# 附录B　参 考 答 案

## 第1章：Java简介——实践与练习

### 1.7.1 填空题

1．Java源程序文件的后缀是　*.java　，Java字节码文件的后缀名称是　*.class　。

2．Java程序实现可移值性，依靠的是　JVM　。

3．Java语言的三个分支是：　Java SE　、　Java ME　、　Java EE　。

4．Java程序由　类　组成，如果Java使用　publicclass声明类，则文件名称必须与类名称一致。

5．Java执行是从　main()　方法开始执行的，此方法的完整定义是　public static void main(String args)。

6．Java类名的每个单词的首字母通常要求　大写　。

### 1.7.2 选择题

1．推出Java语言的公司是　B　。

A．IBM　　B．SUN　　C．Apple　　D．Microsoft

2．下面的哪个环境变量是java解释时所需要的？　B　

A．path　　B．classpath　　C．JAVA_HOME　　D、TEMP

### 1.7.3 判断题

1．Java语言属于编译型的开发语言。　（ × ）

2．Java Application程序不是由main()方法开始执行的。　（ × ）

### 1.7.4 简答题

1．简述Java实现可移值性的基本原理。

答：Java属于编译型和解释型的编程语言，所有的*.java程序必须编译为*.class文件后才可以在电脑上执行，而执行*.class文件的电脑并不是一台真实的电脑，而是利用软件和硬件模拟

出来的一台虚拟电脑，称为 Java虚拟机，而针对不同的操作系统平台，有不同版本的 Java虚拟机，即由 Java虚拟机去适应不同的操作系统，只要 Java虚拟机的支持没有改变，同一个*.class可以在不同的平台上运行。

Note

2．简述 Java语言的三个程序分支。

答：Java SE（Java标准版）：提供基础的 Java类库及平台支持。

Java ME（Java微型版）：提供嵌入式开发支持。

Java EE（Java企业版）：提供企业平台支持。

3．简述 Java中 path及 classpath的作用。

答：Path：是操作系统的环境属性，指的是可以执行命令的程序路径。

CLASSPATH ：是所有*.clas文件的执行路径，java命令执行的时候将利用此路径加载所需要的*.clas文件。

4．简述 Java语言的主要特点。

答：Java语言的主要特点如下：

（1）Java语言是一门面向对象语言，且语法足够简单；

（2）Java避免了 C/C++ 中复杂的指针关系，而使用了更为简单的引用方式来进行内存传递；

（3）Java是为数不多的支持多线程开发的编程语言；

（4）Java提供了自动的垃圾收集机制，可以定期释放出无用的垃圾空间；

（5）Java语言的安全性较高；

（6）Java最大的特点是具备可移植性，即：同一个程序在不同的操作系统上都可以运行。

## 1.7.5 编程题

1．在屏幕上输出：“我喜欢学习 Java”的信息。

答案：

```
public class TestDemo {
    public staticvoid main(String[] args) {
        System.out.println("我喜欢学习 Java");
    }
}
```

2．在屏幕上打印出以下图形：

```
**************************************
*********    Java 程序设计 *********
**************************************
```

答案：

```
public class TestDemo {
    public staticvoid main(String[] args) {
        System.out.println("**************************************");
        System.out.println("*********Java 程序设计      *********");
        System.out.println("**************************************");
    }
```

```
}
```

