全国计算机继续教育研究会推荐用书

计算机新技术丛书

JDBC 与 Java 数据库程序设计

刘丽珏　张龙祥　编著



人 民 邮 电 出 版 社

需要全本请访问 www.ertongbook.com

《计算机新技术丛书》编委会

名誉主任：邹海明

主　　任：张凤祥

副 主 任：杨学良　袁开榜　李茂青　张大方　孙大高
　　朱　凯　邹桂章　林　勇

秘 书 长：王　虹

编　　委：王汝传　陈文伟　钱 进　程佩青　顾玉昆
　　俞思伟　承厚之　韩启明　汲伟民　陈　旭
　　王树亮　迟成文　胡学联　王文耀　陈道员
　　相万让　石建斌　徐竹青　傅献祯　王爱军
　　顾元刚　陈大正　刘高嵩　李顺福

图书在版编目(CIP)数据

JDBC 与 Java 数据库程序设计/张龙祥，刘丽珏编著.北京：人民邮电出版社，2001.7
(计算机新技术丛书)

ISBN 7-115-09399-7

Ⅰ.J… Ⅱ. ①张… ②刘… Ⅲ. JAVA 语言—数据库—程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2001)第 035796 号

内 容 简 介

本书围绕 Internet 上 Web 数据库访问这一专题，着重剖析 JDBC 的工作原理，介绍了运用 JDBC 进行 Java 数据库程序设计的方法、JDBC 在 Java Servlet 方面的应用，以及在 JBuilder 和 Visual J++中使用 JDBC 进行数据库编程等实用技术。

本书是一本技术参考书，适合计算机软件开发人员、计算机项目管理人员、大专院校相关专业的师生阅读，也可作大专院校相关专业的教材和计算机继续教育的教材。

全国计算机继续教育研究会推荐用书
计算机新技术丛书
JDBC 与 Java 数据库程序设计

◆ 编　　著　刘丽珏　张龙祥
责任编辑　邹文波
执行编辑　苗　颖

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号
邮编　100061　　电子函件　315@ pptphn
网址　http://www.pptph.com.cn
北京汉魂图文设计有限公司制作
北京　　印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：17
字数：409 千字　　2000 年 7 月第 1 版
印数：1 – 0 000 册　　2000 年 7 月北京第 1 次印刷

ISBN 7-115-09399-7/TP・2288

定价：35.00 元(附光盘)

《计算机新技术丛书》
出版说明

计算机科学和技术的迅猛发展，强有力地推动了世界经济、社会的飞速发展。当今世界各国毫无例外地将大量的人力、财力、物力投向计算机科学、技术、工业、教育，以争夺这个对军事、政治、经济、社会发展具有决定意义的“制高点”。在这样的形势下，计算机科技的交流、推广成为急迫而又繁重的任务。

近几年来，我们与电子工业出版社、湖北科技出版社、西南交大出版社合作出版了《计算机继续教育》、《当代微机技术》、《今日微机技术》等丛书。从 1994 年起，与西南交通大学出版社合作编辑、出版了《全国计算机新科技与计算机继续教育论文集》，该论文集从 1995 年起每年出版上、下两集，到 2000 年已出版 13 集。这些丛书、论文集对于推广、交流计算机新技术和开展计算机继续教育起到了积极的作用，已经成为我国的计算机新技术交流、推广、教育的一个阵地。

但是，如何更快、更好地使从事计算机工作的专业工作者以及所有在自己的研究和工作中应用计算机的非计算机专业工作者，尽快地获取国外最新的计算机理论和技术，是一个值得进一步探讨的问题。2000 年 8 月“全国计算机新科技与计算机继续教育学术会议”（新疆会议）讨论认为，我们能做到的一个好的办法是，组织人力、物力尽快地编写《计算机新技术丛书》，以尽快地引进先进国家的计算机新技术、新理论。关键在一个“新”字。不论层次高低，只要是新出现的，对我国有益的新技术、新理论我们都应尽快引进，以缩短我国科技队伍的计算机及其应用的整体水平与先进国家之间的差距，为我国的计算机事业出一把力。

