

为您开启步入 Java 殿堂的快捷之门

- ▶ 由国内 Java 专家精心编著，融会作者多年的实际开发经验
- ▶ 本书内容全面、系统，知识讲解深入浅出，堪称 JBuilder 的权威教材
- ▶ 每个知识点都辅以实例进行讲解，内容实用、步骤详细、可操作性强

读者对象：

本书既适合作为大中专院校及培训班学生理想的 JBuilder 教材，也是初中级读者自学 JBuilder 8 的必备参考书

刘 斌 / 编著

深入浅出

Borland

JBuilder 8

程序设计与开发

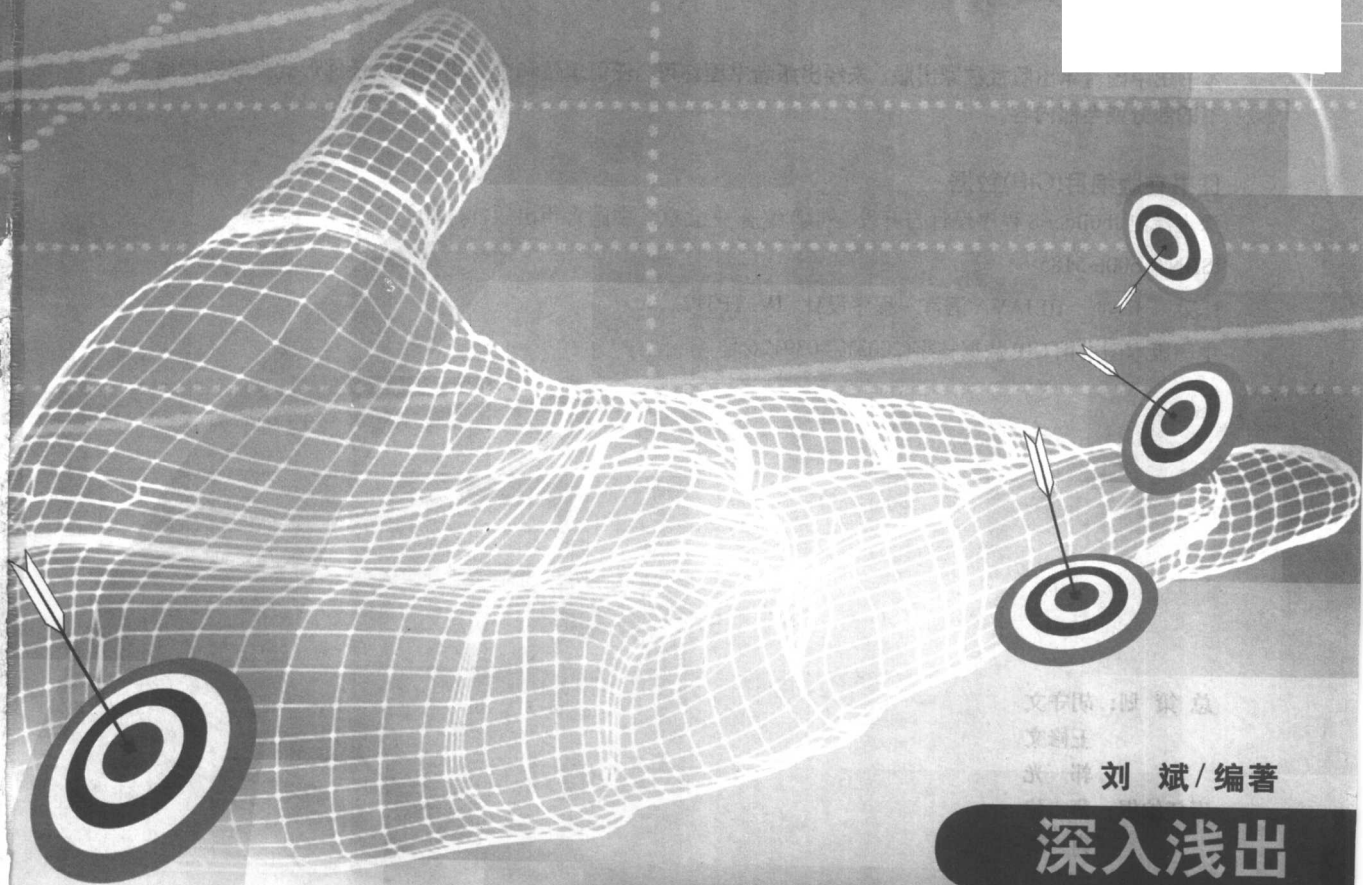


随书附赠光盘，内含书中所讲范例的完整源程序，以及相关 Java 开发工具



中国青年出版社

<http://www.21books.com> <http://www.cgchina.com>



文字版：刘斌
刘斌 / 编著

深入浅出

Borland **JBuilder 8**

程序设计与开发

 **中国青年出版社**
CHINA YOUTH PRESS
<http://www.21books.com> <http://www.cgchina.com>

(京)新登字083号

本书由中国青年出版社独家出版。未经出版者书面许可,任何单位和个人不得以任何形式复制或传播本书的部分或全部内容。

图书在版编目(CIP)数据

深入浅出Jbuilder 8 程序设计与开发 / 刘斌 编著 - 北京: 中国青年出版社, 2003

ISBN 7-5006-5185-6

I. 深... II. 刘... III. JAVA 语言 - 程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 039126 号

总策划: 胡守文

王修文

郭光

责任编辑: 曹建

肖辉

刘利平

责任校对: 王志红

书名: 深入浅出Jbuilder 8 程序设计与开发

编著: 刘斌

出版发行: 中国青年出版社

地址: 北京市东四十二条21号 邮政编码: 100708

电话: (010) 84015588 传真: (010) 64053266

印刷: 山东高唐印刷有限责任公司

开本: 787 × 1092 1/16 印张: 21.75

版次: 2003年7月北京第1版

印次: 2003年7月第1次印刷

书号: ISBN 7-5006-5185-6/TP · 331

定价: 36.00元 (1CD)

