第1章 Java入门

实验1 一个简单的应用程序

1. 相关知识点

Java语言的出现是源于对独立于平台语言的需要，即这种语言编写的程序不会因为芯片的变化而无法运行或出现运行错误。目前，随着网络的迅速发展，Java语言的优势越显明显，Java已经成为网络时代最重要的语言之一。

Sun公司要实现“编写一次，到处运行”(write once，run anywhere)的目标，就必须提供相应的Java运行平台，目前Java运行平台主要分为下列3个版本。

(1) Java SE：称为Java标准版或Java标准平台。Java SE提供了标准的JDK开发平台。利用该平台可以开发Java桌面应用程序和低端的服务器应用程序，也可以开发Java Applet程序。当前成熟的新的JDK版本为JDK1.6。

(2) Java EE：称为Java企业版或Java企业平台。使用JavaEE可以构建企业级的服务应用，Java EE平台包含了Java SE平台，并增加了附加类库，以便支持目录管理、交易管理和企业级消息处理等功能。

(3) Java ME：称为Java微型版或Java小型平台。Java ME是一种很小的Java运行环境，用于嵌入式的消费产品中，如移动电话、掌上电脑或其他无线设备等。

上述Java运行平台都包括了相应的Java虚拟机(Java Virtual Machine)，虚拟机负责将字节码文件(包括程序使用的类库中的字节码)加载到内存，然后采用解释方式来执行字节码文件，即根据相应硬件的机器指令翻译一句执行一句。Java SE平台是学习掌握Java语言的最佳平台，而掌握Java SE又是进一步学习Java EE和Java ME所必须的。

2. 实验目的

本实验的目的是让学生掌握开发Java应用程序的三个步骤：编写源文件、编译源文件和运行应用程序。

3. 实验要求

编写一个简单的Java应用程序，该程序在命令行窗口输出两行文字：“你好，很高兴学习Java”和“We are students”。

4. 程序效果示例

程序运行效果如图1.1所示。

```
C:\1000>javac Hello.java

C:\1000>java Hello
你好，很高兴学习Java
We are students
```

图1.1 简单的应用程序

5. 程序模板

按模板要求，将【代码】部分替换为Java程序代码。

```
//Hello.java
public class Hello {
public static void main (String args[ ]) {
    【代码 1】                                    //命令行窗口输出"你好,很高兴学习 Java"
        A a = new A();
        a.fA();
   }
}
class A {
   void fA() {
     【代码 2】                                   //命令行窗口输出"We are students"
   }
}
```

6. 实验指导

打开一个文本编辑器。如果是 Windows 操作系统,打开“记事本”编辑器。可以通过“程序”→“附件”→“记事本”来打开文本编辑器;如果是其他操作系统,请在指导老师的帮助下打开一个纯文本编辑器。

按着“程序模板”的要求编辑输入源程序。

(1) 保存源文件

将源文件命名为 Hello.java。要求将源文件保存到 C 盘的某个文件夹中,例如 C:\1000。

(2) 编译源文件

打开命令行窗口,对于 Windows 操作系统,打开 MS-DOS 窗口。对于 Windows 2000/XP 操作系统,可以通过单击“开始”按钮,选择“程序”|“附件”|MS-DOS 打开命令行窗口,也可以选择“开始”|“运行”命令,在打开的“运行”对话框中输入 cmd,打开命令行窗口。如果当前 MS-DOS 窗口显示的逻辑符是“D:\”,输入“C:”,按 Enret 键确认,使得当前 MS-DOS 窗口的状态是“C:\”。如果当前 MS-DOS 窗口的状态是 C 盘符的某个子目录,请输入“cd\”,使得当前 MS-DOS 窗口的状态是“C:\”。当 MS-DOS 窗口的状态是“C:\”时,输入进入文件夹目录的命令,例如,“CD 1000”。然后执行下列编译命令:C:\1000> javac Hello.java。

在编译源文件时如果遇到错误提示:“Command not Found”,请检查是否正确设置了系统变量 Path。如果 JDK 的安装目录是 C:\jdk1.6,可以在命令行临时设置系统变量 Path:path C:\jdk1.6\bin。

在编译源文件时如果遇到错误提示:“File not Found”,请检查源文件是否保存在当前目录中。

在编译源文件时可能遇到一些语法错误的提示,例如:“非法字符:\65307”,其原因是在汉语输入状态下输入了程序中需要的语句分号。Java 源程序中语句所涉及的小括号及标点符号都是英文状态下输入的,如"你好,很高兴学习 Java"中的引号必须是英文状态下的引号,而字符串里面的符号不受汉语或英文的限制。

(3) 运行程序

```
C:\1000 > java Hello
```

运行程序如果出现错误提示:Exception in thread main java.lang.NoCalssFondError,

请检查是否正确设置了系统变量 ClassPath，或检查是否运行的是主类的名字。

7. 实验后的练习

(1) 编译器怎样提示丢失大括号的错误。

(2) 编译器怎样提示语句丢失分号的错误。

(3) 编译器怎样提示将 System 写成 system 这一错误。

(4) 编译器怎样提示将 String 写成 string 这一错误。

8. 填写实验报告

实验编号：101　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2　教室、老师和学生

1. 相关知识点

一个 Java 应用程序(也称为一个工程)是由若干个类所构成，这些类可以在一个源文件中，也可以分布在若干个源文件中。Java 应用程序有一个主类，即含有 main 方法的类，Java 应用程序从主类的 main 方法开始执行。在编写一个 Java 应用程序时，可以编写若干个 Java 源文件，每个源文件编译后产生若干个类的字节码文件。经常需要进行如下的操作。

• 将应用程序涉及的 Java 源文件保存在相同的目录中，分别编译通过，得到 Java 应用程序所需要的字节码文件。

• 运行主类

当使用解释器运行一个 Java 应用程序时，Java 虚拟机将 Java 应用程序需要的字节码文件加载到内存，然后再由 Java 的虚拟机解释执行，因此，可以事先单独编译一个 Java 应用程序所需要的其他源文件，并将得到的字节码文件和主类的字节码文件存放在同一目录中。如果应用程序的主类的源文件和其他的源文件在同一目录中，也可以只编译主类的源文件，Java 系统会自动地先编译主类需要的其他源文件。

2. 实验目的

熟悉 Java 应用程序的基本结构，并能联合编译应用程序所需要的类。

3. 实验要求

编写 3 个源文件：ClassRoom. java、Teacher. java 和 Student. java，每个源文件只有一个类。ClassRoom. java 含有应用程序的主类(含有 main 方法)，并使用了 Teacher 和 Student 类。将 3 个源文件保存到同一目录中，例如 C:\1000，然后编译 ClassRoom. java。

