

高等院校信息技术规划教材

# Java程序设计

于万波 等 编著

INFORMATION TECHNOLOGY  
INFORMATION TECHNOLOGY  
INFORMATION TECHNOLOGY

清华大学出版社



高等院校信息技术规划教材

# Java程序设计

于万波 等 编著

清华大学出版社  
北京

## 内 容 提 要

本书讲解 Java 的一些基本内容。

第 1 章是 Java 程序介绍,通过第 1 章的学习可以初步了解 Java。第 2 章讲解图形、图像、动画程序设计以及如何利用 Java 制作图形用户界面。第 3 章给出了一些客户端与服务器的通信程序。第 4 章讲解数据库程序设计并且给出了一个实例。第 5 章讲述语言规范、语法规则。第 6 章介绍 Web 网络程序设计。第 7 章介绍如何使用目前流行的 Eclipse 软件。

本书可以作为高等院校各专业学生学习 Java 的教材,也可以作为高级程序设计以及面向对象课程的参考书。

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 13701121933

## 图书在版编目(CIP)数据

Java 程序设计/于万波等编著. --北京:清华大学出版社,2009.6

(高等院校信息技术规划教材)

ISBN 978-7-302-19448-4

I. J… II. 于… III. JAVA 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 016227 号

责任编辑:袁勤勇 李 晔

责任校对:李建庄

责任印制:孟凡玉

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, [c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

质 量 反 馈:010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者:北京四季青印刷厂

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260 印 张:27

字 数:635 千字

版 次:2009 年 6 月第 1 版

印 次:2009 年 6 月第 1 次印刷

印 数:1~3000

定 价:35.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:010-62770177 转 3103 产品编号:030759-01

# 前言

## foreword

Java 是一种面向对象编程语言,它囊括了计算机网络程序设计、分布式程序设计的所有思想方法。

与其在没有基础的情况下学习软件理论,还不如扎扎实实学习 Java 程序设计、Java 软件开发和 Java 系统设计等。在设计与开发的过程中总结、体会各种理论,培养使用计算机解决实际问题的能力。

本书是用来学习基本的 Java 程序设计的,学好本书内容是进一步学习 Java 所必需的。

书中有些内容是介绍性的,例如第 3 章、第 4 章和第 6 章等。这些内容是非常实用的,通常在专门的教程或专著中讲解,现在尝试性地引入本书中。如何描述这些内容,请读者读后给出建议。作者将感谢那些提出修改意见的读者。

本书充分考虑了初学者的需求,利用逐层深入、递进嵌套的方式有机地组织内容。把一些重点难点分散,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。特别是希望读者学会修改完善程序。

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,这是作者最希望的。书中缺点一定有很多,还希望读者指正(作者邮箱 yu\_wb@126.com)。

陈涛编写了本书的第 6 章与第 7 章,黄艳峰编写了本书的第 5 章,迟春见编写了第 2 章,其余章节由于万波编写。设计、统稿和修正等由于万波完成,于硕调试修改了其中的一些程序,在此一并表示感谢。

作 者

2009 年 4 月

于大连大学

# 目录

# Contents

|       |                                |    |
|-------|--------------------------------|----|
| 第 1 章 | Java 程序介绍                      | 1  |
| 1.1   | JDK                            | 1  |
| 1.2   | 应用程序示例                         | 4  |
| 1.2.1 | 算术计算                           | 4  |
| 1.2.2 | 比较与排序                          | 9  |
| 1.2.3 | 计算分段函数的值                       | 12 |
| 1.2.4 | 窗体及其他组件                        | 13 |
| 1.3   | 类的基本知识                         | 16 |
| 1.3.1 | 类的结构                           | 16 |
| 1.3.2 | Frame 类                        | 19 |
| 1.3.3 | 用于鼠标编程的 MouseAdapter 类         | 24 |
| 1.3.4 | TextField 类、Button 类与 Canvas 类 | 28 |
| 1.3.5 | Applet 类                       | 37 |
| 1.4   | 各种运算与表达式                       | 41 |
| 1.4.1 | 数据类型                           | 42 |
| 1.4.2 | 运算符                            | 45 |
| 1.4.3 | 表达式                            | 47 |
| 1.5   | 文件操作                           | 48 |
| 1.5.1 | 目录的创建与删除                       | 49 |
| 1.5.2 | 文件的创建与删除                       | 51 |
| 1.5.3 | 读文件                            | 54 |
| 1.5.4 | 写文件                            | 57 |
| 1.6   | 显示时间日期                         | 59 |
| 1.7   | 音乐播放                           | 62 |
| 1.8   | 浏览网页                           | 65 |
|       | 习题 1                           | 71 |