前 言

Borland 公司在上一个版本 JBuilder 7 推出仅隔半年，就又将崭新的 JBuilder 8 展现到用户面前。如此频繁地推出新版本，大大缩短了每个版本的产品的销售周期，让用户觉得有应接不暇的感觉。用户甚至会责怪 Borland 公司：“半年前，我用 JBuilder 7 开始学习 Java，现在，我刚对 Java 领域有所认识，我的开发工具居然已经过时了。”但这并不怪 Borland 公司，因为这是 Java 的缘故，选择了 Java 就意味着选择了永不停息地学习。

Java 的出现成为计算机信息交换的一个里程碑，为迅速发展的计算机信息世界增添了新的活力。面对一个如此充满吸引力，而且又如此前景光明的领域，谁又能不兴奋雀跃，迫不及待地投身其中。本书将为您开启一扇步入 Java 殿堂的快捷之门。

目前 JBuilder 是应用最广、用户最多的 Java 开发工具，众多的软件公司都在使用 JBuilder 进行项目开发，对 Java 程序员来讲仅仅掌握 Java 语言本身的语法是远远不够的，根本无法开发大型的应用程序，所以 JBuilder 是其必须熟练使用的工具。本书就是针对那些刚刚接触 Java 语言或已经掌握了 Java 语言基本语法的用户，从实际应用的角度出发，全面而系统地介绍了利用 JBuilder 8 进行 Java 程序设计的知识，内容涵盖当前 JBuilder 8 最主流的应用。在具体的讲解过程中，注重结合实例，帮助读者充分掌握 JBuilder 8 的用法以及 Java 的实际编程应用，使读者能够迅速提高 Java 语言使用技术，进而将 Java 编程应用到实际工作之中。

全书共分 16 章。第 1 章对 Java 和 JBuilder 8 进行总体概述。第 2, 3 章讲解了 Java 语言的构成及相关实例。第 4, 5, 6 章分别讲解了利用 JBuilder 8 进行界面设计、文本处理和图形处理。第 7 章讲述多线程的实现过程。第 8 章讲解了 JavaBeans 的创建与设计。第 9, 10, 11 章详细地讲解了 JBuilder 8 在数据库开发方面的具体应用。第 12 章详述了 Applet 的应用和 JBuilder 8 创建 Applet 的方法。第 13 章则主要讲解的是网络通信知识，为后面的网络应用程序的开发做准备。第 14, 15, 16 章分别讲解了 JBuilder 8 在 Servlet, JSP 和 EJB 方面的开发应用。基本上涵盖了最常用和最实用的技术，使读者对 JBuilder 8 有一个全面而系统的认识。

本书第 1, 2 章由刘斌、刘海波编写，第 3, 4 章由齐伟、张九凌编写，第 5, 6, 7 章由刘阳、马涛、侯鹏飞、谭敏海编写，第 8, 9 章由贾德坤、张胜洪、白琳编写，第 10, 11 章由于江华、李永松、王博编写，第 12, 13 章由钟建华、张庆东、洪海编写，第 14, 15 章由罗权、沙华力、王华明编写，第 16 章由郭志煌、肖回春编写，附录部分由姚克、陶芳玲整理编写。由于作者水平有限，虽竭尽全力，但错误和不足之处在所难免，欢迎读者朋友将意见和建议反馈给我们，以便修改和共同提高。

作 者

2003 年 5 月

目 录

第1章 初识 Jbuilder 8

1.1 Java 简介	2
1.1.1 Java 语言的特点	2
1.1.2 Java 语言与 Internet	5
1.2 JBuilder 8 简介	6
1.3 JBuilder 8 的安装	11
1.4 JBuilder 8 的运行环境	13
1.4.1 应用浏览器	13
1.4.2 使用帮助	21

第2章 Java 语言的构成

2.1 程序结构	26
2.2 语法问题	26
2.2.1 注释	26
2.2.2 标识符	26
2.2.3 自变量	27
2.2.4 运算符及特殊分隔符	28
2.3 数据类型	28
2.3.1 数值类型	28
2.3.2 布尔类型	30
2.3.3 数组	30
2.3.4 类	31
2.4 运算符和表达式	33
2.4.1 算术运算符	33
2.4.2 关系运算符	34
2.4.3 布尔逻辑运算符	35
2.4.4 位运算符	35
2.4.5 条件运算符	35
2.4.6 赋值运算符	36
2.4.7 表达式	36
2.5 控制语句	36
2.5.1 条件语句 if...else	37
2.5.2 多分支语句 switch	37
2.5.3 返回语句 return	38
2.5.4 while 语句	38

2.5.5 do...while 语句	38
2.5.6 for 语句	39
2.5.7 continue 语句	39

第3章 从 Hello World 开始

3.1 创建一个应用程序	42
3.1.1 创建工程	42
3.1.2 创建和添加应用程序	44
3.1.3 创建用户界面	46
3.1.4 添加事件	48
3.2 部署应用程序	49
3.2.1 什么是部署	49
3.2.2 创建 JAR 文件	50