由于这套丛书定位于“新”，必然涉及到计算机科技各个层次、各个领域，因此，该丛书将适用于各科研、工程机构，各院校研究生、大学生，以及急迫需要使用计算机新技术的工作人员。我会向全国各层次希望学习、使用计算机新技术的人员推荐这套丛书。我们欢迎已经掌握某项计算机新科技的学者加盟该丛书的编写工作，使该丛书更具有活力。

经过作者、出版社以及我们学会的努力，这套丛书终于开始出版了，我们预祝该丛书能达到预期的效果。

在本丛书的组织、编辑、出版过程中，得到了许多专家、教授、院校及出版社的支持，我们表示深切的感谢！

全国计算机继续教育研究会

2001 年 6 月

目　　录

第 1 章　Java 编程基础 1

1.1　Java 语言概要 1
1.1.1　Java 语言简史 1
1.1.2　Java 虚拟机（JVM） 2
1.1.3　Java 语言的特点 3
1.2　学习 Java 5
1.2.1　词法 5
1.2.2　数据类型 9
1.2.3　变量 13
1.2.4　赋值和初始化 13
1.2.5　运算符和表达式 14
1.2.6　字符串 20
1.2.7　流程控制 22
1.2.8　类 36
1.2.9　数组 44
1.2.10　接口和包 49
1.2.11　Java API 简介 51
1.2.12　JavaBeans 概述 59
1.3　简单的 Java 应用实例 59

第 2 章　数据库基础知识 63

2.1　关系型数据库 63
2.1.1　关系型数据库概念 63
2.1.2　SQL 语言 66
2.2　网络数据库系统 73
2.2.1　Client/Server 系统 73
2.2.2　Browser/Server 系统 75
2.3　数据库访问接口 77
2.3.1　ODBC 77
2.3.2　JDBC 78
2.4　Linux 数据库 79

需要全本请访问 www.ertongbook.com

第 3 章　JDBC 原理.....81
3.1　什么是 JDBC.....81
3.2　JDBC API.....82
3.2.1　概述.....82
3.2.2　有关 JDBC 的类.....84
3.2.3　实现 JDBC 连接.....85
3.3　JDBC 驱动程序.....86
3.4　基本的 JDBC 数据库访问流程.....90
3.5　通过 JDBC 创建和修改数据库.....97
第 4 章　Web 数据库技术.....103
4.1　Java Applet 中的数据库访问.....103
4.1.1　使用 Applet.....103
4.1.2　JDBC 在 Applet 中的应用.....107
4.2　JSP 技术.....110
4.2.1　与数据库的连接.....110
4.2.2　ASP 与 JSP 的比较.....112
第 5 章　Java Web 服务器.....117
5.1　Java Web 服务器.....117
5.1.1　JavaServer.....117
5.1.2　管理（Administration）.....118
5.1.3　访问 Java Web 服务器.....123
5.2　Servlet.....124
5.2.1　什么是 Servlet.....124
5.2.2　Servlet 示例.....128
第 6 章　JDBC 在 Java Servlet 中的应用.....135
6.1　Servlet 结构.....135
6.1.1　Servlet 接口.....135
6.1.2　HttpServlet 类.....135
6.1.3　ServletRequest 类.....137
6.1.4　HttpServletRequest 接口.....138
6.1.5　ServletResponse 接口.....139
6.1.6　HttpServletResponse 接口.....139
6.2　JDBC Servlet 示例.....140
第 7 章　数据库存取优化.....155
7.1　基本概念.....155

7.2 SQL 程序优化 155
7.2.1 prepared 语句 155
7.2.2 存储过程 157
7.2.3 SQL 调用与存储过程比较 159
7.3 动态存取 160
7.3.1 java.sql.ResultSetMetaData 161
7.3.2 java.sql.DatabaseMetaData 163
7.3.3 动态存取示例 163

