



高职高专“十一五”规划教材

计算机类

Java程序设计 项目化教程

朱作付 韩永印 主编



冶金工业出版社

www.cnmp.com.cn

高职高专“十一五”规划教材·计算机类

Java 程序设计项目化教程

主 编	朱作付	韩永印
副主编	孙力群	瞿高粤
	陈明忠	

北 京
冶金工业出版社
2009

内 容 简 介

本书以项目为载体,介绍了 Java 语言的专业理论与实用技能。全书共分 12 个项目,通过项目主要介绍了 Java 编程基础、用户界面 GUI 设计、网络与通信编程、线程、数据库编程、JSP 的使用等知识。

本书内容翔实,紧贴项目工作的实际要求,通过项目的完成来学习知识。每个项目都设置了知识目标、能力目标、工作任务和预备知识等,同时将一个项目分解成多个模块,方便了师生的教学互动。本书针对高职高专学生的特点,做到理论知识适用、够用,专业技能实用、管用,密切联系实际。

本书实务性强,适合作为高职高专院校计算机及相关专业的教材,也适合作为程序开发人员的自学参考书。

图书在版编目(CIP)数据

Java 程序设计项目化教程/朱作付,韩永印主编. —北京:冶金工业出版社,2009.7

ISBN 978-7-5024-5032-8

I. J… II. ①朱…②韩… III. JAVA 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 126158 号

出 版 人 曹胜利

地 址 北京北河沿大街嵩祝院北巷 39 号,邮编 100009

电 话 (010)64027926 电子信箱 postmaster@cnmip.com.cn

责任编辑 刘 源

ISBN 978-7-5024-5032-8

北京天正元印务有限公司印刷;冶金工业出版社发行;各地新华书店经销

2009 年 7 月第 1 版,2009 年 7 月第 1 次印刷

787mm×1092mm 1/16; 15.75 印张; 371 千字; 244 页; 1-3000 册

29.00 元

(本书如有印装质量问题,本社发行部负责退换)

前 言

近年来我国高等职业教育发展迅猛，目前，高等职业院校已占全国高等学校的半数以上，高职学生数已超过全国大学生的半数。发展高职、培养大量技术型和技能型人才，是国民经济发展的迫切需要，是高等教育大众化的要求，是促进社会就业的有效措施。为了打破旧的框架，编写一套独具一格的、体现高职高专教育新理念和教学特点的教材，我们本着体系得当，循序渐进，分解难点，正确选择典型项目，项目分解成模块，选好切入点，以及语言通俗易懂、易于理解的原则，在编写此书的过程中力求做到从实际到理论，从具体到抽象，知识贯穿于项目，知识的学习过程就是项目的完成过程，培养学生的学习能力、工作能力和创造能力。

以往的 Java 语言教材，普遍存在以下几个问题：一是过于重视语法，把简单的问题复杂化。二是过于追求系统性，长篇累牍，致使书的内容太多，教师在讲授知识时不易把握和运用。三是没有实际的项目，致使学生不知道 Java 语言有什么作用，与现实生活有什么联系。

本书在编写上突出了以下 3 个方面的特点：

(1) 精简教学内容。考虑到各学校的课时限制和高职高专学生的实际情况，内容上不贪多求全，合理舍去不常使用的内容，但对于 Java 语言的基本内容予以细致的介绍，做到重点突出，易于理解。

(2) 以项目为导向，以模块为驱动。通过项目的学习培养学生分析问题、解决问题的能力 and 团队协作精神，围绕项目将语法和规则渗透到教学过程中，增强课程内容与职业岗位能力要求的相关性。