第4章 用户界面设计

4.1 在程序中使用组件	54
4.1.1 Swing 中的组件和容器	54
4.1.2 组件选项卡 (Component palette)	56
4.1.3 使用组件	57
4.1.4 组件应用范例	61
4.2 布局管理器	66
4.2.1 FlowLayout 布局管理器	66
4.2.2 VerticalFlowLayout 布局管理器	67
4.2.3 BorderLayout 布局管理器	68
4.2.4 CardLayout 布局管理器	68
4.2.5 GridLayout 布局管理器	70
4.2.6 GridBagLayout 布局管理器	70
4.3 应用程序的外观	71
4.3.1 在设计时改变外观	72
4.3.2 在代码中改变程序的外观	72

第5章 文本处理程序

5.1 菜单设计器	76
5.2 流的应用	78
5.2.1 流的基本概念	78
5.2.2 常用的字节流	78
5.2.3 常用的字符流	79

5.3 文件	80	7.1.1 什么是线程	122
5.4 Swing 中的事件处理	80	7.1.2 Java 中线程的实现	122
5.5 剪贴板的应用	82	7.1.3 线程的优先级	123
5.6 创建一个文本编辑器	83	7.1.4 线程的调度	123
5.6.1 使用向导建立一个应用程序	83	7.1.5 线程间的同步	123
5.6.2 设计用户界面	84	7.2 线程的生命周期	123
5.6.3 为框架添加菜单和菜单项	84	7.3 线程的实现方法	124
5.6.4 为 File 菜单添加事件处理代码	85	7.4 多线程范例	125
5.6.5 为 Edit 菜单添加事件处理代码	87	第 8 章 创建 JavaBeans	
5.6.6 实现 openFile, saveFile, saveAsFile 等方法	91	8.1 JavaBeans 概述	134
5.6.7 实现对文档内容更改的监听	93	8.1.1 概念及特点	134
5.6.8 实现 Undo/Redo 操作	94	8.1.2 使用 BeansExpress	138
5.6.9 菜单的失效处理	95	8.2 创建 JavaBeans	145
第 6 章 图形处理		8.2.1 创建 Bean 类	145
6.1 抽象类	98	8.2.2 增加属性	146
6.2 矢量 Vector 类	98	8.2.3 为 Bean 添加事件	149
6.3 事件监听机制	100	8.2.4 设计属性编辑器	151
6.3.1 鼠标事件	100	8.2.5 创建 BeanInfo 类	151
6.3.2 MouseListener 和 MouseAdapter	101	8.2.6 校验 JavaBeans 的合法性	152
6.3.3 MouseMotionListener 和 MouseMotion Adapter	101	8.2.7 发布组件	152
6.4 图形绘制实例程序	102	8.3 测试 JavaBean	153
6.4.1 创建应用程序框架	103	第 9 章 数据库的创建和连接	
6.4.2 设计界面	104	9.1 理解 JBuilder 8 的数据库应用	158
6.4.3 实现抽象类 Figure 和它的子类	104	9.1.1 JDBC 的基本概念	159
6.4.4 DrawFigurePanel 类的实现	106	9.1.2 JDBC 的组成部分	160
6.4.5 添加事件处理代码	108	9.1.3 4 种 JDBC 驱动程序	161
6.4.6 在 DrawFigurePanel 类中覆盖 Paint Component()方法	109	9.1.4 为 JBuilder 8 添加 JDBC 驱动	162
6.5 MVC	110	9.2 JBuilder 的 DataExpress 结构	164
6.6 JTree 树	111	9.3 数据库浏览器程序	165
6.7 图像浏览程序	112	9.3.1 设置 ODBC 数据源	165
6.7.1 程序界面设计	113	9.3.2 设置 DataExpress 组件	166
6.7.2 组件的初始化	115	9.3.3 添加并设置数据感知组件	170
6.7.3 下拉列表组件的事件处理	117	第 10 章 数据库的访问	
6.7.4 图像的显示	119	10.1 关系数据库基础	174
第 7 章 Java 中的多线程		10.1.1 实体—关系模型	174
7.1 基本概念	122	10.1.2 表	174
		10.1.3 视图	175
		10.1.4 索引	175
		10.1.5 同义词	176