Note

# 第 2 章　简单 Java程序——实践与练习

Note

## 2.9.1　填空题

1．Java中的标识符组成原则：___由字母、数字、下划线、$所组成，其中不能以数字开头，不能是 Java的关键字___。

2．__assert__关键字是在 JDK 1.4时加入的，__enum__关键字是在 JDK 1.5时加入的。

3．列举出已经知道的 5 个关键字：__public、static、void、int、double__。

4．Java注释分为以下三种：__单行注释（//）__、__多行注释（/*…*/）__、__文档注释__。

5．Java中使用__int__关键字，可以定义一个整型数据。

6．在一个 Java源文件中定义了 3 个类和 15 个方法，编译该 Java源文件时会产生__3__个字节码文件，其扩展名是__*.class__。

7．布尔型数据类型的关键字是__boolean__，有__true__和__false__两种取值。

8．整型数可以采用__byte__、__short__、__int__和__long__4 种类型表示。

9．根据占用内存长度的不同将浮点型分为__float__和__double__两种。

10．Java程序结构分为：__顺序结构__、__分支结构__、__循环结构__3 种。

11．逻辑表达式：true&&false&&true的结果是__false__。

12．逻辑表达式：!true||false的结果是__false__。

13．在方法中可以使用__return__语句来结束方法的执行。

14．在 Java中数组排序的方法是_____。

15．方法中的__void__关键字用来表示方法不返回任何值。

## 2.9.2　选择题

1．下面哪些标识符是正确的？__D__

A．class　　B．hello world　　C．123$temp　　D．Demo

2．下面哪些关键字是 Java中未使用到的关键字？__A、B__（多选）

A．const　　B．goto　　C．int　　D．assert

3．public static void main方法的参数描述是：__A__。

A．String args[]　B．int[] args　　C．Strings args[]　D．String args

4．下面说法正确的是__C__。

A．Java程序的源文件名称与主类（puublic class）的名称相同，后缀可以是.java或.tx等

B．JDK 的编译命令是 java

C．一个 java源文件编译后可能产生多个 class文件

D．在命令行编译好的字节码文件，只需在命令行直接输入程序名即可运行该程序

5. 下面说法不正确的是____A____。
   A. Java语言是面向对象的、解释执行的网络编程语言
   B. Java语言具有可移植性，是与平台无关的编程语言
   C. Java语言可对内存垃圾自动收集
   D. Java语言执行时需要 Java的运行环境
6. 下面____A____不是 Java的关键字。
   A. integer　　B. double　　C. float　　D. char
7. 在 Java中，字节数据类型的关键字和默认值是____A____。
   A. byte和 0　　B. byte和 1　　C. boolean和 true　　D. boolean和 false
8. 3.15E2表示的数据是____C____。
   A. 3.15 × 2　　B. 3.15 × $10^{-2}$　　C. 3.15 × $10^{2}$　　D. 0.315 × $10^{-2}$
9. 程序 System.out.println("1 + 1 = " + 1);的输出结果是____C____。
   A. 1　　B. 1 + 1 = 2　　C. 1 + 1 = 11　　D. 2
10. 程序 System.out.println(10 / 3);的输出结果是____B____。
    A. 1　　B. 3　　C. 3.3　　D. 3.33333
11. 执行下面的语句后，a、b、c 的值分别是____C____。

```
int a = 2 ;
int b = (a++) * 3 ;
int c = (++a) * 3 ;
```

    A. 2、6、6　　B. 4、9、9　　C. 4、6、12　　D. 3、9、9
12. 以下的____B____能正确表示 Java语言中的一个整型常量。
    A. 35.d　　B. -20　　C. 1,234　　D. "123"
13. 下面的数据类型____D____是 float型
    A. 33.8　　B. 129　　C. 89L　　D. 8.6F
14. 下列关于自动类型转换的说法中，正确的一个是____B____。
    A. int类型数据可以自动转换为 char类型数据
    B. char类型数据可以被自动转换为 int类型数据
    C. boolean类型数据不可以做自动类型转换，但是可以做强制转换
    D. long类型数据可以被自动转换为 short类型数据
15. 一个方法在定义过程中又调用自身，这种方法称为____B____。
    A. 构造方法　　B. 递归方法　　C. 成员方法　　D. 抽象方法

## 2.9.3 判断题

1. 变量的内容可以修改，常量的内容不可修改。（ √ ）
2. goto是 Java中未使用到的关键字。（ √ ）
3. enum 关键字是在 JDK 1.4版本中增加的。（ × ）
4. 使用 public class定义的类，文件名称可以与类名称不一致。（ × ）
5. 主方法编写：public void main(String。arg)（ × ）
6. 字符$不能作 Java标识符的第一个字符。（ × ）
7. System.out.print()输出后是不加换行的，而 System.out.println()输出后是加换行的。（ × ）

8. 使用 break语句可以跳出一次循环。 ( × )
9. byte的取值范围是：0~255。 ( × )
10. int和 double进行加法操作，int会自动转换为 double类型。 ( × )
11. 使用“&”操作时，如果第一个条件是 false 则后续的条件都不再判断。 ( × )
12. 使用“&&”操作时，如果第一个条件是 false 则后续的条件都不再判断。 ( √ )
13. 使用“|”操作时，如果第一个条件是 true 则后续的条件都不再判断。 ( × )
14. 使用“||”操作时，如果第一个条件是 true 则后续的条件都不再判断。 ( √ )
15. 定义多个同名方法时，可以依靠返回值区别同名方法。 ( × )

Note

## 2.9.4 简答题

1. 请解释常量与变量的区别。

答：常量就是一个固定的数值，是不可改变的，如数字 1、2 就是一个整型的常量。

变量是利用声明的方式，将内存中的某个内存块保留下来以供程序使用。可以声明的数据类型为整型、字符型、浮点型或是其他数据类型，作为变量的保存之用。变量在程序语言中扮演了最基本的角色。变量可以用来存放数据，而使用变量之前必须先声明它的数据类型。