4. 程序效果示例

程序运行效果如图 1.2 所示。

5. 程序模板

请按模板要求，将【代码】部分替换为 Java 程序代码。

```
教学活动从教室开始
我是张老师
我是学生,名字是:奖励
```

图 1.2　只编译主类

```
//ClassRoom.java
public class ClassRoom {
    public static void main (String args[ ]) {
      【代码 1】                                    //命令行窗口输出"教学活动从教室开始"
      Teacher zhang = new Teacher();
      Student jiang = new Student();
      zhang.introduceSelf();
      jiang.introduceSelf();
    }
}
//Teacher.java
public class Teacher {
  void introduceSelf() {
    【代码 2】                                    //命令行窗口输出"我是张老师"
   }
}
//Student.java
public class Student {
  void introduceSelf() {
    【代码 3】                                    //命令行窗口输出"我是学生,名字是:奖励"
   }
}
```

6. 实验指导

编译 ClassRoom.java 的过程中，Java 系统会自动地先编译 Teacher.java、Student.java，编译通过后，C:\1000 中将会有 ClassRom.class、Teacher.class 和 Student.class 三个字节码文件。

当运行上述 Java 应用程序时，虚拟机将 ClassRom.class、Teacher.class 和 Student.class 加载到内存。当虚拟机将 ClassRom.class 加载到内存时，就为主类中的 main 方法分配了入口地址，以便 Java 解释器调用 main 方法开始运行程序。如果编写程序时错误地将主类中的 main 方法写成：public void main(String args[])，那么，程序可以编译通过，但却无法运行。

7. 实验后的练习

(1) 将 ClassRoom.java 编译通过以后，不断地修改 Teacher.java 源文件中的【代码 2】，比如，在命令行窗口输出“我是数学老师”或“我是物理老师”。要求每次修改 Teacher.java 源文件后，单独编译 Teacher.java，然后直接运行应用程序(不要再编译 ClassRoom.java)。

(2) 如果需要编译某个目录下的全部 Java 源文件，如 C:\1000 目录，可以使用如下命令：

```
C:\1000> javac *.java
```

请练习上述命令。

8. 填写实验报告

实验编号：102　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实 验 答 案

实验 1

【代码 1】 System. out. println("你好，很高兴学习 Java ");

【代码 2】 System. out. println("We are students ");

实验 2

【代码 1】 System. out. println("教学活动从教室开始");

【代码 2】 System. out. println("我是张老师");

【代码 3】 System. out. println("我是学生，名字是：奖励");

第 2 章　基本数据类型

实验 1　输出特殊偏旁的汉字

1. 相关知识点

Java 的简单数据类型(也称基本数据类型)包括 byte、short、int、long、float、double 和 char。简单数据类型按精度级别由低到高的顺序是：

```
byte  short  char  int  long  float  double
```

掌握简单类型的数据转换规则：当把级别低的变量的值赋给级别高的变量时，系统自动完成数据类型的转换；当把级别高的变量的值赋给级别低的变量时，需用类型转换运算。

要观察一个字符在 Unicode 表中的顺序位置，需使用 int 类型转换，如(int)'a'。要得到一个 0～65 535 之间的数所代表的 Unicode 表中相应位置上的字符需使用 char 型转换。char 型数据和 byte、short、int 运算的结果是 int 型数据。

2. 实验目的

掌握 char 型数据和 int 型数据之间的互相转换，同时了解 Unicode 字符表。

3. 实验要求

编写应用程序，在命令行窗口输出 5 个“石”字旁的汉字和 5 个“女”字旁的汉字。

4. 程序效果示例

运行效果如图 2.1 所示。

```
输出5个石字旁的汉字:
研(30740)砕(30741)砖(30742)砗(30743)砘(30744)砚(30745)
输出5个女字旁的汉字:
妈(22920)妩(22921)妊(22922)妹(22923)妍(22924)妍(22925)
```

图 2.1　输出特殊偏旁的汉字

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

InputChinese.java

```
public class E {
   public static void main (String args[ ]){
       char ch = '研',zifu = 0;
       int p = 22920,count = 5,position = 0;
       System.out.printf("输出 %d 个石字旁的汉字:\n",count);
       for(char c = ch;c <= ch + count;c++) {
        【代码 1】                    //c 进行 int 型转换数据运算,并将结果赋值给 position
         System.out.printf(" %c( %d)",c,position);
       }
       System.out.printf("\n 输出 %d 个女字旁的汉字:\n",count);
       for(int n = p;n <= p + count;n++) {
```

```
            【代码 2】                    // n 做 char 型转换运算,并将结果赋值给 zifu
             System.out.printf(" % c( % d)",zifu,n);
           }
      }
}
```

6. 实验指导

Uincode 表将偏旁相同的汉字按顺序排列。

7. 实验后的练习

(1) 将一个 double 型数据直接赋值给 float 型变量,程序编译时系统会提示怎样的错误?

(2) 在应用程序的 main 方法中增加语句:

```
float x = 0.618;
```

程序能编译通过吗?

(3) 在应用程序的 main 方法中增加语句:

```
byte y = 128;
```

程序能编译通过吗? 在应用程序的 main 方法中增加语句:

```
int z = (byte)128;
```

程序输出变量 z 的值是多少?

8. 填写实验报告

实验编号:201　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2　输入、输出学生的基本信息

1. 相关知识点

Scanner 是 JDK1.5 新增的一个类,可以使用该类创建对象:

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

然后 reader 对象调用下列方法,读取用户在命令行(如 MS-DOS 窗口)输入的各种简单类型数据:

```
nextBoolean();nextByte(),nextShort(),nextInt(),nextLong(),nextFloat(),nextDouble().
```

上述方法执行时都会堵塞,程序等待用户在命令行输入数据后按 Enter 键确认。

2. 实验目的

掌握从键盘为简单型变量输入数据。

3. 实验要求

编写 Java 应用程序，使用 Scanner 对象输入学生的基本信息，并输出基本信息。

4. 程序效果示例

程序运行效果如图 2.2 所示。

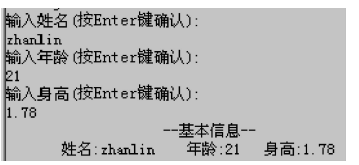

图 2.2 输入输出学生基本信息

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

InputMess.java

```
import java.util.Scanner;
public class InputMess {
   public static void main(String args[]) {
       Scanner reader = new Scanner(System.in);
       System.out.println("输入姓名(按 Enter 键确认):");
       String name =【代码 1】                    //从键盘为 name 赋值
       System.out.println("输入年龄(按 Enter 键确认):");
       byte age =【代码 2】                       //从键盘为 age 赋值
       System.out.println("输入身高(按 Enter 键确认):");
       float height =【代码 3】                   //从键盘为 height 赋值
       System.out.printf("%28s\n","--基本信息--");
       System.out.printf("%10s%-10s","姓名:",name);
       System.out.printf("%4s%-4d","年龄:",age);
       System.out.printf("%4s%-4.2f","身高:",height);
   }
}
```

