据上面的代码，下列正确的说法是（）

- A、编译错误，提示无法从静态上下文中引用非静态变量 `arr`  
B、编译时正确，运行时将产生错误      C、输出 0      D、输出空

21. 设有下面的两个类定义：

```
class A{  
    void Show(){System.out.println("A study Java ! ");  
}  
class BB extends A{  
    void Show(){System.out.println("BB study C++ ! ");  
}}}
```

则顺序执行如下语句后输出结果为：（a）

```
A   a=new A();      A b=new BB(); a.Show();   b.Show();
```

- A、A study Java!      B、A study C++!  
BB study C++!      BB study Java!  
C、A study Java!      D、A study C++!  
BB study Java!      BB study C++!

22. Swing 与 AWT 的区别不包括：（d）

- A、Swing 是由纯 Java 实现的轻量级构件      B、Swing 没有本地代码  
C、Swing 不依赖操作系统的支持      D、Swing 支持图形用户界面 都支持

23. 容器 `JFrame` 及 `JPanel` 默认的布局管理器分别是（b）

- A、`FlowLayout` 和 `GridLayout`      B、`BorderLayout` 和 `FlowLayout`  
C、`FlowLayout` 和 `FlowLayout`      D、`CardLayout` 和 `BorderLayout`

24. 在开发一个 JAVA GUI 程序时，通常都要对按键事件作出响应和处理，一般需要在程序的开头写上（d）语句。

- A、`import java.awt.*;`      B、`import java.applet.*;`  
C、`import java.io.*;`      D、`import java.awt.event.*;`

25. 每个使用 Swing 组件的程序必须有一个（d）

- A、按钮      B、标签      C、菜单      D、容器

## 二、 填空题（每空 1 分，共 20 分）

1. JAVA 源程序是由 包定义语句、导入语句、类定义、方法定义和执行语句等五部分组成的。
2. 根据类定义的格式，类体里一般有内部类、属性（成员变量）和方法。
3. 面向对象程序设计的三个基本特征分别是抽象、封装、继承、多态。
4. 算术运算异常、下表数组越界异常类都源于 Exception 这个父类。
5. 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。
6. 接口的所有成员属性都默认且必须是 public、static 和 final 的。
7. 方法的 重载 是指在同一个类中，声明方法名称相同但参数不同的多个方法。
8. Java 语言的各种数据类型之间提供两种转换：自动类型转换和强制转换。
9. 在类的构造方法和实例方法中，使用关键字 this 可以解决局部变量与实例变量同名的问题。
10. this 关键字代表当前对象，而 super 关键字是指 当前对象的父类。
11. 抽象方法不能使用修饰符 final 和 static 修饰。
12. 在进行 Java GUI 程序设计时，一般都要用到 swing 特性，因此必须在程序的开头写上 import javax.swing.\*; 语句。

## 三、 判断题（每小题 1 分，共 7 分），请将正确的打 √，错误的打 ×。

- 1、在异常处理中，若 try 块中的代码可能产生多种异常则可以对应多个 catch 语句，如 try 块中没有产生异常的代码，那么可以在 try 块后不对

应 **catch** 块（即可无 **catch** 块）（不能没有 **catch**），而 **finally** 块是无论异常发生与否都要执行的语句块，它是可有可无的。（**finally** 可以没有）

（ 错 ）

2、Java 程序是由若干类定义组成的，类定义包括定义类头和定义类体。

（ 对 ）

3、有的类定义时可以不定义构造方法，因为当用户不定义一个或以上的构造方法时系统会为其自动创建一个默认的无参构造方法，且含有一条 **super()** 语句。（ 对 ）

4、抽象类中的抽象方法必需在该类的子类中具体实现。（不是必需实现）

（ 错 ）

5、终态类不能派生子类，终态方法不能被覆盖。（对）

6、一个 Java 类可以有一个父类，并实现多个接口。（对）

7、接口是特殊的抽象类，所以接口也可以继承，子接口将继承父接口的所有常量和抽象方法。（对）

#### 四、 简答题（每题 3 分，共 9 分）

1、简述静态成员与非静态成员的区别

静态是被所有对象共享。怎么访问静态，可以通过类名访问。非静态不能访问静态

2、简述构造方法的特性；略

3、在一般情况下，J2SE GUI 程序设计的六步流程

创建顶层容器及中间容器

设置容器状态：大小，颜色..

创建组件及设定其状态

为组件及容器添加相应的监听器。

按照容器的布局管理方式添加组件。

将顶层容器显示。

#### 五、 写程序结果及程序填空（第 1、2 题每题 3 分，第 3 题程序填空每空 2 分【小计 8 分】，共 14 分）

```
1、 class Animal {  
    public Animal() {  
        System.out.print ("Animal—");  
    }  
}  
public class Dog extends Animal {  
    private String name;  
    public Dog(){this( “无名狗” );}  
    public Dog(String name) {this.name=name; }  
    public String getName(){return name;}  
    public void printName(){System.out.println(getName());}  
    public static void main(String[] args) {  
        Dog snoppy= new Dog ( “史努比” );snoppy. printName();  
    }  
}
```

程序结果为: **Animal—史努比**

2、 public class abc

```
{
    public static void main(String args[])
    { SubSubClass x = new SubSubClass(10 , 20 , 30);
      x.show();
    }
}
class SuperClass
{ int a,b;
  SuperClass(int aa , int bb)
  { a=aa; b=bb; }
  void show( )
  { 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); }}
```

程序结果为:      a=60

                    b=20

                    c=30

3、 以下程序实现了如下图示的计算器界面：将 16 个按钮放置在面板中（面板的布局为 4\*4 的 **GridLayout** 布局），再将面板放置在框架中的中间（框架布局为缺省布局 **BorderLayout**）。但还有 4 处代码并未完成，请你在答题纸上完成程序填空。



```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class AppMain
{
    public static void main (String[] args)
    {
        new MyFrame("计算器");
    }
}
class MyFrame extends JFrame
{
    JTextField  tf_Output=null;//输出结果文本域
    JPanel      p=new JPanel();
    JButton[]  b=new JButton[16];

    MyFrame(String title)
    {
        super(title);
        【代码 1】 this.setBounds(200,190,179,180); //设置窗体的位置距离
        屏幕顶端距离 190，距离左端 200，高度 180，宽度 179
        //由于 JFrame 的缺省布局为 BorderLayout 所以不需设置其布局管理方式了
        Container c=this.getContentPane();//获取当前窗体对象的内容窗格
        tf_Output =new JTextField(20);
        【代码 2】 c.add(tf_Output,"North"); //按照图片提示，将 tf_Output
        文本域加入窗体对应位置

        String name[]={"1","2","3","+";//第 1 行
                        "4","5","6","-";//第 2 行
                        "7","8","9","*";//第 3 行
                        "0",".", "=", "/"//第 4 行
                        };

        p.setLayout(new GridLayout(4,4));//设置面板 p 的布局管理方式为 4*4
        的网格布局
        for(int i=0;i<name.length;i++)
        {
            b[i]=new JButton(name[i]);
            p.add(b[i]);
        }
        c.add(p,BorderLayout.CENTER);//将面板放置在窗体中的中间
        this.addWindowListener(new WindowAdapter() //侦听窗体的关闭操作，
        实现关闭窗体时退出程序
        {public void 【代码 3】 windowClosing(WindowEvent e)
            {System.exit(0);}ew
        });
    }
}

```



```
        【代码 4】 this.setVisible(true);    //将窗体显示  
    }  
}
```