

最新 JBuilder 8.0 程序员指南

北京希望电子出版社 总策划
顾明皓 赵文武 等 编 著



中国科学技术出版社
CHINA SCIENCE AND TECHNOLOGY PRESS

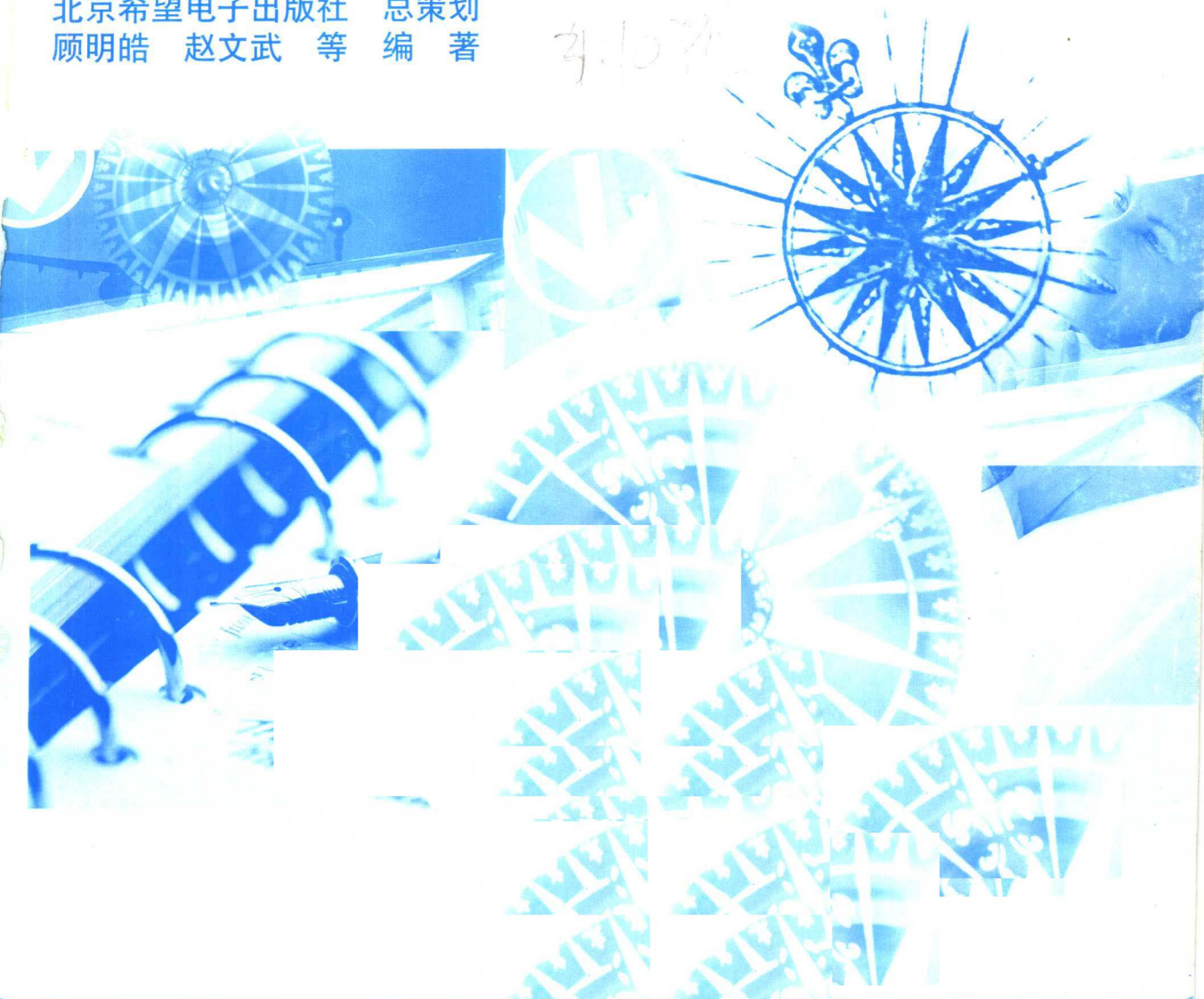


北京希望电子出版社
Beijing Hope Electronic Press
www.bhep.com.cn

最新 JBuilder 8.0 程序员指南

北京希望电子出版社 总策划
顾明皓 赵文武 等 编 著

41071



中国科学技术出版社
CHINA SCIENCE AND TECHNOLOGY PRESS



北京希望电子出版社
Beijing Hope Electronic Press
www.bhpe.com.cn

图书在版编目 (CIP) 数据

最新 JBuilder 8.0 程序员指南/顾明皓 等编著. —北京:
中国科学技术出版社, 2003.8
ISBN 7-5046-3551-0

I. 最... II. 顾... III. JAVA 语言—程序设计
IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 039068 号

书 名: 最新 JBuilder 8.0 程序员指南

总 策 划: 北京希望电子出版社

文 本 著 作 者: 顾明皓 赵文武 等

责 任 编 辑: 周凤明 沈葆华

出版、发行者: 中国科学技术出版社 北京希望电子出版社

地 址: 北京市海淀区中关村南大街 16 号 100081

北京市海淀区知春路甲 63 号卫星大厦三层 100080

网址: www.bhp.com.cn E-mail: lwm@bhp.com.cn zwb@bhp.com.cn

电话: 010-62520290, 62521724, 62528991, 62630301, 62524940, 62521921,
62521724 (发行) 010-82675588-318, 62532258, 62562329 (门市)

010-82675588-501, 82675588-201 (编辑部)

