

## 习题-第一章

### 一、选择题(答案可能不止一个)

1. ( B )是以对象为特征的可视化程序组件?  
A. 过程化语言    B. 面向对象的语言    C. 机器语言    D. 以上都不是
2. Java 语言在体系结构上是( C )。  
A. 过程化语言    B. 抽象的    C. 与平台无关的    D. 特定的
3. 所有 Java 应用程序必须包含下面哪个方法才能运行?( D )  
A. start()    B. begin()    C. main()    D. main(String args[])
4. 执行编译好的 Java 程序的命令是( B )。  
A. javac    B. java    C. run    D. execute
5. 在 Java 中, 类可以具有任何数目的超类?( B )  
A. 正确    B. 错误
6. 以下哪种 Java 程序可访问 JVM 之外的资源?( D )  
A. Java 应用程序    B. Java Applet    C. Java Servlet    D. 以上都不是
7. Java 源代码文件可使用下列哪个扩展名存储?( A )  
A. .java    B. .javac    C. .javax    D. .src
8. 子类也称为( C )。  
A. 内联类    B. 嵌套类    C. 派生类    D. 隐藏类
9. 以下哪个选项使 Java 得以在因特网上流行? A  
A. Java Applet    B. Java 应用程序    C. Java 字节码    D. 以上都不是
10. 以下哪个是关于 Java 的正确说法?( D )  
A. 仅具有编译器    B. 仅具有翻译器    C. 具有编译器和翻译器    D. 具有字节码阅读器

### 二、填空题

1. Java 程序有\_\_\_\_\_Application\_\_\_\_\_和\_\_\_\_\_Applet\_\_\_\_\_两种类型;
2. 现有一个 Java 类 MyJava, 其权限为 public, 则包含该类的 Java 程序文件名为\_MyJava.java\_; 用 javac 命令编译后, 得到类文件是\_MyJava.class\_; 要运行该程序, 在命令行环境下应输入 \_java MyJava\_ 命令;
3. 在 Eclipse 集成开发环境中开发 Java 程序, 先要点击\_\_File\_\_菜单创建 Java 项目, 然后点出 \_\_New→Class\_\_菜单在 Java 项目中新建 Java 类, 接着编写程序代码, 之后点出\_\_Run 菜单\_\_或点击\_\_Run\_按钮运行程序;
4. 通常, 主类程序都有一个\_\_main\_方法, 它是 Java 应用程序的入口点。

## 习题-第二章

### 一、选择题(答案可能不止一个)

1. 下列哪一个不是 Java 语言的关键字?( B )  
A. if    B. sizeof    C. private    D. null
2. 下列关于基本数据类型取值范围的描述中, 正确的是( BD )  
A. byte 类型的取值范围为-128~128    B. boolean 类型的取值为 true 和 false  
C. char 类型的取值范围为 0~65536    D. short 类型的取值范围为-32768~32767

3. 下列赋值语句中, 不正确的是( D )
- A. `boolean b=(5>6);`    B. `double d=0.0f;`    C. `float f=1.23;`    D. `char c='NBA';`
4. 表达式 `true || false && true || false && !true` 的值是( C )
- A. 0    B. 1    C. true    D. false
5. 已知: `int x=3; int y=6; z=x|y;` 则 z 的值为( B )
- A. 6    B. 7    C. 8    D. 9
6. 在一个应用程序中定义了数组 `a: int [] a={1,2,3,4,5,6};` 能够打印输出数组最后一个元素的语句是( B )
- A. `System.out.println(a[6]);`    B. `System.out.println(a[5]);`  
 C. `System.out.println(a[4]);`    D. `System.out.println(a[a.length]);`
7. 观察下列代码段, 其运行结果中包括( BCDF )选项
- ```
for (int i=0;i<2;i++){
    for (int j=0;j<3;j++){
        if (i==j)
            continue;
        System.out.println("i="+i+", j="+j);
    }
}
```
- A. i=0, j=0    B. i=0, j=1    C. i=0, j=2  
 D. i=1, j=0    E. i=1, j=1    F. i=1, j=2
8. 执行如下代码段后, 程序的输出结果是( C )
- ```
int x=1;
switch (x){
    case 1:
        ++x;
        System.out.print(x);
    case 5:
        x++;
        System.out.print(x);
        break;
    default:
        System.out.print(x);
}
```
- A. 1    B. 12    C. 23    D. 123    E. 234    F. 3

## 二、填空题

1. 请将程序所缺代码补充完整:

```
public __class__ App1{
    public __static__ void __main__(_String__ arg[]){
        __System____.out.println("你好!");
    }
}
```

2. 设 i、d、c 的数据类型分别为 int、double 和 char 型, 它们的值为: i=10, d=23.4, c='a'。请将表达式的值填入表格中(假定各表达式互不影响):

表 1 表达式对应的运行结果

表达式	运算结果	表达式	运算结果
r1=i++	10	r2=++i	11
i%3	3	d/2	11.7
d > 123.45	false	i-=0	false
i+20 <= d	false	(i<20) && (c!='A')	true
c>='a' && c<='z'	true	(int) (c+3)	100

### 三、思考题

- Java 中的数据类型可分哪两大类？基本类型包括哪 8 种？
  - 内置数据类型和引用数据类型；
  - 分别为整型 int，短整型 short，长整型 long，字节型 byte，布尔型 boolean，字符型 char，单精度浮点数 float，双精度浮点数 double
- Java 中的包名、类名、接口类、方法名的命名有什么约定俗成的规则？
 

包名：包的名字都是由小写单词组成；

类名：通常由名词组成，名称内所有单词第一个字母都大写，其他字母小写；

方法名：通常第一个单词为动词，第一个字母应该小写，如果有其他单词，则其他单词第一个字母大写，其余字母小写；

接口名：大小写规则与类名相似；

### 四、编程题

- 现有一函数：

$$y = \begin{cases} x & (x < 1) \\ 2x - 1 & (1 \leq x < 10) \\ 3x - 11 & (x \geq 10) \end{cases}$$

请编写一个程序，输入 x 值，输出 y 值。

```
public class Two {
    public static void main(String[] args) {
        Two t = new Two();
        System.out.print("Please input x:");
        Scanner in = new Scanner(System.in);
        double x = in.nextDouble();
        System.out.print("Result:");
        System.out.print(t.function1(x));
    }

    public double function1(double x){
        if (x<1){
            return x;
        }
        else if (x>=1&& x<=10){
            return 2*x-1;
        }
        else {
            return 3*x-11;
        }
    }
}
```

```

    }
}

}