第 8 章 Java 开发工具 Visual J++的应用 175

8.1 Visual J++ 6.0 的特点 175
8.1.1 Developer Studio 开发环境 175
8.1.2 WFC（Windows Foundation Classes） 175
8.1.3 Form Designer（Form 设计器） 176
8.1.4 ADO（ActiveX Data Object） 177
8.1.5 COM 支持 177
8.1.6 Object Browser（对象浏览器） 177
8.1.7 IntelliSense（智能传感） 177
8.1.8 编译错误显示 177
8.1.9 集成的调试器 177
8.1.10 HTML 支持 178
8.1.11 多项目解决方案 178
8.1.12 基于目录的项目管理 178
8.1.13 向导和创建器 178
8.1.14 打包和发布功能 178
8.2 Visual J++ 6.0 概览 179
8.2.1 安装和卸载 179
8.2.2 Visual J++ 6.0 主界面 183
8.3 创建 Java 程序 189
8.3.1 创建 Java 应用程序 189
8.3.2 创建 Java Applet 程序 200
8.4 创建数据库程序 205

第 9 章 Java 开发工具 JBuilder 的应用 223

9.1 JBuilder 3.0 的特点 223
9.1.1 代码浏览功能 223
9.1.2 基于组件的编程 223
9.1.3 多组件库 223
9.1.4 强大的调试器 224
9.1.5 广泛的定制能力 224

9.2 JBuilder 3.0 概览 224
9.2.1 安装和卸载 224
9.2.2 JBuilder 3.0 主工具栏 229
9.2.3 File 菜单向导 231
9.2.4 Wizard 菜单向导 231
9.3 创建 Java 程序 233
9.3.1 创建 Java 应用程序 233
9.3.2 创建 Java Applet 241
9.4 创建数据库程序 246
9.4.1 创建工程和应用程序 246
9.4.2 建立数据库连接 246
9.4.3 建立查询 248
9.4.4 创建显示查询结果的用户界面 248
9.4.5 运行查询 250
9.4.6 保存数据 253
附录 参数化查询示例 255

第 1 章　Java 编程基础

Java 语言是在 1995 年由 Sun Microsystems 公司提出的一种编程语言，它具有了革命化的“编写一次，随处运行”的跨平台能力，并具有强大的网络能力，这些特点足以使它引起很多人的注意。此外，Java 语言的开发者在消灭常见的程序设计错误、减轻程序编制负担以及保证代码可读性及可维护性方面做了大量的工作，提高了编程的效率。

Java 语言之所以受到众人瞩目，是由于其强大的移植能力、多线程处理和联网能力。同时它也正在进入嵌入式系统，逐渐成为手持设备、车内计算机等系统的标准。

Java 语言看起来很像大家熟悉的 C/C++语言，但与之相比，Java 语言去掉了 C 语言中许多容易使人犯错的地方。更重要的是，用 Java 语言编写的程序可以运行在任何支持 Java 语言的任何平台上，而且，Java 从一开始就是为网络编程准备的。Java 语言还具有让任何人使用 Applet 程序的能力。Applet 是一个小巧、安全、动态、跨平台、活跃、网络化的应用程序。Java 语言的内置安全性检查机制，使从网上下载的 Java Applet 对本机系统不会造成任何危害。

1.1　Java 语言概要

Java 语言是一种新型的网络语言，它是和 Internet 同步发展起来的，其引人注目的特点，使得它很快被广大用户所接受，逐渐成为主流编程语言之一。

1.1.1　Java 语言简史

Java 语言诞生于 1991 年。最初在 Sun 公司由 Patrick Naughton 和 James Gosling 领导的一组工程师打算设计一种小型的计算机语言。为了实用，这种语言要能产生简短、高效的执行代码，用它编制的程序必须容易运行在不同的几种处理器上。同时，为了减少使用和维护的成本，这种语言必须是简单和面向对象的。这个项目在 Sun 公司中被称为 Green（绿色项目），而这种语言被称为 Oak。

早期对可移植性的强调使得开发在 Internet 上下载的程序成为自然而然的事，因为这些程序可以不加修改地运行在任何能够解释 Java 语言的计算机上。用 Java 语言进行开发意味着一个 Web 站点的设计者不必再编制同一程序的多个版本。

在 1995 年 5 月召开的 SunWorld’95 大会上，Sun 公司发布了用 Java 语言开发的浏览器 HotJava，显示了 Java 的威力，从此引发了至今不衰的 Java 热。同年秋，Netscape 公司决定让他们的浏览器提供对 Java 语言的支持，并从 1996 年发布的 Netscape 2.0 开始，提供了对 Java 语言的支持。注册了 Java 使用许可证的还有 IBM、Symantec、Inprise 和其他许多公司。即使是微软，也注册并宣布支持 Java 语言。IE 浏览器早已支持 Java 语言，同时 Windows 本身也配套提供了 Java 虚拟机。