经 销: 各地新华书店、软件连锁店

排 版: 希望图书输出中心 马 君

印 刷 者: 北京媛明印刷厂

开本 / 规格: 787 毫米×1092 毫米 1/16 25 印张 607 千字

版次 / 印次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-5046-3551-0/TP·181

定 价: 38.00 元

前 言

Java 语言是 SUN 公司推出的一种面向对象的、多线程的、交互式的编程语言，它功能强大，表达能力强，应用广泛，是当前最为流行的编程语言。Java 语言的平台无关特性非常适用于网络和分布式应用，因此一经推出就备受青睐；其卓越的设计思想也使 Java 成为国际互联网中的“世界语”，并将网络的发展带入一个新的纪元。

JBuilder 是 Borland 公司推出的一种基于组建技术的可视化 Java 集成开发环境，适用于各种层次的项目开发，包括数据库开发、网络应用、多媒体编程等；并集程序的编写、调试、扩展、运行于一体，极大地提高了开发人员的工作效率。JBuilder8.0 是 JBuilder 的最新产品，它支持 JDK1.4.1，并可使用 Java Soft 提供的丰富组件资源，实现多种 Java 应用程序的开发。JBuilder8.0 的可视化开发系统具有界面友好、高度集成、资源丰富、执行高效等特点，且同以往版本相比功能更趋完善。

全书共分 12 章，第 1 章介绍 Java 语言的发展历史，应用前景，语言基本特点及实现机制。第 2 章系统地介绍了 JBuilder8.0 开发环境的建立及使用方法。第 3 章重点介绍 Java 语言的语法体系。第 4 章介绍 Java 语言的基本实现方式，如何在 JBuilder8.0 可视化开发环境中实现 Java 图形用户界面编程，重点讲解 AWT 组件和布局管理器的使用。第 5 章介绍 Java 小应用程序及其在 JBuilder8.0 中的实现，内容涉及图形绘制及使用字体和颜色类。第 6 章介绍 JavaBean 组件技术的基础知识以及 JBuilder8.0 提供的组件资源，重点学习 Swing 和 Swing Container 组件的使用方法。第 7 章介绍 Java 对数据库的支持，重点讲解如何在 JBuilder8.0 中，通过 DataExpress 和 dbSwing 等组件实现数据库开发。第 8 章介绍多线程的基本概念和应用，着重讲解如何创建线程，实现线程同步，避免发生死锁，以及利于线程实现通信功能。第 9 章介绍 Java 网络编程的基础知识，重点讲解如何获取网络资源，如何创建服务器端/用户端程序，以及如何实现网络通信。第 10 章综合介绍 Java 语言对多媒体技术的支持，内容涉及图像处理、动画效果及声音技术。第 11 章介绍 XML 语言的基础知识，及如何使用 JBuilders8.0 开发 XML 代码。在读者对 Java 应用程序开发有了一定的理解后，第 12 章将介绍应用程序的调试方法，使读者的编程水平能进一步提高。

本书以 JBuilder8.0 在 Java 应用程序开发中的应用为主线，侧重基础性和实用性。本书各章节内容相对独立，从基础语法到工程应用与开发，深入浅出地介绍了 Java 语言的各项基本要素和功能应用，内容丰富，通俗易懂。本书立足读者的编程基础，语言简洁流畅，逻辑性强，可以引导读者循序渐进地掌握各知识点。

本书由顾明皓和赵文武共同执笔编写。此外，张华东、李晓、范之誉、王宏、李琦、王瑾、吴君华、付鑫育、李龙、钱少伟、刘荣强、李伟光、朱峰、许大中、魏勇、萧玉、丁桦、李林、邵华刚、朱莉和肖育新等同志在整理材料方面给予了编者很大的帮助。在此，编者向他们表示深深的谢意。

由于时间有限，加之作者水平有限，书中的错误和不足之处在所难免，竭诚欢迎广大读者对本书提出批评和建议。

目 录

第1章 Java 概述..... 1	3.2 Java 的数据类型..... 52
1.1 Java 语言简介..... 1	3.2.1 基本数据类型..... 52
1.1.1 Java 的起源和发展..... 1	3.2.2 数据类型的转换..... 56
1.1.2 Java 的应用前景..... 2	3.2.3 基本数据类型的例子..... 57
1.2 Java 的特性..... 3	3.3 Java 的数组..... 58
1.2.1 Java 语言的特点..... 3	3.3.1 一维数组..... 59
1.2.2 Java 和 C++ 的比较..... 5	3.3.2 多维数组..... 60
1.3 Java 的设计目标和实现机制..... 7	3.3.3 字符串..... 62
1.3.1 Java 的设计目标..... 7	3.4 Java 的流程控制..... 64
1.3.2 Java 的实现机制..... 7	3.4.1 分支选择语句..... 64
1.4 本章小结..... 8	3.4.2 循环迭代语句..... 67
1.5 习题..... 8	3.4.3 跳跃转移语句..... 69
第2章 JBuilder 8.0 开发环境..... 9	3.5 Java 的类和对象..... 72
2.1 JBuilder 8.0 的新特点..... 9	3.5.1 面向对象编程的几个基本概念..... 72
2.2 JBuilder 的安装..... 11	3.5.2 类的定义和实现..... 74
2.2.1 JBuilder 8.0 Enterprise 的安装环境..... 11	3.5.3 类的构造函数及重载..... 78
2.2.2 JBuilder 8.0 Enterprise 的安装过程..... 12	3.5.4 对象..... 81
2.3 JBuilder 8.0 的界面组成..... 16	3.5.5 访问权限..... 84
2.3.1 JBuilder 8.0 主窗口..... 16	3.6 类的继承和接口..... 87
2.3.2 主菜单..... 19	3.6.1 类的继承..... 87
2.3.3 工具栏..... 34	3.6.2 接口..... 90
2.4 编写自己的 Java 程序..... 35	3.7 Java 的包..... 93
2.4.1 创建 MyApplet 程序..... 35	3.7.1 包的创建和引用..... 93
2.4.2 实现 Applet 程序..... 42	3.7.2 包的访问权限..... 95
2.5 本章小结..... 45	3.7.3 Java 的常用包..... 95
2.6 习题..... 45	3.7 异常处理..... 100
第3章 Java 的语法..... 46	3.7.1 try-catch-finally 结构..... 100
3.1 Java 的词法..... 46	3.7.2 异常的类型..... 102
3.1.1 Java 的关键字..... 46	3.7.3 throw 语句..... 102
3.1.2 Java 的标识符..... 46	3.8 注意事项..... 103
3.1.3 Java 的注释..... 47	3.9 本章小结..... 103
3.1.4 Java 的常量..... 48	3.10 习题..... 103
3.1.5 Java 的运算符..... 49	第4章 Java 图形用户界面编程..... 104
3.1.6 Java 的分隔符..... 51	4.1 建立 Java 应用的方法..... 104
	4.1.1 Applet 程序..... 104
	4.1.2 Application 程序..... 104