```

2. 利用公式：
$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

求 $\pi$ 的近似值，直到最后一项的绝对值小于  $10^{-4}$  为止。(用 while 语句实现)

```

public class Four {
    public static void main(String[] args) {

        Four f = new Four();
        System.out.print(f.forPI());
    }

    public double forPI(){
        double n = 0;
        double sum = 0;
        do {
            n++;
            if (n%2==1)
                sum += 1/(2*n-1);
            else
                sum -= 1/(2*n-1);
            System.out.println(n);

        } while ((1/(2*n-1))>=0.0001);
        return sum*4;
    }
}

```

3. 使用 Math.random()方法产生 50 个 0—100 的随机整数，并将它们存放到一维数组 score 中，然后统计各分数段上的次数(即分别在 0~9, 10~19, 20~29, …, 80~89, 90~100 上的次数)，并将分类统计结果存入到数组 result 中，最后输出数组 result 各元素的值。

```

public class Six {

    public static void main(String[] args) {
        int[] score = new int[50];
        for (int i=0;i<50;i++){
            score[i] = (int)(Math.random()*100);
        }
        int[] result = new int[10];
        for (int i=0;i<10;i++){
            result[i] = 0;

```

```

    }
    classify(score,result);
    for (int r:result){
        System.out.println(r);
    }
}

```

```

public static void classify(int a[], int result[]){
    for (int x:a){
        if (x<=9){
            result[0]++;
            continue;
        }
        if (x<=19) {
            result[1]++;
            continue;
        }
        if (x<=29){
            result[2]++;
            continue;
        }
        if (x<=39){
            result[3]++;
            continue;
        }
        if (x<=49){
            result[4]++;
            continue;
        }
        if (x<=59){
            result[5]++;
            continue;
        }
        if (x<=69){
            result[6]++;
            continue;
        }
        if (x<=79){
            result[7]++;
            continue;
        }
        if (x<=89){
            result[8]++;
            continue;
        }
    }
}

```

```

        }
        if (x<=100) result[9]++;
    }
}
}

```

## 习题-第三章

### 一、选择题(答案可能不止一个)

1. 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( )

2. 对于构造方法, 下列叙述正确的是( ACD )

- A. 构造方法是类的一种特殊方法, 它的方法名必须与类名相同  
B. 构造方法的返回类型只能是 void 型  
C. 构造方法的主要作用是完成对类的对象的初始化工作  
D. 一般在创建新对象时, 系统会自动调用构造方法

3. 若需要定义一个类变量或类方法, 应使用哪种修饰符?( A )

- A. static                      B. package                      C. private                      D. public

4. 若在某一个类定义中定义有如下的方法: abstract void performDial(); 该方法属于( D )

- A. 本地方法                      B. 最终方法                      C. 静态方法                      D. 抽象方法

5. 不使用 static 修饰符限定的方法称为实例方法。下列关于实例方法的说法中, 正确的是( D )

- A. 实例方法可以直接调用父类的实例方法  
B. 实例方法可以直接调用父类的类方法  
C. 实例方法可以直接调用其他类的实例方法  
D. 实例方法不可以直接调用本类的类方法

6. 在 Java 中, 一个类可同时定义许多同名的方法, 这些方法的形式参数的个数、类型或顺序各不相同, 传回的值也可以不相同。这种面向对象程序特性称为( C )

- A. 隐藏                      B. 覆盖                      C. 重载                      D. Java 不支持此特性

7. 设有下面的一个类定义:

```

class AA{
    static void Show(){
        System.out.println("我喜欢 Java!");
    }
}
class BB{
    void Show(){
        System.out.println("我喜欢 C++!");
    }
}

```

若已经使用 AA 类创建对象 a 和 BB 类创建对象 b, 则下面哪一个方法调用是正确的?( AC )

- A. a.Show( )                      B. AA.Show( )                      C. AA.Show( )                      D. a.Show( )  
b.Show( )                      BB.Show( )                      b.Show( )                      BB.Show( )

8. 关于下列代码, 哪个语句的描述是正确的? ( C )

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

- A. 编译时将产生错误                      B. 编译时正确, 运行时将产生错误  
C. 输出零                                      D. 输出空

9. 下列方法定义中, 正确的是( A )

- A. `int x() { char ch='a'; return (int)ch; }`    B. `void x() { ...return true; }`  
C. `int x() { ...return true; }`                      D. `int x( int a, b){ return a+b; }`

10. 在某个类 A 中存在一个方法: `void getSort(int x)`, 以下能作为这个方法的重载的声明的是 ( AC )

- A. `void getSort(float x)`                      B. `int getSort(int y)`  
C. `double getSort(int x,int y)`                      D. `void get(int x,int y)`

11. 有一个类 A, 以下为其构造方法的声明, 正确的是( B )

- A. `void A(int x){...}`                      B. `A(int x){...}`  
C. `a(int x){...}`                                      D. `void a(int x){...}`

12. 符合对象和类关系的是( C )

- A. 人和老虎    B. 书和汽车    C. 楼和建筑物    D. 汽车和交通工具

## 二、填空题

1. 定义一个类包括类的声明和 定义 两个部分。

2. Java 中类成员的访问修饰符有以下几种: `private`, `public`, 缺省, `protected`; 类的访问修饰符只能使用缺省和 缺省。

3. 类的类型修饰符包含了缺省、`abstract` 及 `final` 三种, 分别对应具体类、抽象类和最终类三种类型。

4. `this` 除了可以用作对当前对象的引用外, 还可以使用 `this` 调用 当前类 构造方法以及访问 当前对象 的成员变量。

5. 如果子类中的某个方法的名字、返回值类型和参数列表与它的父类中的某个方法完全一样, 则称子类中的这个方法 覆盖 了父类的同名方法。

6. 创建一个名为 `MyPackage` 的包的语句是 `package MyPackage`, 该语句应该放在程序的位置为 源文件的开头。

## 三、思考题

1. 面向对象编程语言的基本特征是什么?

面向对象的特点是: 封装. 多态. 继承

2. 什么是对象? 什么是类? 对象与类的关系是什么?