2. 解释方法重载的概念，并举例说明。

答：方法重载指的是多个方法的方法名称相同，但是方法中的参数类型及个数不同。

代码说明：

```
public static int add(int x, int y) {
     return x + y;
}
public static int add(int x, int y, int z) {
     return x + y + z;
}
public static double add(double x, double y) {
     return x + y;
}
```

## 2.9.5 编程题

1. 打印出 100~1000 范围内的所有“水仙花数”，所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数本身。例如，153 是一个“水仙花数”，因为 153=1 的三次方＋5 的三次方＋3 的三次方。

答案：

```
public class TestDemo {
     public staticvoid main(String[] args) {
          int i, j, k;
          for (int x = 100; x < 1000; x++) {
               i = x / 100;                    /计算百位数字
               j = (x / 10) % 10;              /计算十位数字
               k = x % 10;                     /计算个位数字
               if(x == i * i * i + j * j * j + k * k * k) {
```

```
                System.out.print(x + "");
            }
        }
    }
}
```

程序运行结果：

153、370、371、407、

2．通过代码完成两个整数内容的交换。

答案：

实现一：引用第三方变量

```
public class TestDemo {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        int temp = x;
        x = y;
        y = temp;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
}
}
```

实现二：利用数学计算完成

```
public class TestDemo {
    public static void main(String[] args) {
        int x = 10;
        int y = 20;
        x += y ;
        y = x - y;
        x = x - y ;
        System.out.println("x = " + x);
        System.out.println("y = " + y);
}
}
```

程序运行结果：

x = 20

y = 10

3．判断某数能否被 3，5，7 同时整除。

答案：

```
public class TestDemo {
    public static void main(String[] args) {
        int data = 105;
        if(data % 3 == 0 && data % 5 == 0 && data % 7 == 0) {
            System.out.println(data 可以同时被 3、5、7 整除。");
        }else {
            System.out.println(data 不可以同时被 3、5、7 整除。");
        }
    }
}
```

程序运行结果：

105 可以同时被 3、5、7 整除。

4．编写程序，分别利用 while、do…while和 for循环求出 100~200 的累加和。

答案：

实现一：使用 while循环

```
public class TestDemo {
    public static void main(String[] args) {
        int sum = 0;
```

```
            int x = 100;
            while (x <= 200) {
                sum += x;
                x++;
            }
            System.out.println("累加结果：" + sum);
        }
}
```

Note

实现二：使用 do…while循环

```
public class TestDemo {
    public static void main(String[] args) {
        int sum = 0;
        int x = 100;
        do {
            sum += x;
            x++;
        } while (x <= 200);
        System.out.println("累加结果：" + sum);
    }
}
```

实现三：使用 for循环

```
public class TestDemo {
    public static void main(String[] args) {
        int sum = 0;
        for (int x = 100; x <= 200; x++) {
            sum += x;
        }
        System.out.println("累加结果：" + sum);
    }
}
```

# 第 3 章　面向对象——实践与练习

## 3.34.1　填空题

1．面向对象的三大特征：____封装____、____继承____、____多态____。

2．类由____属性____和____方法____组成。

3．____new____运算符的作用是根据对象的类型分配内存空间。当对象拥有内存空间时，会自动调用类中的____构造方法____为对象____实例化____。

4．使用____private____修饰的类成员称为私有成员。私有成员只能在____类____中使用。

5．构造方法的名称与____类名称____相同。

6．____private____关键字可以让类中的属性和方法对外部不可见。

7．this关键字可以调用本类中的___属性___、___方法___、___构造方法___，调用___构造方法___时必须放在___构造方法___的首行。

8．Java中通过___extends___关键字实现继承。

9．一个类只能继承___一___个父类，但能实现___多个___接口。

Note

10．___Object___类是所有类的父类，该类中判断两个对象是否相等的方法是___public boolean equals(Object oth)___，取得对象完整信息的方法是___public String toString()___。

11．Integer类是对___int___基本数据类型的封装。Float类是对___float___基本数据类型的封装。Double 类是对___double___基本数据类型的封装。字符类 Character是对___char___基本数据类型的封装。

12．当子类中定义的方法与父类方法同名且参数类型及个数、返回值类型相同时，称子类方法___覆写___父类方法，子类默认使用___本类已经覆写___方法，使用父类的同名方法，必须使用___super___关键字说明。

13．当子类定义的成员变量与父类的成员变量同名时，称子类___覆盖___父类的成员变量，子类默认使用___本类___属性。使用父类的同名成员变量，必须用___super___关键字说明。