4.2 AWT 组件包.....	110	6.1.3 JavaBean API	188
4.2.1 AWT 包简介	110	6.2 JavaBean 组件的构成.....	190
4.2.2 AWT 的事件处理.....	113	6.2.1 创建自己的 JavaBean 组件... 191	
4.2.3 按钮 (Button) 组件.....	117	6.2.2 JBuilder 8.0 中的 JavaBean 组件	196
4.2.4 标签 (Label) 组件.....	121	6.3 JavaBean 组件的应用.....	199
4.2.5 文本域 (TextArea) 和文 本框 (TextField) 组件	121	6.3.1 JPanel 和 JtabbedPane 组件的应用.....	199
4.2.6 列表框 (List) 组件	125	6.3.2 JSplitPane、JScrollPane 和 JTree 组件的应用	202
4.2.7 下拉框 (Choice) 组件	129	6.3.3 JMenuBar、JPopupMenu、 JToolBar 和 JFileChooser 组件 的应用.....	209
4.2.8 复选框 (Checkbox) 组件... 131		6.4 本章小结.....	218
4.2.9 单选框 (CheckboxGroup) 组件	135	6.5 习题.....	218
4.2.10 滚动条 (Scrollbar) 组件 ... 138		第 7 章 Java 数据库编程	219
4.3 布局管理器.....	141	7.1 数据库概述.....	219
4.3.1 BorderLayout 布局管理器 141		7.1.1 数据库的基本概念.....	219
4.3.2 FlowLayout 布局管理器	144	7.1.2 数据库模型.....	220
4.3.3 XYLayout 布局管理器.....	146	7.1.3 几种数据库产品	221
4.3.4 GridBagLayout 布局管理器.. 147		7.1.4 SQL 语言简介	222
4.3.5 PaneLayout 布局管理器.....	150	7.1.5 ODBC 数据库.....	224
4.3.6 其他的布局管理器.....	153	7.2 JDBC 概述	225
4.4 本章小结.....	153	7.2.1 JDBC 的特点	226
4.5 习题.....	153	7.2.2 JDBC 的接口、类和异常	227
第 5 章 Java 的小应用程序.....	154	7.2.3 使用 DriverManager 实现数 据库的连接.....	233
5.1 小程序简介.....	154	7.3 数据库组件的使用	240
5.1.1 小程序的创建和执行.....	155	7.3.1 DataExpress 组件.....	240
5.1.2 小程序的 HTML 文档	159	7.3.2 dbSwing 组件和 More db- Swing 组件	241
5.1.3 HTML 和小程序间的参数 传递	161	7.3.3 数据库组件的应用.....	242
5.2 小程序的应用.....	164	7.4 本章小结.....	247
5.2.1 小程序的组件应用.....	164	7.5 习题.....	247
5.2.2 小程序的图形绘制.....	167	第 8 章 Java 多线程编程	248
5.2.3 颜色类在小程序中的应用 177		8.1 关于线程.....	248
5.2.4 字体类在小程序中的应用 181		8.1.1 线程概述.....	248
5.3 本章小结.....	184	8.1.2 Java 与多线程.....	249
5.4 习题.....	184	8.1.3 线程的管理.....	250
第 6 章 JavaBean 组件编程.....	185		
6.1 JavaBean 简介	185		
6.1.1 JavaBean 的基本概念	185		
6.1.2 JavaBean 的组件模型	186		