所谓对象就是真实世界中的实体, 对象与实体是一一对应的, 也就是说现实世界中每一个实体都是一个对象, 它是一种具体的概念。

类是具备某些共同特征的实体的集合, 它是一种抽象的概念, 用程序设计的语言来说, 类是一种抽象的数据类型, 它是对所具有相同特征实体的抽象。

类与对象的关系: 类是对象的集合, 对象是类的实例; 对象是通过 `new className` 产生的通常情况下, 类称为模板或蓝本, 对象是由这些模板或蓝本产生的。

### 3. 类变量与实例变量的区别？

类变量是所有对象共有，其中一个对象将它值改变，其他对象得到的就是改变后的结果；而实例变量则属对象私有，某一个对象将其值改变，不影响其他对象；

实例变量是类实例化后，系统为其创建的一个类的实例变量的拷贝，即一个实例拥有一个实例变量。

类变量则是用 static 声明，系统为每个类变量分配一次存储空间。即所有的实例共享一个类变量。

### 4. 类方法与实例方法的区别？

类方法又叫静态方法，实例方法又叫非静态方法；

类方法可以直接通过类名调用，实例方法必需先初始化类的实例，然后通过类的实例才能调用。

### 5. 什么是方法的重载？

在同一个类中，有两个或两个以上具有相应名称，不同参数序列的方法。

### 6. 构造方法有什么作用？

它与所在类的名字完全相同，创建对象时会自动调用它，它用来初始化一个对象的内部状态。

### 7. Java 中引入包的优点是什么？

可以将不同类组织起来进行管理，借助于包可以方便地组织自己的类代码，并将自己的代码与别人提供的代码库分开管理。

### 8. 如何将需要的外部类引入程序中？如何引用包中的某个类？如何引用整个包？

导入该外部类所在的包；import 包名.类名；import 包名

## 四、编程题

### 1. 根据下面的要求实现圆类 Circle：

Circle 类的成员变量：radius 半径

Circle 类的成员方法：

- (1) Circle() 构造方法，将半径置为 0
- (2) Circle(double r) 构造方法，创建 Circle 对象时将半径初始化为 r
- (3) double getRadius() 获得圆的半径值
- (4) void setRadius(double r) 设置圆的半径
- (5) double getPerimeter() 获得圆的周长
- (6) double gerArea() 获得圆的面积

接着再定义一个类 CircleTest，用来调用 Circle 类，具体如下：

在 CircleTest 类的 main()方法中创建一个半径为 0 的对象 c1，再调用它的 setRadius()修改其半径值为 50，然后输出 c1 的周长和面积。

```
public class CircleTest {  
  
    public static void main(String[] args) {  
        Circle c1 = new Circle(0);  
        c1.setRadius(50);  
        System.out.println("圆的周长为：" + c1.getPerimeter());  
        System.out.println("圆的面积为：" + c1.getArea());  
    }  
}
```



```

class Circle{

    public double radius;

    /**
     * construction function
     * set the radius to 0 when initializing
     */
    public Circle(){
        radius = 0;
    }

    /**
     * construction function
     * set the radius to r
     * @param r radius of the circle
     */
    public Circle(double r){
        radius = r;
    }

    public double getRadius(){
        return radius;
    }

    public void setRadius(double r){
        radius = r;
    }

    public double getPerimeter(){
        return 2*Math.PI*radius;
    }

    public double getArea(){
        return Math.PI*radius*radius;
    }

}

```

- 2 . 先定义一个教师类 Teacher，它封装了：  
三个成员变量：

strNo	//表示工号
strName	//表示姓名

```

        intWorkAge                                //表示工龄
    四个成员方法：
        Teacher(String no, String name, int workAge)    //构造方法
        getTeacherNo()                                //获取教师工号
        getTeacherName()                              //获取教师姓名
        getTeacherWorkAge()                           //获取教师工龄

```

接着再定义一个类 TeacherTest，用来调用 Teacher 类，具体如下：

在 Teacher 类的 main()方法中创建一个工号为“0234”、姓名为“张三”、工龄为 35 年的对象 t1，然后输出 t1 的工号、姓名和工龄。

```

public class TeacherTest {
    public static void main(String[] args) {
        Teacher t1 = new Teacher("0234","张三",35);
        System.out.println("工号：" + t1.getTeacherNo());
        System.out.println("姓名：" + t1.getTeacherName());
        System.out.println("工龄：" + t1.getTeacherWorkAge());
    }
}

```

```

class Teacher{

    private String strNo;
    private String strName;
    private int intWorkAge;

    Teacher(String no, String name, int workAge){
        strNo = no;
        strName = name;
        intWorkAge = workAge;
    }

    public String getTeacherNo(){
        return strNo;
    }

    public String getTeacherName(){
        return strName;
    }

    public int getTeacherWorkAge(){
        return intWorkAge;
    }
}

```

## 习题-第四章

### 一、选择题(答案可能不止一个)

1. 现有 :String str1="Java"; String str2=new String("Java"); String str3="Java"; String str4=new String("Java"); 在下列表达式中, 结果为 true 的有( B )  
A. str1==str2;      B. str1==str3;      C. str2==str4;      D. str1.compareTo(str4);
2. 若有 :String str1="Java"; String str2=new String("Java"); String str3="Java"; String str4=new String("Java"); 在下列表达式中, 结果为 true 的有( ABC )  
A. str1.equals(str2);      B. str1.equals(str3);  
C. str2.equals(str4);      D. str1.compareTo(str3);
3. 现有 : String str="Tel:(020)87818998"; 若要得到第二个“8”的位置, 可调用 ( C )方法  
A. str.indexOf('8');      B. str.lastIndexOf('8');  
C. str.indexOf('8', str.indexOf('8')+1);      D. str.charAt('8');
4. 若有 : String str=new String("广州亚运会")转换为字节数组, 正确的语句( A )  
A. byte b[]=str.getBytes();      B. byte b[]=str.getChars();  
C. byte b[]=str.getBytes("GBK");      D. byte b[]=new Byte(str);
5. 以下表达式中, 正确的有( ABC )  
A. String str=""+"6";      B. "test".trim();      C. "nba".toUpperCase();  
D. String words[]="What is your name?".split();
6. 具有 append()方法的是( B )类  
A. String      B. StringBuffer      C. Math      D. StringTokenizer
7. 要产生[1,100]的随机整数,可通过( A )语来实现  
A. (int) (1 + Math.random() \* 100);      B. (int) (1 + Math.random() \* 99);  
C. (int) (1 + Math.random() \* 101);      D. (int) (Math.random() \* 101);
8. 要将字符串"678" 转换成 int 类型数据, 正确的语句是( C )  
A. Double.parseDouble("678");      B. Int.parseInt("678");  
C. Integer.parseInt("678");      D. Integer.valueOf("678");
9. 要获得当前系统时间的毫秒数, 正确的方法有( B )  
A. Date now=new Date(); long n=now.getTime();      B. System.currentTimeMillis();  
C. Calendar cal=new GregorianCalendar();long n=cal.getTimeInMillis();  
D. Calendar cal=new Calendar();long n=cal.getTime();
10. 假设系统时间为 2010 年 10 月 1 日 0 时 0 分 0 秒 0 毫秒, 则执行如下语句 :Calendar cal=new GregorianCalendar(); System.out.println(cal.get(Calendar.MONTH));后将输出( A )  
A. 9      B. 4      C. 5      D. 6