6. 实验指导

JDK1.5 后续版本新增了和 C 语言中 printf 函数类似的数据输出方法，其格式为：

```
System.out.printf("格式控制部分",表达式 1,表达式 2,…,表达式 n)
```

输出数据时也可以控制数据在命令行的位置，例如，%md：输出的 int 型数据占 m 列。%m.nf：输出的浮点型数据占 m 列，小数点保留 n 位。

7. 实验后的练习

编写一个 Java 应用程序，在主类的 main 方法中声明用于存放矩形的宽和高的 2 个 double 型变量：width、height 以及存放矩形面积的 double 型变量 area。

使用 Scanner 对象调用 nextDouble()方法，让用户从键盘为 width，height 变量输入值，然后程序计算出矩形的面积，并输出矩形的宽和高以及面积。

8. 填写实验报告

实验编号：302　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 3　超大整数的加法

1. 相关知识点

对于特别大的整数，无法使用 long 型变量来处理大整数的加法。一种简单的处理办法是使用数组。可以将一个大整数的各个位上的数字存放到一个数组中。那么只需将存放大整数各个位上的数字的两个数组的各个元素按着一定的算法进行加法运算，将结果存放到另一个数组中即可。

2. 实验目的

本实验的目的是让学生掌握使用数组处理大整数的加法。

3. 实验要求

声明 3 个 int 型数组：a、b、c，要求三者的长度相同。将其中的 a、b 初始化为大整数的表示，但大整数的数字的长度必须小于数组的长度，以便保证数组 a 和 b 的首元素的值是 0。将 a 和 b 的各个元素进行加法运算(需要进位时，需改变 a 的元素的值)，结果依次存放到数组 c 中，输出数组 c。

4. 程序效果示例

程序运行效果如图 2.3 所示。

```
99794567234564789876 9
加上:
591645623457213403 29
等于:
1057110234691369239098
```

图 2.3　大整数的加法

5. 程序模板

仔细阅读模板代码，完成实验后的练习。

HandleLargeNumber.java

```
public class HandleLargeNumber {
   public static void main(String args[]) {
      int a[] = {0,9,9,7,9,4,5,6,7,2,3,4,5,6,4,7,8,9,8,7,6,9};
      int b[] = {0,0,5,9,1,6,4,5,6,2,3,4,5,7,2,1,3,4,0,3,2,9};
      int c[] = new int[a.length];
      int i = 0,result = 0,k = 0;
      for(i = 0;i < a.length;i++) {
         if(a[i] != 0) {
            k = i;
            break;
         }
      }
      for(i = k;i < a.length;i++) {
          System.out.printf("%d",a[i]);
      }
      System.out.printf("\n 加上:\n");
      for(i = 0;i < b.length;i++) {
         if(b[i] != 0) {
            k = i;
            break;
         }
      }
      for(i = k;i < b.length;i++) {
          System.out.printf("%d",b[i]);
```

```
        }
        for(i = a.length - 1;i >= 0;i -- ) {
           result = a[i] + b[i];
           if(result >= 10) {
                 c[i] = result % 10;
                 a[i - 1] = a[i - 1] + 1;
           }
           else
                  c[i] = result;
        }
        System.out.printf("\n 等于:\n");
        for(i = 0;i < c.length;i++) {
           if(c[i]! = 0) {
               k = i;
               break;
           }
        }
        for(i = k;i < c.length;i++) {
              System.out.printf(" % d",c[i]);
        }
    }
}
```

6. 实验指导

数组 a 和数组 b 的单元进行加法操作时，需要进行进位处理，所以要求数组 a 和 b 的首元素必须是数字 0。

7. 实验后的练习

参考本实验，编写程序计算两个大整数的减法。

8. 填写实验报告

实验编号：804　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实 验 答 案

实验 1

【代码 1】 position = (int)c;

【代码 2】 zifu=(char)n;

实验 2

【代码 1】 reader.nextLine();

【代码 2】 reader.nextByte();

【代码 3】 reader.nextFloat();

第3章 运算符、表达式与语句

实验1 托运行李

1. 相关知识点

类型转换运算符是单目运算符，其运算所得数据的类型可能不同于操作元的类型。类型转换运算符不改变操作元本身的类型，操作元经常是数值型数据。例如，(float)12 的结果是 12.0f，(int)45.98 的结果是 45，(double)(int)68.89 的结果是 68.0。

2. 实验目的

本实验的目的是让学生掌握类型转换运算符。

3. 实验要求

(1) 火车在计算托运行李费用时以 kg 为单位计算费用(12 元/kg)，忽略重量中的小数部分，即忽略不足 1kg 的部分。

(2) 汽车在计算托运行李费用时以 kg 为单位计算费用(22 元/kg)，将重量中的小数部分进行四舍五入，即将不足 1kg 的部分进行四舍五入。

(3) 飞机在计算托运行李费用时以 g(1kg 等于 1000g)为单位计算费用(0.062 元/g)，将重量中的小数部分，即不足 1g 的部分进行四舍五入。

用 double 型变量 weight 存放行李重量，用 charge 存放托运费用，程序让用户从键盘输入 weight 的值，该值是行李的重量(kg 为单位)，然后程序将分别计算出用火车、汽车和飞机托运行李的费用。

4. 程序效果示例

程序运行效果如图 3.1 所示。

```
输入行李重量:9.62567
行李重量:9.625670公斤(kg)
需要计费的重量:9(kg)
用火车托运(12元/kg),费用:108.000000元
需要计费的重量:10(kg)
用汽车托运(22元/kg),费用:220.000000元
行李重量:9625.670000克(g)
需要计费的重量:9626(g)
用飞机托运(0.062000元/g),费用:596.812000元
```

图 3.1 托运行李

5. 程序模板

按模板要求，将【代码】替换为程序代码。

BaggageAndMony.java

```
import java.util.Scanner;
public class BaggageAndMony {
   public static void main(String args[]) {
       int trainCharge = 12;                    //火车托运计费:每公斤 12 元
       int carCharge = 22;                      //汽车托运计费:每公斤 12 元
       double planeCharge = 0.062 ;             //飞机托运计费:每克 0.062 元
       Scanner reader = new Scanner(System.in);
       double weight,charge;
       System.out.printf("输入行李重量:");
```

```
        weight = reader.nextDouble();
        System.out.printf("行李重量:%f 公斤(kg)\n",weight);
        System.out.printf("需要计费的重量:%d(kg)\n",(int)weight);
        【代码 1】          //将表达式(int)weight * trainCharge 的值赋值给 charge
        System.out.printf("用火车托运(%d 元/kg),费用:%f 元\n",trainCharge,charge);
        System.out.printf("需要计费的重量:%d(kg)\n",(int)(weight + 0.5));
        【代码 2】          //将表达式(int)(weight + 0.5) * carCharge 的值赋值给 charge
        System.out.printf("用汽车托运(%d 元/kg),费用:%f 元\n",carCharge,charge);
        System.out.printf("行李重量:%f 克(g)\n",weight * 1000);
        System.out.printf("需要计费的重量:%d(g)\n",(int)(weight * 1000 + 0.5));
        【代码 3】          //将表达式(int)(weight * 1000 + 0.5) * planeCharge 的值赋值给 charge
        System.out.printf("用飞机托运(%f 元/g),费用:%f 元\n", planeCharge,charge);
    }
}
```

