数据库高级开发与专业应用系列

Java 数据库开发与专业应用

敬铮　主编

李伟　编著

国防工业出版社

·北京·

需要全本请访问 www.ertongbook.com

内容简介

本书全面系统的讲述了关于使用 JDBC 进行网络开发的方法以及相关的问题。全书共十四章：第一章概括了 JAVA 的发展历程；第二章简单介绍了它的开发环境；第三章简单介绍了 JAVA 的语法知识；第四章介绍了和本书相关的数据库基础知识；第五——第八章讲述了 JDBC 的相关知识；后面的章节向读者介绍了同 JDBC 相关的一些应用方式。

本书使用于从事网络数据库开发应用的工程技术人员和大专院校的师生，以及对于 JDVA 语言感兴趣的编程爱好者。

目　　录

第 1 章　Java 概述……1
1.1　Java 的起源……1
1.2　Java 的发展现状……2
1.3　Java 的特点……3
1.4　Java 的应用程序简介……5
1.5　小结……9
第 2 章　Java 开发环境……10
2.1　JDK 简介……10
2.2　JDK 的安装……10
2.3　JDK 的配置……13
2.4　小结……13
第 3 章　Java 语言……15
3.1　第一个简单的程序……15
3.2　Java 的数据类型……16
3.3　Java 的注释风格……17
3.4　Java 的标识符……18
3.5　Java 的分隔符……18
3.6　Java 的运算符……19
3.7　Java 的控制语句……23
3.8　Java 的数组……27
3.9　Java 的类……31
3.10　Java 中的异常处理……36
3.11　Java 的多线程……38
3.12　小结……42
第 4 章　数据库基础知识……43
4.1　关系数据库概述……43
4.1.1　关系数据库产生的历史……43
4.1.2　关系数据库的定义……43
4.1.3　关系数据库的基本术语……44
4.1.4　关系数据库的特点……45
4.2　SQL 语言……46
4.2.1　SQL 的历史……46
4.2.2　SQL 的简单查询功能……46

4.3 SQL 的多表查询 ····· 58
4.4 子查询操作 ····· 62
4.5 SQL 的数据操作功能 ····· 67
4.6 创建存储过程 ····· 77
4.7 小结 ····· 81
第 5 章 JDBC 概述 ····· 82
5.1 JDBC 简介 ····· 82
5.2 Java 的数据库模型 ····· 83
5.3 同 SQL 语言的一致性 ····· 84
5.4 JDBC 与 ODBC 和其他 API 的比较 ····· 84
5.5 JDBC 的产品 ····· 85
5.6 JDBC 使用场合 ····· 86
5.7 JDBC 的安全性 ····· 87
5.8 JDBC 的开发环境 ····· 89
5.9 小结 ····· 89
第 6 章 使用 JDBC 进行数据连接 ····· 90
6.1 建立数据源 ····· 90
6.2 使用 JDBC 与数据库建立连接 ····· 95
6.3 小结 ····· 101
第 7 章 使用 JDBC 对数据库进行操作 ····· 102
7.1 数据插入 ····· 102
7.1.1 一个简单的插入例子 ····· 102
7.1.2 代码详解 ····· 103
7.1.3 关于 Statement ····· 104
7.2 数据查询 ····· 106
7.3 关于 ResultSet 对象 ····· 114
7.4 使用 JDBC 进行数据删除操作 ····· 118
7.5 使用 JDBC 进行数据更新操作 ····· 125
7.6 使用 PreparedStatement 对象 ····· 131
7.6.1 PreparedStatement 对象概述 ····· 131
7.6.2 一个例子程序 ····· 132
7.6.3 PreparedStatement 对象用法详解 ····· 138
7.7 使用 CallableStatement 对象 ····· 140
7.7.1 CallableStatement 概述 ····· 140
7.7.2 一个简单的例子 ····· 141
7.7.3 CallableStatement 对象的使用方法 ····· 142
7.8 小结 ····· 144
第 8 章 使用 JDBC 进行数据库定义操作 ····· 146
8.1 使用 JDBC 建立数据表 ····· 146
8.2 使用 JDBC 释放数据表 ····· 148