### 二、填空题

1. 调用\_\_\_\_Math.PI\_\_\_\_表达式可得到圆周率  $\pi$  的近似值 ;
2. 执行 StringBuffer buffer=new StringBuffer(); 语句后, buffer 的初始容量为\_\_16\_\_字节 ;
3. 若要创建一个指定时间 : 2008 年 8 月 8 日 20 时 0 分 0 秒 的对象, 可用\_ Calendar cal=new GregorianCalendar(2008,Calendar.August,8,20,0,0)\_语句来实现 ;
4. SimpleDateFormat 类的功能是\_ 对日期字符串进行解析和格式化输出\_\_。

### 三、思考题

1. 字符串字面变量与使用 new String("...")方法构造的字符串有什么不同 ?  
当使用字面变量形式创建字符串对象时, JVM 会对这个字面量进行检查, 如果字符串常量

池中存在相应内容的字符串对象的引用，则将这个引用返回，否则新的字符串对象被创建，然后将这个引用放入字符串常量池并返回该引用。

当使用 `new` 来构造字符串对象的时候，不管字符串常量池中有没有相同内容的对象的引用，新的字符串对象都会被创建。

2. 比较字符串时，使用 “==” 运算符与调用 `equals()` 方法有什么差异？

`equals()` 可以用来检测两个字符串是否相等；

`==` 运算符只能确定两个字符串是否存储在同一位置上。

3. `StringBuffer` 与 `String` 相比，有什么优点？

`StringBuffer` 允许对字符串内容进行修改；

`StringBuffer` 是线程安全的

4. 基本数据类型的包装类有什么用途？

(1) 提供一种机制将基本数据类型包装到对象中，从而使基本数据类型能够包含在为对象而保留的操作中；

(2) 为基本数据类型提供分类功能，大多数用于基本数据类型与 `String` 对象间相互转换。

5. 比较 `Date` 类与 `Calendar` 类的区别。

`Date` 类用来表示时间点；

`Calendar` 类描述了日历的一般属性，表示大家熟知的日历，更加通用。

#### 四、编程题

1. 编写一个 Java 程序，实现下列功能：

(1) 声明一个名为 `str` 的 `String` 对象，并使它的内容初始化为 “How are you?”；

(2) 打印整个字符串内容；

(3) 输出字符串的长度；

(4) 输出字符串的第一个字符；

(5) 输出字符串的最后一个字符；

(6) 输出字符串中的第一个单词(提示：先查找第一个空格位置，再获取该位置之前的子串)；

(7) 以大写字母方式输出字符串内容。

```
public class One {  
    public static void main(String[] args) {  
        String str = "How are you?";  
        System.out.println(str);  
        System.out.println(str.length());  
        System.out.println(str.charAt(0));  
        System.out.println(str.charAt(str.length()-1));  
        System.out.println(str.substring(0, str.indexOf(" ")));  
        System.out.println(str.toUpperCase());  
    }  
}
```

2. 编程实现：输入任意一个正整数，输出其对应的二进制、八进制、十六进制字符串。

```
public class Four {  
    public static void main(String[] args) {  
        System.out.print("请输入一个正整数：");  
        Scanner input = new Scanner(System.in);  
        int i = input.nextInt();  
        System.out.println("二进制：" + Integer.toBinaryString(i));  
    }  
}
```

```

        System.out.println("八进制：" + Integer.toOctalString(i));
        System.out.println("二进制：" + Integer.toHexString(i));
    }
}

```

3. 借助 SimpleDateFormat 类输出系统的当前时间，输出格式为：

北京时间：xxxx 年 xx 月 xx 日 xx 时 xx 分 xx 秒。

```

public class Five {
    public static void main(String[] args) {
        SimpleDateFormat df = new SimpleDateFormat("北京时间：yyyy 年 MM 月 dd 日
HH 时 mm 分 ss 秒");//设置日期格式
        System.out.println(df.format(new Date()));// new Date()为获取当前系统时间
    }
}

```

## 习题-第五章

一、选择题(答案可能不止一个)

1. 设有下面两个类的定义：

```

class Person {}
class Student extends Person {
    long id;           // 身份证号
    int score;          // 入学总分
    String name;        // 姓名
    int getScore(){return score;}
}

```

问：类 Person 和类 Student 的关系是( B )。

A.包含关系      B.继承关系      C.关联关系      D.无关系，上述类定义有语法错误

2. 设有下面的两个类定义：

```

class AA {
    void Show(){
        System.out.println("我喜欢 Java!");
    }
}
class BB extends AA {
    void Show(){
        System.out.println("我喜欢 C++!");
    }
}

```

则顺序执行如下语句后输出结果为：( A )

```

AA a;           BB b;
a.Show();       b.Show();

```

A.我喜欢 Java!                      B.我喜欢 C++!

我喜欢 C++!                      我喜欢 Java!  
C.我喜欢 Java!                      D.我喜欢 C++!  
我喜欢 Java!                      我喜欢 C++!

3. 在 Java 中，一个类可同时定义许多同名的方法，这些方法的形式参数的个数、类型或顺序各不相同，传回的值也可以不相同。这种面向对象程序特性称为( C )。

A、隐藏      B、覆盖      C、重载      D、Java 不支持此特性

4. 对于子类的构造方法说明，下列叙述中不正确的是( D )。

A.子类无条件地继承父类的无参构造方法。

B.子类可以在自己的构造方法中使用 super 关键字来调用父类的含参数构造方法，但这个调用语句必须是子类构造方法的第一个可执行语句。

C.在创建子类的对象时，将先执行继承自父类的无参构造方法，然后再执行自己的构造方法。

D.子类不但可以继承父类的无参构造方法，也可以继承父类的有参构造方法。

5. 给出下面代码：

```
1) class Parent {  
2) private String name;  
3) public Parent(){  
4) }  
5) public class Child extends Parent {  
6) private String department;  
7) public Child() {}  
8) public String getValue(){ return name; }  
9) public static void main(String arg[]) {  
10) Parent p = new Parent();  
11) }  
12) }
```

那些行将引起错误？( D )