1996 年初，Sun 公司发布了 Java 的第一个版本，不过早期的 Java 并不适合用来进行正规的应用程序开发。但在很短的时间内，Java 就抛掉了以前的粗糙，取而代之的是功能全面、具有高度扩展能力的新版本。由于 Sun 对 Java 抱有无限的信心，使得它把 Java 的详细说明甚至源代码都公布于众，这样也就创造了第一个真正的标准的跨平台语言。

1.1.2 Java 虚拟机（JVM）

Java 之所以能够“编写一次，随处运行”，其原因就在于 Java 虚拟机（Java Virtual Machine，JVM）。JVM 位于 Java 程序和用户的计算机系统之间，如图 1-1 所示。

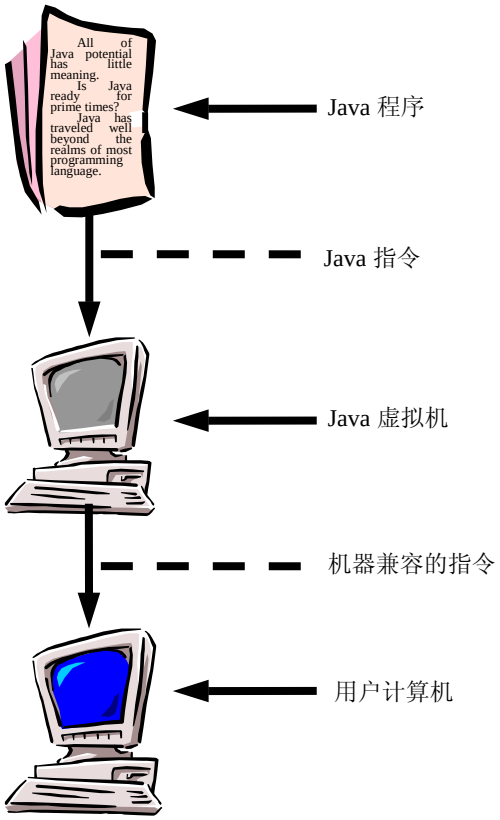


图 1-1 Java 虚拟机

所谓虚拟机，其实在现实中并不存在，而是计算机的一种抽象设计，是由计算机程序来实现的。Java 虚拟机，指的就是一种知道 Java 语言的各种指令，并能够将这些指令翻译成用户计算机所能理解的指令的计算机程序，也就是一种“抽象计算机”。

Java 语言在执行时既被解释又被编译，这是由 JVM 结构所决定的。用户编写的 Java 程序首先被编译成能在 Java 虚拟机上运行的虚拟指令，编译后的 Java 程序像用其他任何语言编写的程序一样，被当成可执行文件，不同的是，这种可执行文件必须运行于 JVM 上。JVM 在不同的系统中以同样的方式工作，因此 Java 程序不必为每种平台单独进行编译。JVM 为不同的平台解释 Java 程序，所有和移植有关的问题都在 JVM 中得到解决，这样用户的 Java 程序在理论上就可以不加修改地运行在任何拥有 JVM 的平台上。

事实上，早在 20 年前，UCSD Pascal 就已使用了与 Java 虚拟机相类似的技术，但由于当时硬件条件的限制，虚拟仿真所造成的性能降低现象十分严重，此外，对程序的可移植性也没有强制的要求，因此，这项技术并没有得到重视。

随着 Internet 的出现，人们对可移植性的要求越来越高，同时，当前的高速硬件也使 JVM 真正实用化。另一方面，虚拟机仿真器自身的进步极大地提高了仿真性能。最新的实时编译器（JIT），能够在程序解释时编译它们，同时进行初步的优化，使得 Java 程序取得了和传统编译过的文件相似的性能。

目前，一种新的、智能的、自适应优化的虚拟机正在开发中。这种虚拟机能够指出在什么地方程序消耗了大部分运行的时间，然后便将那些地方进行优化。运行于这些虚拟机上的 Java 程序有可能比某些编译过的程序运行速度更快，并且仍然保持与平台无关的特性。