8.3 建立和删除视图……150
8.3.1 建立视图……150
8.3.2 视图的删除……153
8.4 小结……155
第 9 章 服务器端程序设计初步……156
9.1 Servlet 简介……156
9.2 Servlet 应用范围……156
9.3 Servlet 的结构……157
9.4 Servlet 的生命周期……157
9.5 Servlet 运行及开发环境的建立……158
9.6 Servlet 入门……160
9.7 Servlet 进阶……163
9.8 小结……166
第 10 章 开发 Web 交互式应用程序……167
10.1 网页元素简介……167
10.2 使用 Servlet 开发交互式网络应用程序……177
10.3 小结……184
第 11 章 开发 Web 数据库应用程序……186
11.1 在 Servlet 中进行数据库连接操作……186
11.2 使用 Servlet 进行数据库查询操作……188
11.3 在 Servlet 中插入数据……190
11.4 在 Servlet 中进行数据库删除……193
11.5 在 Servlet 中进行数据库更新操作……196
11.6 在 Servlet 中使用 PreparedStatement 对象……199
11.7 在 Servlet 中使用 CallableStatement 对象……203
11.8 小结……206
第 12 章 Servlet 综合实例……208
12.1 留言版实例……208
12.2 网站计数器实例……220
12.3 人才市场实例……223
12.4 小结……275
第 13 章 Java Server Pages……276
13.1 JSP 概述……276
13.2 JSP 语法讲解……277
13.2.1 语法概貌……277
13.2.2 JSP 语法详解……278
13.3 简单的 JSP 的例子……292
13.4 JSP 与 JDBC……330
13.5 小结……332
第 14 章 JDBC API 参考……333

需要全本请访问 www.ertongbook.com

14.1 Class DriverManager 333
14.2 Interface Driver 336
14.3 Class DriverPropertyInfo 337
14.4 Interface Connection 339
14.5 Interface PreparedStatement 344
14.6 Interface ResultSet 352

第 1 章　Java 概述

1.1　Java 的 起 源

1．Java 是什么

美国硅谷有一句行话，每 10～15 年有一次轮回。最近的一次轮回就是从 Java 开始。

Java 是一个由 Sun 公司开发而成的新一代编程语言。使用它可在不同种机器、不同种操作平台的网络环境中开发软件。不论你使用的是哪一种 WWW 浏览器，哪一种计算机，哪一种操作系统，只要 WWW 浏览器上面注明了“支持 Java”，就可以看到生动的主页。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式，带来了自 PC 机以来又一次技术革命，为迅速发展的信息世界增添了新的活力。

Sun 的 Java 语言开发小组成立于 1991 年，其目的是开拓消费类电子产品市场，例如，交互式电视、烤面包箱等。Sun 内部人员把这个项目称为 Green，那时 World Wide Web 还在图纸上呢。该小组的领导人 James Gosling，是一位非常杰出的程序员。他出生于 1957 年，1984 年加盟 Sun Microsystem 公司，之前在一家 IBM 研究机构工作。他是 SunNeWs 窗口系统的总设计师，也是第一个用 C 语言实现 EMACS 的文本编辑器 COSMACS 的开发者。

在研究开发过程中，Gosling 深刻体会到消费类电子产品和工作站产品在开发哲学上的差异：消费类电子产品要求可靠性高、费用低、标准化、使用简单，用户并不关心 CPU 的型号，也不欣赏专用昂贵的 RISC 处理器，他们需要建立在一个标准基础之上，具有一系列可选的方案，从 8086 到 80586 都可以选取。

2．Java 的诞生

为了使整个系统与平台无关，Gosling 首先从改写 C 编译器着手。但是 Gosling 在改写过程中感到仅 C 语言是无法满足需要的，于是在 1991 年 6 月份开始准备开发一个新的语言，那么给它起一个什么名字呢?Gosling 回首向窗外望去，看见一棵老橡树，于是建一个目录叫 Oak，这就是 Java 语言的前身（后来发现 Oak 已是 Sun 公司另一个语言的注册商标，才改名为 Java，即太平洋上一个盛产咖啡的岛屿的名字）。

Gosling 在开始写 Java 时，并不局限于扩充语言机制本身，更注重于语言所运行的软硬件环境。他要建立一个系统，这个系统运行于一个巨大的、分布的、异构的网格环境中，完成各电子设备之间的通信与协同工作。Gosling 在设计中采用了虚机器码（Virtual Machine Code）方式，即 Java 语言编译后产生的是虚拟机，虚拟机运行在一个解释器上，每一个操作系统均有一个解释器。这样，Java 就成了平台无关语言。这和 Gosling 设计的 SunNeWs 窗口系统有着相同的技术味道。在 NeWs 中用户界面统一用 Postscript 描述，不同的显示器有不同的 Postscript 解释器，这样便保证了用户界面的良好的可移植性。

Patrick Naughton 也是 Sun 公司的技术骨干，曾经是 Open Windows 项目的负责人。当

Naughton 加入该小组后，整个工作进展神速。经过 17 个月的奋战，整个系统胜利完成。它是由一个操作系统、一种语言（Java）、一个用户界面、一个新的硬件平台、三块专用芯片构成的。通常情况下，这样的项目在 Sun 公司要 75 个人干三年。项目完成后，在 Sun 公司内部做了一次展示和鉴定，观众的反应是：在各方面都采用了崭新的、非常大胆的技术。许多参观者对 Java 留下了非常深刻的印象，特别得到 Sun 的两位领导人 ScottMcNealy 和 BillJoy 的关注，但 Java 的前途未卜。