(3) 重点突出，难度适中，趣味性强。采用简单易懂的项目降低教学难度，强调实用性和趣味性，激发学生学习的积极性，并使学生在解决问题的过程中获得成就感以及信心。

本书由朱作付、韩永印任主编，孙力群、瞿高粤、陈明忠任副主编，王永、张雪松、岳敏、徐超、许颖梅参加编写。

由于编者水平所限，书中如有不足之处敬请使用本书的师生与读者批评指正，以便修订时改进。如读者在使用本书的过程中有其他意见或建议，恳请向编者(bjzhangxf@126.com)踊跃提出宝贵意见。

编 者

目 录

项目一 制作万年历 1

一、知识目标 1

二、能力目标 1

三、工作任务 1

四、预备知识 2

模块 1 我的第一个 Java 程序 2

一、知识目标 2

二、工作任务 2

三、操作流程 2

模块 2 判断一个数是否为回文数程序 6

一、知识目标 6

二、工作任务 6

三、操作流程 6

模块 3 随机生成学生数据并显示

最低分学生信息 13

一、知识目标 13

二、工作任务 13

三、操作流程 13

项目二 Box 类的变幻 19

一、知识目标 19

二、能力目标 19

三、工作任务 19

四、预备知识 19

模块 1 创建一个类 20

一、知识目标 20

二、工作任务 20

三、操作流程 20

模块 2 构造函数的使用 25

一、知识目标 25

二、工作任务 25

三、操作流程 25

模块 3 方法重载 27

一、知识目标 27

二、工作任务 27

三、操作流程 27

模块 4 继承 30

一、知识目标 30

二、工作任务 31

三、操作流程 31

项目三 解决约瑟夫环问题 38

一、知识目标 38

二、能力目标 38

三、工作任务 38

四、预备知识 38

模块 1 数组的使用 39

一、知识目标 39

二、工作任务 39

三、操作流程 39

模块 2 字符串处理 46

一、知识目标 46

二、工作任务 46

三、操作流程 46

模块 3 异常处理 54

一、知识目标 54

二、工作任务 54

三、操作流程 54

项目四 会员信息采集窗口制作 59

一、知识目标 59

二、能力目标 59

三、工作任务 59

四、预备知识 60

模块 1 制作一个 Swing 桌面应用程序 64

一、知识目标 64

二、工作任务 65

三、操作流程 65

模块 2 为基本窗口添加控件	72	二、能力目标	114
一、知识目标	72	三、工作任务	114
二、工作任务	72	四、预备知识	114
三、操作流程	73	模块 1 使用 JTable 显示表格数据	115
项目五 用菜单和工具栏控制		一、知识目标	115
文本样式	87	二、工作任务	116
一、知识目标	87	三、操作流程	116
二、能力目标	87	模块 2 为表格添加数据修改按钮	121
三、工作任务	87	一、知识目标	121
四、预备知识	88	二、工作任务	121
模块 1 利用菜单控制文本样式	89	三、操作流程	121
一、知识目标	89	项目八 利用 Applet 制作相册	124
二、工作任务	89	一、知识目标	124
三、操作流程	90	二、能力目标	124
模块 2 为基本窗口添加控件	98	三、工作任务	124
一、知识目标	98	四、预备知识	124
二、工作任务	98	五、操作流程	126
三、操作流程	98	项目九 TCP 协议通信的实现	135
项目六 布局管理器在日历中的		一、知识目标	135
使用	102	二、能力目标	135
一、知识目标	102	三、工作任务	135
二、能力目标	102	四、预备知识	135
三、工作任务	102	模块 1 获取网络信息	137
四、预备知识	102	一、知识目标	137
模块 1 使用 GridBagLayout 布局		二、工作任务	137
组件 1	103	三、操作流程	137
一、知识目标	103	模块 2 TCP 协议通信的客户端实现	141
二、工作任务	103	一、知识目标	141
三、操作流程	103	二、工作任务	141
模块 2 使用 GridBagLayout 布局		三、操作流程	141
组件 2	109	模块 3 TCP 协议通信的服务器端	
一、知识目标	109	实现	149
二、工作任务	109	一、知识目标	149
三、操作流程	109	二、工作任务	149
项目七 利用 JTable 创建学生		三、操作流程	149
信息表	114	项目十 实现线程	154
一、知识目标	114	一、知识目标	154