|                           |     |
|---------------------------|-----|
| <b>第 2 章 图形、图像与界面设计</b>   | 76  |
| 2.1 图形绘制                  | 76  |
| 2.1.1 基本绘图方法              | 76  |
| 2.1.2 颜色与字体设置             | 82  |
| 2.1.3 绘制函数曲线图形            | 87  |
| 2.1.4 继承 Frame 类在窗体上绘图    | 90  |
| 2.1.5 Graphics 2D         | 94  |
| 2.2 图像操作                  | 102 |
| 2.3 动画制作                  | 107 |
| 2.3.1 图像动画制作              | 107 |
| 2.3.2 图形动画制作              | 112 |
| 2.4 常用的 GUI 组件            | 115 |
| 2.4.1 文本与标签               | 116 |
| 2.4.2 按钮                  | 127 |
| 2.4.3 列表、下拉列表与选择按钮        | 132 |
| 2.5 事件                    | 145 |
| 2.5.1 鼠标事件                | 146 |
| 2.5.2 键盘事件                | 152 |
| 2.5.3 焦点事件                | 157 |
| 2.5.4 滚动条调整事件             | 159 |
| 2.5.5 容器事件 ContainerEvent | 163 |
| 2.6 容器                    | 165 |
| 2.6.1 布局                  | 165 |
| 2.6.2 面板                  | 172 |
| 2.6.3 对话框                 | 174 |
| 2.7 菜单                    | 179 |
| 习题 2                      | 188 |
| <b>第 3 章 服务器与客户端程序设计</b>  | 193 |
| 3.1 Socket 聊天程序设计         | 193 |
| 3.1.1 基于命令窗口的聊天程序         | 193 |
| 3.1.2 相关的类与方法             | 200 |
| 3.1.3 继承 Frame 类的聊天程序     | 207 |
| 3.2 数据报发送与接收              | 212 |
| 3.2.1 数据报通信程序设计           | 212 |
| 3.2.2 组播程序设计              | 223 |
| 3.2.3 相关的类与方法             | 226 |

