



21世纪高等学校电子信息类专业规划教材

Java

语言实用教程

主 编 于万波
副主编 刘竹林 刘其昌 刘 君



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类专业规划教材

Java 语言实用教程

于万波 主 编

刘竹林 刘其昌 刘 君 副主编



清华大学出版社
北京交通大学出版社

· 北京 ·

内 容 简 介

该书利用较多的实例由浅入深地讲解了 Java 的一些基本内容。

第 1 章是 Java 程序的简单介绍,通过这一章的学习可以初步了解 Java。第 2 章讲解了如何利用 Java 制作图形用户界面,包括基本的与图形界面相关的类的使用。图形图像动画程序设计放在第 3 章中。第 4 章与第 5 章讲述语言规范、语法规则,第 6 章学习几类网络程序设计,第 7 章对数据库操作进行了介绍。

该书可作为学习 Java 的各专业学生的教材,也可以作为高级程序设计及面向对象课程的参考书,同时也可作为自学者及其他计算机爱好者的参考用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

Java 语言实用教程/于万波主编. —北京:清华大学出版社;北京交通大学出版社,2008.9
ISBN 978-7-81123-402-2

I. J… II. 于… III. JAVA 语言-程序设计-教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 143667 号

责任编辑:郭东青

出版发行:清华大学出版社 邮编:100084 电话:010-62776969

北京交通大学出版社 邮编:100044 电话:010-51686414

印刷者:北京东光印刷厂

经 销:全国新华书店

开 本:185×260 印张:22.75 字数:562 千字

版 次:2008 年 10 月第 1 版 2008 年 10 月第 1 次印刷

书 号:ISBN 978-7-81123-402-2/TP·437

印 数:1~4 000 册 定价:35.00 元

本书如有质量问题,请向北京交通大学出版社质监组反映。对您的意见和批评,我们表示欢迎和感谢。

投诉电话:010-51686043, 51686008; 传真:010-62225406; E-mail: press@bjtu.edu.cn。

前 言

每一本书都有自己的特点，都拥有一些读者。本书的特点是充分考虑了初学程序设计人员的需求，利用逐层深入、递进嵌套的方式有机地组织内容。把一些重点难点知识分散开，把一些相关的例题排列在一起，以此讲解一些知识点，总结一些规律性的东西。

有的地方借鉴了其他一些优秀的教材或专著，这些著作都已经列在参考文献中，在此谨对它们的作者表示敬意与感谢。有个别程序是作者多年教学过程中积累下来的，可能取自于某处，但已无从考证，好在已经经过改进或加入了详尽的解释。

多数程序是根据教学的需要由作者自己设计的。

Java 是庞大的，它见证着计算机软件学科的发展，囊括着计算机网络程序设计、分布式程序设计的所有思想方法。作为一本入门教程，选择内容有一定的难度，也许再有两三本后续教程作为辅助，对进一步掌握 Java 更为有利。

作者认为，在学习的过程中要善于思考、善于钻研、善于实践，本书的设计在很大程度上体现了这种思想。

读者通过学习本书内容，能够得到一些收获，是作者最希望的。书中缺点一定有很多，还希望读者指正。

陈微微参加了本书前 3 章的编写，王凯同学阅读了本书的初稿，于硕同学调试修改了其中的一些程序，在此一并表示感谢。

作 者

2008 年 8 月于大连大学

目 录

第 1 章 Java 程序介绍	1
1.1 Java 应用程序	1
1.1.1 Java 与 JDK	1
1.1.2 应用程序示例	4
1.2 类的基本知识	19
1.2.1 类的结构	19
1.2.2 类的继承与重用	35
1.3 程序设计	38
1.3.1 变量与表达式	39
1.3.2 分支	44
1.3.3 循环	47
1.4 文件操作	50
1.4.1 文件与目录	51
1.4.2 读文件	53
1.4.3 写文件	54
1.5 高层网络编程	55
习题	60
第 2 章 图形用户界面设计	65
2.1 常用的 GUI 组件	65
2.1.1 文本与标签	65
2.1.2 按钮	77
2.1.3 列表、下拉列表与选择按钮	80
2.1.4 滚动条	86
2.2 事件	87
2.2.1 鼠标事件	88
2.2.2 键盘事件	93
2.2.3 焦点事件	98
2.3 容器	101
2.3.1 布局	101
2.3.2 面板	105
2.3.3 Component 类	107
2.3.4 对话框	108
2.4 菜单	111
2.5 swing 组件	121