二、能力目标.....	154	三、操作流程.....	192
三、工作任务.....	154	项目十二 实现 JSP 综合测试网站.....	205
四、预备知识.....	154	一、知识目标.....	205
模块 1 线程创建及基本控制.....	157	二、能力目标.....	205
一、知识目标.....	157	三、工作任务.....	205
二、工作任务.....	157	四、预备知识.....	205
三、操作流程.....	157	模块 1 我的第一个 JSP 网页.....	222
模块 2 线程的管理.....	164	一、知识目标.....	222
一、知识目标.....	164	二、工作任务.....	222
二、工作任务.....	164	三、操作流程.....	223
三、操作流程.....	164	模块 2 页面验证效果.....	227
模块 3 Applet 程序设计.....	175	一、知识目标.....	227
一、知识目标.....	175	二、工作任务.....	227
二、工作任务.....	175	三、操作流程.....	228
三、操作流程.....	175	模块 3 注册页面效果.....	234
项目十一 数据库编程.....	178	一、知识目标.....	234
一、知识目标.....	178	二、工作任务.....	235
二、能力目标.....	178	三、操作流程.....	235
三、工作任务.....	178	模块 4 在线考试页面效果.....	239
四、预备知识.....	178	一、知识目标.....	239
模块 1 数据库的连接.....	179	二、工作任务.....	239
一、知识目标.....	179	三、操作流程.....	239
二、工作任务.....	179	模块 5 投票调查页面效果.....	241
三、操作流程.....	179	一、知识目标.....	241
模块 2 数据库的基本操作.....	192	二、工作任务.....	242
一、知识目标.....	192	三、操作流程.....	242
二、工作任务.....	192	参考文献.....	244

项目一 制作万年历

一、知识目标

(一)项目目标

能够熟练地运用 Java 语言编写程序。

(二)教学目标

- (1) 了解 Java 的特点、发展历程。
- (2) 掌握开发 Java 程序的步骤。
- (3) 学会并熟练掌握创建一个 Java 程序的方法。
- (4) 熟悉 Java 程序的注释以及 Java 编码规范。
- (5) 掌握条件语句的用法。
- (6) 熟悉循环结构的使用。

二、能力目标

- (1) 培养学生收集、处理信息，准备、动手能力。
- (2) 理解什么是程序。
- (3) 具备用 Java 语言编写程序的能力。
- (4) 掌握简单调试与排错技术。
- (5) 了解开发 Java 程序的过程。
- (6) 培养学生规范编码的能力。

三、工作任务

使用 Java 语言编写 Java 程序，输出××××年××月对应的日历。效果如图 1-1 所示。

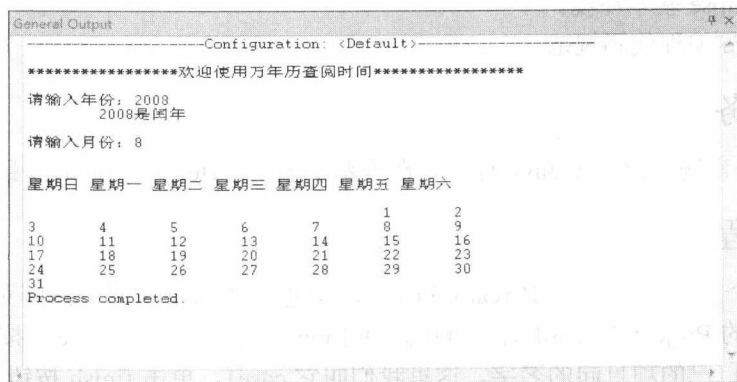


图 1-1 万年历运行图

四、预备知识

(一)Java 简介

Java 是 Sun 公司在 1995 年推出的新的编程语言,它是一种跨平台的、应用于当前高速发展的网络编程语言。在编程语言中,可以认为 Basic 语言促使了 C 语言的出现, C 语言促使了 C++语言的出现,而 C++又促使了 Java 语言的出现。