14．如果子类定义了构造方法，在创建子类对象时首先默认调用___父类无参构造方法___，然后再调用本类的构造方法。

15．在 Java中，数组排序的方法是___java.util.Arrays.sort()___。

## 3.34.2 选择题

1．如果希望方法直接通过类名称访问，在定义时要使用的修饰符是___A___。

A．static　　B．final　　C．abstract　　D．this

2．如果类中没有定义构造方法，系统会提供一个默认的构造方法。默认构造方法的特点是___C___。

A．无参数有操作　　B．有参数无操作

C．即无参数也无任何操作　　D．有参数有操作

3．有一个类 Demo ，对与其构造方法的正确声明是___B___。

A．void Demo(int x){}　　B．Demo(int x){}

C．Demo Demo(int x){}　　D．int Demo(){}

4．以下关于面向对象概念的描述中，不正确的一项是___C___。

A．在现实生活中，对象是指客观世界的实体

B．程序中的对象就是现实生活中的对象

C．在程序中，对象是通过一种抽象的数据类型来描述的，这种抽象数据类型称为类（class）

D．在程序中，对象是一组变量和相关方法的集合

5．下列哪一项不属于面向对象程序设计的基本要素？___D___

A．类　　B．对象　　C．方法　　D．安全

6．下列程序的执行结果是___A___。

```
public class TestDemo {
    public void fun() {
        staticint i = 0;
```

```
        i++;
        System.out.println(i);
    }
    public staticvoid main(String args[]) {
        Demo d = new  Demo();
        d.fun();
    }
}
```

Note

A．编译错误　　B．0　　C．1　　D．运行成功，但不输出

7．顺序执行下列程序语句后，则 b 的值是___C___。

```
String str = "Hello" ;
String b = str.substring(0,2) ;
```

A．Hello　　B．hello　　C．He　　D．null

8．不能直接使用 new 创建对象的类是___B___。

A．静态类　　B．抽象类　　C．最终类　　D．公有类

9．为类定义多个名称相同、但参数的类型或个数不同的方法的做法称为___B___。

A．方法重载　　B．方法覆写　　C．方法继承　　D．方法重用

10．定义接口的关键字是___C___。

A．extends　　B．class　　C．interface　　D．public

11．现在有两个类 A、B，以下描述中表示 B 继承自 A 的是___D___。

A．class A extends B　　B．class B implements A

C．class A implements　　D．class B extends A

12．下面关于子类调用父类构造方法的描述正确的是___C___。

A．子类定义了自己的构造方法，就不会调用父类的构造方法

B．子类必须通过 super关键字调用父类有参的构造方法

C．如果子类的构造方法没有通过 super调用父类的构造方法，那么子类会先调用父类中无参构造方法，之后再调用子类自己的构造方法

D．创建子类对象时，先调用子类自己的构造方法，然后再调用父类的构造方法

13．假设类 X 是类 Y 的父类，下列声明对象 x 的语句中不正确的是___D___。

A．X x = new X() ;　　B．X x = new Y() ;

C．Y x = new Y() ;　　D．Y x = new X() ;

14．编译并运行下面的程序，结果是___B___。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    void test() {
        System.out.print("A");
    }
}
class B extends A {
    void test() {
```

```
        super.test();
        System.out.println("B");
    }
}
```

A．产生编译错误　　B．代码可以编译运行，并输出结果：AB

C．代码可以编译运行，但没有输出　　D．编译没有错误，但会运行时会产生异常

15. 编译运行下面的程序，结果是___A___。

```
public class A {
    public static void main(String args[]) {
        B b = new B();
        b.test();
    }
    public void test() {
        System.out.print("A");
    }
}
class B extends A {
    void test() {
        super.test();
        System.out.println("B");
    }
}
```

A．产生编译错误，因为类 B 覆盖类 A 的方法 test 时，降低了其访问控制的级别

B．代码可以编译运行，并输出结果：AB

C．代码可以编译运行，但没有输出

D．代码可以编译运行，并输出结果：A

16. 下面___B___修饰符所定义的方法必须被子类所覆写。

A．final　　B．abstract　　C．static　　D．interface

17. 下面___A___修饰符所定义的方法不能被子类所覆写。

A．final　　B．abstract　　D．static　　D．interface

18. 下面的程序编译运行的结果是___A___。