1.1.3　Java 语言的特点

Java 是一种强有力的编程语言，它之所以能在世界范围内迅速地传播，是因为它具有一些非常优秀的技术特点，基本上可以归纳为以下几个方面：

1．简单

Java 语言最初是为家用电器产品设计的，为了便于推广，被设计得简单高效，生成的可执行代码也非常短小精悍。Java 语言来源于 C++语言，这使得 C++的程序员可以很容易地转向 Java 语言，但它取消了 C++语言中一些比较复杂的部分，没有了头文件、指针、结构、单元、运算符重载、虚拟基础类等等。

另一方面就是“小”。Java 成功的秘诀之一就是采用了特殊的软件构造方法，可在内存容量小的计算机上独立运行。其基本的解释器以及类支持模块仅为 40KB，即便再加入基本的标准库，以及对线程的支持，也只需多加 175KB。

2．面向对象

面向对象是着眼于数据（即对象）以及对象接口的一种编程技术，Java 语言的面向对象特性和 C++语言存在许多共同之处。两者最主要的区别在于多重继承，Java 语言并不直接支持多重继承，即一个类不能有一个以上的父类，它采用了“元类模型”，通过反射机制以及对象序列化特性，使其更容易实现永久性对象，而 GUI 构建器也更容易集成现有的组件。

Java 是一种完全面向对象的程序设计语言，其程序代码以类的形式组成。Java 语言抛弃了 C++语言中非面向对象的特性，因此，它也被称为是纯面向对象的。

3．分布式

Java 提供了内容广泛的例程库，可以处理像 HTTP 和 FTP 这样的 TCP/IP 协议。Java 应用程序可以通过一个特定的 URL 在网络上访问对象，就像访问本地文件系统一样。

Java 具备强大且易于使用的联网能力，使用 Java 进行 Internet 编程的过程是非常容易的，Java 也使公共网关接口（CGI）的脚本编制更加容易。而另一种称为 Servlet（服务器小应用程序）的机制，使得用 Java 进行服务器端编程时，可获得非常高的效率。许多流行的 Web 服务器都支持 Servlet。

4．健壮性

Java 将重点放在早期的潜在问题检查、以后的动态检查和避免出现潜在错误方面。Java 语言与 C++语言之间最重要、最大的差异是 Java 不允许直接使用指针，而是采用了一个指

需要全本请访问 www.ertongbook.com

针模型，这样可以避免改写内存和毁损数据的情况发生。同时 Java 语言还提供了内存管理机制，即一个自动的“垃圾回收”功能，此功能能够自动释放不再访问的对象所占用的内存，而不像在 C++语言中那样，程序员需要使用 free 或 delete 语句来释放空间。

此外，Java 编译器可以提前侦测到许多问题，而在其他语言中，这些问题只有在运行时才会显露出来。

5．安全性

由于 Java 语言将重点放在网络和分布式环境中使用，这就要求软件具有高度的稳定性和安全性。利用 Java 语言可以轻松构建出防病毒、防黑客的系统。它提供了 4 级安全性保障机制：

- 字节码校验器。
- 类装载器。
- 文件访问控制。
- 运行时内存布局。

所有这些特性说明，Java 的确是目前最安全的一种程序设计语言。

6．中性结构

Java 编译器生成的是一种中性的对象文件格式，编译好的代码理论上可以在任何安装了 Java 运行时间库的机器上运行。Java 采用了完全统一的语言版本，其编译器生成的是特殊字节码指令（JavaByteCode），同任何一种特定的计算机体系都无关。它不仅在任何机器上都易于解释，也易于动态翻译成本机代码。同时 Java 语言环境还定义了一个用于访问底层 OS 功能的扩展类库，以使 Java 语言的应用程序不依赖于具体系统。Java 语言的这种结构上的中立性是保证其可移植性的基础。

7．可移植

Java 定义出自己的一套虚拟机，以及这套虚拟机上所使用的代码——Java 字节码。Java 语言通过预先将源代码编译为接近于机器指令的字节码，有效地克服了传统解释型语言的性能瓶颈；又由于解释执行字节码只需要 Java 运行系统，不需要具体某个平台的支持，因此保证了 Java 语言运行环境与平台的无关性，所以移植很方便。Java 语言的可移植性主要是由以下两条机制保证的：