自 Java 正式推出以后,以其特有的优势迅速发展,经过几年的发展,Java 已经在软件开发和动态网站上占据了相当大的市场。可以说 Java 语言是编程语言的一场革命,它每次的版本升级都会带来不小的轰动。

在 Java 技术领域中应用最为广泛的 Java 技术是: JavaSE 和 JavaEE。

(二)JavaSE

JavaSE 的全称是 Java Platform Standard Edition(Java 平台标准版),是 Java 技术的核心,提供基础 Java 开发工具、执行环境与应用程序接口(API),主要用于桌面应用程序的开发。打个比方,程序员就像一个厨师,刚开始是掂大勺,然后练刀功,之后开始煎炒烹炸,最后才可以自己设计菜色。JavaSE 就是教你成为一个烹饪大师的基本功。

(三)JavaEE

JavaEE 就是教你设计菜色了。JavaEE 的全称是 Java Platform Enterprise Edition (Java 平台企业版),它主要用于网络程序和企业级应用的开发。

需要强调的是任何 Java 学习者都是要从 JavaSE 开始入门,JavaSE 是 Java 语言的核心,而 JavaEE 是在 JavaSE 的基础上扩展的。JavaSE 提供了 Java 的执行环境,使我们开发的应用程序能够在操作系统上运行,就像我们打仗一样,JavaSE 是一把锋利无比的刀,JavaEE 就是我们排兵布阵的方法,在布好的阵当中我们要拿着这把刀去杀敌。

模块 1 我的第一个 Java 程序

一、知识目标

- (1) 熟悉 Eclipse 基本环境。
- (2) 了解 Java 编程代码规范。

二、工作任务

使用 Java 语言编写第一个 Java 程序。在屏幕上输出: Hello 《Java 项目案例教程》!

三、操作流程

(1) 创建一个 Java 项目。在 JCreator Pro 中,创建一个 Java 项目。选择菜单 file | new | Project,在弹出的 Project Wizard 对话框中选择 Empty project,单击 Next 按钮,在 Name 一栏中输入你为自己的项目起的名字,这里我们叫它 case1,单击 finish 按钮,就完成了项目的创建,如图 1-2 所示。

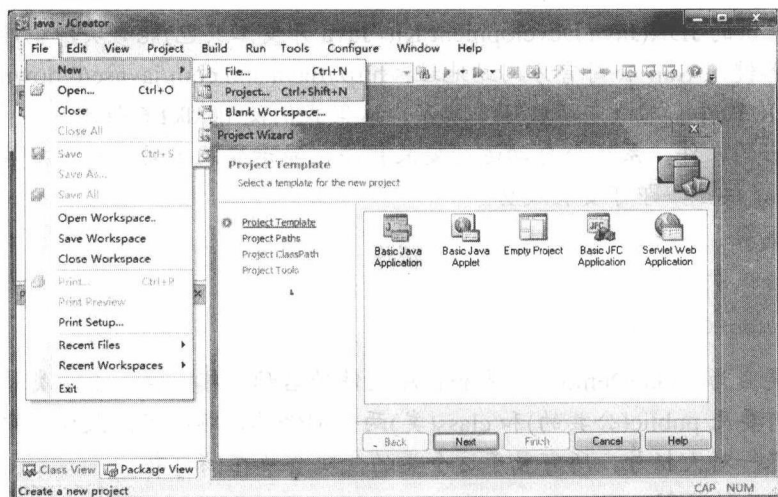


图 1-2 在 Eclipse 中创建一个项目

(2) 手动创建 Java 源程序。在 JCreator Pro 中选中刚才创建的工程 case1，选择菜单 file | new | file，在弹出的 new file 对话框的 java class 栏中，输入你的源程序名字，这里叫它 OneDemo.java，单击“完成”按钮，创建成功。