A.第 3 行      B.第 6 行      C.第 7 行      D.第 8 行

name 变量是父类的私有变量，不会被继承

6. 类 Teacher 和 Student 是类 Person 的子类；

```
Person p;  
Teacher t;  
Student s;  
//p, t and s are all non-null.  
if(t instanceof Person) { s = (Student)t; }
```

最后一句语句的结果是：( C )

A.将构造一个 Student 对象

B.表达式是合法的

C.表达式是错误的

D.编译时正确，但运行时错误

7. 对于下列代码：

```
1) class Person {  
2) public void printValue(int i, int j) { //... }  
3) public void printValue(int i) { //... }  
4) }  
5) public class Teacher extends Person {  
6) public void printValue() { //... }
```

```

7) public void printValue(int i) {...}
8) public static void main(String args[]){
9) Person t = new Teacher();
10) t.printValue(10);
11) }

```

第 10 行语句将调用哪行语句？( D )

A.line 2          B.line 3          C.line 6          D.line 7

8. 对于下列代码：( )

```

public class Parent {
    public int addValue( int a, int b) {
        int s;
        s = a+b;
        return s;
    }
}
class Child extends Parent {
}

```

下述哪些方法可以加入类 Child? ( BC )

- A. int addValue( int a, int b ){// do something...}
- B. public void addValue (int a, int b ){// do something...}
- C. public int addValue( int a ){// do something...}
- D. public int addValue( int a, int b )throws MyException { //do something...}

**解释：**被重写的父类方法没有抛出异常，所以子类不能抛出异常 Java 重写规则，对于异常重写方法不能抛出新的异常或者比被重写方法声明的检查异常更广的检查异常。但是可以抛出更少，更有限或者不抛出异常。

9. 给出下面代码，那个选项是正确的？( A )

```

public class test{
    static int a[] = new a[10];
    public static void main(String args[]) {
        System.out.println(arr[10]);
    }
}

```

- A. 编译时将产生错误
- B. 输出零
- C. 编译时正确，运行时将产生错误
- D. 输出空

## 二、填空题

1. 继承 是一种软件重用形式，在这种形式中，新类获得现有类的数据和方法，并可增加新的功能。
2. 派生类构造方法可以通过关键字 super 调用基类构造方法。
2. 多态是指 类的属性或功能在各个子类中可以具有彼此不同的具体形态，在 Java 中有两种多态，一种是使用方法的 重载 实现多态，另一种是使用方法的 重写 实现多态。
3. Java 中所有类都是类 Object 的子类。
4. 被关键字 final 修饰的方法是不能被当前类的子类重新定义的方法。

## 三、思考题

1. 比较 Overload(方法重载)和 Override(方法覆盖)的区别。

答：Override 是重写：方法名称、参数个数，类型，顺序，返回值类型都是必须和父类方法一致的。

Overload 是重载：方法名称不变，其余的都是可以变更的。它的关系是同一个类，同一个方法名，不同的方法参数或返回值。

重写是父类与子类之间多态性的一种表现，而重载是一个类中多态性的一种表现。

2. 在创建派生类的对象的时候，基类与派生类中构造方法的调用顺序怎样？

答：先调用基类中的构造方法，接下来调用派生类中的构造方法。

3. 比较 this 和 super 的区别

答：super 表示超类，this 代表对象本身

关键字 this 一是引用隐式参数，二是调用该类的其他构造器；

关键字 super 则用来调用超类的方法或构造器

4. 什么是多态？面向对象程序设计为什么要引入多态的特性？使用多态有什么优点？

答：多态就是指程序中定义的引用变量所指向的具体类型和通过该引用变量发出的方法调用在编程时并不确定，而是在程序运行期间才确定，即一个引用变量到底会指向哪个类的实例对象，该引用变量发出的方法调用到底是哪个类中实现的方法，必须在由程序运行期间才能决定。

面向对象程序设计中，利用多态提高程序的抽象性，突出 Java 语言的继承性，大大提高了程序的扩展能力。

多态优点：可替换性、可扩充性、接口性、灵活性、简洁性。

#### 四、编程题

1. 编写一个学生类 Student，要求：

(1) 学生类 Student 属性有：

id : long 型，代表学号

name : String 类对象，代表姓名

age : int 型，代表年龄

sex : boolean 型，代表性别(其中：true 表示男，false 表示女)

phone : String 类对象，代表联系电话

(2) 学生类 Student 的方法有：

Student(long i, String n, int a, boolean s, long p)：有参构造方法，形参表中的参数分别初始化学号、姓名、年龄、性别和联系电话。

int getAge()()：获取年龄作为方法的返回值。

boolean getSex()()：获取性别作为方法的返回值。

long getPhone()()：获取联系电话作为方法的返回值。

public String toString()：以 姓名：联系电话 的形式作为方法的返回值

public class Student {

private long id; //学号

private String name;

private int age;

private boolean sex; //true 代表男生，false 代表女生

private String phone;

/\*\*

\* Construction function



```

    * @param i the Student id
    * @param n the Student name
    * @param a the Student age
    * @param s the Student sex
    * @param p the Student phone number
    */
    public Student(long i, String n, int a, boolean s, String p){

        id = i;
        name = n;
        age = a;
        sex = s;
        phone = p;

    }

    /**
     * @return the age of the student
     */
    public int getAge(){
        return age;
    }

    /**
     * @return the sex of the student
     */
    public boolean getSex(){
        return sex;
    }

    /**
     *
     * @return the phone number of the student
     */
    public String getPhone(){
        return phone;
    }

    /**
     *
     * @return the form of name:phone
     */
    public String toString(){

```

```

        return name + " : " + phone;
    }

    public static void main(String[] args) {
        Student s = new Student(1,"DixonShen",23,true,"18862141735");
        System.out.println(s.toString());
        System.out.println(s.getAge());
        if (s.getSex()==true)
            System.out.println("男");
        else
            System.out.println("女");
        System.out.println(s.getPhone());
    }
}

```

## 习题-第六章

### 一、选择题(答案可能不止一个)

1. 下面的是关于类及其修饰符的一些描述，正确的是：(ACD )。
  - A . abstract 类只能用来派生子类，不能用来创建 abstract 类的对象。
  - B . final 类不但可以用来派生子类，也可以用来创建 final 类的对象。
  - C . abstract 不能与 final 同时修饰一个类。
  - D . abstract 方法必须在 abstract 类中声明，但 abstract 类定义中可以没有 abstract 方法。
2. 在使用 interface 声明一个接口时，只可以使用( D )修饰符修饰该接口。
  - A.private                  B.protected                  C.private    protected                  D.public