|              |                                             |            |
|--------------|---------------------------------------------|------------|
| 3.3          | RMI 程序设计 .....                              | 227        |
| 3.4          | 字节输入输出类 .....                               | 235        |
| 3.4.1        | FileInputStream 类与 FileOutputStream 类 ..... | 236        |
| 3.4.2        | DataInputStream 类与 DataOutputStream 类 ..... | 239        |
| 3.5          | 字符输入输出类 .....                               | 244        |
| 3.5.1        | Reader 类及其子类 .....                          | 245        |
| 3.5.2        | Writer 类及其子类 .....                          | 250        |
| 习题 3         | .....                                       | 253        |
| <b>第 4 章</b> | <b>数据库管理程序设计 .....</b>                      | <b>259</b> |
| 4.1          | 数据库表 .....                                  | 259        |
| 4.2          | 简单的数据库表操作 .....                             | 262        |
| 4.2.1        | 显示数据表中的数据 .....                             | 262        |
| 4.2.2        | 添加记录 .....                                  | 267        |
| 4.2.3        | 更新记录 .....                                  | 270        |
| 4.3          | 学生信息管理系统 .....                              | 276        |
| 4.3.1        | 学生信息管理系统主界面设计 .....                         | 276        |
| 4.3.2        | 数据库类的设计 .....                               | 280        |
| 4.3.3        | 学生信息添加类的设计 .....                            | 282        |
| 习题 4         | .....                                       | 291        |
| <b>第 5 章</b> | <b>Java 语言规范 .....</b>                      | <b>293</b> |
| 5.1          | 数据类型 .....                                  | 293        |
| 5.1.1        | 基本数据类型与字符串类型进行转换 .....                      | 293        |
| 5.1.2        | 字符串 .....                                   | 296        |
| 5.1.3        | 数组 .....                                    | 299        |
| 5.1.4        | 向量 .....                                    | 309        |
| 5.2          | 分支与循环结构 .....                               | 311        |
| 5.2.1        | 分支结构 .....                                  | 311        |
| 5.2.2        | 循环结构 .....                                  | 316        |
| 5.2.3        | 跳转语句 .....                                  | 321        |
| 5.3          | 方法的重载 .....                                 | 323        |
| 5.3.1        | 构造方法 .....                                  | 323        |
| 5.3.2        | 重载 .....                                    | 326        |
| 5.3.3        | 对父类方法的覆盖 .....                              | 326        |
| 5.4          | 接口 .....                                    | 327        |
| 5.4.1        | 接口的实现 .....                                 | 327        |



|              |                                                |            |
|--------------|------------------------------------------------|------------|
| 5.4.2        | 接口的定义 .....                                    | 328        |
| 5.4.3        | 接口的继承 .....                                    | 329        |
| 5.5          | 线程 .....                                       | 331        |
| 5.5.1        | 线程类 Thread .....                               | 331        |
| 5.5.2        | 线程接口 Runnable .....                            | 336        |
| 5.6          | 异常 .....                                       | 338        |
| 5.7          | 类的组织 .....                                     | 340        |
| 5.7.1        | 包 .....                                        | 341        |
| 5.7.2        | AWT 包的组成 .....                                 | 342        |
| 5.7.3        | 内嵌类 .....                                      | 343        |
| 5.8          | 修饰控制符 .....                                    | 344        |
| 5.8.1        | 访问控制符 .....                                    | 344        |
| 5.8.2        | static 修饰符 .....                               | 347        |
| 5.8.3        | Abstract 与 final .....                         | 348        |
| 习题 5         | .....                                          | 348        |
| <b>第 6 章</b> | <b>Web 程序设计 .....</b>                          | <b>358</b> |
| 6.1          | Tomcat 简介及配置 .....                             | 358        |
| 6.1.1        | Tomcat 容器简介 .....                              | 358        |
| 6.1.2        | Tomcat 安装及配置 .....                             | 359        |
| 6.1.3        | 创建和发布 Web 应用 .....                             | 360        |
| 6.2          | Servlet 技术 .....                               | 365        |
| 6.2.1        | Servlet 简介 .....                               | 365        |
| 6.2.2        | Servlet API .....                              | 365        |
| 6.2.3        | Servlet 的生命周期及主要方法 .....                       | 367        |
| 6.2.4        | GenericServlet 和 HttpServlet .....             | 369        |
| 6.2.5        | HttpServletRequest 和 HttpServletResponse ..... | 370        |
| 6.2.6        | 创建和运行 HttpServlet 的步骤 .....                    | 371        |
| 6.3          | JSP 技术 .....                                   | 375        |
| 6.3.1        | JSP 简介 .....                                   | 375        |
| 6.3.2        | JSP 语法 .....                                   | 376        |
| 6.3.3        | JSP 简单示例 .....                                 | 379        |
| 6.4          | JavaBean 技术 .....                              | 381        |
| 6.4.1        | JavaBean 简介 .....                              | 381        |
| 6.4.2        | JavaBean 属性 .....                              | 382        |
| 6.4.3        | JSP 访问 JavaBean 的语法 .....                      | 384        |
| 6.4.4        | JavaBean 的范围 .....                             | 385        |
| 习题 6         | .....                                          | 386        |