习题	127
第3章 图形图像编程与动画制作	132
3.1 图形绘制	132
3.1.1 基本绘图方法	132
3.1.2 颜色设置	135
3.1.3 绘图实例	137
3.2 图像操作	149
3.3 动画制作	151
3.3.1 图像动画制作	151
3.3.2 图形动画制作	154
3.3.3 音乐播放	161
习题	162
第4章 程序设计规范	165
4.1 数据类型	165
4.1.1 常用数据类型之间的转换	165
4.1.2 数组	169
4.1.3 字符串	175
4.1.4 向量	178
4.2 分支与循环结构	181
4.2.1 分支结构	181
4.2.2 循环结构	186
4.2.3 跳转语句	190
4.3 方法的重载	192
4.3.1 构造方法	192
4.3.2 重载	195
4.3.3 对父类方法的覆盖	196
4.4 接口	198
4.4.1 接口的实现	198
4.4.2 接口的定义	199
4.4.3 接口的继承	200
4.5 线程	201
4.6 异常	216
习题	221
第5章 类的构造	230
5.1 类的组织	230
5.1.1 包	230
5.1.2 修饰控制符	232
5.1.3 内嵌类	237

5.2	AWT 包	238
5.2.1	AWT 包的组成	238
5.2.2	Button 类	240
5.3	数学类	246
5.3.1	lang 包中的数学类 Math	246
5.3.2	数学类 Math 的使用	251
5.4	Applet 类	254
5.4.1	Applet 与 HTML	254
5.4.2	网页中的 Applet 通信	257
5.4.3	生命周期	260
5.4.4	Applet 常用方法	262
5.4.5	Applet 类与相关接口定义	266
5.5	输入输出类	273
	习题	281
第 6 章	输入输出与网络程序设计	283
6.1	Socket 聊天程序设计	283
6.1.1	基于命令窗口的聊天程序	283
6.1.2	相关的类与方法	290
6.1.3	继承 Frame 类的聊天程序	293
6.2	数据报发送与接收	304
6.2.1	数据报通信程序设计	304
6.2.2	组播程序设计	314
6.2.3	相关的类与方法	316
6.3	RMI 程序设计	317
6.4	输入输出	323
6.4.1	数据输入流与数据输出流	323
6.4.2	FileInputStream 类与 FileOutputStream 类	325
	习题	330
第 7 章	数据库管理程序设计	335
7.1	数据库表	335
7.2	简单的数据库表操作	338
	习题	352
	参考文献	354

第 1 章 Java 程序介绍

Java 程序设计是该书主要介绍的内容,从本章开始逐步学习 Java 类及面向对象的知识、Java 语言规范、Java 程序设计技术、使用语言解决实际问题等。

1.1 Java 应用程序

1.1.1 Java 与 JDK

与其他语言一样,如果要在机器上运行,必须在机器上装有该语言的编辑编译及运行系统。事实上,编辑软件有很多,如 Windows 所带的记事本就可以编辑 Java 程序。Java 的编译及运行系统也有很多,本书使用 SUN 公司免费提供的系统 JDK (Java Develepment Kit)。

进入 <http://www.sun.com/>,在网站中选择“Download”,在弹出的网页中选择“Java”产品,本书选择了

Java EE 5 SDK Update 4 (with JDK 6u5)

该程序大小为 160 MB 左右。下载完成后,双击该程序的图标进行安装。按照提示要求填写选项,一步一步完成安装。作者的安装目录为 C:\Program Files\Java\jdk1.6.0。

如果在你的机器上已经装有 JDK,那么就可以按照例 1-1 进行 Java 程序编辑、编译与运行了。

【例 1-1】 使用 Java 程序进行简单的输出。

首先,打开 Windows 附件中的记事本,输入下面程序:

```
public class C
{
    public static void main(String args[])
    {
        System.out.println("Hello,Welcome");
    }
}
```