3．Java 真正开始受 IT 业界的关注

到了 1994 年，WWW 已如火如荼地发展起来。Gosling 意识到 WWW 需要一个中性的浏览器，它不依赖于任何硬件平台和软件平台，它应是一种实时性较高、可靠安全、有交互功能的浏览器。于是 Gosling 决定用 Java 开发一个新的 Web 浏览器。

这项工作由 Naughton 和 JonathanPayne 负责，到 1994 年秋天，完成了 WebRunner 的开发工作。WebRunner 是 HotJava 的前身，这个原型系统展示了 Java 可能带来的广阔市场前景。WebRunner 改名为 HotJava，并于 1995 年 5 月 23 日发表后，在产业界引起了巨大的轰动，Java 的地位也随之而得到肯定。又经过一年的试用和改进，Java1.0 版终于在 1996 年年初正式发表。

1.2 Java 的发展现状

Java 是 Sun 公司推出的新的一代面向对象程序设计语言，特别适合于 Internet 应用程序开发，它的平台无关性直接威胁到 Wintel 的垄断地位。一时间，“连 Internet，用 Java 编程”成为技术人员的一种时尚。虽然新闻界的报导有点言过其实，但 Java 作为软件开发的一种革命性的技术，其地位已被确立，这表现在以下几个方面：

(1) 计算机产业的许多大公司购买了 Java 的许可证，包括 IBM, Apple, DEC, Adobe, SiliconGraphics,HP,Oracel,Toshiba，以及最不情愿的 Microsoft。这一点说明，Java 已得到了业界的认可。

(2) 众多的软件开发商开始支持 Java 的软件产品。例如：Borland 公司开发的基于 Java 的快速应用程序开发环境 Latte。Borland 公司的这一举措，推动了 Java 进入 PC 机软件市场。Sun 公司自己的 Java 开发环境 JavaWorkshop 已经发布。数据库厂商如：Illustra,Sysbase,Versant,Oracle 都在开发 CGI 接口，支持 HTML 和 Java。今天是以网络为中心的计算时代，不支持 HTML 和 Java，应用程序的应用范围只能限于同质的环境（相同的硬件平台）。

(3) Intranet 正在成为企业信息系统最佳的解决方案，而其中 Java 将发挥不可替代的作用。Intranet 的目的是把 Internet 用于企业内部的信息系统，它的优点表现在：便宜，易于使用和管理。用户不管使用何种类型的机器和操作系统，界面是统一的 Intrnet 浏览器，而数据库、Web 页面、应用程序(用 Java 编的 Applet)则存在 WWW 服务器上，无论是开发人员，还是管理人员，抑或是用户都可以受益于该解决方案。Java 语言正在不断发展和完善，Sun 公司是主要的发展推动者，较通用的编译环境有 JDK(JavaDevelopKit)与 JWS(JavaWorkshop)。还有很多其他公司正在开发 Java 语言的编译器与集成环境，预计不久 Java 语言的正确性与效率都将会提高，用户用 Java 编程和现在用 C++编程一样方便。

1.3　Java 的特点

Java 到底是一种什么样的语言呢？Java 是一种简单的、面向对象的、分布式的、解释的、健壮的、安全的、结构中立的、可移植的、性能很优异的、多线程的、动态的语言。

1．简单

Java 最初是为对家用电器进行集成控制而设计的一种语言，因此它必须简单明了。Java 语言的简单性主要体现在以下三个方面：

(1) Java 的风格类似于 C++，因而 C++程序员是非常熟悉的。从某种意义上讲，Java 语言是 C 及 C++语言的一个变种，因此，C++程序员可以很快就掌握 Java 编程技术。

(2) Java 摒弃了 C++中容易引发程序错误的地方，如指针和内存管理。

(3) Java 提供了丰富的类库。

2．面向对象

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的，它不支持类似 C 语言那样的面向过程的程序设计技术。Java 支持静态和动态风格的代码，继承及重用。单从面向对象的特性来看，Java 类似于 SmallTalk，但其他特性，尤其是适用于分布式计算环境的特性远远超越了 SmallTalk。

3．分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此，Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。为分布环境尤其是 Internet 提供动态内容无疑是一项非常宏伟的任务，但 Java 的语法特性却使我们很容易地实现这项目标。

4．健壮性

Java 致力于检查程序在编译和运行时的错误。类型检查帮助检查出许多开发早期出现的错误。Java 自己操纵内存减少了内存出错的可能性。Java 还实现了真数组，避免了覆盖数据的可能。这个功能特征大大缩短了开发 Java 应用程序的周期。Java 提供：Null 指针检测，数组边界检测，异常出口字节代码校验。