6. 实验指导

为了实现四舍五入,只需将浮点数据加上 0.5,再将结果进行 int 型转换运算即可。类型转换运算符的级别是 2 级,因此,(int)15.9+0.5 的结果是 15.5,即相当于((int)15.9)+0.5,而(int)(15.9+0.5)的结果才是 16。

7. 实验后的练习

在实验的基础上进行改进,让飞机在托运行李时给用户一定的优惠:免收费用中不足一元、一角或一分的金额。

8. 填写实验报告

实验编号:301　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2　自动售货机

1. 相关知识点

复合语句的形式是:

```
{
   若干语句
}
```

复合语句是一条语句。if 语句、if-else 语句中的 if 操作和 else 操作都是复合语句。由于复合语句由若干条语句构成,因此在复合语句中就可以有各种语句,如可以有 if 语句、if-else 语句、switch 语句等。

2. 实验目的

本实验的目的是让学生掌握在 if-else 分支语句的 if 操作中使用 switch 语句。

3. 实验要求

自动售货机为客户提供各种饮料。饮料的价格有两种：2 元和 3 元。用户投入 2 元钱，可以选择“净净矿泉水”、“甜甜矿泉水”和“美美矿泉水”之一。用户投入 3 元钱，可以选择“爽口可乐”、“清凉雪碧”和“雪山果汁”之一。编写程序模拟用户向自动售货机投入钱币、得到一种饮料。

4. 程序效果示例

程序运行效果如图 3.2 所示。

```
投入金额:2或3元(按Enter键确认):3
选择爽口可乐(1),清凉雪碧(2),和雪山果汁(3)之一:
输入1,2或3:2
得到清凉雪碧
```

图 3.2　自动售货机

5. 程序模板

请编译、运行模板给出的代码，然后完成试验后的练习。

MachineSell.java

```
import java.util.Scanner;
public class MachineSell {
   public static void main(String args[]){
       int money;
       int drinkKind;
       System.out.printf("投入金额:2 或 3 元(按 Enter 键确认):");
       Scanner reader = new Scanner(System.in);
       money = reader.nextInt();
       if(money == 2) {
           System.out.printf("选择净净矿泉水(1),甜甜矿泉水(2)和美美矿泉水(3)之一:\n");
           System.out.printf("输入 1,2 或 3:");
           drinkKind = reader.nextInt();
           switch(drinkKind) {
              case 1 : System.out.printf("得到净净矿泉水\n");
                       break;
              case 2 : System.out.printf("得到甜甜矿泉水\n");
                       break;
              case 3 : System.out.printf("得到美美矿泉水\n");
                       break;
              default: System.out.printf("选择错误");
           }
       }
       else if(money == 3) {
           System.out.printf("选择爽口可乐(1),清凉雪碧(2),和雪山果汁(3)之一:\n");
           System.out.printf("输入 1,2 或 3:");
           drinkKind = reader.nextInt();
           switch(drinkKind) {
              case 1 : System.out.printf("得到爽口可乐\n");
                       break;
              case 2 : System.out.printf("得到清凉雪碧\n");
                       break;
              case 3 : System.out.printf("得到雪山果汁\n");
                       break;
              default: System.out.printf("选择错误");
           }
       }
```

```
            else {
                System.out.printf("输入的钱币不符合要求");
            }
        }
    }
```


6. 实验指导

switch 语句中“表达式”的值应为 byte,short、int、char 型或枚举类型；case 后的“常量值 1”到“常量值 n”也是 byte,short、int、char 型或枚举类型常量,而且要互不相同。

7. 实验后的练习

改进自动售货机。使得用户也可以投入 5 元钱,选择“草原奶茶”、“青青咖啡”和“甜美酸奶”之一。

8. 填写实验报告

实验编号：302　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 3　猜数字游戏

1. 相关知识点

循环是控制结构语句中的最重要的语句之一,循环语句是根据条件反复执行同一代码块。循环语句有下列三种：

(1) while 循环

while 语句的一般格式：

```
while(表达式){
      若干语句                                    //该复合语句称为循环体
}
```

while 语句的执行规则如下：

① 计算表达式的值,如果该值是 true 时,就进行 ②,否则进行③。

② 执行循环体,再进行①。

③ 结束 while 语句的执行。

(2) for 循环

for 语句的一般格式：

```
for (表达式 1; 表达式 2; 表达式 3) {
    若干语句   //该复合语句称为循环体
}
```

for 语句的执行规则如下：

① 计算“表达式 1”,完成必要的初始化工作。

② 判断“表达式 2”的值,若“表达式 2”的值为 true,则进行③,否则进行④。

③ 执行循环体,然后计算“表达式 3”,以便改变循环条件,进行②。

④ 结束 for 语句的执行。

2. 实验目的

本实验的目的是让学生使用 if-else 分支和 while 循环语句解决问题。

3. 实验要求

编写一个 Java 应用程序,在主类的 main 方法中实现下列功能:

- 程序随机分配给客户一个 1～100 之间的整数。
- 用户输入自己的猜测。
- 程序返回提示信息,提示信息分别是:“猜大了”、“猜小了”或“猜对了”。
- 用户可根据提示信息再次输入猜测,直到提示信息是“猜对了”。

4. 程序效果示例

程序运行效果如图 3.3 所示。

```
给你一个1至100之间的整数,请猜测这个数
输入您的猜测:50
猜大了,再输入你的猜测:25
猜小了,再输入你的猜测:36
猜大了,再输入你的猜测:30
猜大了,再输入你的猜测:27
猜大了,再输入你的猜测:26
猜对了!
```