8.2 线程的实现.....	251	10.3.1 实现动画的技巧.....	336
8.2.1 Thread 类.....	251	10.3.2 动画技术的实现.....	338
8.2.2 Runnable 接口的定义和应用...	257	10.4 本章小结.....	342
8.3 线程的同步.....	260	10.5 习题.....	342
8.3.1 Synchronized 方法.....	260	第 11 章 Java 与 XML	343
8.3.2 线程同步的实现.....	262	11.1 XML 概述	343
8.3.3 关于死锁	265	11.1.1 XML、HTML 和 SGML	
8.4 线程的通信.....	267	的比较	343
8.4.1 管道通信	267	11.1.2 XML 语言的优点和缺点	346
8.4.2 标识符通信.....	272	11.1.3 XML 语言的三要素	347
8.5 本章小结.....	275	11.2 XML 的基本语法规则	349
8.6 习题.....	275	11.2.1 XML 文件的基本组成	349
第 9 章 Java 网络编程.....	276	11.2.2 XML 文件的总体结构	353
9.1 计算机网络.....	276	11.2.3 编写一个简单的 XML 文件...	355
9.1.1 计算机网络的起源.....	276	11.3 XML 分析器	357
9.1.2 计算机网络的发展.....	277	11.3.1 分析 XML 文件	357
9.1.3 计算机网络的一些基本概念...	279	11.3.2 XML 文件分析器的应用	359
9.2 URL 类的应用.....	280	11.4 文档类型定义 DTD	364
9.2.1 URL 和 URL 类.....	280	11.4.1 DTD 声明的基本形式.....	364
9.2.2 URL 类的应用.....	282	11.4.2 DTD 与 XML 的结合	367
9.3 Socket 类的应用	287	11.4.3 XML 与 DTD 的转换	370
9.3.1 Socket、Socket 类和		11.5 本章小结.....	372
ServerSocket 类	287	11.6 习题.....	372
9.3.2 Client/Server 程序	289	第 12 章 异常和程序调试	373
9.4 Datagram (数据报) 类的应用	298	12.1 关于异常	373
9.4.1 数据报通信基础.....	298	12.1.1 异常类型.....	373
9.4.2 数据报通信的实现.....	300	12.1.2 捕获异常.....	374
9.5 本章小结.....	308	12.1.3 抛出异常.....	379
9.6 习题.....	308	12.2 程序调试.....	382
第 10 章 Java 多媒体编程	309	12.2.1 调试前准备.....	382
10.1 图像处理.....	309	12.2.2 断点.....	384
10.1.1 图像的载入、显示.....	309	12.2.3 调试窗口.....	386
10.1.2 图像的转换处理.....	316	12.2.4 执行到指定位置.....	388
10.1.3 图像处理的实现.....	320	12.2.5 单步运行操作.....	388
10.2 声音技术.....	332	12.2.6 Evaluate/Modify 窗口.....	390
10.2.1 AudioClip 类.....	332	12.2.7 Watch 窗口.....	391
10.2.2 声音处理的实现.....	333	12.3 本章小结.....	392
10.3 动画技术.....	336	12.4 习题.....	392

第 1 章 Java 概述

本章要点

- 掌握 Java 语言的基本特点
- 明确 Java 语言的设计目标和实现机制

1.1 Java 语言简介

1.1.1 Java 的起源和发展

相信大家对 Internet 一定都不陌生,它是一个由各种不同类型的独立运行和管理的计算机网络组成,采用 TCP/IP 协议互联的全球性计算机网络。组成 Internet 的计算机网络可以是小规模局域网(LAN),城市规模的城域网(MAN),或是大规模的广域网(WAN)等等。Internet 可以为各类用户提供有价值的信息和其他服务。可以说,它是一个世界规模的巨大的信息资源和服务资源,不仅带来了计算机产业的又一次革命,而且也是人类历史发展的一个里程碑。

软件是计算机的灵魂,当计算机业向网络时代迈进的时候,计算机软件业也必然要经受同样的洗礼。互联网的出现虽然极大地方便了信息的共享和交流,但人们渴望能更进一步——共享处理能力和软件!如果可以实现这一步,Internet 对于用户而言就不仅仅是一个巨大的硬盘,而是一台巨大的计算机,操作平台和文件格式的壁垒已经不复存在,取而代之的是透明的应用和自由地获取信息。

20 世纪 90 年代中期,随着 Internet 的迅猛发展和 WWW 的迅速普及,计算机业界急需一种真正意义的网络语言,Java 语言就是以此为契机迅速发展起来的。Java 实际上包含了两方面的含义:从狭义角度讲,Java 只是由 Sun 公司开发的一种面向对象的编程语言,该语言能够通过网络发布并动态执行任务;广义的 Java 除了语言本身外,还包括一个客户机/服务器模式下的开发和执行环境,并代表了一种新的发展思路 and 方向。因此,作为一种编程语言,Java 具有划时代的意义。

Java 语言的前身是一种名为 Oak(橡树)的开发语言,Oak 语言的设计初衷主要是为方便家庭电器产品的交互式操作,不但语言本身非常简洁而高效,生成的可执行代码也非常短小精悍,更为重要的是 Oak 语言中采用了非常巧妙的安全机制,在安全性和可移植性方面有所加强,这使得它很适合在分布式计算环境中使用。虽然 Oak 语言在设计上是非常成功的,但在问世之初并没有引起人们太多的重视,产品发展定位的错误使得这项卓越的技术一度处于夭折的边缘,直到一个关键性人物的介入才改变了它的命运,他就是 Sun 公司创始人之一的 Bill Joy。

Joy 曾在加州大学伯克利分校参与 Unix 的开发,他深知网络对 Unix 的推广与辉煌所起的作用。在他的大力支持和推动下,Sun 将 Oak 的发展定位于互联网,并在 WWW 上让用户免费使用,这使得 Oak 得以迅速推广。在明确了发展方向后,Sun 公司组织力量在 Oak

语言的基础上不断完善，最终推出了现在的 Java 语言。和 Oak 一样，Java 语言最初也是在 Internet 上免费发布的，这使它也能迅速地占领相当的市场份额，其后续的发展也因此变得十分的顺利。