（1）Java 从本质上讲是解释执行的，任何一种平台，任何一台机器，只要配备了 Java 解释器 —— Java 虚拟机，就可以运行 Java 程序。

（2）Java 的数据类型在任何机器上都是一致的，它不支持依赖于特定硬件环境的数据类型。

例如，在 Java 语言中的一个 int 类型数肯定是个 32 位的整数，而在 C/C++语言中，一个 int 类型数可以是个 16 位整数，也可以是个 32 位整数。在 Jave 中，唯一的限制是 int 类型必须拥有和 short int 类型至少一样多的字节，而且不可多过 long int 指定的字节。将数值类型的大小固定下来后，就可以轻松避开移植时可能出现的大量问题。

8．解释型

前面已经讲过，Java 语言在本质上是解释执行的，在已经移植了解释器的任何机器上，都可以直接执行 Java 字节码。由于链接变得轻而易举，因此整个开发过程也快了许多，更容易发挥开发者的创造力。

9．高性能

尽管解释过的字节码通常具有比普通代码更高的性能，但如果只采用标准的 Java 解释器来转换字节码，“高性能”只能是一个“神话”。为解决这个问题，可以通过一个真正的编译器来运行许多 Java 程序，而不是局限在对字节码的解释上面。

目前，许多公司都为 Java 设计了固有代码编译器。所谓固有代码，就是指目标平台专用的代码，它不能跨平台兼容，对于那些设计成在特定机器上，以独立形式运行的任何一个应用程序来说，都可以采用这种做法。还有另一种编译方式可供选择，即“准实时”（Just-in-Time）编译器，或称 JIT 编译器。JIT 编译器首先将字节码编译成固有代码，将结果存下来，在需要时才调用它们。这种方法可以明显改善速度，因为只需要进行一次解释就可以了，而且 JIT 编译器还可以监测出哪些代码经常执行，从而对其进行适当优化，尽管仍然比真正的固有代码编译器要慢，但要显著快于传统的 Java 解释器。

10．多线程

多线程的优点是具有更好的交互性及实时性。只要位于基层的操作系统支持，Java 中的线程便有能力利用多处理器系统。但由于目前主流平台在线程的实施上存在着广泛的差异，因此 Java 在这个方面不可能再做到与平台无关性。在不同的机器中，只有用来调用多线程机制的代码是相同的，Java 为不同的基层操作系统都实现了专门的多线程机制。

11．动态

若想将新代码加入一个正在运行的程序，“动态”无疑是一个非常重要的特点。在许多方面，Java 都比 C 或 C++更像一种动态语言。经过设计，它与不断发展着的工作环境有着很好的相容性。在库里，可以很容易地添加新方法以及新实例变量而不会对它们原来的客户造成任何干扰。在 Java 中，可以简单、直观地查询运行时间类型信息。

1.2　学习 Java

在这一节中，将阐释 Java 编程的一些基本的概念，比如数据类型、循环以及用户自定义方法等。如果你是一名经验丰富的 C++程序员，就可以很快地浏览一下本节的内容。

1.2.1　词法

在程序被编译时，实际上是在对程序进行词法分析，并把源代码分解为词法符号和空白符号。忽略空白符号后，再对剩余的部分进行处理。Java 语言中的词法符号一共有 5 种：关键字、标识符、常量、运算符和分隔符。在源代码中，一般还带有一些注释。

1．Java 关键字

Java 有自己的关键字，关键字不能作为标识符在程序中使用。表 1-1 列出了 Java 的关键字，其中打*号的是现在尚未使用，留作后用的关键字。

表 1-1　　Java 关键字

关 键 字	含　义
abstract	一个抽象类或方法
boolean	布尔类型
break	退出一个 switch 或循环

byte	8 位整数类型

续 表

case	switch 语句中的一种条件
*cast	未使用
catch	try 块的从句，用于捕捉一个异常
char	unicode 字符类型
class	定义一个类的类型
*const	常数，未使用
continue	从循环末尾继续
default	switch 的默认从句
Do	do/while 循环的顶部
double	双精度浮点数类型
else	if 语句的 else 从句
extends	定义一个类的父类
final	一个常数，或者一个不可覆盖和取消类或方法
finally	try 块的一部分，无论如何都会执行
float	单精度浮点数类型
for	一种循环类型
*future	未使用
*generic	未使用
*goto	转到，未使用
If	一个条件语句
implements	定义一个类实现的接口
import	输入一个包
*inner	未使用
instanceof	检测对象是否属于一个类的实例
int	32 位整数类型
interface	一个抽象类型，含有类能够实现的方法
long	64 位整数类型
native	由主机系统实现的一种方法
new	分配一个新对象或数组
null	一个空引用
*outer	未使用