图 3.3　猜数字

5. 程序模板

按模板要求,将【代码】部分替换为 Java 程序代码。

GuessNumber.java

```
import java.util.Scanner;
import java.util.Random;
public class GuessNumber {
   public static void main (String args[]) {
       Scanner reader = new Scanner(System.in);
       Random random = new Random();
       System.out.println("给你一个 1～100 之间的整数,请猜测这个数");
       int realNumber = random.nextInt(100) + 1;//random.nextInt(100)是[0,100)中的随机整数
       int yourGuess = 0;
       System.out.print("输入您的猜测:");
       yourGuess = reader.nextInt();
       while(【代码 1】)                          //循环条件
       {
          if(【代码 2】)                          //猜大了的条件代码
          {
              System.out.print("猜大了,再输入你的猜测:");
              yourGuess = reader.nextInt();
          }
          else if(【代码 3】)                     //猜小了的条件代码
          {
              System.out.print("猜小了,再输入你的猜测:");
              yourGuess = reader.nextInt();
          }
       }
       System.out.println("猜对了!");
   }
}
```

6. 实验指导

经常使用 while 循环“强迫”程序重复执行一段代码,【代码 1】必须是值为 boolean 型数据的表达式,只要【代码 1】的值为 true 就是让用户继续输入猜测。只要用户的输入能使得循环语句结束,就说明用户已经猜对了。

7. 实验后的练习

用 yourGuess>realNumber 替换【代码 1】可以吗? 语句:“System. out. println("猜对了!");”为何要放在 while 循环语句之后? 放在 while 语句的循环体中合理吗?

8. 填写实验报告

实验编号:303　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实 验 答 案

实验 1

【代码 1】　charge = (int)weight * trainCharge;

【代码 2】　charge = (int)(weight+0.5) * carCharge;

【代码 3】　charge = (int)(weight * 1000+0.5) * planeCharge;

实验 2

【代码 1】　yourGuess!=realNumber

【代码 2】　yourGuess>realNumber

【代码 3】　yourGuess<realNumber

第4章 类与对象

实验1 Tank类

1. 相关知识点

类是Java中最重要的数据类型。类的目的是抽象出一类事物的共有属性和行为，即抽象出数据以及在数据上所进行的操作。类的类体由两部分组成：变量的声明和方法的定义，其中的构造方法(方法名与类名相同，无类型)用于创建对象，其他的方法供该类创建的对象调用，如图4.1所示。

抽象的目的是产生类，而类的目的是创建具有属性和行为的对象。使用new运算符和类的构造方法为声明的对象分配变量，即创建对象。对象不仅可以操作自己的变量改变状态，而且能调用类中的方法产生一定的行为。通过使用运算符“.”，对象可以实现对自己的变量访问和方法的调用。

Java程序以类为“基本单位”，即一个Java程序就是由若干个类所构成。一个Java程序可以将它使用的各个类分别存放在不同的源文件中，也可以将它使用的类存放在一个源文件中。因此，要学习Java编程就必须学会怎样去写类，即怎样用Java的语法去描述一类事物共有的属性和行为。

2. 实验目的

本实验的目的是让学生使用类来封装对象的属性和行为。

3. 实验要求

编写一个Java应用程序，该程序中有两个类：Tank(用于刻画坦克)和Fight(主类)。具体要求如下：

(1) Tank类有一个double类型的变量speed，用于刻画坦克的速度；一个int型变量bulletAmount，用于刻画坦克的炮弹数量。定义了speedUp()和speedDown()方法，体现坦克有加速、减速行为；定义了setBulletAmount(int p)方法，用于设置坦克炮弹的数量；定义了fire()方法，体现坦克有开炮行为。Tank类的UML图如图4.2所示。

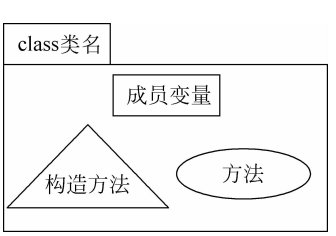

图4.1 类的基本结构

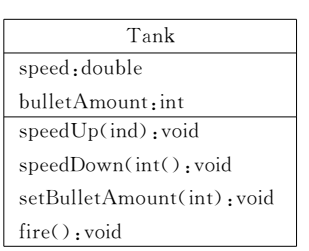

图4.2 Tank类的UML图

（2）在主类 Fight 的 main 方法中用 Tank 类创建坦克，并让坦克调用方法设置炮弹的数量，显示坦克的加速、减速和开炮等行为。

4. 程序效果示例

程序运行效果如图 4.3 所示。

```
tank1的炮弹数量：10
tank2的炮弹数量：10
tank1目前的速度：80.0
tank2目前的速度：90.0
tank1目前的速度：65.0
tank2目前的速度：60.0
tank1开火：
打出一发炮弹
tank2开火：
打出一发炮弹
打出一发炮弹
tank1的炮弹数量：9
tank2的炮弹数量：8
```

图 4.3　Tank 类创建对象

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

Tank.java

```
public class Tank {
        【代码 1】//声明 double 型变量 speed,刻画速度
        【代码 2】//声明 int 型变量 bulletAmount,刻画炮弹数量
         void speedUp(int s) {
            【代码 3】  //将 s + speed 赋值给 speed
         }
         void speedDown(int d) {
             if(speed - d >= 0)
               【代码 4】  //将 speed - d 赋值给 speed
             else
               speed = 0;
         }
         void setBulletAmount(int m) {
             bulletAmount = m;
         }
         int getBulletAmount() {
             return bulletAmount;
         }
         double getSpeed() {
             return speed;
         }
         void fire() {
              if(bulletAmount >= 1){
                 【代码 5】  //将 bulletAmount - 1 赋值给 bulletAmount
                 System.out.println("打出一发炮弹");
              }
              else {
                 System.out.println("没有炮弹了,无法开火");
              }
         }
     }
```

Fight.java

```
public class Fight {
   public static void main(String args[]) {
      Tank tank1,tank2;
      tank1 = new Tank();
      tank2 = new Tank();
      tank1.setBulletAmount(10);
      tank2.setBulletAmount(10);
      System.out.println("tank1 的炮弹数量: " + tank1.getBulletAmount());
      System.out.println("tank2 的炮弹数量: " + tank2.getBulletAmount());
```

```
        tank1.speedUp(80);
        tank2.speedUp(90);
        System.out.println("tank1 目前的速度: " + tank1.getSpeed());
        System.out.println("tank2 目前的速度: " + tank2.getSpeed());
        tank1.speedDown(15);
        tank2.speedDown(30);
        System.out.println("tank1 目前的速度: " + tank1.getSpeed());
        System.out.println("tank2 目前的速度: " + tank2.getSpeed());
        System.out.println("tank1 开火: ");
        tank1.fire();
        System.out.println("tank2 开火: ");
        tank2.fire();
        tank2.fire();
        System.out.println("tank1 的炮弹数量: " + tank1.getBulletAmount());
        System.out.println("tank2 的炮弹数量: " + tank2.getBulletAmount());
    }
}
```

6. 实验指导

创建一个对象时,成员变量被分配内存空间,这些内存空间称做该对象的实体或变量,而对象中存放着引用,以确保这些变量由该对象操作使用。需要注意的是,没有被创建的对象是空对象,那么不能让一个空对象去调用方法产生行为。假如程序中使用了空对象,在运行时会出现异常:NullPointerException。对象是动态地分配实体,Java 的编译器对空对象不做检查。因此,在编写程序时要避免使用空对象。