5．结构中立

另外，为了使 Java 作为网络的一个整体，Java 将它的程序编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码。现在，Java 运行系统有 Solaris2.4(SPARC),Win32 系统(Windows95 和 WindowsNT)等。Java 源程序被编译成一种高层次的与机器无关的 byte-code 格式语言，这种语言被设计在虚拟机上运行，由机器相关的运行调试器实现执行。

6．安全

Java 的安全性可从两个方面得到保证。一方面，在 Java 语言里，像指针和释放内存等 C++功能被删除，避免了非法内存操作。另一方面，当 Java 用来创建浏览器时，语言功能和一个浏览器本身提供的功能结合起来，使它更安全。Java 语言在机器上执行前，要经过代码校验，检查代码段的格式，检测指针操作、对象操作是否过分以及试图改变一个对象

的类型等一系列的测试。

7．完善的校验机制

如果 byte-code 通过代码校验，没有返回错误，我们可知道：代码没有堆栈上溢出和下溢出，所有操作代码参数类型都是正确的，没有发生非法数据转换，如将整数转换成指针，访问对象操作是合法的。

8．类装载

ClassLoader 通过将本机类与网络资源类的名称分开，来保持安全性。因为调入类时总要经过检查，这样避免了特洛伊木马现象的出现。从网络上下载的类被调进一个与源相关的私有的名字域。当一个私有类访问另一个类时，build-in(本机类)首先被检查，然后检查相关的类。这样就避免了破坏本机类情况的出现。

9．可移植

同体系结构无关的特性使得 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行，这成为 Java 应用软件便于移植的良好基础。但仅仅如此还不够。如果基本数据类型设计依赖于具体实现，也将为程序的移植带来很大不便。例如在 Windows3.1 中整数(Integer)为 16bits，在 Windows95 中整数为 32bits，在 DECAlpha 中整数为 64bits，在 Intel486 中为 32bits。通过定义独立于平台的基本数据类型及其运算，Java 数据得以在任何硬件平台上保持一致。Java 语言的基本数据类型及其表示方式如下：byte8-bit 二进制补码 short16-bit 二进制补码，int32-bit 二进制补码，long64-bit 二进制补码，float32-bitIEEE754 浮点数，double32-bitIEEE754 浮点数，char16-bitUnicode 字符。

在任何 Java 解释器中，数据类型都是依据以上标准具体实现的。因为几乎目前使用的所有 CPU 都能支持以上数据类型、8～64 位整数格式的补码运算和单/双精度浮点运算。Java 编译器本身就是用 Java 语言编写的。Java 运算系统的编制依据 POSIX 方便移植的限制，用 ANSIC 语言写成。Java 语言规范中也没有任何“同具体实现相关”的内容。

10．解释

Java 解释器(运行系统)能直接运行目标代码指令。链接程序通常比编译程序所需资源少，所以程序员可以在创建源程序上花上更多的时间。

11．高性能

如果解释器速度不慢，Java 可以在运行时直接将目标代码翻译成机器指令。Sun 用直接解释器一秒钟内可调用 30 万个过程。翻译目标代码的速度与 C/C++的性能没什么区别。

12．多线程

Java 提供的多线程功能使得在一个程序里可同时执行多个小任务。线程——有时也称小进程——是一个大进程里分出来的小的独立的进程。因为 Java 实现的是多线程技术，所以比 C 和 C++更健壮。多线程带来的更大的好处是更好的交互性能和实时控制性能。当然实时控制性能还取决于系统本身(UNIX,Windows,Macintosh 等)，在开发难易程度和性能上都比单线程要好。任何用过当前浏览器的人，都感觉为调一幅图片而等待是一件很烦恼的事情。在 Java 里，你可用一个单线程来调一幅图片，而你可以访问 HTML 里的其他信息而不必等它。

13．动态

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行过程中所需要的类，这是 C++语言进行面向对象程序设计所无法实现的。在 C++程序设计过程中，每当

在类中增加一个实例变量或一种成员函数后，引用该类的所有子类都必须重新编译，否则将导致程序崩溃。Java 从如下几方面采取措施来解决这个问题。Java 编译器不是将对实例变量和成员函数的引用编译为数值引用，而是将符号引用信息在字节码中保存，传递给解释器，再由解释器在完成动态连接类后，将符号引用信息转换为数值偏移量。这样，一个在存储器生成的对象不在编译过程中决定，而是延迟到运行时由解释器确定的。这样，对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时，这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次，随后代码便可以全速执行。在运行时确定引用的好处是可以使用已被更新的类，而不必担心会影响原有的代码。如果程序连接了网络中另一系统中的某一类，该类的所有者也可以自由地对该类进行更新，而不会使任何引用该类的程序崩溃。Java 还简化了使用一个升级的或全新的协议的方法。如果你的系统运行 Java 程序时遇到了不知怎样处理的程序，没关系，Java 能自动下载你所需要的功能程序。