续 表

package	一个类包
private	仅能由这个类的方法使用的一种特性
protected	仅能由这个类、它的子类以及同一个包内的其他类的方法使用的一种特性
public	可由所有类的方法使用的一种特性
*rest	未使用
return	从一个方法返回

short	16 位整数类型
static	类所特有的一种特性，而非类的对象所有
super	超类对象或构建器
switch	一种选择语句
synchronized	同步，线程专用的方法
this	一个方法，或者这个类的一个构建器的隐式参数
throw	产生一个异常
throws	方法可产生的异常
transient	标记那些不应持久的数据
try	用于捕捉异常的一个代码块
*var	未使用
void	指出一个方法不返回任何值
*volatile	未使用
While	一种循环

2．Java 标识符

我们在程序中使用的所有标识符（变量、类、方法、标号和其他用户自定义的对象）的集合就构成了 Java 的符号集。Java 符号集采用的是 unicode 码，用 16 位存储一个字符。在 Java 中，标识符必须以字母、下划线“_”或美元符“$”开头，后面可以不跟、跟一个或多个由字母、下划线、美元符或数字组成的字符数字串。

标识符可以是任意的字符或数字，但要遵循以下限制：

- 首字符必须是字母、下划线“_”或美元符号“$”。
- 首字符后的字符不能为 unicode 字符集中保留的字符，但可以是数字、字母。
- 标识符不能是 Java 关键字。

下面是一些合法标识符的例子：

xyz，JavaJDBC，_lookfor，$38

而以下则是一些不合法的例子：

7command，rode#，null

3．Java 常量

在程序运行过程中，其值不能被改变的量称为常量。它是用来表示固定的数值和字符串值的词法符号。Java 语言支持 5 种常量：整型常量、浮点常量、布尔常量、字符常量和字符串常量。

在 C、C++语言中把一个标识符定义为一个常量有两种方法，即采用宏定义#define 或用关键字 const 来定义。在 Java 语言中，不能用 define 命令来定义常量，const 关键字也暂时不能用，其常量的定义是通过 final 关键字来实现的，例如：

final double PI=3.14159265;

在一般情况下，常量符号用大写来表示，而变量名用小写来表示，以示区别。常量在运行过程中不能被改变，否则会出现无法预知的现象。

4．Java 运算符

运算符是用来对变量或其他对象进行运算操作的符号。Java 内部有 44 个运算符，运算符可分为 4 类：算术操作、位操作、关系操作和逻辑操作。关于各种运算符的详细情况将在

后面的章节中进行介绍。

5．Java 分隔符

Java 语言的分隔符一共有 5 种：“()”、“{}”、“[]”、“，”和“；”。

（1）括号“()” 用来隔开表达式、组合表达式或表示方法调用。

（2）方括号“[]” 用来表示一维或多维数组的下标。

（3）花括号“{}” 用来表示一个程序块的开始和结束。最外层的花括号是接在用 class 关键字进行类定义的语句后使用的，该{}中的所有部分表示一个类程序块。

（4）逗号“，” 用来分隔方法参数中的参数，同一变量类型的连续声明以及用于逗号表达式等。

（5）分号“；” 是句子的终止符，任何合法的 Java 表达式语句后面必须有分号，花括号后面没有分号。

6．注释

注释用来解释程序的正文段，它的作用是对程序作一些说明。和大多数编程语言一样，Java 的注释不会在可执行程序中显示出来，所以应根据需要添加尽可能多的注释，不必担心会造成代码的膨胀。