10.1.6	系统表	176	14.3	Servlet 的生命周期	256
10.1.7	Database Pilot 的应用	176	14.4	JavaMail API 介绍	257
10.1.8	创建数据库结构浏览程序	178	14.5	Servlet 邮件收发范例	259
10.2	为数据库应用程序添加功能	191	14.5.1	创建 Servlet	260
10.2.1	从下拉列表添加数据	192	14.5.2	LoginServlet 的代码实现	263
10.2.2	使用计算列	193	14.5.3	创建 showMail	267
第 11 章 数据库管理任务			14.5.4	创建 sendMail	269
11.1	数据的备份和恢复	196	14.5.5	运行范例程序	270
11.2	使用文本文件存储数据	197	第 15 章 JSP 的应用		
11.3	数据感知组件	198	15.1	JSP 及其特点	276
11.4	用 JBuilder 8 实现数据库管理	199	15.2	JSP 页面的执行过程	278
11.4.1	创建应用程序界面	200	15.3	JSP 基本的语法元素	279
11.4.2	数据备份的实现	203	15.3.1	指令元素	280
11.4.3	数据恢复的实现	205	15.3.2	脚本元素	280
第 12 章 Applet 的应用			15.3.3	动作元素	281
12.1	Applet 的基本概念	212	15.4	JSP 制作注册程序	282
12.2	Applet 的生命周期	213	15.4.1	创建 JSP 文件	283
12.3	用 JBuilder 8 创建 Applet	213	15.4.2	设计 DbBean 类	284
12.3.1	使用 Applet 向导	214	15.4.3	设计登录页面	286
12.3.2	初始化 Applet	216	15.4.4	设计注册页面	288
12.3.3	Applet 的启动和停止	217	15.4.5	设计信息更新页面	290
12.3.4	时间的显示	219	15.4.6	运行范例程序	291
第 13 章 网络通信			第 16 章 EJB 的应用		
13.1	TCP/IP 协议的基本概念	224	16.1	EJB 简介	296
13.2	TCP 套接字	225	16.1.1	为什么要用 EJB	296
13.2.1	IP 地址	225	16.1.2	EJB 中的角色	296
13.2.2	TCP 端口	226	16.1.3	EJB 的体系结构	298
13.3	网络通信程序范例	226	16.1.4	EJB 的类型	299
13.3.1	创建客户端程序	227	16.2	使用 EJB 设计器开发一个会话 bean	299
13.3.2	创建服务器端程序	234	16.2.1	指定目标应用服务器	300
13.3.3	运行程序	238	16.2.2	创建 EJB 模型	302
13.4	UDP 协议	240	16.2.3	构建 bean	303
13.5	Java 中的 UDP 实现	242	16.2.4	编译 bean	310
13.5.1	创建服务器端程序	243	16.3	用客户端应用程序测试 bean	311
13.5.2	创建客户端程序	248	16.3.1	创建用于测试的客户端应用程序	311
13.5.3	运行程序	250	16.3.2	为客户端添加代码	311
第 14 章 Servlet 的应用			16.3.3	编译并运行应用程序	315
14.1	Servlet 简介	254	附录 A Java 2 中的关键字		
14.2	Servlet 的实现结构	255	附录 B 常用的 Java 类		



第 1 章

初识 JBuilder 8

- 1.1 Java 简介
- 1.2 JBuilder 8 简介
- 1.3 JBuilder 8 的安装
- 1.4 JBuilder 8 的运行环境

JBuilder 是 Borland 公司开发的一个功能强大的可视化 Java 开发工具。它的主要应用对象是使用 Java 进行程序开发的程序员。它强大的功能和友好的用户界面使 Java 程序员从繁琐的界面设计和重复的代码录入中解脱出来,快速高效地建立各种 Java 应用程序,包括小应用程序、需要与网络数据库连接的应用程序、Web 应用程序以及企业级分布式应用程序等。JBuilder 8 是其最新发布版本。

1.1 Java 简介

Java 是 Sun MicroSystem 公司开发的一代功能强大的面向对象程序设计语言,特别适合于 Internet 应用程序开发,它具有小巧、稳定、与平台无关等特点,是当今风靡世界的网络编程语言。

1.1.1 Java 语言的特点

Java 到底是一种什么样的语言呢?在 Sun 公司的 Java 白皮书中是这样被定义的,Java 是一种简单的、面向对象的、分布式的、健壮的、解释型的、安全的、中性结构的、可移植的、高性能的、多线程的、动态的语言。这个定义概括了 Java 语言的特点,下面将简单地做出解释。

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 3 个方面:

- (1) Java 的风格类似于 C++,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

面向对象程序设计(Object-Oriented Programming)技术是人们创造出来的程序设计的一种新思维。它基于类,将应用程序分解成相互独立的对象,通过集成这些对象来建立应用程序。而且这些对象可以在以后反复地使用或改动,实现了程序设计工作中的高效率。Java 所有的数据类型都有相应的类,整个语言都基于对象。从严格意义上来说,一种面向对象的编程语言至少应遵守以下 4 个基本原则:

- 封装性,即实现了信息隐藏和模块化。将信息与处理信息的能力组合起来并将它们封装于对象之中。
- 多态性,即不同对象对同一种信息,可以按照对象本身的性质进行响应。类的多态性是通过方法的重载、覆盖和界面实现的。
- 继承性,即下层继承上层、子辈继承父辈的特性。这样可以有效地组织程序代码并减少重复编写。
- 动态绑定,即允许在运行时而非编译时按具体数据类型和参数来选用函数。

按照以上的原则来衡量,Java 语言完全是面向对象的,而其他许多程序设计语言都不能被称为面向对象的,例如,因为 C++没有实现动态绑定,Visual BASIC 没有实现封装性而不

能被称为严格意义上的面向对象的编程语言。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 的分布式特点体现在 Java 提供了 URL 对象, 从而使数据分布得以实现。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 对象打开并访问分布于网络上的远程对象, 其访问方式与访问本地文件系统几乎完全相同。Java 支持 WWW 客户 / 服务器计算模式, 利用远程方法唤醒应用程序接口可以调用远程对象的方法。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。我们可以通过 Socket 类建立低级别的网络连接, 并且利用小服务程序 (Servlet) 机制来进行 Java 服务器高效编程。所以, 可以把运算从服务器端分散到客户端, 从而提高了系统的执行效率, 避免了瓶颈制约, 增加了动态可扩充性。

4. 健壮性

由于 Gosling 等设计人员在开发 Java 语言时就把保证程序运行的绝对可靠作为设计目标之一, 因此, Java 致力于检查程序在编译和运行时的错误以及避免潜在错误的发生上。