7. 实验后的练习

(1) 改进 speedUP 方法,使得 Tank 类的对象加速时不能将 speed 值超过 220。

(2) 增加一个刹车方法:void brake(),Tank 类的对象调用它能将 speed 的值变成 0。

8. 填写实验报告

实验编号:401　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2　计算机与光盘

1. 相关知识点

类的成员变量可以是某个类的对象,如果用这样的类创建对象,那么该对象中就会有其他对象,也就是说该类的对象将其他对象作为自己的组成部分,这就是人们常说的 Has-A。一个对象 a 通过组合对象 b 来复用对象 b 的方法,即对象 a 委托对象 b 调用其方法。当前对象随时可以更换所组合对象,使得当前对象与所组合的对象是弱耦合关系。

2. 实验目的

本实验的目的是让学生掌握对象的组合以及参数传递。

3. 实验要求

编写一个 Java 应用程序，模拟在计算机中放入光盘，即计算机将 CD 类型的对象作为自己的一个成员变量。具体要求如下。

(1) 有三个源文件：Computer.java、CD.java 和 User.java，其中 CD.java 中的 CD 类负责创建光盘对象。Computer.java 中的 Computer 类有类型是 CD，名字是 includeCD 的成员变量，Computer 类负责创建计算机对象。User.java 是主类。

(2) 在主类的 main 方法中首先使用 CD 类创建一个对象：dataCD，然后使用 Computer 类再创建一个对象：computerIMB，computerIMB 对象将 CD 类的实例“dataCD”的引用传递给 computerIMB 对象的成员变量“includeCD”。

Computer 类组合 CD 类的实例的 UML 图如图 4.4 所示。

4. 程序效果示例

程序运行效果如图 4.5 所示。

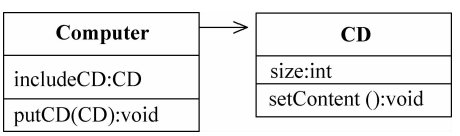

图 4.4　Computer 组合 CD 的 UML 图

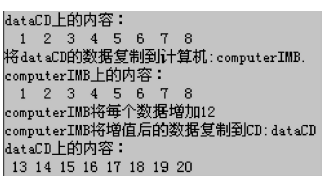

图 4.5　计算机与 CD

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

```
//CD.java
public class CD {
   int size;
   int content[];
   public void setSize(int size) {
       this.size = size;
       content = new int[size];
   }
   public int getSize() {
       return size;
   }
   public int [] getContent() {
       return content;
   }
   public void setContent(int [] b) {
       int min = Math.min(content.length,b.length);
       for(int i = 0;i < min;i++)
         content[i] = b[i];
   }
}
//Computer.java
```

```
public class Computer {
   int data[];
   CD includeCD;
   public void putCD(CD cd) {
       includeCD = cd;
       int size = includeCD.getSize();
       data = new  int[size];
   }
   void copyToComputer() {
       int [] b = includeCD.getContent();
       int min = Math.min(data.length,b.length);
       for(int i = 0;i<min;i++) {
          data[i] = b[i];
       }
   }
   public void addData(int m) {
       for(int i = 0;i<data.length;i++) {
         data[i] = data[i] + m;
       }
   }
   void copyToCD() {
       includeCD.setContent(data);
   }
   void showData() {
       for(int i = 0;i<data.length;i++) {
          System.out.printf("%3d",data[i]);
       }
   }
}
//User.java
public class User {
   public static void main(String args[]) {
       CD dataCD = new CD();
       int b[] = {1,2,3,4,5,6,7,8};
       dataCD.setSize(b.length);
       dataCD.setContent(b);
       int a[] = dataCD.getContent();
       System.out.println("dataCD 上的内容:");
       for(int i = 0;i<a.length;i++)
          System.out.printf("%3d",a[i]);
       Computer computerIMB = new Computer();
       【代码 1】//computerIMB 调用 putCD(CD cd)方法,将 dataCD 的引用传递给 cd
       System.out.println("\n 将 dataCD 的数据复制到计算机:computerIMB.");
       【代码 2】//computerIMB 调用 copyToComputer()方法
       System.out.println("computerIMB 上的内容:");
       computerIMB.showData();
       int m = 12;
       System.out.println("\ncomputerIMB 将每个数据增加" + m);
       computerIMB.addData(m);
       System.out.println("computerIMB 将增值后的数据复制到 CD:dataCD");
       【代码 3】//computerIMB 调用 copyToCD()方法
```

```
        System.out.println("dataCD 上的内容: ");
        a = dataCD.getContent();
        for(int i = 0;i < a.length;i++)
           System.out.printf(" % 3d",a[i]);
    }
}
```

6. 实验指导

当参数是引用类型时,“传值”传递的是变量中存放的“引用”,而不是变量所引用的实体。需要注意的是,对于两个同类型引用型变量,如果具有同样的引用,就会用同样的实体,因此,如果改变参数变量所引用的实体,就会导致原变量的实体发生同样的变化。通过组合对象来复用方法也称“黑盒”复用,因为当前对象只能委托所包含的对象调用其方法,这样一来,当前对象对所包含的对象的方法的细节是一无所知的。

7. 实验后的练习

主类中再增加一个 CD 的对象,然后将计算机中的数据(data 数组)复制到 CD 对象中。

8. 填写实验报告

实验编号:402　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 3　家族的姓氏

1. 相关知识点

类有两种基本的成员:变量和方法。变量用来刻画对象的属性;方法用来体现对象的行为(功能),即方法使用某种算法操作变量来实现一个具体的行为(功能)。

成员变量用来刻画类创建的对象的属性,其中一部分成员变量称为实例变量,另一部分称为静态变量或类变量。类变量是与类相关联的变量,而实例变量是仅仅和对象相关联的变量。不同对象的实例变量将被分配不同的内存空间,如果类中有类变量,那么所有对象的这个类变量都分配给相同的一处内存,改变其中一个对象的这个类变量会影响其他对象的这个类变量。也就是说,对象共享类变量。

方法是类体的重要成员之一。其中的构造方法是具有特殊地位的方法,供类创建对象时使用,用来给出类所创建的对象的初始状态,另一部分方法可分为实例方法和类方法,类所创建的对象可以调用这些方法形成一定的算法,体现对象具有某种行为。一个类的类方法也可以用该类的类名调用。

当对象调用方法时,方法中出现的成员变量就是指分配给该对象的变量。类中的方法可以操作成员变量,当对象调用方法时,方法中出现的成员变量就是指分配给该对象的变量,方法中出现的类变量也是该对象的变量,只不过这个变量和所有的其他对象共享而已。