Java 提供了 3 种方法来显示注释：

（1）//注释 表示自“//”标记开始，一直到本行结束，所有内容都会被当作注释对待。

例如：//这之后到本行末尾都是注释

（2）/* */注释 表示从“/*”开始到第一个“*/”之间的内容都为注释。

例如：/*这之间是注释*/

（3）/** */注释 这是 Java 语言特有的注释方式，同“/* */”注释一样，表示从“/**”开始到第一个“*/”之间的字符为注释，它只能在一个相邻的声明之前使用，该注释的内容将被包含进 javadoc 程序所自动生成的文档中。

例如：/**这里的注释将被包含在 javadoc 程序自动生成的文档中*/

在使用注释的过程中，必须注意几点：

（1）注释不允许嵌套使用。

（2）注释串不能出现在字符值或字符串值内。

（3）在“//”注释中，字符串“/*”、“*/”和“/**”不再表示注释，而是普通字符串。

（4）在“/* */”和“/** */”注释中，字符串“//”也不再表示注释。

1.2.2 数据类型

Java 是一种严格的“类型化”语言。这意味着每个变量都必须有一个已经声明好的类型。在 Java 语言中，数据类型可分为两大类：基本数据类型和引用类型。基本数据类型是不能再分解的数据类型，数据在作为函数参数时是以“传值”方式工作的；引用类型有时也称复合类型，它是可以分解为基本类型的数据类型，其数据在作为函数参数时是以“传址”的方式工作的。

Java 语言提供了 8 种基本类型：byte、short、int、long、char、float、double 和 boolean。这 8 种类型可分成 4 组：

- 整型：byte、short、int 和 long，用来表示正负数。
- 浮点型：float 和 double，用来表示小数。

- 字符型：char，用来表示字母或数字。
- 布尔型：boolean，用来表示逻辑值。

Java 的引用类型包括：

- 类：class。
- 接口：interface。
- 数组：array。

在本节中，我们将介绍 Java 语言的基本数据类型。

1．整型

整型即整数类型，是指那些没有小数部分的数字。Java 取消了 unsigned 含义，所以 Java 语言的整数类型都是有符号的。因为 Java 语言取消了无符号型，所以整数类型减少了一半。Java 语言共提供了 4 种整数类型，表 1-2 对这 4 种类型进行了说明。

表 1-2　　Java 整数类型

类　型	长　度	取 值 范 围
int	32	-2147483648~2147483647
short	16	-32768~32767
long	64	-9223372036854775808L~9223372036854775807L
byte	8	-128~127

在大多数情况下，int 类型是最实用的，范围从-2147483648~2147483647，刚好超过 20 亿。如果要以“毫秒”为单位来表示一年，就需要使用 long 类型，long 类型有一个后缀 L。byte 和 short 类型主要应用于特殊的应用程序，比如低级文件控制，或者用于对存储空间要求极大的大型数组。同 C/C++一样，十六进制数和八进制数则分别带有前缀 0x 和 o，比如 0xCAED 和 o437。

这里最关键的一点是，在 Java 语言中，整数类型的取值范围并不是由最终运行 Java 代码的机器来决定。这使得程序员可以方便自如地将一个软件移植到另一个平台，甚至能在相同平台的不同操作系统之间移植。在 C/C++语言中情况就不是这样了。比如说 int 类型，在 C/C++语言中，int 类型意味着整数类型将依赖于目标机器。对 16 位处理器来说，整数用 2 个字节表示。在 Interl Pentium 处理器上，整数类型则由具体的操作系统决定：对于 MS-DOS 和 Windows 3.1 来说，整数用 2 个字节表示；而对使用了 32 位模式的 Windows 9x 或 Windows NT 程序来说，整数就变成 4 个字节。在 Java 中这种情况是不存在的，它的所有数据类型的大小都是与平台无关的。

2．浮点型

浮点类型是指那些含有小数部分的数字，共有两种浮点类型，如表 1-3 所示。

表 1-3　　浮点类型

类　型	长　度	取 值 范 围
float	32	约±3.40282347E+38F（6～7 个有效的十进制数位）
double	64	约±1.79769313486231570E+308（15 个有效数位）

float 类型的数值有一个后缀 F，比如 4.516F。如果发现一个浮点数没有 F 后缀，就表示它默认为精度更高的 double 类型。当然，也可以明确指定一个 D 后缀，这同样表示双精度

需要全本请访问 www.ertongbook.com