在 OneDemo.java 文件中输入：

```
/*一个简单的 Java 应用程序*/  
public class OneDemo {  
    public static void main(String[] args) {  
        /*输出 Hello 《Java 项目案例教程》!*/  
        System.out.println("Hello 《Java 项目案例教程》!");  
    }  
}
```

(3) 编译 Java 源程序。这一步不用我们手工来做，JCreator Pro 会自动编译。如果有错误，会给出相应的提示。

(4) 运行 OneDemo.java 文件，选择菜单 RUN | run project 或按 F5 键，如果你看到如图 1-3 所示的输出结果，恭喜你，第一个 Java 程序编写成功。

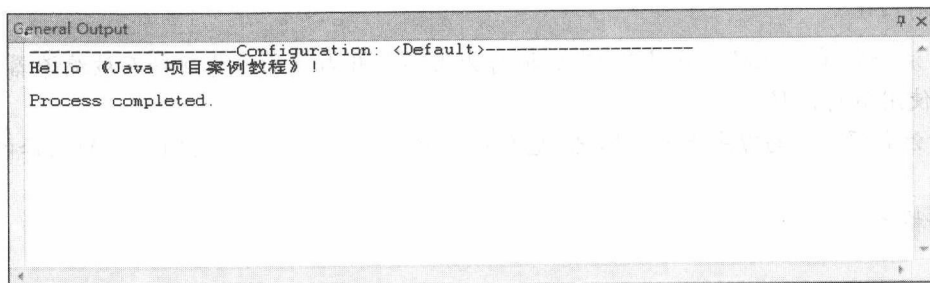


图 1-3 程序输出结果



Sun 公司提供的 JDK(Java Development Kit, Java 开发工具包)能够实现编译和运行的功能。打开浏览器键入 Sun 公司官方网站地址: <http://java.sun.com/javase/6/download.jsp>, 即可免费获取 JDK 软件。下载完毕就可安装了, 这里我们使用 jdk1.6 版本。

打开下载好的软件, 系统会自动运行安装程序, 显示安装画面; 根据安装向导提示或者一直单击 Next 按钮, 即可安装成功。

● 程序框架

程序框架:

```
public class OneDemo{}
```

这里命名类名为: OneDemo, 它要和程序文件的名称一模一样。至于类是什么将在下章学习。类名前要用 public(公共的)和 class(类)两个词修饰, 顺序不能改变, 中间用空格隔开。类名后面跟一对大括号, 所有属于这个类的代码全放在{和}中。

(1) main 方法的框架。

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

main 方法有什么用呢? 就像我们的房子一样不管多大都要从门进去, 程序也要从一个固定的位置开始执行, 在程序中我们把它叫做“入口”。而 main 方法就是 Java 程序的入口, 是所有 Java 应用程序的起始点。

在编写 main 方法时, 要求按照上面的格式和内容进行书写, main 方法前面使用 public、static、void 修饰, 它们都是必需的, 而且顺序不要改变, 中间用空格分开。另外, main 后面的小括号和其中的内容 String[] args 必不可少。

(2) 填写的代码。

```
System.out.println("Hello 《Java 项目案例教程》!");
```

这一行的作用就是打印一句话。System.out.println()是 Java 语言自带的功能, 使用它可以向控制台输出信息。

● Java 程序的注释

(1) 使用单行注释。

注释格式:

```
//注释内容
```

该方法进行单行注释, 从符号//直到换行为止的所有内容均作为注释而被编译器忽略。

(2) 使用多行注释。

多行注释通过/*与*/符号进行标记, 它们必须成对出现, 在它们之间输入的注释内容可以换行。

注释格式:

```
/*  
注释内容 1  
注释内容 2  
...  
*/
```

为了使程序界面美观, 程序开发人员习惯上在每行的注释内容前面加上一个*号。

注释格式:

```
/*
*注释内容 1
*注释内容 2
*...
*/
```

同单行注释一样, 在/*与*/之间被注释的所有内容, 编译器都不会做任何处理。

(3) 文档注释。

当这类注释出现在任何声明(如类的声明、类的成员变量声明或者类的成员方法声明)之前时, 会被 JavaDoc 文档工具读取作为 JavaDoc 文档内容, 文档是对代码结构和功能的描述。