1.4　Java 的应用程序简介

Java 的应用程序类型多种多样，适合于不同的场合，下面我们给大家一一介绍。

1．JavaApplet

在 Internet 上使用最多的 Java 应用就是 Java 小程序（Applet）了。它的特性介绍如下：

简单来讲,Java 应用程序指独立的由 Java 语言编制、并能够由 Java 解释器运行的程序。Java 应用程序一般以命令行方式运行。

而 Javaapplet,一般可在 WWW 浏览器上运行。开发人员使用特殊的 HTML 标记,把 applet 嵌入到 Web 页面中,用户使用支持 Java 的浏览器从 Internet 中某一服务器中读入一个包含 Applet 的页面,然后在本地运行该 Applet。

由于 Javaapplet 可在支持 Java 的浏览器上运行,所以可以使用浏览器的一些复杂的图形显示和图像处理包,也可以使用用户界面(UI)元素、网络编程和事件处理机制。而 Java 应用程序在利用这些特性时相对困难一些。

由于 Applet 可从网络上任何地方下载并在本地机器上运行,从安全性角度考虑,比如防病毒,有必要对 applet 采取一些限制:

- 除非在访问控制列表中指定文件目录，否则不允许 Applet 在本地文件系统上完成读写操作。
- Applet 只能与原始保存该程序的服务器之间进行通信。
- Applet 不能再启动和运行客户的程序。在 Unix 系统上,则不能创建进程。
- Applet 也不能在本地平台上加载本地程序,包括使用共享库,比如 DLL 等。

为了给大家一个感性的认识，我们这里给出一个 Java Applet 小程序的演示画面，如图 1.1 所示。

2．Java 应用程序

Java 应用程序是使用的比较多的一种，特点是丰富的 GUI 界面元素，适用于绝大部分系统。它与传统的 C/C++程序的入口方式基本一致。

这里我们给出一个简单的字符界面应用程序的例子。

这是一个简单的 Hello World 式的应用程序，如图 1.2 所示。

需要全本请访问 www.ertongbook.com



图 1.1　一个 Java Applet 演示

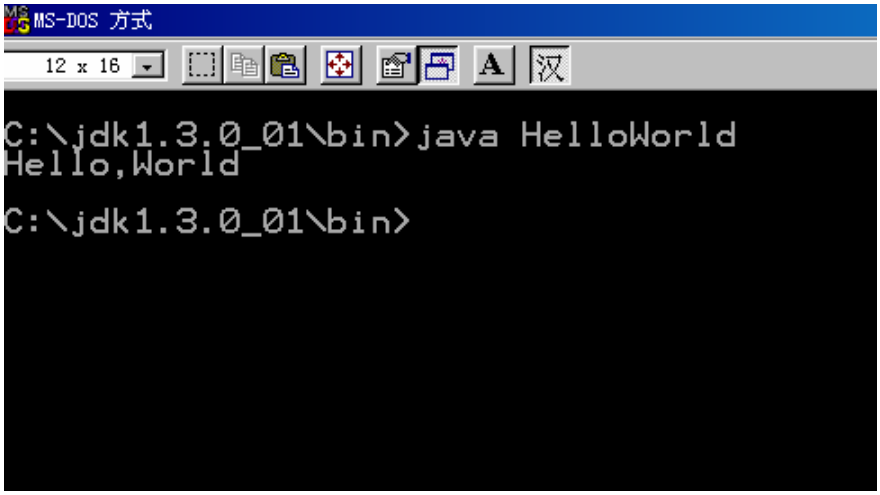


图 1.2　控制台应用程序演示

下面我们再给出一个图形用户界面的例子，让读者看看 Java 强大的 GUI 功能。

这个例子演示了在不同的系统下面图形界面的一致性。具体的演示效果如图 1.3~图 1.5 所示。

3．服务器端的应用程序

随着互联网的发展，越来越多的计算任务要由服务器来进行处理，“肥服务器，瘦客户端”是目前最为流行的互联网计算模式。Java 也推出了自己独有的服务器端应用程序开发模式。Serverlet 便是 Java 的服务器端应用程序的简称。它具有不可比拟的特性，效率之高令其他同类的开发模式汗颜。下面就是一个简单的服务器端计算的演示。