3. 给定下列程序，哪些是正确的？( B )

```

public abstract interface Frobnicate{
    public void twiddle(String s);
}

```

- A. public abstract class Frob implements Frobnicate{
  - public abstract void twiddle(String s){}
- B. public class Frob extends Frobnicate{
  - public void twiddle(String s){}
  - public void twiddle(Integer i){}
- C. public abstract class Frob implements Frobnicate{}
- D. public class Frob extends Frobnicate{
  - public void twiddle(Integer i){}

### 二、填空题

1. 抽象方法是一种仅有方法声明，没有方法体的方法，该方法必须在抽象类中定义。
2. 如果某个子类没有覆盖抽象方法，则必须被声明为抽象类。
3. 如果一个类没有覆盖接口定义中的全部方法，则必须被声明为抽象类。

### 三、判断题

1. 接口是特殊的类，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。( √ )
2. 抽象方法必须在抽象类中，所以抽象类中的方法都必须是抽象方法。( × )
3. 一个类如果实现了某个接口，那么它必须重载该接口中的所有方法。( × )
4. 类在实现接口的方法时，必须显示地使用 public 修饰符。( × )
5. 实现一个接口，则在类中一定要实现接口中的所有方法。( × )
6. 抽象类不能实例化。( √ )

### 四、思考题

1. 为什么要使用抽象类？如何定义抽象类和抽象方法？抽象类中是否可以不包含抽象方法？  
使用抽象类，通过继承实现多态，提高重用性。

```
public abstract class ClassName{, abstract FunName();
```

抽象类中可以不包含抽象方法。

2. 什么是接口？为什么要定义接口？接口与抽象类有何异同？

接口是 java 中的一种引用数据类型。接口也成为一种特殊的、完全没有实现的类，与抽象类不同，接口中定义的所有方法都是抽象方法，并且接口中定义的所有域都是常量

3. 一个类如何实现接口？实现接口的类是否一定要重写该接口中的所有抽象方法？

当类实现接口的时候，类要实现接口中所有的方法。否则，类必须声明为抽象的类。

类使用 implements 关键字实现接口。在类声明中，Implements 关键字放在 class 声明后面。

实现一个接口的语法，可以使用这个公式：

```
Public class ClassName implements 接口名称[, 其他接口, 其他接口..., ...] ...
```

## 习题-第七章

### 一、选择题

1. 以下哪个是所有异常和错误的基类( B )  
A . Exception      B . Throwable      C . Error      D . RuntimeException
2. 以下哪个关键字用于抛出一个异常( A )  
A . throw      B . thorwn      C . throws      D . cacth
3. 以下哪一个为用户定义的异常类的正确声明( B )  
A . Class A extends B{...}  
B . Class A extends Exception{...}  
C . Class A extends B implements Exception{...}  
D . Class A extends extends Exception implements Exception {...}
4. 下列说法中错误的是( D )  
A . 子类可以与超类方法抛出相同类型的异常；  
B . 子类可以抛出超类方法所抛出的异常的子类；  
C . 子类中的方法抛出的异常数目可以少于超类中方法所抛出的异常数目，或者不需要抛出任何异常。  
D . 自定义异常不一定要继承 Exception 及其子类；

5. 下列说法中错误的是( D )

- A . try、catch、finally 这三个关键字都不能单独使用。
- B . try 语句 finally 组成 try...finally 结构。
- C . catch 语句可以有一个或多个。
- D . finally 语句可以有一个或多个。

## 二、填空题

- 1 . Throwable 类有两个直接类：\_Error\_ 指 JVM 系统内部错误、资源耗尽等严重情况;\_Exception\_ 指其它因编程错误或偶然的外在因素导致的一般性问题。
- 2 . 在 Exception 类中，返回当前异常对象信息的描述的方法是\_\_toString\_\_；返回当前异常对象信息的详细描述的方法是：\_getMessage\_；用来跟踪异常事件发生时执行堆栈的内容的方法是\_\_printStackTrace\_\_。
- 3 . 如果 Java 运行时系统找不到可以捕获异常的方法，则运行时系统将终止，相应的 Java 程序也将\_\_退出\_\_。
- 4 . 在 Java 中用关键字\_\_throws\_\_ 声明抛出的异常。
- 5 . 无论是否抛出异常都要执行的语句应用放到\_\_finally\_ 语句中。

## 三、思考题

1 . 关键字 throw 的作用是什么？关键字 throws 的作用是什么？

throw 的作用是由语句抛出异常，throw 总是出现在函数体中，程序会在 throw 语句后立即终止。

throws 的作用是声明方法可能抛出的异常，总是出现在一个函数头中，用来表明该成员函数可能抛出的各种异常。

2 . 假设 statement2 引起以下 try-catch 代码块的异常。

```
try {  
    statement1;  
    statement2;  
    statement3;  
} catch (Exception1 e1) {}  
catch (exception2 e2) {}  
statement4;
```

回答以下问题：

- (1) statement3 将被执行吗？
- (2) 如果没有捕获异常，还执行 statement4 吗？
- (3) 如果在 catch 子句中捕获到异常，statement4 将被执行吗？

答：(1) 不会被执行

(2) 执行

(3) 不一定

3 . 假设 statement2 引起以下 try-catch 代码块的异常。

```
try{  
    statement1;  
    statement2;  
    statement3;  
} catch (Exception1 e1) {}
```

```
catch(exception2 e2){}
catch(exception3 e3){throw e3;}
finally{ statement4;}
Statement5;
```

回答以下问题：

- (1) 如果没有捕获异常，还执行 statement5 吗？
- (2) 如果异常类型是 statement3，statement4 执行吗？statement5 将被执行吗？

答：(1) 执行

(2) statement4 会执行，statement5 不确定会不会被执行。

## 习题-第八章

### 一、判断题

1. Java 泛型的本质是参数化类型。( √ )
2. Java 数组的协变性(covariant)是指如果类 Base 是类 Sub 的基类，那么 Base[]就是 Sub[]的基类。Java 泛型也是协变的，就是说如果类 Base 是类 Sub 的基类，那么 List<Base>就是 List<Sub>的基类。( × )
3. 根据 Java 集合框架中的定义，Java Set 集合中的元素是可以重复的。( × )
4. Java List 集合与 Java 数组相似，Java List 集合的长度也是固定的。( × )
5. List 主要特征是其元素以线型方式存储，集合中可以存放重复的对象。( √ )
6. 泛型类的构造方法仍然和普通类的构造方法形式是一样的，不带参数。( √ )
7. Java 集合中 Map 是一种把键对象和值对象进行映射的集合，他的每一个元素都包含一个键对象，一个值对象，键对象相当于值对象的索引，而且值对象不可以是 Map 类型的。( × )
8. Java 集合中 ArrayList 是长度可变的数组。可以对元素快速的随机访问。( √ )
9. Java 集合中 Set 最为主要的特征是集合中的对象不按照特定的方式排序，但是可以有重复的对象。( × )
10. Java 语言在使用泛型类时必须指定泛型类型参数。( × )