```
public class A implements B {
    public static void main(String args[]) {
        int m, n;
        A a = new A();
        m = a.K;
        n = B.K;
        System.out.println(m +"" + n);
    }
}
interface B {
    intK = 5;
}
```

A．5，5　　B．0，5

C．0，0　　D．编译程序产生编译结果

19. 下面关于接口的说法中不正确的是___C___。

A．接口所有的方法都是抽象的
B．接口所有的方法一定都是 public类型
C．用于定义接口的关键字是 implements
D．接口是 Java中的特殊类，包含全局常量和抽象方法

Note

20. 下面关于 Java的说法不正确的是___A___
A．abstrac和 fina能同时修饰一个类
B．抽象类不光可以做父类，也可以做子类
C．抽象方法不一定声明在抽象类中，也可以在接口中
D．声明为 fina的方法不能在子类中覆写

## 3.34.3 判断题

1．没有实例化的对象不能使用。（ √ ）
2．不可以为类定义多个构造方法。（ × ）
3．使用 stati声明的方法可以调用非 stati声明的方法。（ × ）
4．非 stati声明的方法可以调用 stati声明的属性或方法。（ √ ）
5．String对象可以使用==进行内容的比较。（ × ）
6．垃圾是指无用的内存空间，会被垃圾收集机制回收。（ √ ）
7．构造方法可以有返回值类型的声明。（ × ）
8．匿名对象是指使用一次的对象，使用之后将等待被垃圾回收。（ √ ）
9．使用 stati定义的内部类就成为外部类。（ √ ）
10．多个实例化对象之间不会互相影响，因为保存在不同的内存区域之中。（ √ ）
11．fina声明的类可以有子类。（ × ）
12．一个类继承了抽象类，则抽象类中的抽象方法需要在其子类中覆写。（ √ ）
13．fina类型的变量是常量，其内容不可改变。（ √ ）
14．一个类不能既是子类又是父类。（ √ ）
15．子类只能继承父类的成员，但不能修改父类成员。（ × ）
16．Java语言只支持单继承，不支持多继承。（ √ ）
17．子类可以继承父类的所有成员。（ √ ）
18．一个接口可以继承一个抽象类。（ × ）
19．一个接口可以同时继承多个接口。（ √ ）
20．在程序中，this和 super调用构造方法时可以同时出现。（ × ）

## 3.34.4 简答题

1．String类的操作特点。
答：String类的对象有两种实例化方式。
方式一：直接赋值，只开辟一块堆内存空间，并且对象可以入池；
方式二：构造方法，开辟两块堆内存空间，有一块将称为垃圾，不会自动入池，使用

intern()方法手工入池。

String对象的比较方法如下所示。

==：比较的是两个字符串对象的内存地址数值；

equals()：字符串内容比较。

字符串对象一旦声明，则内容不可改变，改变的只能是字符串对象的地址指向。

2．简述垃圾对象的产生。

答：垃圾指的是一块无用的引用内存，当将变量设置为 null或者长时间不使用时，就将成为垃圾。

3．static方法如何调用？非 static方法如何调用？

答：static方法可以使用类名称或实例化对象调用，而非 static方法只能依靠实例化对象才可以调用。

4．类与对象的关系是什么？如何创建及使用对象？

答：类规定了对象所具有的属性及行为（方法），类只有通过产生对象才可以分配属性或者是调用方法，对象的创建依靠关键字 new 创建。

5．举例说明子类对象的实例化过程。

答：当通过关键字 new 实例化子类对象时，会默认调用父类的无参构造方法，为父类对象实例化，而后才会调用子类的构造方法，为子类对象实例化。

6．简述 this与 super关键字的区别。

答：this和 super都可以调用类中的属性、方法、构造方法，但是 this调用的是本类操作，而 super是由子类调用父类操作。

7．简述方法的重载与覆写的区别。

答：方法重载是发生在一个类中，方法名称相同、参数的类型及个数不同，不受权限的限制。而覆写是发生在继承关系之中，子类和父类定义了方法名称相同、参数类型及个数、返回值类型完全相同的方法时所发生的操作，在子类覆写父类方法时，被覆写的方法不能拥有比父类更严格的访问权限。

8．在已有类的基础上派生新的类有什么好处？

答：扩充已有类的功能，并且利用方法的覆写扩充已有方法的功能。

9．如何区分子类和父类？子类可以继承父类的哪些内容？

答：子类使用 extends继承父类或使用 implements实现多个接口，子类可以继承父类中的全部内容，但是对于私有操作属于隐式继承，而非私有操作属于显式继承。

10．什么是多态？实现多态的方法有哪些？

答：多态是面向对象的最后一个主要特征，它本身主要分为两个方面。

方法的多态性：重载与覆写

重载：同一个方法名称，根据不同的参数类型及个数可以完成不同的功能；

覆写：同一个方法，根据操作的子类不同，所完成的功能也不同。

对象的多态性：父子类对象的转换。

向上转型：子类对象变为父类对象，格式：父类 父类对象 = 子类实例，自动；

向下转型：父类对象变为子类对象，格式：子类 子类对象 = (子类) 父类实例，强制；

11．接口有哪些特征？如何定义和实现接口？

答：接口之中全部由全局常量及抽象方法所组成，一个类可以同时实现多个接口，在 Java中使用 interface定义接口，子类使用 implements实现接口。

12．接口和抽象类有哪些区别？

答：抽象类及接口区别如下。

| No. | 区　　别 | 抽　象　类 | 接　　口 |
|---|---|---|---|
| 1 | 定义关键字 | abstract class | interface |
| 2 | 组成 | 常量、变量、抽象方法、普通方法、构造方法 | 全局常量、抽象方法 |
| 3 | 权限 | 可以使用各种权限 | 只能是 public |
| 4 | 关系 | 一个抽象类可以实现多个接口 | 接口不能够继承抽象类，却可以继承多接口 |
| 5 | 使用 | 子类使用 extend继承抽象类 | 子类使用 implements实现接口 |
| | | 抽象类和接口的对象都是利用对象多态性的向上转型，进行接口或抽象类的实例化操作 | |
| 6 | 设计模式 | 模板设计模式 | 工厂设计模式、代理设计模式 |
| 7 | 局限 | 一个子类只能够继承一个抽象类 | 一个子类可以实现多个接口 |

Note

13. 简述基本数据类型的自动装箱及自动拆箱操作。

答:在 JDK 1.5后,基本数据类型可以采用直接赋值的方式为包装类进行对象的实例化操作,而包装类的对象也可以通过直接赋值的方式变回基本数据类型。

## 3.34.5 编程题

1. 编写并测试一个代表地址的 Address类，地址信息由国家、省份、城市、街道和邮编组成，并可以返回完整的地址信息。

答案:

```
class Address {
     private String national;
     private String provincial;
     private String city;
     private String street;
     private String zipcode;
     public Address() {
     }
     public Address(String national, String provincial, String city,
               String street, String zipcode) {
          super();
          this.national = national;
          this.provincial = provincial;
          this.city = city;
          this.street = street;
          this.zipcode = zipcode;
     }
     public String toString() {
          return "国家：" +this.national +,"省份：" +this.provincial +, 城市："
                    + this.city +,"街道：" +this.street +,"邮政编码：" +this.zipcode;
     }
     //settergetter略
}
public class TestDemo {
```

```
        public staticvoid main(String args[]) {
            Address ad =new  Address("中国", "北京", "北京市", "MLDN", "100088");
            System.out.println(ad);
        }
}
```

Note

程序运行结果：

国家：中国，省份：北京，城市：北京市，街道：MLDN ，邮政编码：100088

2．定义并测试一个代表员工的 Employee 类，员工属性包括编号、姓名、基本薪水和薪水增长额，还包括计算增长后的工资总额。

答案：

```
class Employee {
    privateint empno ;                    /雇员编号
    private String ename ;                /雇员姓名
    privatedouble  sal ;                  /基本工资
    privatedouble  rate ;                 /工资增长额
    public Employee() {
    }
    public Employee(int empno, String ename,double  sal,double  rate) {
        super ();
        this.empno = empno;
        this.ename = ename;
        this.sal = sal;
        this.rate = rate;
    }
    public String toString() {
        return "雇员编号:" +this.empno + ", 雇员姓名:" +this.ename + ", 基本工资:" +
this.sal ;
    }
    public void growthin() {              /增长薪水
        this.sal =this.sal *this.rate ;
    }
    //setergetter略
}
public class TestDemo {
    public staticvoid main(String args[]) {
        Employee emp =  new  Employee(7369, "SMITH", 1000, 1.5);
        emp.growthin() ;                  /工资增长
        System.out.println(emp);
    }
}
```

程序运行结果：

雇员编号：7369，雇员姓名：SMITH ，基本工资：1500.0

3．编写程序，从字符串“want you to know one thing”中统计出字母“n”的出现次数。

答案：

```
public class TestDemo {
    public staticvoid main(String args[]) {
        String str = "want you to know one thing" ;/定义字符串
```

```
        int sum = 0 ;
        while (str.indexOf("n") != -1) {          //是否还有字母 n
            sum ++ ;                              //数据统计量增加
            str = str.substring(str.indexOf("n") + 1) ;//改变字符串内容
        }
        System.out.println("字母 n 的出现次数：" + sum);
    }
}
```

Note

程序运行结果：

字母 n 的出现次数：4

4. 设计一个 Dog 类，有名字、颜色、年龄等属性，定义构造方法来初始化类的这些属性，定义方法输出 Dog 信息。编写应用程序使用 Dog 类。

答案：

```
class Dog {
    private String name ;
    private String color ;
    private int age ;
    public Dog() {
    }
    public Dog(String name, String color, int age) {
        super();
        this.name = name;
        this.color = color;
        this.age = age;
    }
    public String toString() {
        return "狗的名字：" +this.name + ",狗的颜色：" +this.color +,"狗的年龄：" +this.age ;
    }
    // setter、getter略
}
public class TestDemo {
    public static void main(String args[]) {
        Dog dog = new  Dog("金毛", 金黄色", 3);
        System.out.println(dog);
    }
}
```

程序运行结果：

狗的名字：金毛，狗的颜色：金黄色，狗的年龄：3

5. 字符串操作：

从字符串“MLDN　中心 Java技术学习班 20130214”中提取开班日期。

答案：