图 1.6 显示的页面是让我们填入一些信息。

图 1.7 显示的我们刚刚填入的信息，已经被 Servlet 传递到了另外一个页面上了，由此我们可以看出 Servlet 强大的服务器计算功能。

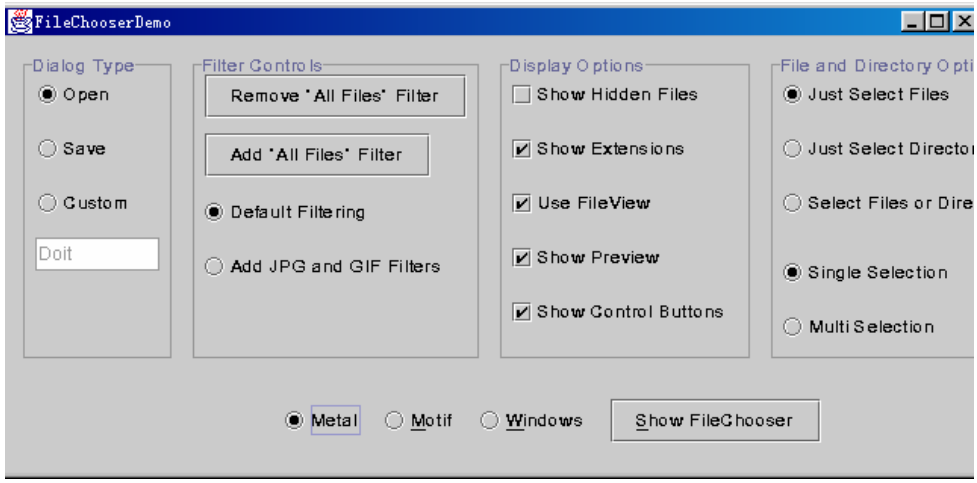


图 1.3　Metal 系统的 GUI 界面

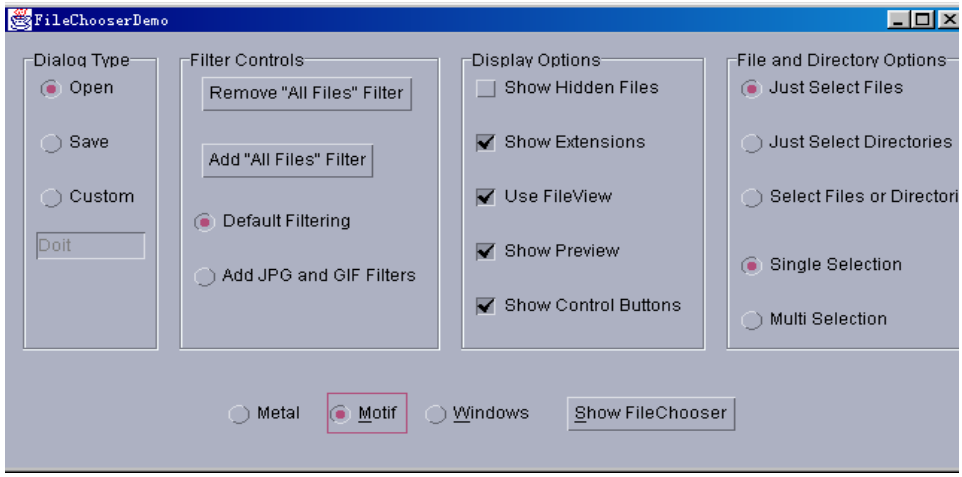


图 1.4　Motiff 系统的 GUI 界面

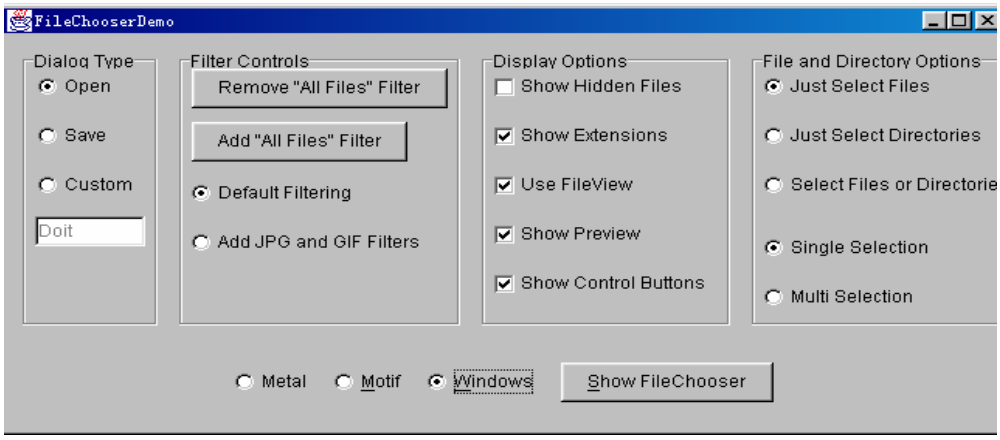


图 1.5　Windows 系统的 GUI 界面

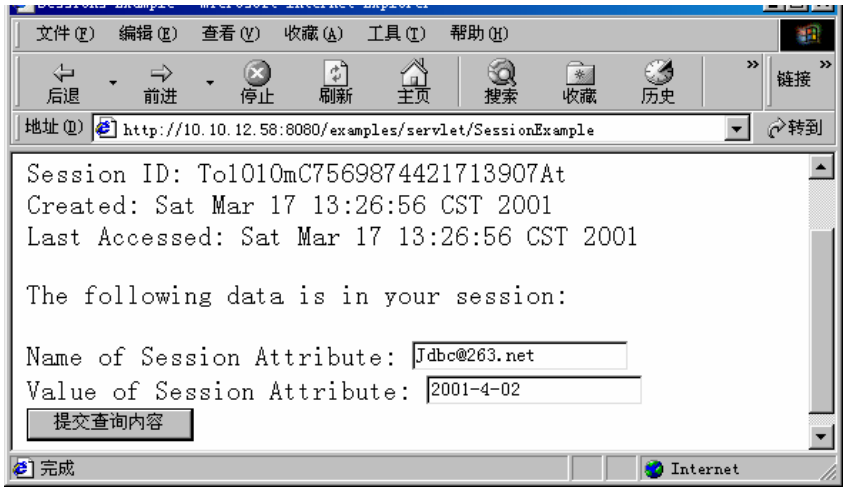


图 1.6　填入信息

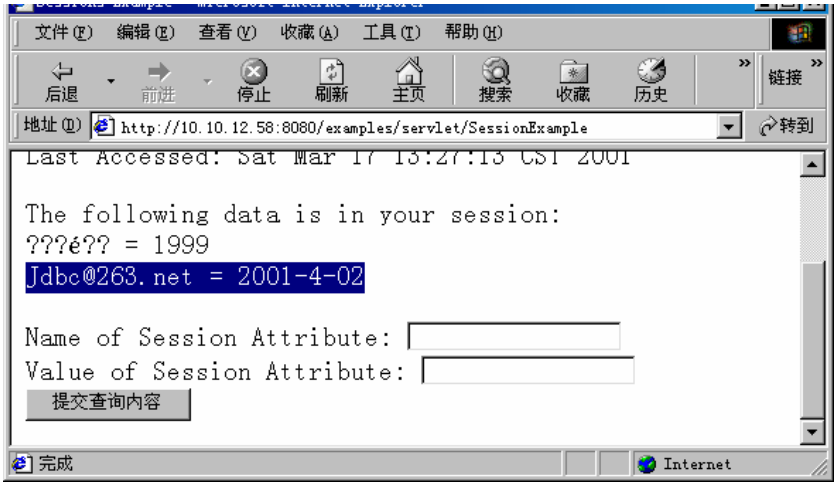


图 1.7　简单的计算功能

4．最新的计算模式 JSP

前一阶段流行的非常火热的服务器端脚本开发语言 ASP 读者一定听说过。我想很可能读者当中的一部分人使用过 ASP。的确，ASP 作为一种简单快捷的脚本语言，有着自身的优点，它最为突出的部分就是可以直接将 ASP 代码放入到 HTML 语句中。光凭这一点，就足以让其他的 CGI 程序自叹不如。但是 ASP 也有着它自身的不足。

(1) 仅仅支持 Windows 操作系统

目前作为网络服务器的操作系统一般以 Unix 为主，Windows NT 仅仅占有了低端计算的部分份额，所以这令 ASP 应用大为受限。