1995 年，Sun 公司完成了完整的 Java 技术规范，并立即得到包括 Netscape 在内的各 WWW 厂商的广泛支持。同年 6 月，Sun、Silicon Graphics 和 Macromedia 三家公司宣布将联合指定一套新的基于 Java 的开发式多媒体格式和 API。同年秋季，Netscape 获准在其浏览器 2.0 的 32 位版本中支持 Java 语言，这也使 Netscape 成为 Java 技术的第一个商业客户；随后，Java 得到了其他各大计算机公司的支持，并入选了《商业周刊》1995 年度最佳新产品。1996 年，Sun 公司专门成立 Javasoft 分公司来发展 Java 技术。短短的几年里，Java 凭借着它自身无以伦比的优越性，在全球信息网中得到了广泛的应用。JavaOne 是全世界 Java 开发者的盛会，从 1996 年开始在美国旧金山举行，而 2002 年的参加人数是 2 万人，全世界现有 Java 程序员 300 万人，Java 语言已成为最为流行的网络编程语言，但对于 Java 的发展来说还只是一个开始。

1.1.2 Java 的应用前景

虽然 Java 的问世具有一定的偶然性，但它所展现的强大的生命力和非凡的优越性决定了它必然成为软件产业又一次变革的推动力，它为软件业向更高层次的发展扫清了障碍，IBM 和 Microsoft 对 Java 的支持充分地说明了这一点。

Java 语言是专为网络应用开发的，因此特别适合在 WWW 上应用。使用 Java 语言编写的代码经过 Java 解释器的编译后就具有了平台无关性，可以在使用任何操作系统的计算机上运行，这样就从根本上消除了旧软件体系的弊端，使应用软件变得更加简洁、实用。将 Java 解释器放在网络计算机（以与 Internet 连接等为主要目的，应用程序可从网络上下载的计算机）中，客户端就不必放置复杂的程序，一切应用从服务器上索取。而且，当软件和硬件平台出现变动时（如软件的升级），用户都无须改动应用程序，从而大大简化了客户端的维护和备份，节约了大量的服务费用。

可以这样说，Internet 为跨平台的沟通铺设了基本的体系结构，而 Java 和网络计算机的结合，则给了用户一个“以不变（应用软件）应万变（平台）”的具体方案。这三者的结合共同架设了通往未来网络时代的桥梁。

Java 语言的应用前景十分广阔，主要包括以下几个方面：

- 所有面向对象的应用开发，包括面向对象的事件描述、处理和综合；
- 计算过程的可视化，可操作化软件的开发；
- 动态效果的设计，包括图形图像的处理；
- 网络交互流程的设计和实现；
- Web 页面的动态设计及客户端和服务端交互操作的实现；
- 各类数据库的连接、查询等；
- 其他应用程序的开发。

正如 Web 的发明人 Tim Berners 所说：“不要只把 Java 看成一种新的编程工具，而应把 Java 看作一场革命。”只有为世界带来全新景象的技术，才能称得上是一场技术革命，Java 就做到了这一点。Java 语言给未来的软件业带来了新的思路，它改变了人们的生活方式和

科学研究的环境，它将成为未来网络世界的“通用语言”。

1.2 Java 的特性

1.2.1 Java 语言的特点

Java 语言是一种完全面向对象的程序设计语言。它可以用于编写独立程序，但其最主要的功能是开发能在 WWW 中执行的程序，这在很大程度上得益于 Java 的跨平台特性。Java 语言的特点主要有以下几个方面：简单、面向对象、分布式、解析执行、健壮、安全、可移植、高性能、多线性及动态性。

1. 简单性

Java 语言是一种非常实用的编程语言，它通过一些最基本的方法来完成指定的任务。只要了解一些 Java 的基本概念和语法结构，就可以用它来编写出适合各种情况的应用程序。

Java 不使用头文件，摒弃了 C 语言中容易出错的预处理程序，不再支持运算符重载、多重继承等模糊的概念，取消了 C 语言中的 Struct 和 Union 等结构，不支持可能破坏整个逻辑结构的 goto 语句，并通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。

Java 语言从 C++ 而来，使得原来的 C++ 程序员很容易转向 Java 编程；同时 Java 语言中取消了 C++ 中很多复杂但不实用的部分，所以 Java 比 C++ 更简单、实用，更容易学习，程序的可读性更强。

此外，Java 对空间的需求也不大，很适合在小型机上运行，其基本解释器及类的支持只有 400KB 左右，对资源的需求很小。

2. 纯面向对象

Java 语言是一种“完全”面向对象的程序设计语言，它的设计集中在对象及其接口，同时也提供了简单的类机制以及动态的接口模型。在 Java 语言中，所有的程序和数据都存在对象中，这就是说对象是数据以及对数据进行操作的程序的集合。这些数据就是对象的变量，而这些程序就是对象的方法 (Method)，都是专门为对象而编写的。数据和方法可以描述对象的本质。

在 Java 语言中，对象是由类来构建的，类的使用非常普遍，编写任何一个有意义的 Java 应用程序都是和类分不开的。因此，必须熟悉类的基本结构和使用方法。类中的变量和方法可以封装在对象当中，实现了模块化和相关信息的隐藏。通过定义类还可提供一种对象的原型，在父类中定义大致的结构和功能，再通过类的继承机制，使子类可以使用父类所有的方法，从而实现了代码的充分利用。

3. 结构中立性

Java 程序采用了完全统一的语言文本，如 Java 的基本数据类型不会随机器类型的变化而改变（整型数据总是 32 位，不会随机器的不同而不同）。在解释执行的时候，Java 解释器生成与计算机体系结构无关的字节码指令，只要安装了 Java 运行系统，Java 程序就可在

任意的平台上运行。同时 Java 语言还定义了一个用于访问底层 OS 功能的扩展类库, 以使 Java 的应用程序能不依赖具体系统。Java 语言的结构中立性是保证 Java 的可移植性的基础。