|                                         |     |
|-----------------------------------------|-----|
| <b>第 7 章 Eclipse 开发环境</b> .....         | 387 |
| 7.1 Eclipse 平台安装及配置 .....               | 387 |
| 7.2 Java 集成开发环境 .....                   | 388 |
| 7.2.1 开发界面简介 .....                      | 388 |
| 7.2.2 创建 Java 项目 .....                  | 388 |
| 7.2.3 编译和运行 Java 项目 .....               | 390 |
| 7.2.4 调试 Java 项目 .....                  | 392 |
| 7.3 使用 SWT 开发 GUI 程序 .....              | 396 |
| 7.3.1 SWT 概述 .....                      | 396 |
| 7.3.2 SWT 包简介 .....                     | 396 |
| 7.3.3 SWT Designer 下载及安装 .....          | 397 |
| 7.3.4 SWT Designer 的注册激活 .....          | 398 |
| 7.3.5 使用 SWT Designer 创建简单 GUI 程序 ..... | 401 |
| 习题 7 .....                              | 406 |
| <b>附录 A Java 的发展历史</b> .....            | 407 |
| <b>附录 B Java 的体系结构</b> .....            | 410 |
| <b>附录 C Java 语法补充</b> .....             | 412 |
| <b>附录 D 实用功能拾遗</b> .....                | 416 |
| <b>参考文献</b> .....                       | 418 |

# 第 1 章

# chapter 1

## Java 程序介绍

Java 从 C 与 C++ 演化而来,更加适合于网络程序设计与分布式程序设计。Java 及其相关产品已经成为程序设计与开发的主要工具之一。Java 既是工具,也是一种重要的、实用的语言规范。本章通过一些简单的例子介绍最基本的 Java 程序设计,以便读者对 Java 程序设计有最基本的认识,为以后各章的学习做一些准备。

### 1.1 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以及运行系统。事实上,编辑软件是很多的,例如 Windows 自带的记事本就可以编辑 Java 程序。Java 的编译以及运行系统也有很多,本书主要使用 SUN 公司免费提供的 JDK(Java Development Kit)编译与运行 Java 程序。

进入 <http://www.sun.com/> 网站,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 160MB 左右。下载完成后,双击该程序的图标开始安装。按照提示填写选项,一步一步完成安装。作者的安装目录为 E:\Java\jdk1.6.0\example。

安装完成后,需要设置系统环境变量。按照下面的方法设置(添加)系统环境变量,这样就可以在其他目录中使用编译器(javac.exe)与解释器(java.exe)编译运行程序了。

在 Windows 2000 以上版本中,首先进入控制面板,然后选择“系统”选项,打开“系统属性”对话框,在“高级”选项卡中单击“环境变量”按钮,在打开的对话框中的“系统变量”选项区域(见图 1-1)中选择 Path 变量,然后单击“编辑”按钮,出现如图 1-2 所示的“编辑系统变量”对话框。

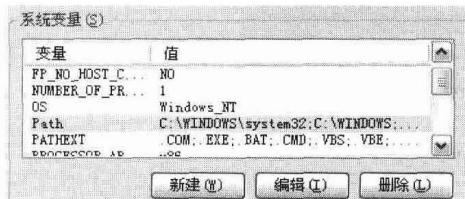


图 1-1 系统变量 Path

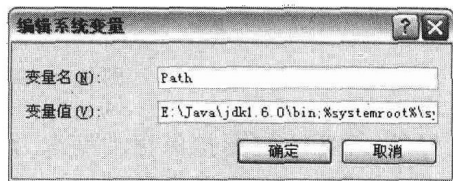


图 1-2 Path 系统变量编辑

把 Java 中 JDK 的 bin 路径写入变量值栏,作者机器上的路径为 E:\Java\jdk1.6.0\bin。因为用来进行编译与运行的 java 与 javac 等文件都在 bin 中,所以把 bin 的路径写入 Path。当在命令窗口中输入 java 或 javac 命令时,系统就会到 E:\Java\jdk1.6.0\bin 中去找 java 或 javac 程序,然后运行。