```
public class TestDemo {
    public static void main(String args[]) {
        String str = "MLDN中心 Java 技术学习班 20130214" ;
        System.out.println(str.substring(str.indexOf("20130214")));
    }
```

```
}
```

程序运行结果：

```
20130214
```

将“MLDN JAVA 高端技术培训”字符串中的“JAVA ”替换为“JAVA EE ”。

答案：

```
public class TestDemo {
    public staticvoid main(String args[]) {
        String str = "MLDN JAVA高端技术培训" ;
        System.out.println(str.replaceAll("Java".toUpperCase(), "JAVA EE"));
    }
}
```

程序运行结果：

```
MLDN JAVA EE 高端技术培训
```

取出“Java技术学习班 20130214”中的第 8 个字符。

答案：

```
public class TestDemo {
    public staticvoid main(String args[]) {
        String str = "Java技术学习班 20130214" ;
        System.out.println(str.charAt(8));
    }
}
```

程序运行结果：

```
班
```

清除“Java技术学习班 20130214”中的所有“0”。

答案：

```
public class TestDemo {
    public staticvoid main(String args[]) {
        String str = "Java技术学习班 20130214" ;
        System.out.println(str.replaceAll("0", ""));
    }
}
```

程序运行结果：

```
Java技术学习班 213214
```

从任意给定的身份证号码中提取此人的出生日期。

答案：

```
public class TestDemo {
    public staticvoid main(String args[]) {
        String str = "110105197609190052O" ;
        System.out.println(str.substring(6,14));
    }
}
```

程序运行结果：

```
19760919
```

6．编写一个银行账户类，类的构成包括：

数据成员

用户的账户名称、用户的账户余额；

方法

开户（设置账户名称及余额），利用构造方法完成

查询余额

Note

答案：

```
class Account {
     private String name ;
     privatedouble  balance ;
     public Account() {
     }
     public Account(String name,double  balance) {
          super ();
          this.name = name;
          this.balance = balance;
     }
     public String toString() {
          return "账户名称：" +this.name + ", 余额：" +this.balance;
     }
     public double  getBalance() {
          return balance;
     }
     // setter和getter略
}
public class TestDemo {
     public staticvoid main(String args[]) {
          Account acc =new  Account("张三", 5000.0);
          System.out.println(acc);
          System.out.println("账户余额：" + acc.getBalance());
     }
}
```

程序运行结果：

```
账户名称：张三，余额：5000.0
账户余额：5000.0
```

7．定义一个 ClassName 接口，接口中只有一个抽象方法 getClassName()，设计一个类 Company ，该类实现接口 ClassName 中的方法 getClassName()，功能是获取该类的类名称。编写应用程序使用 Company 类。

答案：

```
interface ClassName {
     public String getClassName() ;
}
class Company  implements  ClassName {
     public String getClassName() {
          return "Company";
     }
}
```

```
public class TestDemo {
     public staticvoid main(String args[]) {
          ClassName name =  new  Company() ;
          System.out.println(name.getClassName());
     }
}
```

程序运行结果：

```
Company
```

8. 建立一个人类（Person）和学生类（Student），其功能要求：

（1）Person中包含4个保护型的数据成员name、address、sex、age，分别为字符串、字符串、字符及整型，表示姓名、地址、性别和年龄。一个 4 参构造方法、一个无参构造方法及一个输出方法用于显示4种属性。

（2）Student继承Person，并增加输出成员math、english存放数学和英语成绩。一个6参构造方法、一个两参构造方法，一个无参构造方法，重写输出方法用于显示全部6种属性。

答案：

```
class Person {
     private String name ;
     private String address ;
     privatechar sex ;
     privateint age ;
     public Person() {
     }
     public Person(String name, String address,char sex,int age) {
          super();
          this.name = name;
          this.address = address;
          this.sex = sex;
          this.age = age;
     }
     public String toString() {
          return "姓名：" +this.name + ", 地址：" +this.address +"性别：" +this.sex
                    + ", 年龄：" +this.age;
     }
     //setter、getter略
}
class Studentextends  Person {
     privatedouble  math ;
     privatedouble  english ;
     public Student() {
     }
     public Student(String name, String address,char sex,int age,double  math,
               double  english) {
          super(name, address, sex, age);
          this.math = math;
          this.english = english;
     }
     public String toString() {
```

Note

Note