4. 可移植性

Java 程序是在 Java 虚拟机中运行的, 任何一台机器, 任何一种平台, 只要装配了 Java 解释器, 就可以运行 Java 程序。同时, Java 的类库中提供了不同平台的接口, 可以方便地移植类库。另外, Java 编译器是由 Java 语言实现的, Java 运行时系统由标准 C 语言实现, 因此 Java 系统本身也具有可移植性。

Java 的可移植性有着十分深远的意义, 它不仅使软件开放者实现了“一次开发, 处处运行”的方便, 还迎合了网络分布式计算的思想。Java 语言出现后, 整个 Internet 就成为了一个巨大的操作系统, 而 Java 就是这个系统的语言。Java 语言编写的任何程序都可以通过 Internet 被其他得到许可的用户下载并使用, 这样不仅大大节省了开发时间和存放空间, 也省去了管理程序的麻烦。

5. 健壮性

Java 在编译和运行程序时, 要对可能出现的隐藏问题进行检查, 以消除错误的产生, 把错误产生的机率降到最小。通过面向对象的异常处理机制, 使得 Java 在编译程序时, 提示可能出现但未被处理的异常情况, 以防止系统的崩溃。

此外, Java 在编译时还可捕获类型声明中的许多常见错误, 防止动态运行时出现不匹配的问题。另外, 它提供的垃圾自动收集机制可以避免人为管理内存时可能出现的错误, 提高了系统的安全性。

6. 安全性

Java 不支持指针, 一切对内存的访问都必须通过对象的实例变量来实现, 这样就防止了程序员使用所谓的“特洛伊”木马等欺骗手段访问对象的私有成员, 同时也避免了指针操作中容易产生的错误问题。使得安全性大大提高。

Java 语言除了语言本身的安全性外, 运行环境还提供了严格的安全保障机制, 如字节码校验器、类装载器、文件访问控制和运行时内存布局等, 这些都是为适应分布式网络计算所设计的, 可以保障良好的网络安全性。

7. 分布性

Java 是一种面向网络的语言, 很适合在 Internet 和分布式环境下运行, 它支持 WWW 客户机/服务器模式, 通过 Java 提供的类库可以方便地实现 TCP/IP 协议, 用户可以通过 URL 地址访问网络资源。

8. 多线程

Java 内部具有对多任务的支持。Java 的多线程机制使得应用程序能够并行执行, 而且同步机制也保证了对共享数据的正确操作。通过使用多线程, 程序设计者可以分别使用不



同的线程来完成特定的行为，而不需要采用全局的事件循环机制，这样就很容易实现网络的实时交互行为。

9. 类库丰富

Java 的类库资源非常丰富，类库不仅可以向用户提供许多实用而且功能强大的成熟工具，省去了独立开发的麻烦，还可以在已有标准类库的基础上开发用户自己的类库资源，最大限度地提高编程效率。

10. 静态和动态

Java 的静态体现在 Java 程序中的对象（数值、字符和数组）类型必须定义才能使用。动态性则是指 Java 的设计特点使它一直处于一个不断发展的环境中。例如，可以在类库中自由加入新的方法和实例变量而不会影响用户程序的开发。

1.2.2 Java 和 C++的比较

Java 语言借鉴了 C++语言的很多优点，例如变量声明、参数传递、操作符和流程控制等等。C++语言虽然适应了软件工程界的面向对象的潮流，但它还不能真正满足面向对象的编程。换句话说，C++并不是一种完全面向对象的语言，其中还保留了很多非面向对象的特征，而 Java 语言则实现了纯面向对象的特点。

Java 语言的设计摒弃了 C 和 C++中很多不很合理或不绝对需要的内容，这些内容主要有以下几个部分。

1. 指针

指针或内存地址是 C++中最有效也是最有危害的特性，不正确的指针操作会带来很多意想不到的错误。同时通过指针对某个内存地址进行显式类型转换后，可以访问一个 C++中的私有成员，造成系统的崩溃。因此，在 Java 中不能直接使用指针。

2. 内存分配

C++中的内存分配和指针操作有同样的危险。其内存分配是通过 `malloc()` 和 `free()` 库函数以及 `new` 和 `delete` 两个运算符来实现的，程序员可以根据自己的要求进行内存的分配和释放，但再次释放已被释放的内存或未被分配的内存，可能会造成系统的崩溃。而 Java 中没有 `malloc()` 和 `free()` 函数，由于每个负责的数据结构都是对象，它可通过 `new` 运算符在内存堆上分配空间，一旦不再访问对象，占据的内存空间就会被自动回收，因此不需要使用 `free` 和 `delete`，这种机制被称为“垃圾回收”。

3. 全局变量

C++中的全局变量作为程序的状态信号没有很好地封装，虽然全局变量的作用十分明显，但也给系统的运行带来了很多的不稳定因素。而 Java 程序中的所有变量都必须在类内定义，但可通过在类中定义公用、静态的变量来实现和全局变量同样的功能，这样就很好地解决了不完全封装的缺陷。

4. 数据类型

不同的 C++编译器根据不同机器的时间配置分配给数据类型以不同的字长（16 位、32 位或 64 位），例如 `int` 类型的数据在 IBM PC 中为 16 位，在 VAX-11 中为 32 位。而在 Java 中，所给定的基本数据类型确定一个合理的字长并保持不变，如 `int` 类型的变量总是占 32 位字长。Java 解释器的这种严格的数据类型虽然很难实现对代码的优化，但这却是保证跨平台实现的惟一途径。