(2) 执行效率比较低

针对这种情况 Java 推出了 JSP，它具有 ASP 的大部分优点，同 Java Beans 组件技术相互结合使得使用 Java 进行 Web 应用开发如虎添翼。

下面给出一个 JSP 的例子，以便使大家对于 JSP 有一个感性的认识。

图 1.8 演示的是一个使用 JSP 输出网页的例子。



图 1.8　简单的 Jsp 程序

看了这么多关于 Java 的介绍，读者一定已经跃跃欲试了，从下一章开始我们将全面介绍关于 Java 的开发和应用。

1.5　小　　结

1．问：Java 语言能够干什么？

答：Java 语言几乎无所不包，能够胜任的范围非常广泛：消费类电子产品、嵌入式操作系统、手机、网络、电子商务等等。

2．问：Java 语言容易学习吗？

答：那要看同什么语言进行比较，还得看看学习者的自身素质如何。总体来说 Java 比 C++要简单许多，因为它去掉了 C++许多晦涩的语法规范，但是也需要学习者能够深刻体会到面向对象的编程思想精髓。

3．问：C/C++程序员学习 Java 困难吗？

答：从总体上来说如果程序员能够领会到面向对象的思想，那么学习 Java 应该是易如反掌的事情。因为毕竟 Java 是从 C/C++那里继承过来的。

4．问：目前 Java 语言本身有什么缺陷吗？

答：Java 由于是一种解释执行的语言，它的执行速度肯定赶不上 C/C++这样的编译语言。所以速度是目前影响 Java 推广的最大的障碍，也使得它不能很好地在桌面操作系统上应用。