然后,选择记事本“文件”菜单中的“另存为”,在出现的对话框中把文件名写为 C.java,如图 1-1 所示。此处的文件名必须与类(class)名称 C 一致,如果你在程序中定义的类名称为:

```
public class C1
```

那么另存为文件时,文件的名称应该为 C1.java。

另外，注意一定把保存类型选为所有文件，如图 1-1 所示。

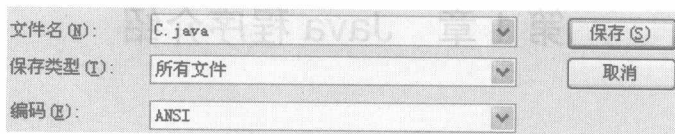


图 1-1 在记事本中保存 Java 文件

文件保存的位置是有要求的，在这个例子中，把所做的 C.java 存储在 C:\Program Files\Java\jdk1.6.0\example 目录下。同时把 C:\Program Files\Java 中的名称为 java 与 javac 的文件，还有 jli.dll 文件复制到 C:\Program Files\Java\jdk1.6.0\example 目录中，如图 1-2 所示。



图 1-2 把 java 与 javac 等文件复制到工作文件夹中

打开附件中的命令窗口，进入 C:\Program Files\Java\jdk1.6.0\example 目录，如图 1-3 所示。然后输入命令 javac C.java 对程序进行编译，编译成功后，使用命令 java C 运行程序，在屏幕上输出了 Hello,Welcome。

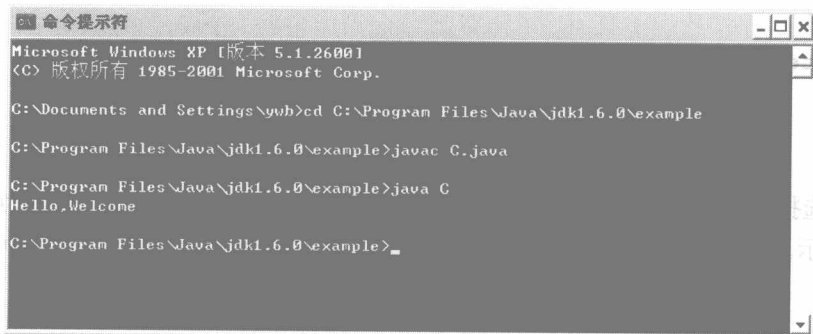


图 1-3 在命令窗口中进入工作目录并编译运行程序

注意 若文件存放在 D:\Program Files\java\jdk1.6.0_06\example 中，则需先用命令 D:

转换到 D:\。

在例 1-1 中讲解了如何编辑、运行 Java 应用程序，输出了一个字符串。例 1-2 也是输出一个字符串，字符串由数字字符与空格组成。

【例 1-2】 使用 Java 程序输出数字。

```
public class shuzi
{
    public static void main(String args[])
    {
        System.out.println("0 1 2 3 4 5 6 7 8 9");
    }
}
```

注意 语句中的标点符号如"与;等都必须是在英文状态下的符号。

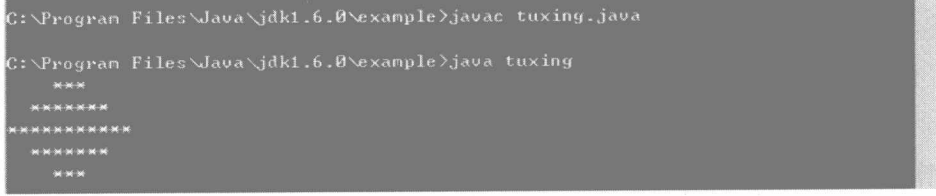
Java 程序都是由类构成的，在一个类中常有一个主函数 main，一般情况下用关键字 public、static、void 等修饰，String args[] 是主函数的参数，String 的第一个字母要大写。

【例 1-3】 使用 Java 程序输出简单图形。

设计下面程序：

```
public class tuxing
{
    public static void main(String args[])
    {
        System.out.println("    ***    ");
        System.out.println("  *      *  ");
        System.out.println("*****");
        System.out.println("  *      *  ");
        System.out.println("    ***    ");
    }
}
```