```
        return super.toString() + "数学成绩：" + this.math + "，英语成绩：" + this.english;
    }
    //setter、getter略
}
public class TestDemo {
    public static void main(String args[]) {
        Student stu = new Student("张三", "北京西城区甲11号德外大街德胜科技园美江大厦 A座 - 6层", '男', 25, 90.0, 99.0);
        System.out.println(stu);
    }
}
```

程序运行结果：

姓名：张三，地址：北京西城区甲11号德外大街德胜科技园美江大厦 A 座 - 6层，性别：男，年龄：25，数学成绩：90.0，英语成绩：99.0

9. 定义员工类，具有姓名、年龄、性别属性，并具有构造方法、显示数据方法、定义管理层类、继承员工类，并有自己的属性：职务和年薪。定义职员类，继承员工类，并有自己的属性：所属部门和月薪。

答案：

```
class Employee {
    private String name ;
    private int age ;
    private char sex ;
    public Employee() {
    }
    public Employee(String name, int age, char sex) {
        super();
        this.name = name;
        this.age = age;
        this.sex = sex;
    }
    public String toString() {
        return "雇员姓名：" + this.name + "，年龄：" + this.age + "，性别：" + this.sex;
    }
    //setter、getter略
}
class Manager extends Employee {
    private String job ;
    private double income ;
    public Manager() {
    }
    public Manager(String name, int age, char sex, String job, double income) {
        super(name, age, sex);
        this.job = job;
        this.income = income;
    }
    public String toString() {
        return super.toString() + "职位：" + this.job + ",年薪：" + this.income ;
    }
```

```
        //setter、getter略
    }
    class Staff extends  Employee {
        private String dept ;
        private double  salary ;
        public Staff() {
        }
        public Staff(String name, int age, char  sex,  String dept, double  salary) {
            super(name,  age,  sex);
            this.dept = dept;
            this.salary = salary;
        }
        public String toString() {
            return super.toString() + "部门：" +this.dept +"，月薪：" +this.salary ;
        }
        //setter、getter略
    }
    public class TestDemo {
        public static void main(String args[]) {
            Employee ea = new  Manager("张三", 30,"男","总监", 200000.0);
            Employee eb = new  Staff("李四", 25,"女","业务部", 1500.0);
            System.out.println(ea);
            System.out.println(eb);

        }
    }
```

程序运行结果：

雇员姓名：张三，年龄：30，性别：男，职位：总监，年薪：200000.0

雇员姓名：李四，年龄：25，性别：女，部门：业务部，月薪：1500.0

10．定义 Shape 类表示一般二维图形。Shape 具有抽象方法 area 和 perimeter 分别计算形状的面积和周长。试定义一些二维形状类（如矩形、三角形、圆形等），这些类均为 Shape 类的子类。

答案：

```
abstract class Shape {
    public abstract double area();
    public abstract double perimeter();
}
class Rectangle extends Shape {                    /矩形
    private double wide ;                          /宽
    private double longs ;                         /长
    public Rectangle() {
    }
    public Rectangle(double wide, double longs) {
        super();
        this.wide = wide;
        this.longs = longs;
    }
    public double area() {
```



Note

Note

```
            return this.longs * this.wide ;
        }
        public double perimeter() {
            return (this.longs + this.wide) * 2;
        }
}
class Triangle extends Shape {                          /三角形
        private double edgea ;                          /边长
        private double edgeb ;                          /边长
        private double edgec ;                          /边长
        public Triangle() {
        }
        public Triangle(double edgea, double edgeb, double edgec) {
            super();
            this.edgea = edgea;
            this.edgeb = edgeb;
            this.edgec = edgec;
        }
        public double area() {
            return this.edgea * this.edgeb / 2 ;
        }
        public double perimeter() {
            return this.edgea + this.edgeb + this.edgec ;
        }
}
class Round extends Shape {                             /圆形
        private double radius ;                         /半径
        public Round() {
        }
        public Round(double radius) {
            super();
            this.radius = radius;
        }
        public double area() {
            return this.radius * this.radius * Math.PI;
        }
        public double perimeter() {
            return this.radius * 2 * Math.PI;
        }
}
public class TestDemo {
        public static void main(String args[]) {
            Shape rectangle = new Rectangle(10.5, 20.6);
            Shape triangle = new Triangle(10.1, 20.2, 30.3);
            Shape round = new Round(30.3) ;
            System.out.println("矩形面积：" + rectangle.area(),+"矩形周长：" + rectangle.perimeter());
            System.out.println("三角形面积：" + triangle.area(), "三角形周长：" + triangle.perimeter());
            System.out.println("圆形面积：" + round.area() +,"圆形周长：" + round.perimeter());
        }
}
```