5. 头文件

C 程序中可使用头文件来声明类、全局变量、库函数等，但在系统中维护这些头文件是很困难的。Java 不支持头文件的使用，类成员的类型和访问权限都封装在类中，运行时对访问进行控制，防止访问私有变量。

6. 预处理

C 和 C++程序中用宏定义来实现的代码虽然给开发人员带来的方便，但也破坏了代码的可读性。在 Java 中，不支持宏定义，但可使用关键字 `final` 声明一个特殊的变量，以实现宏定义中最常用的常量定义。

7. 多重继承

C++中具有多重继承的特性，这在一定程度上增加了类间的关联性，但同时也扰乱了整个的类层次结构，降低了代码的可读性。Java 中不再支持多重继承，但可以使用接口（`Interface`）来实现相似的功能。

8. 结构和联合

在 C 语言中可以定义 `Struct`（结构）和 `Union`（联合），但由于其成员均为公有，在使用中可能会带来一系列的安全问题。在 Java 中只有类才是全局的，而 `Struct` 的定义显然与之不符，因此 Java 中取消了 `Struct` 和 `Union`。

9. goto 语句

在 C++引入异常处理之前，`goto` 语句常被用来在出现异常情况时跳出循环，但这种随意的跳转也破坏了整个程序的逻辑结构。在 Java 语言中，可通过异常处理语句来处理出现的异常情况，`goto` 语句就再没有存在的必要了。取消这种随意跳转的语句有利于优化代码并保持系统的稳定性和安全性。

10. 类型转换

在 C、C++中，可以使用指针进行任意的类型转换，但这样会引发很多安全问题。在 Java 中，运行时系统会进行类型相容性检查，以防止不安全的类型转换。

1.3 Java 的设计目标和实现机制

1.3.1 Java 的设计目标

Java 语言的设计目标主要有几个方面:

- 创建一种完全面向对象的程序设计语言;
- 提供一个程序运行的解释环境,使程序代码独立于平台;
- 吸收 C 和 C++ 的优点,使程序员容易掌握;
- 去掉 C 和 C++ 中影响封装、安全性和健壮性的部分,如指针,内存申请和释放;
- 提供多线程机制;
- 提供动态下载程序代码的机制;
- 提供代码检验机制以保证安全性。

从 Java 的设计目标中可以看出,一种新的语言的产生可以借鉴已有的成熟语言,但更为重要的是要有自己独特的设计定位和先进的设计理念,Java 语言在这方面堪称典范,因为在 Java 出现之前还不存在其他的跨平台语言。

1.3.2 Java 的实现机制

Java 之所以能具备这么多的优点是和 Java 语言的实现机制紧密相关的,下面就介绍 Java 语言的三个实现机制。

1. Java 虚拟机 (Java Virtual Machine)

虚拟机是在一台计算机上由软件模拟的假想的计算机,它在机器和编译程序之间加入了一层抽象、虚拟的中间件,虚拟机在任何平台上都提供给编译程序同样的接口。编译程序只面对虚拟机,而由虚拟机来负责和计算机的连接,这样就实现了 Java 代码的平台无关性,而 Java 虚拟机就是平台无关的关键。

Java 编译器针对 Java 虚拟机产生 class 文件, class 文件也称为字节码文件,其中存储着 Java 虚拟机可识别的代码,而由 Java 解释器负责将 Java 虚拟机的代码解释成特定系统的机器码,并在特定的平台上执行。可以说,每一种 Java 解释器都是 Java 虚拟机的一个实例,帮助它完成同硬件设备的连接。

可以说,Java 虚拟机是 Java 语言的基础,它并不能识别 Java 语言,它只能识别编译后的 Java 字节码。Java 虚拟机是一个抽象的计算机,但它和实际的计算机一样,可以执行指令,管理数据、内存等。它的逻辑组件包括:指令集、寄存器组、32 位栈、无用单元收集堆和方法区域,这几个部分不依赖于任何实现技术和机器平台。

2. 无用内存收集机制 (Garbage collection)

在 Java 和 C++ 语言的比较中已经介绍过,Java 语言不支持指针操作和内存管理功能,这样做最大的好处就是可以提高代码的安全性和系统运行的稳定性。为提高内存的使用效率,Java 中引入了独特的内存管理机制,即一个自动的“垃圾回收”功能。Java 可自动扫描内存资源,并自动释放不再使用的内存块,简化了内存管理。

3. 代码安全性检测 (Code Security)

Java 语言为了提高代码运行的安全性, 其运行环境提供了多种安全性保障机制, 其中很重要的一个就是代码安全性检测, 它的任务是在代码执行前进行自动检查, 可以及时发现程序的错误和病毒, 起到“防患于未然”的作用。另外, 文件访问控制机制可以很好地限定程序的访问范围, 防止出现侵占系统资源或修改系统文件等行为。由此可以看出, Java 语言比以往任何一种程序设计语言更重视安全性和稳定性, 是一种真正“安全”的语言。

1.4 本章小结

本章主要介绍了 Java 语言的一些基本概况, 其中包括 Java 的起源和发展, Java 的特点和优势, 以及 Java 的实现机制等。Java 语言的应用范围很广, 如图形处理、网络应用、数据库开发等等, 所有这些内容都将在后续的章节中分别向大家介绍。

1.5 习 题

1. 比较 C 语言和 Java 语言的异同。
2. 简述 Java 语言的基本特点 and 设计目标。

第 2 章 JBuilder 8.0 开发环境

本章要点

- 掌握 JBuilder 8.0 开发环境中各开发工具的用途
- 创建第一个 Java 应用程序

JBuilder 8.0 是 JBuilder 系列的最新版本,本章将详细介绍 JBuilder 8.0 的安装方法、界面组成、基本使用方法以及 JBuilder 8.0 的特点。通过本章的学习,读者可以对 JBuilder 8.0 的开发界面有一个全面的了解。