用同样的方法编辑类文件所在路径以及当前工作路径。

在“系统变量”选项区域中找到 classpath,把当前工作路径与类文件所在路径添加进去。作者机器上当前工作路径是 E:\Java\jdk1.6.0\example;. ; ,类文件所在路径是 E:\Java\jdk1.6.0\jre\lib\rt.jar;. ; 。

注意,在填写这两条路径的时候在最后两个分号之间有“.”,如图 1-3 和图 1-4 所示。

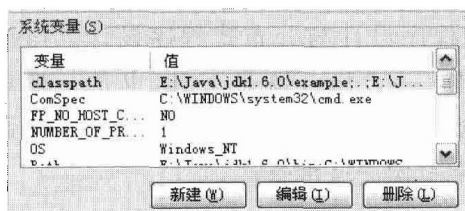


图 1-3 系统变量编辑窗口

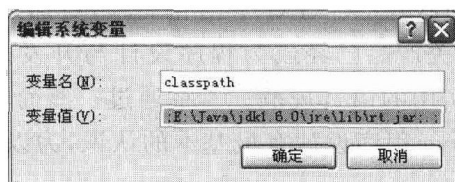


图 1-4 classpath 变量编辑

如果你的机器上已经安装了 JDK,并且已经设置好环境变量,那么就可以按照下面的例题进行 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,输入下面的程序:

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

然后,选择记事本菜单中的“文件”→“另存为”命令,在出现的对话框中把文件名写为 C.java,如图 1-5 所示。此处的文件名必须与类(class)名称 C 一样,如果你在程序中定义的类型名称为 public class C1,那么另存为文件时,文件的名称应该为 C1.java。

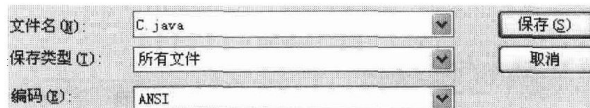


图 1-5 在记事本中保存 java 文件

另外,注意一定要把保存类型选为“所有文件”。

在这个例子中,把所做的 C.java 存储在 E:\Java\jdk1.6.0\example 目录中。

打开附件中的命令窗口,进入 E:\Java\jdk1.6.0\example 目录,如图 1-6 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`。

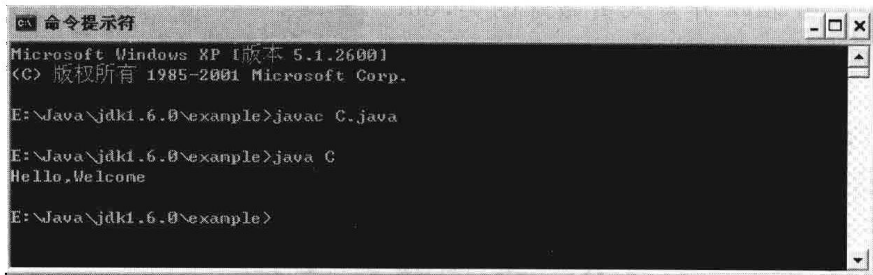


图 1-6 在命令窗口进入工作目录并编译运行程序

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。下面的例 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 程序输出数字与空格。

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

**【注】** 语句中的标点符号如 `"` 与 `;` 等都必须是英文状态下的符号。

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` 和 `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 应用程序输出简单图形。