### 二、填空题

1. 定义泛型时，泛型参数放到尖括号中，其功能类似于方法的形参。在实例该类时要为参数传入实际参数，泛型参数的实际参数只能是类类型，包括自定义的类型，而不能是\_\_简单\_\_类型。
2. Java 泛型就是在定义\_\_类\_\_、\_\_接口\_\_时指定类型形参，类型形参将在声明该类型变量、创建该类对象时确定，即传入类型实参。Java 的参数化类型称为泛型(Generic Type 或 Generic)。
3. Java SE 5.0 改写了集合框架中的全部接口和类，提供泛型支持，在声明集合变量、创建集合对象时，传入\_\_类型参数\_\_。
4. 如果 Foo 是 Bar 的一个子类型，G 是一个 Java 泛型，那么，G<Foo>\_\_不是\_\_G<Bar>的子类，Foo[]\_\_是\_\_Bar[] 的子类。
5. 泛型可以被继承或实现，但是泛型作为父类或父接口时，不能再包含有\_\_类型实参\_\_，而只能包含有泛型实参。
6. Java 编译器允许在使用泛型是不传入实际参数，在使用 Java 编译器 javac 编译时会产生\_\_泛型\_\_警告。可以通过使用编译器的 -Xlint:unchecked 选项查看详细信息。

7. 泛型的类型参数的实参只能是类类型, 不能是 简单类型。
8. Java 集合类和数组不同, 数组元素既可以是 基本数据 类型, 也可以是对象类型; 而集合中的元素只能是 对象引用。
9. Java 集合类主要由 2 个接口派生, 一个是 Collection, 另一个是 Iterator。
10. Java 集合可分成 3 类, 其中 Set 集合中的元素不能 重复; List 集合非常像 数组; Map 集合的元素由 2 个值组成, 一个是 Key 对象, 一个是 Value 对象。
11. Iterator 接口是 Java 集合框架中的成员, 和 Clection、Map 集合不一样, Iteratoer 接口主要用于 遍历 Clection 集合中的元素。Iterator 又称 迭代器。
12. Java 中除了可以使用 Iterator 接口的方式访问 Clection 集合类中的元素, Java 1.5 还提供 foreach 循环来迭代访问 Clection 集合中的元素。

### 三、编程题

1. 在程序中声明一个 HashSet 集合类的对象 hset, 向 hset 中添加字符串元素"hello", "world", "你好", "Java", "Java 泛型"和"Java 集合"。用 hset 的 iterator()方法生成迭代器 it, 使用 it 的 hasNext()方法迭代输出集合中的元素。
2. 在 main()方法中分别声明 HashSet、ArrayList 和 HashMap()集合, 向集合中添加若干元素, 然后输出各集合中所有元素。查看 Java API 文档, 找到各种集合添加元素的方法, 注意 Map 集合的元素为一个“键-值对”。Map 集合输出元素的方法也不同 Clection 集合类型。

## 习题-第九章

### 一、选择题(答案可能不止一个)

1. 下列选项中, 哪些属于 File 类的功能?( ABC )
- A. 查看文件大小      B. 创建目录      C. 删除文件      D. 读取文件中的数据
2. File 类可以表示( AB )
- A. 文件      B. 目录      C. 输入流      D. 输出流
3. 要在磁盘上创建一个文件, 可以使用( D )类的实例。
- A. File      B. FileOutputStream      C. RandomAccessFile      D. 以上全对
4. 下列选项中, 哪些类使用了缓冲区技术?( C )
- A. FileInputStream      B. FileReader      C. BufferedOutputStream      D. PrintWriter
5. InputStreamReader 的直接父类是( B )
- A. Object      B. Reader      C. InputStream      D. Writer
6. DataOutputStream 实现了( B )接口
- A. DataInput      B. DataOutput      C. Serializable      D. RandomAccessFile
7. 下列选项中, 哪些不是 BufferedReader 的方法?( C )
- A. read()      B. readLine()      C. newLine()      D. close()
8. 要向 myfile.txt 文件写入内容, 可选用的类有哪些?( BCD )
- A. FilterOutputStream      B. FileOutputStream      C. FileWriter      D. PrintWriter

### 二、填空题

1. 在文本编辑器中, 通常包含“文件”菜单, 请指出下列菜单项中, 使用的流是属于输入流还是输

出流？应如何操作？

(1)“新建”：\_\_输出流\_\_；

(2)“打开”：\_\_输入流\_\_；

(3)“保存”：\_\_输出流\_\_；

(4)“另存为”：\_\_输出流\_\_。

2．要创建 RandomAccessFile 类对象，以只读方式打开 c:\mymobile\blacknames.txt，正确的语句是\_\_RandomAccessFile raf = new RandomAccessFile(" c:\mymobile\blacknames.txt","r");\_\_。

3．标准输入流对象 System.in、标准输出流对象 System.out 是\_\_InputStream 与 PrintStream\_\_类的实例。

4．readObject()和 writeObject()分别是\_\_ObjectInputStream\_\_类和\_\_ObjectOutputStream\_\_类的方法。

5．InputStreamReader、OutputStreamWriter 类分别实现\_\_从字节流到字符流的转换\_\_和\_\_从字符流到字节流的转换\_\_功能。

6．FileInputStream 实现对磁盘文件的读取操作，在读取字符的时候，它一般与\_\_InputStreamReader\_\_类和\_\_BufferedReader\_\_类一起使用。

### 三、思考题

1．用 File 表示的文件或目录是否一定要存在？若不存在，应如何创建？

不一定。若不存在，使用 createNewFile()方法创建新文件。

2．比较下列三组概念的差异：

(1)输入流与输出流；

输入流表示数据从输入设备（如键盘、磁盘、网络）流向内存，输出流则是数据从内存流向输出设备（如屏幕、磁盘、网络）。

(2)字节流与字符流；

字节流是以字节为基本单位，将数据看作是由一个个字节构成的序列，可处理任何类型的数据（包括二进制数据和文本信息），这是较低层次的操作；字符流是以字符为基本单位，将数据看作是一个个字符组成的序列，适用于字符、文本类型数据的操作。