实例方法可操作实例成员变量和静态成员变量，静态方法只能操作静态成员变量，如图4.6所示。

2. 实验目的

本实验的目的是让学生掌握类变量与实例变量，以及类方法与实例方法的区别。

3. 实验要求

编写程序模拟一个家族成员的姓名，姓名由两部分构成：姓氏和名字。编写一个FamilyPerson类，该类有一个静态的String型成员变量surname，用于存储姓氏、一个实例的String型成员变量name，用于存储名字。在主类MainClass的main方法中首先用类名访问surname，并为surname赋值，然后FamilyPerson创建3个对象：father，sonOne和sonTwo，并分别为father，sonOne和sonTwo的成员变量name赋值。

4. 程序效果示例

程序运行效果如图4.7所示。

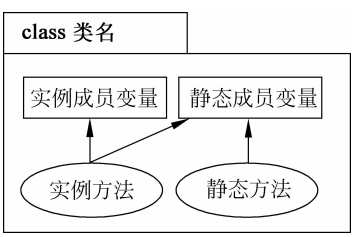

图4.6　实例成员与类成员

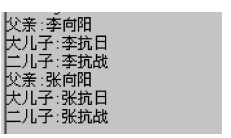

图4.7　家庭的成员的名字

5. 程序模板

按模板要求，将【代码】部分替换为Java程序代码。

```
//FamilyPerson.java
public class FamilyPerson {
    static String surname;
    String name;
    public static void setSurname(String s){
       surname = s;
    }
    public void setName(String s) {
      name = s;
    }
}
//MainClass.java
public class MainClass {
   public static void main(String args[]) {
       【代码1】//用类名FamilyPerson访问surname,并为surname赋值:"李"
       FamilyPerson father,sonOne,sonTwo;
       father = new  FamilyPerson();
       sonOne = new  FamilyPerson();
       sonTwo = new  FamilyPerson();
       【代码2】//father调用setName(String s),并向s传递"向阳"
       sonOne.setName("抗日");
```

```
        sonTwo.setName("抗战");
        System.out.println("父亲:" + father.surname + father.name);
        System.out.println("大儿子:" + sonOne.surname + sonOne.name);
        System.out.println("二儿子:" + sonTwo.surname + sonTwo.name);
        【代码 3】// father 调用 setSurName(String s),并向 s 传递"张"
        System.out.println("父亲:" + father.surname + father.name);
        System.out.println("大儿子:" + sonOne.surname + sonOne.name);
        System.out.println("二儿子:" + sonTwo.surname + sonTwo.name);
    }
}
```

6. 实验指导

当 Java 程序执行时,类的字节码文件被加载到内存,如果该类没有创建对象,类的实例变量不会被分配内存。但是,类中的类变量,在该类被加载到内存时,就分配了相应的内存空间。如果该类创建对象,那么不同对象的实例变量互不相同,即分配不同的内存空间,而类变量不再重新分配内存,所有的对象共享类变量。

7. 实验后的练习

(1)【代码 3】是否可以是 FamilyPerson.setSurname("张");

(2) 能否将主类中的代码:

```
sonOne.setName("抗日");
```

修改为:

```
FamilyPerson.setName("抗日");
```

8. 填写实验报告

实验编号:403　　学生姓名:　　实验时间:　　教师签字:

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实 验 答 案

实验 1

【代码 1】 double speed;

【代码 2】 int bulletAmount;

【代码 3】 speed=s+speed;

【代码 4】 speed=speed-d;

【代码 5】 bulletAmount = bulletAmount-1;

实验 2

【代码 1】 computerIMB.putCD(dataCD);

【代码 2】 computerIMB. copyToComputer();

【代码 3】 computerIMB. copyToCD();

实验 3

【代码 1】 FamilyPerson. surname="李";

【代码 2】 father. setName("向阳");

【代码 3】 father. setSurname("张");

第5章 继承与接口

实验1 中国人与美国人

1. 相关知识点

由继承而得到的类称为子类，被继承的类称为父类(超类)，Java 不支持多重继承，即子类只能有一个父类。人们习惯地称子类与父类的关系是 is-a 关系。

如果子类和父类在同一个包中，那么，子类自然地继承了父类中不是 private 的成员变量作为子类的成员变量，并且也自然地继承了父类中不是 private 的方法作为子类的方法，继承的成员变量或方法的访问权限保持不变。子类和父类不在同一个包中时，父类中的 private 和友好访问权限的成员变量不会被子类继承，也就是说，子类只继承父类中的 protected 和 public 访问权限的成员变量作为子类的成员变量；同样，子类只继承父类中的 protected 和 public 访问权限的方法作为子类的方法。

子类声明的成员变量的名字和从父类继承来的成员变量的名字相同时，将隐藏掉所继承的成员变量。方法重写是指：子类中定义一个方法，这个方法的类型和父类的方法的类型一致或者是父类的方法的类型的子类型，并且这个方法的名字、参数个数、参数的类型和父类的方法完全相同。子类如此定义的方法称为子类重写的方法。

子类继承的方法所操作的成员变量一定是被子类继承或隐藏的成员变量。重写方法既可以操作继承的成员变量、调用继承的方法，也可以操作子类新声明的成员变量、调用新定义的其他方法，但无法操作被子类隐藏的成员变量和方法。

2. 实验目的

本实验的目的是让学生巩固下列知识点：

- 子类的继承性；
- 子类对象的创建过程；
- 方法的继承与重写。

3. 实验要求

编写程序模拟中国人、美国人，北京人。除主类外，程序中有 4 个类：People、ChinaPeople、AmericanPeople 和 BeijingPeople 类。要求如下：

(1) People 类有权限是 protected 的 double 型成员变量：height 和 weight，以及 public void speakHello()、public void averageHeight()和 public void averageWeight()方法。

(2) ChinaPeople 类是 People 的子类，新增了 public void chinaGongfu()方法。要求 ChinaPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法。

(3) AmericanPeople 类是 People 的子类，新增 public void americanBoxing()方法。要求 AmericanPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法。

(4) BeijingPeople 类是 ChinaPeople 的子类，新增 public void beijingOpera()方法。要求 ChinaPeople 重写父类的 public void speakHello()、public void averageHeight()和 public void averageWeight()方法。