设计下面的程序:

```
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,绘制出图 1-7 所示图形。

程序中,语句 `System.out.println` 是用来输出的,是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】** 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。

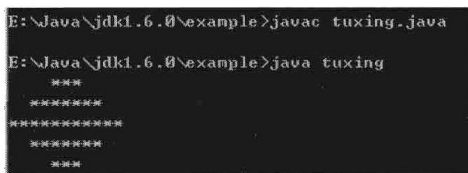


图 1-7 使用输出语句绘制简单图形

## 1.2 应用程序示例

### 1.2.1 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` 是定义整型变量,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】** 变量中每次只能装一个数据。变量中的数据是有范围的。

关于输出语句 `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,只需写为 `System.out.println(c)`,此时不需要给 `c` 加双引号。

当既要计算加法又要计算减法时,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);
    }
}
```

计算乘除时,对该程序稍加修改就可以了。

**【注】** 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 1~10 这 10 个自然数的乘积。

首先设计下面的程序：

```
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);
    }
}
```

程序中乘法使用符号 `*` 表示。上面程序能够计算 1~10 这 10 个数的积，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，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=1,i;
        for(i=1;i<=10;i++)
            c=c*i;
        System.out.println(c);
    }
}
```

该程序也是计算 1~10 这 10 个自然数的积。语句 `Integer c=1` 是在定义整形变量 `c` 的同时给 `c` 赋值为 1。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 1，`i` 的值为 1，所以把 `1*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, $i$  变为了 2。

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  $i \leq 10$ ;看是否成立。此时  $i$  的值为 2,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  $i \leq 10$ ;,不满足,那么退出循环。

**【注】** 程序中等号“=”是用来赋值的。在这个程序中,语句  $\text{for}(i=1;i \leq 10;i++)$  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 1~100 之间自然数中奇数的和。

使用下面的程序就可以计算 1~100 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. 计算自然数 1~20 的和。
2. 计算 1~100 之间自然数中偶数的和。

例 1-4 的程序需要事先把两个要计算的数写在程序中,有很多时候我们希望能够从键盘等随意输入数据,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、积与商等。

设计下面的程序:

```
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 55 78`,然后按 Enter 键,

得到计算结果 133。

该程序只完成了求和运算,很容易修改使其能够计算差、积和商等。例如,下面的程序可以输入两个数,然后计算两个数的商。不过因为两个数都是整数,所以计算结果都取整数。

```
public class C2
{
    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);
    }
}
```

程序中 a/b 中的“/”表示除法。程序编译后,得到的一些运行结果如图 1-8 所示。

```
E:\Java\jdk1.6.0\example>javac C2.java
E:\Java\jdk1.6.0\example>java C2 8 2
4
E:\Java\jdk1.6.0\example>java C2 8 3
2
E:\Java\jdk1.6.0\example>java C2 8 5
1
E:\Java\jdk1.6.0\example>java C2 8 9
0
E:\Java\jdk1.6.0\example>java C2 8 0
Exception in thread "main" java.lang.ArithmeticException: / by zero
at C2.main(C2.java:8)
```

图 1-8 计算两个整数的商

在这个例题中看到了主函数参数 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。

如果要计算两个小数的商,或者希望商是准确的小数,那么只需要把程序修改为如下形式:

```
public class C3
{
    public static void main(String args[])
    {
        double a=0,b=0,c;
        a=Double.parseDouble(args[0]);
        b=Double.parseDouble(args[1]);
```



```

        c=a/b;
        System.out.println(c);
    }
}

```

编译运行时,虽然输入的仍然是 8 与 3,但是因为定义为 double 类型,所以计算机结果就是比较准确的小数,见图 1-9。ParseDouble 是 Double 类的一个方法,可以把字符串变为双精度浮点数(一种比较精确的小数类型)。

```

E:\Java\jdk1.6.0\example>javac C3.java
E:\Java\jdk1.6.0\example>java C3 8 3
2.6666666666666665

```

图 1-9 计算两个数的商

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 C4
{
    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,有些时候必须使用异常捕获功能。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