编译、运行程序后，绘制出如图 1-4 所示图形。



```
C:\Program Files\Java\jdk1.6.0\example>javac tuxing.java
C:\Program Files\Java\jdk1.6.0\example>java tuxing
    ***
  *      *
*****
  *      *
    ***
```

图 1-4 使用输出语句绘制简单图形

程序中，语句 System.out.println 是用来输出的，即在输出设备屏幕上打印一行内容。

System 是一个由 Java 开发人员做好的类，out 类是 System 类的一个子类，println 是 out 类的一个方法。

注意 在使用命令窗口进行编译或运行时，可以用向上或向下箭头调用已经输入的命令。

1.1.2 应用程序示例

1. 算术计算

【例 1-4】 计算两个数的加减乘除。

```
public class jisuan
{
    public static void main(String args[])
    {
        Integer a=1001,b=202,c;
        c=a+b;
        System.out.println(c);
    }
}
```

Integer 定义整型变量，变量类似容器，是用来装数据的，整型变量就是用来装整数的。例题中的 a 可以装 1001，当然也可以装其他的整数；b 与 c 也一样。

注意 变量中每次只能装一个数据。变量中的数据是有范围的。

输出语句 System.out.println(); 在输出变量的值时，只需把变量名称写上就可以了。如例题输出变量 c 的值，只需写成 System.out.println(c);，此时不需要给 c 加双引号。

当既要计算加法也要计算减法的时候，可以使用下面的程序。

```
public class jisuan
{
    public static void main(String args[])
    {
        Integer a=1001,b=202,c;
        c=a+b;
        System.out.println(c);
        c=a-b;
        System.out.println(c);
    }
}
```

计算乘除的时候，可以对该程序依次稍加修改就可以了。

注意 Java 程序体中语句后面必须要写上分号“;”，有分号的目的是为了编译器容易区分各个语句。而 public class jisuan 与 public static void main(String args[]) 后面没有分号。

`public class jisuan` 是用来声明定义类的，该程序中第一个大括号与最后一个大括号中间都属于该类。`public static void main(String args[])` 是用来定义主函数的，也是用大括号把属于它的程序括起来。

【例 1-5】 利用循环语句进行计算。

计算 1 到 10 这 10 个自然数的和。

首先设计下面程序：

```
public class Jisuan1
{
    public static void main(String args[])
    {
        Integer c;
        c=1+2+3+4+5+6+7+8+9+10;
        System.out.println(c);
    }
}
```

上面程序能够计算 1 到 10 这 10 个数的和，不过存在一个缺点，当数据多时，表达式书写起来不方便。观察这些数据之间存在的规律，发现，可以利用 Java 提供的循环语句改写程序如下。

```
public class Jisuan2
{
    public static void main(String args[])
    {
        Integer c=0,i;
        for(i=1;i<=10;i++)
            c=c+i;
        System.out.println(c);
    }
}
```

该程序也是计算 1 到 10 这 10 个自然数的和。语句 `Integer c=0` 是在定义整型变量 `c` 的同时给 `c` 赋值为 0。语句段

```
for(i=1;i<=10;i++)
    c=c+i;
```

是一个整体，称为循环语句。

在执行该循环语句时，首先让 `i` 的值为 1，然后检查 `i<=10` 是否成立，如果成立，那么执行语句 `c=c+i`。此时 `c` 的值为 0，`i` 的值为 1，所以把 `0+1` 得到的 1 赋给变量 `c`，`c` 的值变为 1，然后再执行语句 `i++`。语句 `i++` 相当于 `i=i+1`，就是 `i` 增加 1，即 `i` 变为 2 了。

上面描述的这个过程完成了一次循环,接下来不再执行语句 `i=1;`,直接检查条件 `i<=10;` 看是否成立,此时 `i` 的值为 1,成立,所以继续执行循环体中的语句 `c=c+i;` 把 1+2 得到的 3 赋给变量 `c`,`c` 的值变为 3。然后继续执行语句 `i++`,又完成了一次循环。