Java 是一种强类型语言, 即它需要显式的类型声明, 而不是隐式的声明, 并且不支持 C 及 C++ 中的自动类型转换, 这样, 类型检查帮助检查出许多开发早期出现的错误。Java 不支持指针, 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。异常处理保证了程序长时间的可靠运行而不发生致命错误。

5. 解释型

Java 是一种半编译半解释的语言。严格地说, Java 是被解释的。Java 程序的执行分为两个过程: 首先 Java 源程序经过编译器编译生成字节码 (Bytecode), 即 .class 文件, 然后经过 Java 解释器将字节码转换成实际应用的机器和操作系统上的机器码执行。图 1-1 所示为 Java 程序的执行流程图。

虽然 Java 程序的被解释执行影响了其运行速度, 可它的速度仅为同级被编译执行程序的 1/6~1/2, 但是正是因为把编译和解释结合在一起, Java 程序具有了可靠的安全性和稳固性, 并且程序的执行消除了平台之间的差别。如图 1-1 所示。

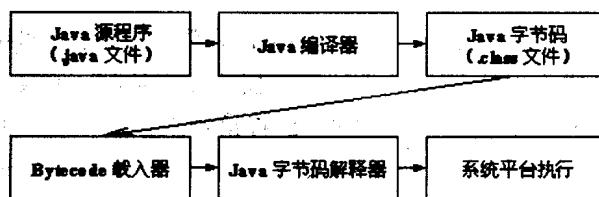


图 1-1 Java 程序的执行流程图

6. 安全性

任何语言的安全性都值得重视。Java 必须保证下载的程序代码在本地运行时不会破坏本

地的机器及文件系统，所以 Java 程序员必须把安全性放在重要的位置。

Java 的安全性可以从以下 3 个方面得到保证：

首先，Java 语言不支持指针，使用户无法直接管理内存，避免了非法内存操作，从而避免了对系统资源的恶意使用。类装载（Class Loader）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，避免了文件或者数据未经授权的修改和删除、正在使用的内存内容的更改、系统内存空间的不断占有、对运行中的程序造成的影响等的问题。

其次，Java 语言中，对于与系统有关的对象和变量，可以加以不同的使用限制，以避免被其他对象修改或继承形式的重新编写。这样就可以避免偷窃系统或个人密码及个人资料。在 Java 程序执行时，字节码载入器要对载入的字节码作一次安全性检查，这个安全性检查即 Bytecode 校验，包括检查程序中对象的使用权限是否合法，检查是否使用了正确的参数类型来调用对象的方法，以及检查系统代码堆栈有否溢出等。

再次，当 Java 用来创建浏览器时，Java 的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。一个支持 Java 的 WWW 浏览器可以阻止 Java 的小应用程序对系统资源的不正当使用，还可以调整对小程序的处理方式。最极端的情况就是完全禁止 Java 小应用程序的执行。当用户的屏幕上出现奇怪的图案或发出奇怪的声音而受到骚扰时，只要关闭浏览器，就可以中止恶意程序的运行。

Java 语言的安全性有目共睹，不能说它是绝对安全的语言，但它肯定是当今流行中最安全的程序设计语言。

7. 中性结构

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。Java 源程序被编译成一种高层次的与机器无关的字节码格式的语言，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。前面已经提到过，Java 程序在任何机器、任何平台上都易于解释，这使 Java 程序可以在任何系统上运行，只要系统支持 Java 虚拟机。也就是说，使用 Java 编写的程序可以在所有的平台上运行；使用了 JFC（Java Foundation Classes）的 Java 程序运行在任何平台都具有相同的外观和行为。

8. 可移植性

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 Windows 3.1 中整数(Integer)为 16 bits，在 Windows 95 中整数为 32 bits，在 DEC Alpha 中整数为 64 bits，在 Intel 486 中为 32 bits。通过定义独立于平台的基本数据类型及其表达方式，Java 数据得以在任何硬件平台上保持一致。所以，Java 在不同的系统中运行不会产生不同的目标代码，也就是说，Java 的可移植性相当好。

9. 高性能

Java 采用半编译、半解释的执行方式，速度虽然比不了编译运行的程序，但已经比其他

解释运行的程序快了不少。Java 还可以在运行时直接将目标代码编译成机器指令。Java 凭借 JIT (Just In Time) 编译器, 只进行一次解释, 将字节码编译成平台专用的机器码并保存下来, 然后再适时调用。Sun 用 JIT 一秒种内可调用 300 000 个过程, 翻译目标代码的速度与 C 和 C++ 的性能没什么区别。

10. 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 实现的多线程技术, 所以比 C 和 C++ 更健壮。多线程带来的好处是更好的交互性能和实时控制性能。这在 Internet 上显得尤为重要。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。在 Java 里, 你可以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。Java 的基本函数库中定义了 Thread 这一基本类, 并内置了一组方法, 只要继承这个类, 就可以利用已有的方法生成一个新的线程, 执行或中止一个线程的任务, 或查看执行状态。Java 采用了监控器 (monitor) 和状态变量 (condition variable) 的技巧来解决数据同步化的问题, 控制各个线程公用的数据不被同时修改, 从而避免数据的丢失。

11. 动态性

Java 的动态特性是其面向对象设计方法的扩展。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下传递给解释器, 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。