2.1 JBuilder 8.0 的新特点

在众多的 Java 语言开发工具中, JBuilder 是一款非常优秀的 Java 可视化开发环境,它是由 Borland 公司推出的一种基于组件技术的可视化 Java 开发工具,并适用于各个层次上的 Java 程序开发,其中包括开发 JavaBeans 组件、编写 XML 文档、开发数据库应用程序以及实现网络链接和多媒体技术等等。

JBuilder 8.0 为 Java 程序开发者提供了灵活的开发方式。它支持 JavaSoft 规定的所有技术标准,如 JavaBean、EJB、J2EE、CORBA、JDBC 和 ODBC 等,用户可使用 JBuilder 8.0 创建各种用途的 Java 应用程序。此外, JBuilder 8.0 还提供了一个集成开发环境,可在同一个界面中实现工程创建、属性设置、程序编译以及代码调试等功能。

JBuilder 8.0 的组件和类库资源丰富,并为不同类型的应用程序提供了开发模板,开发人员使用较少的代码就可编写出功能复杂的程序。这样,不仅可以帮助开发人员快速确立程序的整体框架,最大限度地避免重复劳动;还使开发人员可以集中精力完成程序关键部分代码的组织和编写。

JBuilder 8.0 的使用和维护也很方便。整个开发界面统一、明晰,功能向导丰富、实用,同时 JBuilder 8.0 还提供了丰富的帮助文件,可以帮助解答开发过程中的疑难问题。

同以往 JBuilder 的各版本相比, JBuilder 8.0 的功能更加完善,下面就分几个方面介绍 JBuilder 8.0 的一些新特点。

1. JDK 1.4.1

JBuilder 8.0 使用 JDK 1.4.1,该版本能为 Java 应用程序的开发提供更为有力的支持。例如,Java 客户端应用程序的执行更为快捷,支持鼠标移动或拖动等一系列操作,改善焦点管理,在 Linux 环境下的用户界面操作更为合理等。

2. 工程管理 (Project Management)

JBuilder 8.0 的工程管理中加入了以下几项新功能:

(1) Drag-and-Drop (拖放) 操作: 在 JBuilder 8.0 开发界面的工程窗口 (如图 2-11 所示) 中, 可实现文件的拖放操作, 即可将桌面中 JBuilder 8.0 界面外的文件拖入当前工程的工程窗口中, 并成为该工程的组成部分之一。需要注意的是, 实现此操作的文件必须为 JBuilder 可识别的文件类型之一。此外, 可使用工程窗口中的 Add Files/Packages 按钮, 将指定的文件或包加入当前的工程中。

(2) 创建空文件: 使用开发界面中 File 菜单的 New File 命令, 打开一个创建新文件的对话框, 并在当前工程中创建指定文件名称和类型的空文件。

(3) 工程组: JBuilder 8.0 中引入了工程组 (Project Group) 的概念。工程组是若干相关工程的集合, 各工程间相互协调, 完成较复杂的功能, 例如实现 Server/Client 操作等。此外, 使用相同源文件的工程也可视为同一分组, 但各工程的设置不同, 如应用的服务器类型或 JDK 版本等。

(4) Required Libraries: 当某工程的实现需要依赖其他工程的支持时, 可将该工程所需要的工程加入其 Required Libraries (所需程序库) 列表中, 相关设置可在工程属性对话框中完成。

(5) 调整工程内容: 工程创建后, 可在工程窗口中进一步调整工程的内容, 如优化工程引入文件包, 修改文件包名称、修改类或方法名称, 修改方法参数等。

3. 调试器 (Debugger)

JBuilder 8.0 的调试器加入了以下三项新功能:

(1) Smart Swap: 在程序的调试过程中, 修改程序代码并重新编译时, 可使用开发界面中 Run 菜单的 Smart Swap 命令, 该命令专门用于修改、编译并更新修改后的文件。执行该命令后, 开发人员可检查代码的变化对执行的影响, 此项功能基于 JDK 1.4 的 Hot Swap 技术。

(2) Set Execution Point: 在调试过程中, 可使用开发界面中 Run 菜单的 Set Execution Point 命令, 为当前分步执行的线程设置执行点, 并观察该点处的线程执行情况。

(3) Smart Source: 在调试过程中, 可使用开发界面中 Run 菜单的 Smart Source 命令改变代码的显示, 可选择查看 Java 源代码或调试过程中的非 Java 源代码, 该项功能基于 JDK 1.4 的 JSR-45 技术。

4. Unit Testing

为实现对非本地运行程序的测试, JBuilder 提供了 Unit Testing 技术。为增强对 Unit Testing 技术的支持, JBuilder 8.0 加入了以下新功能:

(1) Cactus: 为测试服务器端的 Java 应用程序执行情况, JBuilder 8.0 提供了 Cactus 测试工具。Cactus 扩展了 Unit Testing 的应用, 可通过 Wizard 菜单的 Cactus Setup 命令打开构建测试环境的向导, 并为当前工程提供 Cactus 支持。

(2) EJB Test Client: EJB 是 Enterprise JavaBean 的简写, 代表企业级的 JavaBean 组件。为测试其执行情况, JBuilder 8.0 中提供了 EJB Test Client Wizard, 该向导可生成两种新的测试客户端程序类型: Junit test client 和 Cactus Junit test client, 它们可帮助实现对 EJB 的测试。