注释格式:

```
/**
*提示信息 1
*提示信息 2
* ...
*/
```

● Java 编码规范

编码规范对一个程序员来说非常重要。为什么这么说呢? 一个软件在开发和使用过程中, 80%的时间是花费在维护上面的, 而且软件的维护工作通常不是由最初的开发人员完成的。编码规范可以增加代码的可读性, 使软件的开发和维护更加方便。

(1) 规范要求类名前用 public 修饰。

(2) 一行只写一条语句。

(3) 用 {} 括起来的部分通常表示程序的某一层结构。{ 一般放在这一结构开始行的最末, } 与该结构的第一个字母对齐, 并单独占一行。低一层次的语句或注释应该比高一层次的语句或注释缩进若干格后书写, 使程序结构更加清晰, 增加程序的可读性。

(4) Java 编程的习惯约定。为了提高程序的可读性, Sun 公司推荐在 Java 编程时使用以下的习惯约定。

类名(class): 由一个或若干个名词组成, 开头大写, 名词间的区分也用大写, 其他小写。如:

```
class AccountBook
class ComplexVariable
```

接口(Interfaces): 规则同类名。如:

```
Interface Account
```

方法(methods): 由一个或多个动词组成, 开头小写, 动词间区分用大写, 其他小写。如:

```
balance Account()
```

常量: 所有字母大写, 单词间用下划线区分。如:

MAXIMUM_SIZE



模块 2 判断一个数是否为回文数程序

一、知识目标

- (1) 掌握 Java 语言中变量的概念。
- (2) 会使用常用数据类型、基本运算符。
- (3) 熟练掌握条件控制语句 if 和 switch 语句的用法。

二、工作任务

从键盘输入数字，判断是否是回文数。运行效果如图 1-4 所示。

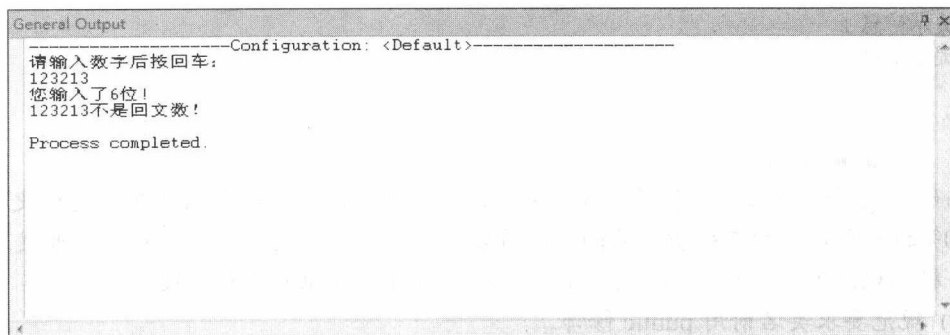


图 1-4 程序输出结果

三、操作流程

- (1) 根据前面的步骤建立新的 Java 程序。
- (2) 打开 JCreator Pro 环境后，建立新的项目，这里命名为 case1，在里面建立新的 Java 源程序，命名为 OneDemo。