5．问：Java 的主要应用系统是什么？

答：由于 Java 执行速度的欠缺、移植性强的特点，使得 Java 主要应用在高端服务器系统——Unix、Solaries、AIX 等。

第 2 章　Java 开发环境

2.1　JDK 简 介

Java 开发工具（JDK）是许多 Java 专家最初使用的开发环境。尽管许多编程人员已经使用第三方的开发工具，但 JDK 仍被当作 Java 开发的重要工具。

JDK 由一个标准类库和一组建立、测试及建立文档的 Java 实用程序组成。其核心 JavaAPI 是一些预定义的类库，开发人员需要用这些类来访问 Java 语言的功能。JavaAPI 包括一些重要的语言结构以及基本图形、网络和文件 I/O。一般来说，JavaAPI 的非 I/O 部分对于运行 Java 的所有平台是相同的，而 I/O 部分则仅在通用 Java 环境中实现。

作为 JDK 实用程序，工具库中有七种主要程序:

Javac：Java 编译器，将 Java 源代码转换成字节码。

Java：Java 解释器，直接从类文件执行 Java 应用程序字节代码。

Appletviewer：小程序浏览器，一种执行 HTML 文件上的 Java 小程序的 Java 浏览器。

Javadoc：根据 Java 源代码及说明语句生成 HTML 文档。

Jdb：Java 调试器，可以逐行执行程序，设置断点和检查变量。

Javah：产生可以调用 Java 过程的 C 过程，或建立能被 Java 程序调用的 C 过程的头文件。

Javap：Java 反汇编器，显示编译类文件中的可访问功能和数据，同时显示字节代码含义。

2.2　JDK 的安装

下面就请跟着我一起来安装 JDK。

首先运行安装程序。出现如图 2.1 所示的界面。JDK 的安装界面非常友好,它首先向用户简要介绍 JDK 的基本的情况,由用户决定是否继续安装。用户在每一步都可以决定是否退出安装。我们这里点击“ Next ”按钮，进行下一步操作。

这时出现如图 2.2 所示的界面。这是一个使用协议，虽然 JDK 是一个免费的软件，但是，它还是有很多协议需要遵守的，读者不妨仔细阅读一下。这里，我们点击“ Next ”按钮，进行下一步操作。

出现如图 2.3 所示的界面。这是一个安装路径选择的界面，缺省的安装路径是 C:盘 jdk1.3.0_01 目录，您可以点击“ Browse ”按钮来进行目录的选择。这里我们点击“ Next ”按钮进行下一步操作。

出现如图 2.4 所示的界面。这里列出的是要安装的 JDK 组件，你可以自由选择。我们这里将各个组件的作用说明如下：

(1) Programe Files。包括了 JDK 所需的一些工具文件，比如说 java.exe，Javac.exe，Javasoc.exe 等等。



图 2.1　初始安装界面

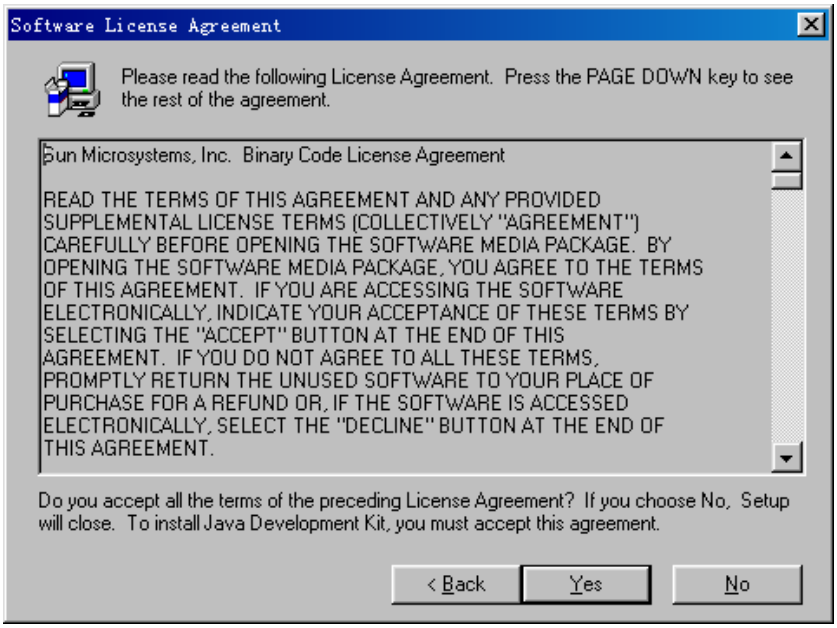


图 2.2　点击下一步出现的界面

(2) Native Interface Header Files。包含的是本地化的接口文件。这个组件的作用不言自明，就是使得 Java 生成的代码能够本地化，加快运行速度。

(3) Old Native Interface Header Files。是为了和以前的 JDK 版本的代码相互兼容。软件

需要全本请访问 www.ertongbook.com