1.1.2 Java 语言与 Internet

Java 的成功很大程度上依赖于 Internet, 这从前面 Java 的历史中就可以了解到。但是 Internet 的发展也离不开 Java。可以说, Java 和 Internet 是相辅相成共同促进的。

早期的 Internet 只有简单的命令行界面, 只有专家和科研人员使用。1989 年, 位于日内瓦的欧洲基本粒子实验室 (CERN) 的科学家发明了 WWW 网络传输协议, 使在文本中插入声音和图像成为可能。1993 年美国伊利诺依大学的 Marc Andreessen 设计开发了第一个 WWW 浏览器 Mosaic, 并在网上免费发送。这样, Internet 进入了寻常百姓家。

虽然 Web 页面中可以插入声音和图像, 但它不具备交互性, 它仍然是静态的。要使 Web

页成为动态的可交互的，就需要在其中嵌入程序。由于网络传输带宽的限制，要求程序简练精悍；由于 Internet 是由数以千计的不同种类的计算机构成，要求程序的运行于平台无关；由于存在于网络环境，要求程序有优异的安全性。而 Java 正好符合这些条件，也就被很自然地应用到了 Internet。

HTML / WWW 浏览器技术只限于文本和图像。如果你想播放一种声音或运行一个演示程序，你不得不下载那个文件并用你本机上能理解和运行那个文件格式的程序来播放它。Java 程序和它的浏览器 HotJava 提供了可让你的浏览器运行程序的方法。你能从你的浏览器里直接播放声音。你还能播放页面里的动画。Java 还能告诉你的浏览器怎样处理新的类型文件。当我们能在 2400 baud 线上传输视频图像时，HotJava 将能显示这个视频。

Java 语言使 Web 实现了真正的交互，而且可以在世界几乎任意一台与 Internet 相连的计算机上运行。因此，用 Java 开发的网络应用系统大大增加了开发效率，减少重复劳动。Java 使 Internet 充满了魅力。

1.2 JBuilder 8 简介

Jbuilder 8 是 Borland 公司开发的一个功能强大的可视化 Java 开发工具。JBuilder 8 提供了 4 种版本：Personal（个人版）、SE（专业版）、Enterprise（企业版）以及 Enterprise Studio for Java。其中，专业版较之企业版不支持数据库、企业级应用及 Web 和 XML 发布，而个人版较之专业版又缺少对分布式应用和团队开发的支持，Studio 较之企业版的优点就是集成了完整的电子商务开发平台。

Borland 公司在 2002 年 11 月向外界宣布了 JBuilder 的新版本 JBuilder 8，并于 2003 年 1 月在北京召开 JBuilder 8 的发布会。JBuilder 8 不仅支持 Java 2 企业版（Java 2 Enterprise Edition, J2EE），而且集成了 Borland Web Services Kit for Java，JBuilder 8 还提供了更好的工具来测试 EJB 应用程序。在其他方面还增加了对 Jakarta Struts 的支持，让目标管理和调试更加容易，增加了对非 Java 代码的支持。JBuilder 8 与 Optimizeit5 紧密集成，能够满足应用开发周期（ALM）中不同阶段的需要，因而能够帮助企业加快软件开发速度、降低成本，并提供更好的投资回报。

与以前版本相比，JBuilder 8 增加了一些激动人心的新特性。

1. JDK 1.4.1

在 JDK 1.4.1 上运行的 JBuilder 8 大大提高了开发的效率，在客户端可以更高速度地执行应用程序，而且也增加了对鼠标拖动的支持，在以前版本的 JBuilder 中，只有靠拖动滚动条来修改程序，现在可以直接用鼠标的滚动键来浏览代码了。还提供了更多关于集中管理，在 Linux 上的 UI 支持。

2. 工程管理

JBuilder 8 在工程管理方面也增加了很多新的特性。

(1) 工程面板的文件拖动功能 可以从外部用拖动的方式拖一个文件到 JBuilder 8 的工程面板里面来，对此文件进行编辑，不过这个文件必须是 JBuilder 识别的文件。如果想使它成为工程的一部分，可以单击 Add Files/Packages 按钮。

(2) 创建一个空的新文件 可以通过“File|New”来创建一个新的空文件。

(3) 工程组 可以在两个相关的工程中创建一个工程组来关联两个工程, 两个工程可以在共用相同代码的基础上, 使用自己不同的应用服务器或者 JDK。

(4) 目录视图 可以在工程里面添加一个目录视图, 目录视图列出了一个目录下面的所有文件, 当外界改变了这个文件的时候, 目录视图会相应地更新。

(5) VisualCafe 工程的导入 JBuilder 8 能够用向导来顺利导入 VisualCafe 的工程和文件。

(6) 在 Required Library 里新增一个工程 如果一个工程依靠另外一个工程的话, 可以在 Required Library 里加入一个工程。

3. 调试功能的增强

JBuilder 8 企业版里还有以下的新增功能。

(1) Smart Swap 有了 Smart Swap 就可以在 debugging 的情况下修改代码了。Smart Swap 可以改变、修改并更新你所修改过的文件。有了它, 可以在当前的执行点上在同一个 debugging Session 下测试代码。(JBuilder 8 的 Smart Swap 技术是基于 JDK 1.4 的 Hot Swap 技术。)

(2) 设定 Execution Point 可以在当前被调试的线程上增加一个 Execution Point, 这将改变当前位置的执行断点。

(3) Smart Source Smart Source 能够改变代码的视图, 这样在调试的时候既可以浏览 Java 代码, 也可以浏览非 Java 代码。(JBuilder 8 的 Smart Swap 技术是基于 JDK 1.4 的 JSR45 技术。)