(3)文件字节流与文件字符流。

文件字节流，FileInputStream 与 FileOutputStream 两个类是文件读写的两个类

文件字符流，FileReader 与 FileWriter 是文件读写的字符流类

3．如何理解 I/O 流操作中的“逐层包装”思想？

逐层包装思想，用一个已存在的流来构造另一个流，也就是说，一个流的构造方法的参数是另一个流，之后又可以在这个新流的基础上再构造别的流，如此下去，直至满足要求为止，包装的目的是实现在更高层次上对数据的简便操作。

4．FilterInputStream/FilterOutputStream 有哪些典型子类？请各举出 2 个例子，并说明它们的功能。

FilterInputStream 子类

(1) DataInputStream，读取基本类型数据和字符串

(2) BufferedInputStream，带缓冲区的输入流，能够减少访问磁盘的次数，提高文件读取性能

FilterOutputStream 子类

(1) DataOutputStream，写入基本数据类型和字符串

(2) BufferedOutputStream，带缓冲区的输出流，能够提高文件的写入效率

5．当创建 FileInputStream 对象时，如果对应的文件不存在，会抛出何种异常？

FileNotFoundException

6. 有人说“RandomAccessFile 类既是输入流，又是输出流”，你是否同意这种说法？请说明原因。

RandomAccessFile 不是输入流、输出流，它是处理文件的另一种类，可集读写操作于一身，是 Object 的直接子类。

#### 四、编程题

1. 利用 File 类和递归方法，列出指定目录下的文件及各级子目录包含的内容。

```
import java.io.File;

public class ListDirectory {
    public static void main(String[] args) {
        String fileName = "D:\\downloads" + File.separator;
        File f = new File(fileName);
        print(f);
    }

    public static void print(File f) {
        if (f!=null){
            if (f.isDirectory()){
                File[] fileArray = f.listFiles();
                if (fileArray!=null){
                    for (int i=0;i<fileArray.length;i++){
                        print(fileArray[i]);
                    }
                }
            }
            else {
                System.out.println(f);
            }
        }
    }
}
```

2. 使用字节流实现文件的拷贝，要求一次能够读写多个字节(如：512 字节等)。

```
public class CopyFile {
    public static void main(String[] args) throws IOException{
        long len = 0L;
        byte[] b = new byte[1024];
        FileInputStream fin = new FileInputStream("src/Ch9/a.txt");
        FileOutputStream fout = new FileOutputStream("src/Ch9/b.txt");
        try{
            int i = fin.read(b);
            while(i!=-1){
                fout.write(b,0,i);
                len += i;
                i = fin.read(b);
            }
        }
    }
}
```



```

    }
    System.out.println("从源文件复制了" + len + "字节到目标文件，文件复制完毕！");
} catch (IOException e){
    System.out.println(e.getMessage());
} finally {
    fin.close();
    fout.close();
}
}
}
}

```

## 习题-第十章

### 一、选择题

- 容器 Panel 和 Applet 缺省使用的布局编辑策略是？( B )  
A、BorderLayout    B、FlowLayout    C、GridLayout    D、CardLayout
- AWT 组件，从各组件的本来功能角度考虑，最好使用哪种组件：  
A、TextArea                      B、Panel  
C、Applet                          D、Canvas
- 下列属于图形用户界面元素的有：( C )  
A. Applet              B. 窗口              C. 按钮              D. 线程
- 如果希望所有的控件在界面上均匀排列，应该使用下列那种布局管理器？( B )  
A. BorderLayout                      B. GridLayout  
C. BorderLayout                      D. FlowLayout
- 下列用户图形界面组件在软件安装程序中常见的是 ( B )  
A、滑块                                  B、进度条  
C、对话框                                D、标签
- 每个使用 Swing 组件的程序必须至少有一个 ( B )  
A. 按钮              B. 顶层容器              C. 标签              D. 菜单

### 二、填空题

- Java 中有提供丰富的图形功能的 java.awt 和 javax.swing 两个包。AWT：是 Abstract Window Toolkit 的缩写，中文意思是“抽象窗口工具包”；Swing：又称为轻量级组件，它是 Java 的改进版本。
- AWT 包中的组件可分为三大类：容器类，如窗口(Window)等；独立组件，如标签(Label)等；菜单类，如菜单条(MenuBar)等。
- Java 把 Component 类的子类或间接子类创建的对象称为 一个组件，把 Container 的子类或间接子类创建的对象称为一个容器。
- 在 Swing 用户界面的程序设计中，容器 可以被添加到其它容器中去。
- 对话框有 模态 和 非模态 两种。这两种对话框的区别是：当 模态 对话框弹出时，只有关闭此对话框才能继续进行其它窗口的操作；当 非模态 对话框弹出时，即使不关闭，也可以对后面的窗口进行操作。

### 三、思考题

1. Java GUI 中有哪些容器类?哪些是组件类?

Frame, Window, Panel, Dialog, ScrollPanel, JFrame, JApplet, JDialog, JWindow, JPanel, JScrollPane, 其中 Frame, JApplet, JDialog, JWindow, JPanel, JScrollPane 是组件类

2. Swing 和 AWT 是什么关系?Swing 有什么特点?

AWT 是 Java 的早期版本, 本身存在很多不完善的地方, Swing 是一个全新的 GUI 用户界面库, 提供许多比 AWT 更好地屏幕显示元素。Swing 组件都是 AWT 的 Container 类的直接子类或间接子类  
Swing 特点: (1) 拥有丰富便捷的用户界面元素集合; (2) 对底层平台以来很少; (3) 给予不同平台的用户一致的感觉。

3. Java GUI 中有哪几种常用的布局方式, 各有什么特点?

(1) FlowLayout, Panel、Applet 的缺省布局管理器。器组件的放置规律是从上到下、从左到右进行放置;

(2) BorderLayout 是 Window、Frame 和 Dialog 的缺省布局管理器。BorderLayout 吧容器分成五个区域: North、South、East、West 和 Center, 每个区域只能显示一个组件, 若要显示多个组件, 可利用 Panel 实现嵌套。

(3) GridLayout 是容器中各个组件呈网格状布局, 平均占据容器的空间;

(4) CardLayout 能够帮助用户处理两个以至更多的成员共享同意显示空间, 把容器分成许多层, 每层的显示空间占据整个容器的大小, 但是每层只允许放置一个组件。