4. 运行效果示例

程序运行效果如图 5.1 所示。

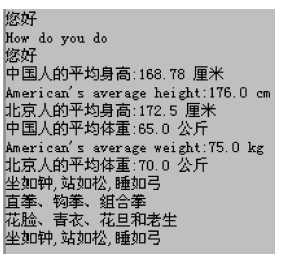

图 5.1　成员的继承与重写

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

```
//People.java
public class People {
   protected double weight,height;
   public void speakHello() {
       System.out.println("yayayaya");
   }
   public void averageHeight() {
       height = 173;
       System.out.println("average height:" + height);
   }
   public void averageWeight() {
       weight = 70;
       System.out.println("average weight:" + weight);
   }
}
//ChinaPeople.java
public class ChinaPeople extends People {
     public void speakHello() {
         System.out.println("您好");
     }
     public void averageHeight() {
         height = 168.78;
         System.out.println("中国人的平均身高:" + height + " 厘米");
     }
    【代码 1】//重写 public void averageWeight()方法，输出:"中国人的平均体重:65 公斤"
     public void chinaGongfu() {
         System.out.println("坐如钟,站如松,睡如弓");
     }
}
//AmericanPeople.java
public class AmericanPeople extends People {
    【代码 2】//重写 public void speakHello()方法，输出"How do you do"
    【代码 3】//重写 public void averageHeight()方法，输出"American's average height:176 cm"
     public void averageWeight() {
         weight = 75;
         System.out.println("American's average weight:" + weight + " kg");
     }
```

```
    public void americanBoxing() {
        System.out.println("直拳、钩拳、组合拳");
    }
}
//BeijingPeople.java
public class BeijingPeople extends ChinaPeople {
    【代码 4】//重写 public void averageHeight()方法,输出:"北京人的平均身高:172.5 厘米"
    【代码 5】//重写 public void averageWeight()方法,输出:"北京人的平均体重:70 公斤"
   public void beijingOpera() {
       System.out.println("花脸、青衣、花旦和老生");
   }
}
//Example.java
public class Example {
   public static void main(String args[]) {
       ChinaPeople chinaPeople = new ChinaPeople();
       AmericanPeople americanPeople = new AmericanPeople();
       BeijingPeople beijingPeople = new BeijingPeople();
       chinaPeople.speakHello();
       americanPeople.speakHello();
       beijingPeople.speakHello();
       chinaPeople.averageHeight();
       americanPeople.averageHeight();
       beijingPeople.averageHeight();
       chinaPeople.averageWeight();
       americanPeople.averageWeight();
       beijingPeople.averageWeight();
       chinaPeople.chinaGongfu();
       americanPeople.americanBoxing();
       beijingPeople.beijingOpera() ;
       beijingPeople.chinaGongfu();
   }
}
```

6. 实验指导

如果子类可以继承父类的方法,子类就有权利重写这个方法,子类通过重写父类的方法可以改变方法的具体行为。方法重写时一定要保证方法的名字、类型、参数个数和类型同父类的某个方法完全相同,只有这样,子类继承的这个方法才被隐藏。子类在重写方法时,不可以将实例方法更改成类方法;也不可以将类方法更改为实例方法,即如果重写的方法是 static 方法,static 关键字必须要保留;如果重写的方法是实例方法,重写时不可以用 static 修饰该方法。

7. 实验后的练习

People 类中的

```
public void speakHello()
public void averageHeight()
public void averageWeight()
```

三个方法的方法体中的语句是否可以省略?

8. 填写实验报告

实验编号：501　　学生姓名：　　实验时间：　　教师签字：

实验效果评价	A	B	C	D	E
模板完成情况					
实验后练习效果评价	A	B	C	D	E
练习完成情况					
总评					

实验 2　银行与利息

1. 相关知识点

子类一旦隐藏了继承的成员变量，那么子类创建的对象就不再拥有该变量，该变量将归关键字 super 所拥有，同样子类一旦重写了继承的方法，就覆盖（隐藏）了继承的方法，那么子类创建的对象就不能调用被覆盖（隐藏）的方法，该方法的调用由关键字 super 负责。因此，如果在子类中想使用被子类隐藏的成员变量或覆盖的方法就需要使用关键字 super。比如 super.x、super.play()就是访问和调用被子类隐藏的成员变量 x 和方法 play()。

2. 实验目的

本实验的目的是让学生掌握重写的目的以及怎样使用 super 关键字。

3. 实验要求

假设银行 Bank 已经有了按整年 year 计算利息的一般方法，其中 year 只能取正整数。如按整年计算的方法：

```
double computerInterest() {
   interest = year * 0.35 * savedMoney;
   return interest;
}
```

建设银行 ConstructionBank 是 Bank 的子类，准备隐藏继承的成员变量 year，并重写计算利息的方法，即 ConstructionBank 类声明一个 double 型的 year 变量，比如，当 year 取值是 5.216 时，表示要计算 5 年零 216 天的利息，但希望首先按银行 Bank 类的方法 computerInterest()计算出 5 整年的利息，然后 ConstructionBank 类再计算 216 天的利息。那么，ConstructionBank 类（建设银行）就必须把 5.216 的整数部分赋给隐藏的 year，并让 super 调用隐藏的、按整年计算利息的方法。

要求 ConstructionBank 和 BankOfDalian 类是 Bank 类的子类，ConstructionBank 和 BankOfDalian 都使用 super 调用隐藏的成员变量和方法。

4. 运行效果示例

程序运行效果如图 5.2 所示。

```
8000元存在建设银行8年零236天的利息:2428.800000元
8000元存在大连银行8年零236天的利息:2466.560000元
两个银行利息相差37.760000元
```

图 5.2　银行计算利息

5. 程序模板

按模板要求，将【代码】部分替换为 Java 程序代码。

```
//Bank.java
public class Bank {
   int savedMoney;
   int year;
   double interest;
   double interestRate = 0.29;
   public double computerInterest() {
       interest = year * interestRate * savedMoney;
       return interest;
   }
   public void setInterestRate(double rate) {
       interestRate = rate;
   }
}
//ConstructionBank.java
public class ConstructionBank extends Bank {
   double year;
   public double computerInterest() {
       super.year = (int)year;
       double r = year - (int)year;
       int day = (int)(r * 1000);
       double yearInterest = 【代码 1】//super 调用隐藏的 computerInterest()方法
       double dayInterest = day * 0.0001 * savedMoney;
       interest = yearInterest + dayInterest;
       System.out.printf("%d 元存在建设银行 %d 年零 %d 天的利息:%f 元\n",
                         savedMoney,super.year,day,interest);
       return interest;
   }
}
//BankOfDalian.java
public class BankOfDalian extends Bank {
   double year;
   public double computerInterest() {
       super.year = (int)year;
       double r = year - (int)year;
       int day = (int)(r * 1000);
       double yearInterest = 【代码 2】// super 调用隐藏的 computerInterest()方法
       double dayInterest = day * 0.00012 * savedMoney;
       interest = yearInterest + dayInterest;
       System.out.printf("%d 元存在大连银行 %d 年零 %d 天的利息:%f 元\n",
                         savedMoney,super.year,day,interest);
       return interest;
   }
}
//SaveMoney.java
public class SaveMoney {
   public static void main(String args[]) {
       int amount = 8000;
```