此外, JBuilder 8 对在调试中的编辑、Change Watch command, ExpressionInsight Window, File Modified dialog box 等都进行了优化。

4. 在单元测试方面的改进

JBuilder 8 利用 Cactus 扩展了它在服务器端的单元测试。当然还是和以前一样支持 JUnit, 以下是 JBuilder 8 在测试方面支持的新的特性。

(1) Cactus 的支持 有了 Cactus, JUnit 可以支持在服务器端的 Java 代码的测试。

(2) Cactus 设置向导 您可以从“Wizard\Cactus Setup”菜单来设置 Cactus。

(3) EJB 客户端测试向导 EJB 客户端测试向导现在可以产生两种类型的 Test Client: Junit Test Client 和 Cactus Test Client。

(4) JUnit Text Collector 在 RunTime Configuration 属性对话框里有一个新的运行期测试类型选项, 让用户更加简单地验证测试用例。

(5) Unit testing Stack Trace Filter 能够集中对自己有用的栈跟踪信息。

(6) Junit 3.8 JBuilder 8 里的 JUnit 的版本是 3.8。

5. UML

原来 Project Property 里的 UML 选项卡在 JBuilder 8 里被去掉了, 里面的两个选项被移到了其他的选项卡上去了。其中 UML diagrams 被移到 Project Property 里的 Class Filters 选项卡中, 可以在 Class Filters 选项卡里的 Name 旁边的下拉框中选择 UML diagrams 来加入或者执行一些类和包。Diagram References From Generated Source 被放在 General 选项卡的最下方。

6. Web development

JBuilder 8 另外一个很重要的新特性是增加了对 Struts 的支持。Struts 是基于 Model2 或 MVC 增加了软件的设计。Model 包含的数据, View 是数据代表, Controller 控制 Model 和

View 之间的交互。最典型的 View 就是一个 JSP 页面，Controller 是一个叫 ActionServlet 的 Servlet，Model 可以是任意的数据访问技术，从 JDBC 到 EJB。整个框架包括了 classes, JSP, Servlet, tag library 在 HTML 和 Java 代码之间建立了一个清晰的分层和业务逻辑的形象表达。JBuilder 8 提供了工具和向导，可以快速地建立一个基于 Struts 的应用。

可以在 JSP 里选择手工加入 JSP 标签库和框架支持，JBuilder 8 自带的 Tomcat 版本是 4.1，Tomcat4.1 不支持 JSP 的调试。另外在 JSTL 等方面也有一些改进。

7. Web Services

JBuilder 8 自带了 Borland Web Services for Java，并提供了向导和工具以便能够更快地开发 Web Services。JBuilder 8 可以和各种各样的 Web Services 工具包集成工作。只要选择了 Web Services 工具包，JBuilder 8 就可以使工程支持导入服务，输出 Web Services 的 Java Classes。

只需要把工程设置成支持 Web Services，就可以输出 Web Services 的 EJB，JBuilder 8 默认把所有的无状态 Session Bean 远程接口显示成 Web Services 的。也可以不用默认行为，而手工设置想要显示成 Web Services 的 EJB Modules，Beans 和方法。

JBuilder 8 的 Web Services 功能和所选的开发包有关，包括：

- 可以设置工程支持 Web Services。
- 建立一个 WSDL 文件来表示所开发的 Web Services。
- 创建调用 Web Service 的客户端。
- 导入一个描述 Web Services 的 Ear 或者 WSDL 来创建调用服务的类。
- 创建服务器端的代码来本地化 Services 类。
- 把 Java 类转换成 Web Services。
- 自动地转换无状态 Session Bean 远程接口里的方法。
- 在调用服务的时候监听客户端和服务端端的 SOAP 信息。
- 利用 Web Services Explorer 来查找或者发布一个 Web Services 作为一个 UDDI (Universal Description, Discovery, and Integration) 注册。

8. XML

(1) SAX Handler wizard JBuilder 8 的 SAX Handler wizard 使用了 JDK1.4 里的 JAXP。

(2) Cocoon 在 Web Application 向导里可以选择 Cocoon 作为 Framework，如果你使用 Cocoon Web Application 向导的话，Cocoon 前的框是自动选择上的。

(3) 在 IDE 里的 XML 选项 在 JBuilder 8 的“Tools|IDE Options”里有个新的选项——Ignore DTD。这个选项是默认选择，这样 JBuilder 不解析 DTD，也不报告结构面板里的错误，如果不选的话，JBuilder 每次都解析 DTD 并报告结构面板里的错误。

9. 运行设置

JBuilder 8 重新设计了运行期设置属性对话框，在 Run 这一页里面包含了一个下拉框，可在里面选择需要设置的应用程序的类型。当选择 Edit 或者 Copy 功能时，这个框是不能被改动的。

10. 编译系统

(1) Apache Ant 的支持 Apache Ant 的版本更新至 1.5.1，新增了几项功能。

Build.xml 以相对路径显示在工程面板里，可以右击 build.xml 文件后选择 Properties，在 Ant 页里修改这个选项。

使用 Ant 向导来向工程里新增编译文件，如果使用向导，系统会自动识别编译文件为 Ant 编译文件。

如果想要 Ant 使用 JDK 工程，在 Project Properties 里的 Build 页的 Ant 页里选择 User Project JDK When Running Ant。