当 `i` 增长到 11 的时候,检查条件 `i<=10;`,不满足,那么退出循环。

注意 程序中等号“=”是用来赋值的;在这个程序中,语句 `for(i=1;i<=10;i++)` 的后面没有写分号,括号内有两个分号。

【例 1-6】 编写程序计算 1 到 100 之间自然数中奇数的和。

使用下面程序就可以计算 1 到 100 之间自然数中奇数的和。

```
public class Jisuan3
{
    public static void main(String args[])
    {
        Integer c=0,i;
        for(i=1;i<=100;i=i+2)
            c=c+i;
        System.out.println(c);
    }
}
```

【思考题】 1. 计算 1 到 10 之间自然数的积。

2. 计算 1 到 100 之间自然数的和。

3. 计算 1 到 100 之间自然数中偶数的和。

例 1-3 等需要事先把两个或多个要计算的数写在程序中,这样有很多不方便,有很多时候我们希望能够从键盘等输入数据,然后进行计算。

【例 1-7】 编写程序,程序运行后从键盘输入两个数,就可以计算出这两个数的和、差与积。

设计下面程序:

```
public class C1
{
    public static void main(String args[])
    {
        int a=0,b=0,c;
        a=Integer.parseInt(args[0]);
        b=Integer.parseInt(args[1]);
        c=a+b;
        System.out.println(c);
    }
}
```

输入 `javac C1.java` 按 Enter 键编译成功后, 再输入 `java C1 56 78`, 然后按 Enter 键, 得到计算结果 134。

该程序只完成了求和运算, 很容易修改使其能够计算差或者积等。

在这个例题中我们看到了主函数参数 `args[]` 的使用, 其实 `args` 后面加上中括号 “`[]`” 表示 `args` 是一个数组, 可以存储一组数据。`args[0]` 为这个数组中的第一个元素, `args[1]` 为第二个元素。因为该数组为字符串型, 所以需要使用整数类中的方法 `parseInt` 把输入的数据转换为整型数。

也可以用例 1-8 完成例 1-7 同样的功能。

【例 1-8】 使用 `InputStreamReader` 类与 `BufferedReader` 类完成读取与临时存储操作, 实现从键盘输入两个数然后计算的功能。

```
import java.io.*;

public class C1
{
    public static void main(String args[])
    {
        int a=0,b=0,c;
        String s;
        try
        {
            BufferedReader br=new BufferedReader(new InputStreamReader
                (System.in));
            s=br.readLine();
            a=Integer.parseInt(s);
            s=br.readLine();
            b=Integer.parseInt(s);
        }
        catch(IOException e)
        { }
        c=a+b;
        System.out.println(c);
    }
}
```

运行程序后, 在光标显示行输入 35, 然后按 Enter 键在新的一行输入 78, 再按 Enter 键后得到计算结果 113。

在该程序中, `try` 与 `catch` 是配套使用的捕获异常语句, 含义为: 做 `try` 后面大括号中的语句, 如果异常便执行 `catch` 后面大括号中的语句。

在 Java 中, 有一些时候必须使用异常捕获功能。另外, 异常与错误是不同的, 例如, 在访问网络上某个节点时, 由于对方防护措施没有访问成功, 那么不是程序有错误, 而是异常。由于太多这类网络异常问题存在, Java 作为网络程序设计语言引入了异常的概念与

处理方式。异常的英文为 Exception。

程序中, 语句 `String s;` 是定义一个字符串变量 `s`, 用来存储字符串。

语句 `BufferedReader br=new BufferedReader(new InputStreamReader(System.in));` 中含有非常丰富的内容。`BufferedReader br` 是用来定义对象的, 定义了一个 `BufferedReader` 类型的对象 `br`; `new` 是用来创建对象的, 使用与 `BufferedReader` 类同名的 (创建) 方法 `BufferedReader()` 来创建。`new InputStreamReader(System.in)` 也是创建了一个 `InputStreamReader` 对象, `System.in` 是调用 `System` 类中的 `in` 方法, 负责输入任务。

当定义并创建完对象 `br` 后, 就可以调用 `br` 对象的 `readLine` 方法, 使用语句 `s=br.readLine();` 进行读取工作, 当执行到该语句时, 系统便停留在输入界面等待输入。