(3) 编写代码：

```
package demo2;
import java.lang.*;
import java.io.BufferedReader;
import java.io.InputStreamReader;
/*判断一个数是否为回文数程序*/
/*利用 StringBuffer reverse() 反转变量, 判断是否是回文数*/
public class OneDemo {
    public static void main(String[] args)
    { // 操作提示
        System.out.println("请输入数字后按回车: ");
        BufferedReader br = new BufferedReader(new InputStreamReader
        (System.in));
```

```
try
{
    //接收输入值赋值给 str
    String str1 = br.readLine();
    //输出 str 变量的位数
    System.out.println("您输入了" + str1.length() + "位!");
    //通过类型转换判断是否是数字,如果是数字,正常转换类型;如果不是数字,会出现异常,
    下面会捕捉异常处理
    int num = Integer.parseInt(str1);
    //把成功转换成 int 类型的变量再转回 String 类型
    String str2 = num + "";
    StringBuffer sb = new StringBuffer(str2);
    //把 str2 通过 reverse() 方法反转,再利用 equals() 方法和 str1 比较
    if(sb.reverse().toString().equals(str1))
    { //做出提示
        System.out.println(str2 + "是回文数!");
    }
    else
    { //做出提示
        System.out.println(str2 + "不是回文数!");
    }
}
catch(Exception e)
{ //异常捕获,做出提示
    System.out.println("只能输入数字!谢谢!");
}
}
```

(4) 运行程序。效果如图 1-4 所示。



知识要点



● Java 的基本符号

Java 的基本符号有 5 种: 关键字、标识符、常量、分隔符和操作符。

(1) Java 的字符集。

Java 采用一种称为 unicode 的字符集,该字符集是一种新的编码标准,与常见的 ASCII 码的区别在于:

unicode 使用 16 位二进制而不是 8 位来表示一个字符。

unicode 字符集中增加了许多非拉丁语字符。

(2) 标识符。

变量、类和方法都需要一定的名称,我们将这种名称叫做标识符。Java 中对标识符有一定的限制。首先,所有标识符的首字符必须是字母(大小写)、下划线 _ 或美元符 \$; 其次,标识符是由数字(0~9)、大写字母(A~Z)、小写字母(a~z)和下划线 _、美元符 \$ 以及所有在十六进制 0xC0 前的 ASCII 码等构成的; 再次,注意不能使用系统保留的关键字做标识符。

以上是标识符命名的基本规则,以下是一个正误对照表,通过它会对标识符的命名规

则有一个更好的了解:

合法标识符	非法标识符
Try	try# (注: 不能用#作为标识符)
group_7	7group (注: 不能用数字符号开头)
opendoor	open-door (注: 不能用-作为标识符)
boolean_1	boolean (注: boolean 为关键字, 不能用关键字做标识符)

(3) 关键字。

关键字是为特定目的而保留的保留字, 程序员不要将关键字作为自己的标识符。Java 的关键字有:

abstract, boolean, break, byte, case, catch, char, class, continue, do, double, else, extends, false, final, finally, float, for, implements, import, instanceof, int, interface, long, native, new, null, package, private, public, return, short, static, super, switch, synchronized, this, throw, true, try, void, while.

(4) 操作符。

1) 算术操作符。

一元: +、-、++、--、-

二元: +、-、*、/、%

值得注意的是++和--操作符。如:

```
int a, b, x=1, y=5;
a=++x; b=y++;
```

此时 a 的值为 2(先加 1, 后赋值), b 的值为 5(先赋值, 后加 1)。

二元操作符两侧的操作对象数据类型不同时, 先自动进行类型转换, 再进行操作。

2) 赋值操作符与复合赋值操作符。

可将

变量=变量 op 表达式

写成

变量 op=表达式

如: $x+=20$ 与 $x=x+20$ 结果一致, 但更简洁。

注意: =与==的不同。

3) 位操作符和移位操作符。

位操作符:

&(按位与)、|(按位或)、^(按位异或)。

移位操作符:

$E \ll n$ 左移 n 位, 空位补 0, 相当于 $E * 2^n$ 。

$E \gg n$ 右移 n 位, 空位用原最高位的位值补足, 相当于 $E / 2^n$ 。

$E \ggg n$ 右移 n 位, 空位补 0。

4) 关系操作符。

关系操作符共 6 个:

>(大于)、>=(大于等于)、<(小于)、<=(小于等于)、!=(不等于)、==(相等)。

关系操作符的结果为 boolean 型数据(true 或 false)。

注: ==操作符只有在比较双方均完全一致时, 其值为 true, 如比较的是两个对象, 即使两个对象的内容相同, 结果也为 false, 只有这两个对象为同一对象时才为 true。

5) 逻辑操作符。

逻辑操作符的操作对象和结果均为 boolean 型, 共 6 个:

!(逻辑非)、&&(逻辑与)、||(逻辑或)、^(逻辑异或)、&(按位与)、|(按位或)。

按位与“&”也可作为逻辑与使用, 但未作优化, 而“&&”操作符是经过优化的。对“|”操作符也类似。

6) 其他操作符。

① 条件操作符。

E1?E2:E3

表达式 E1 若成立, 执行表达式 E2, 否则执行 E3。

② 逗号操作符。

“,” 可用于分隔语句。如:

```
int x,y;
for (x=0,y=0;x<10;x++) {...};
```

③ 转义字符。

转义字符见表 1-1。

表 1-1 转义字符

转义字符	描述
\ddd	八进制模式
\uxxxx	Unicode 双字节字符
\'	单引号字符
\"	双引号字符
\\	反斜杠字符
\r	回车
\n	换行
\f	走纸换页
\d	退格
\t	横向跳格

● 简单数据类型

Java 有着不同的数据类型, Java 的数据类型有: 整型、浮点型、布尔型、字符型、字符串型。

(1) 整型(int)。

1) 整型常量。

十进制整数: 如 123, -456, 0。八进制整数: 以 0 开头, 如 0123 表示十进制数 83,

-011 表示十进制数 -9。十六进制整数：以 0x 或 0X 开头，如 0x123 表示十进制数 291，-0X12 表示十进制数 -18。

2) 整型变量。

整型变量见表 1-2。

表 1-2 整型变量

数据类型	所占位数	数的范围
byte	8	$-2^7 \sim 2^7 - 1$
short	16	$-2^{15} \sim 2^{15} - 1$
int	32	$-2^{31} \sim 2^{31} - 1$
long	64	$-2^{63} \sim 2^{63} - 1$

每一个整型数据占有 32 位的存储空间，即 4 个字节。这意味着整型数据所表示的范围在 -2 147 483 648 和 2 147 483 648 之间，假如由于某些原因，必须表示一个更大的数，64 位的长整型应该是足够的。如果你想把一个整数强制存为一个长型(long)，你可以在数字后面加字母 l。

(2) 浮点型(float)。

1) 浮点型常量。

十进制数形式：由数字和小数点组成，且必须有小数点，如 0.123，1.23，123.0。

科学计数法形式：如 123e3 或 123E3，其中 e 或 E 之前必须有数字，且 e 或 E 后面的指数必须为整数。float 型的值，必须在数字后加 f 或 F，如 1.23f。

2) 浮点型变量。

浮点型变量见表 1-3。

表 1-3 浮点型变量

数据类型	所占位数	数的范围
float	32	$3.4e-038 \sim 3.4e+038$
double	64	$1.7e-038 \sim 1.7e+038$

(3) 布尔型(boolean)。

布尔型是最简单的一种数据类型，布尔数据只有两种状态：真和假，通常用关键字 true 和 false 来表示这两种状态。

(4) 字符型(char)。

1) 字符型常量。

字符型常量是用单引号括起来的一个字符，如 'a'，'A'。

2) 字符型变量。

字符型变量的类型为 char，它在机器中占 16 位，其范围为 0~65 535。字符型变量的定义如：

```
char c='a'; /*指定变量c为char型，且赋初值为'a'*/
```