(2) Project Properties 里的 Build 页 在 Project Properties 里的 Build 页里有了一些变化：Java 页——可以选择编译的编译器。把 General 页里的 Enable Assert Keyword 选项放到这里来了。

General 页——新增了 Autosave All Files Before Compiling, Refresh Project Before Building, Generate Source To Output Path 等选项。

ANT 页——新增了 User Project JDK When Running Ant 选项。

Menu Item 页——可以在这里新增一些指定的对象（例如设置 Ant 命令）。

Web Service 页——新增了 Regenerate Deployment 选项，覆盖 Web Services 的配置文件。

(3) 编译工程组 可以在 Project Group Properties 改变一个工程组里各子工程的编译顺序和定制菜单项目。

(4) 资源管理 可以让你用 JBuilder 打开你不认识的文件，并在编译的时候跟随工程一起复制到相应的目录。

11. 配置

(1) 排除档案文件的依赖 当使用 Archive Builder 和 Native Executable Builder 创建档案文件和创建 WAR 文件的时候，Archive Builder 和 Native Executable Builder 在档案文件属性里的 Content 页，Web App 里的 Class 页里指定了用不同方式把类和资源整合在一起的几种方式：

Classes: (1)Specified Only (2)Specified And Dependent (3)All

Resource: (1)Specified Only (2)All

例如，如果想在档案文件中包括所有的类和资源的话，可以在 Classes 面板和 Resources 面板里都选择 All。如果不想加入这些资源，而只是想加入自己指定的一些东西的话，可以在两个面板里都选择 Specified Only 选项，然后可以用 Add Classes 按钮加入想要的东西。如果添加类的话必须加工程输出路径的类，如果是加文件的话必须加入工程的 Source 路径里的文件。

(2) Archive Builder Archive Builder 现在有了一种新的档案类型，可执行的 JAR 文件。如果要把一个已经存在的 JAR 文件变成可执行的话，可以选择这种类型。也可以为执行选择一种运行期的设置。

如果在 Configuration 页里为一个应用程序选择主类的时候选择了 <Auto Select>，工程在运行的时候就找到默认的运行设置，包括默认的参数来运行应用程序。如果没有默认的设置或者默认的不是一个应用程序的设置，系统就自动使用第一个应用程序运行期设置。

(3) 创建本地执行 如果选择了 Native Executable 或 Executable JAR 档案类型的话，就可以使用 Archive Builder 来创建可执行的文件。也可以使用 Native Executable Builder 来创建。

(4) 设置本地执行文件 当用 Archive Builder 或 Native Executable Builder 创建了可执行的文件后, 也可以选择属性面板里的 Runtime 来设置创建或者重写配置文件。有以下的设置类型:

- Create executable configuration.
- Create executable configuration and save a copy in the specified file.
- Override the executable configuration with the specified file.

如果选择第二种或者第三种, 设置文件将被加到工程里去。

12. 生产力的增加

(1) 消息框 可以使消息框在屏幕的任何位置变成一个自由移动的窗口。

(2) 状态栏消息 可以使用“Tools|IDE Options|Browser|Status Message Timeout”选项设置状态栏信息保持时间。

(3) 使用 Regular expressions 选项查找 在 JBuilder 8 里的查找选项里新增了使用 Regular expressions 查找。

(4) 拖放 可以拖放文本, 选择上了一段文本后, 可以用鼠标把被选择的部分拖放到指定的地方。这样的功能以前常出现在微软的工具里。

(5) 行数 在代码的左边默认显示了各行的行数, 可以在“Tools|Editor Options”里去掉这个选项。也可以在 Line Number 页边直接按右键来快速取消这个选项。

(6) 选择文本 可以利用左边的行数来选择一整行或者一行的一部分。

(7) 代码格式 可以在“Project|Project Properties”里的 Formatting 选项卡中选择相关的选项, 来制定代码格式的偏好和自动对齐代码。在旁边的一个窗子里会显示出设置的改变。

(8) 缩放工具 在编辑器的底部有一个放大镜的按钮, 用这个按钮可以缩放代码。

(9) 快捷键的改进 可以在编辑器的底部改变快捷键的设置, 可以设置不同的方式, 也可以在 Editor Option 里改变具体的某个键。

13. 团队开发的改进

(1) Merge Conflicts 在 History 面板里有一个 Merge Conflicts, 可以在这里查看或者解决在工作台和版本控制库里相应的文件的不一致情况。(Merge Conflicts 只支持 CVS 和 VSS。)

(2) CVS 的改进 CVS 有两个关键的改进: 一是在 JBuilder 8 里用 pserver 方式连接的时候, 可以指定端口号。二是能在 Pull Project From CVS 向导里检查一系列的模块和分支。

(3) Clear case 的改进 Clear case 在版本控制设置管理方面也有很多改进, Status Browser 主要是一个浏览工具: 可以浏览激活的工程, 显示每个文件的版本控制状态, 显示可用的版本和不同版本之间的差别。Commit Browser 提供了 Status Browser 浏览功能, 提供改变文件版本控制操作的一般的访问, 有了 Commit Browser, 能够为每一个文件设定想要设定的版本控制命令。输入一些单个文件或者整个组的注释, 然后单击一下就可以执行所有的命令了。另外还提供动态和快照支持、控制台输出、对于一个文件多个人的修改合并功能、用命令来访问 ClearCase 工具和提供 UCM 的支持。

14. 支持 J2EE 服务器

以下是 JBuilder 8 支持的 J2EE 服务器:

- 支持 Sybase EAServer。