【例 1-9】 计算球表面积与体积。

```
import java.io.*;
public class Qiutiji
{
    public static void main(String args[])
    {
        int a=0;
        double b,c;
        String s;
        try{
            BufferedReader br=new BufferedReader(new InputStreamReader
                (System.in));
            s=br.readLine();
            a=Integer.parseInt(s);
        }
        catch(IOException e)
        { }
        b=4*3.14*a*a;
        c=4/3*3.14*a*a*a;
        System.out.println("表面积为"+b+" "+"体积为"+c);
    }
}
```

程序的执行结果如图 1-5 (a) 所示。由于使用了 `BufferedReader` 对象进行输入, 所以在输入命令 `java Qiutiji` 按 Enter 键后, 再输入球半径 5, 然后按 Enter 键, 计算结果显示在下一行。

2. 比较与排序

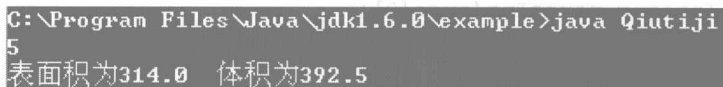
比较与排序是常见的数值计算工作。

【例 1-10】 比较两个数的大小。

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

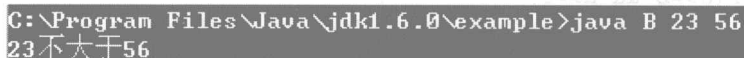
```
{
    int a,b,c;
    a=Integer.parseInt(args[0]);
    b=Integer.parseInt(args[1]);
    if(a>b)
        System.out.println(a+"大于"+b);
    else
        System.out.println(a+"不大于"+b);
}
```

程序的执行结果如图 1-5 (b) 所示。这个例题输入参数的时候就需要把两个参数写在 java B 后面。



```
C:\Program Files\Java\jdk1.6.0\example>java Qiutiji
5
表面积为314.0 体积为392.5
```

(a) 例 1-9 运行结果



```
C:\Program Files\Java\jdk1.6.0\example>java B 23 56
23不大于56
```

(b) 例 1-10 运行结果

图 1-5 两个程序的运行结果

程序中的 if 与 else 都是关键词 (关键词是不可以用作自己定义的变量名称的, 例如 int if 是不可以的)。if 与 else 很多时候一起配对出现, 有时只有 if。

语句 `System.out.println(a+"大于"+b);` 中的加号是完成连接, 属于字符串操作。

【例 1-11】求三个数中最大的。

```
public class bijiaodaxiao
{
    public static void main(String args[])
    {
        int a,b,c;
        a=Integer.parseInt(args[0]);
        b=Integer.parseInt(args[1]);
        c=Integer.parseInt(args[2]);
        if(a>=b && a>=c)
            System.out.println(a+"最大");
        if(b>=a && b>=c)
            System.out.println(b+"最大");
        if(c>=a && c>=b)
            System.out.println(c+"最大");
    }
}
```

程序语句 `if(a>=b && a>=c)` 中符号 “&&” 表示并且。

编译后, 运行可以找出三个数中最大的。不过, 执行命令 `java bijiaodaxiao 71 71 71` 时, 将会打印出 3 次 “71 最大”。

【思考题】 如何修改程序, 使其避免在输入的三个数相同时打印多次。

【例 1-12】 输入三个数, 将三个数从小到大排序。

设计下面程序:

```
public class C5
{
    public static void main(String args[])
    {
        int a,b,c,t1=0,t2=0,t3=0;
        a=Integer.parseInt(args[0]);
        b=Integer.parseInt(args[1]);
        c=Integer.parseInt(args[2]);
        if(a>=b && a>=c)
        {
            t3=a;
            if(b>c)
                {t2=b;t1=c;}
            else
                {t2=c;t1=b;}
        }
        if(b>=a && b>=c)
        {
            t3=b;
            if(a>c)
                {t2=a;t1=c;}
            else
                {t2=c;t1=a;}
        }
        if(c>=a && c>=b)
        {
            t3=c;
            if(a>b)
                {t2=a;t1=b;}
            else
                {t2=b;t1=a;}
        }
    }
}
```