

A close-up photograph of several blue and white computer keyboard keys, including 'O', 'P', 'L', and 'E', arranged in a row at the top of the page.

JBUILDER 8

基础编程

李 樱 编著

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

TP312
1180

JBUILDER 8

基础编程

李 樱 编著

北方工业大学图书馆



00535814

人民邮电出版社

图书在版编目 (CIP) 数据

JBuilder 8 基础编程/李樱编著. —北京: 人民邮电出版社, 2003.6

ISBN 7-115-11268-1

I. J... II. 李... III. JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 036546 号

内容提要

JBuilder 8 是 Borland 公司最新推出的可视化 Java 语言编程工具, 可以满足不同层次开发者的需求。本书结合大量实例详细介绍 JBuilder 8 编程方法和编程技巧。本书共 10 章, 内容按照由浅入深、从常规编程到高级编程的方式展开。第 1 章介绍 JBuilder 8 的基础知识; 第 2 章讲解 JBuilder 8 开发 Java 常规程序的方法; 第 3 章讲解 JBuilder 8 对于 Java 小程序的开发过程; 第 4 章讲解了 JBuilder 8 开发界面程序时所提供的所见即所得的编辑界面; 第 5 章介绍 Java 多媒体编程; 第 6 章至第 10 章分别讲解了 Java 网络编程、JBuilder 数据库编程、服务器端小程序 Servlet 的开发、JBuilder 中开发 EJB 和用 JBuilder 开发 JSP。

本书实例丰富, 重点突出, 实用性强, 适合广大初、中级 Java 程序员学习和参考。

JBuilder 8 基础编程

◆ 编 著 李 樱

责任编辑 刘 浩

◆ 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@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67132692

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 21

字数: 502

2003 年 6 月第 1 版

印数: 1-5 000 册

2003 年 6 月北京第 1 次印刷

ISBN7-115-11268-1/TP · 3449

定价: 32.00 元

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

前 言

根据 Evans Data 公司的调查表明, Borland 公司在 Java 开发工具市场中占有 18.1% 的市场份额, 名列众 Java 开发工具商之首。在 2002 年初 Borland 公司推出 JBuilder 7, 由于 Sun 公司将 Java 的版本升级至 1.4.1, 因此 2002 年 11 月, Borland 公司及时地推出了符合 Java 1.4.1 规范的 JBuilder 8。

本书是一本通过实例讲解 JBuilder 8 开发知识与开发技巧的参考书。通过对每个实例编程思路和代码实现进行详细分析和讲解, 引导读者快速地学会 JBuilder 8 的开发技术, 规范编程思路、培养良好的编码习惯, 提高程序设计水平。

对于初级读者, 通过本书的学习能迅速的掌握 JBuilder 8 的开发技能, 打破了初级读者必须先学习 Java 基础知识, 然后再学习专用开发工具的学习定式。对于中、高级读者, 例如具有一定的 Java 程序设计基础、熟悉 Java 的语法、有一定的程序设计经验、有一定的项目经验的读者, 通过本书可以了解到 JBuilder 8 的最新进展, 并能通过详细的程序分析学习到快速开发的必备知识, 领略到高水平开发人员的精妙思路, 提高编程技巧和开发能力。

本书的实例内容涉及面比较广泛, 基本上囊括了通常程序设计中的各个方面, 凝聚了作者多年的 JBuilder 编程经验, 相信对启发读者的思路并提高读者的编程水平会有很大帮助。本书立足经典但更突出新颖, 不是简单的 Java 基础类图书。

在本书的编写过程中, 得到很多同行和朋友的热情帮助, 很多朋友为本书的编写提供了部分宝贵资源和参考意见, 在此表示由衷的感谢。

本书主要由李樱执笔编写, 贺军、贺民、龚亚萍、陈安南、李晓春、戴军、李志云、陈伊文、孟丽艳、李志伟、周里文、王学龙、徐江、纪红、孙燕、吴红燕、王巧红、王春桥、陈雷等人在试用、通读、实例调试、校对等方面做了大量的工作, 在此深表感谢。由于时间仓促、作者水平有限, 书中未免会有不足和疏漏之处, 敬请广大读者提出宝贵意见。

编者

2003 年 5 月

目 录

第 1 章 走进 JBuilder 8	1
1.1 Java 介绍	1
1.1.1 Java 的历史	1
1.1.2 Java 的特点	2
1.1.3 Java 语言的应用前景	4
1.2 JBuilder 8 介绍	4
1.2.1 JBuilder 的历史	5
1.2.2 JBuilder 特性	5
1.2.3 JBuilder 8 的安装	7
第 2 章 Java 程序入门	11
2.1 JDK 开发环境	11
2.1.1 使用 JDK 编译 Java 程序	11
2.1.2 执行 Java 程序	12
2.1.3 调试 Java 程序	13
2.2 JBuilder 开发环境	14
2.3 Java 程序开发基础	21
2.3.1 程序结构	21
2.3.2 注释	22
2.3.3 标识符	22
2.3.4 数据类型	22
2.3.5 表达式	23
2.3.6 Java 程序控制	24
2.4 Java 与面向对象	25
2.4.1 对象实例	26
2.4.2 实例变量	26
2.4.3 new 操作符	27
2.4.4 点操作符	27
2.4.5 成员函数定义	27
2.4.6 成员函数调用	28
2.4.7 this	28
2.4.8 构造函数	29
2.4.9 继承	29
2.4.10 静态	30
2.4.11 抽象	31
2.5 经典入门实例	32
2.5.1 Hello World 小程序 (Applet)	32

2.5.2	Hello World 程序 (Application)	35
2.5.3	基本语法练习	41
2.5.4	面向对象实例	48
第 3 章	Java 小程序	51
3.1	Java Applet 介绍	51
3.1.1	小程序结构分析	51
3.1.2	小程序与网页的结合	53
3.1.3	小程序与多线程	54
3.1.4	动画小程序	60
3.1.5	缩短小程序装入时间	65
3.2	抽象窗口工具包 AWT	66
3.2.1	使用复选框, 单选按钮和标签	67
3.2.2	使用列表、文本域和选择框	73
3.3	线程	77
3.3.1	Java 线程的几个概念	77
3.3.2	线程实例	78
第 4 章	Java 高级用户界面编程	85
4.1	Swing 介绍	85
4.2	Swing 小程序	86
4.3	事件响应	89
4.4	Swing 组件实例演示	89
4.4.1	图标和标签	89
4.4.2	按钮、复选框、单选框	101
4.4.3	列表和组合框	109
4.4.4	表	114
4.4.5	树	118
第 5 章	Java 多媒体编程	123
5.1	关于 JMF 技术	123
5.2	最简单的播放器	125
5.3	JMF 基础知识	129
5.3.1	javax.media 包	130
5.3.2	ControllerListener 界面	131
5.3.3	Player 界面	131
5.3.4	Manager 类	132
5.3.5	Manager.createPlayer 方法	132
5.3.6	Player.addControllerListener 方法	133
5.3.7	Player.prefetch 方法	133
5.3.8	RealizeCompleteEvent 事件	133
5.3.9	Player.getVisualComponent 方法	133

5.3.10	Player.start 方法	134
5.4	程序的执行过程	134
5.4.1	开始运行	134
5.4.2	构造函数	134
5.4.3	响应事件	135
5.4.4	运行分析	136
5.5	响应 DurationUpdateEvent 事件	136
5.5.1	问题分析	136
5.5.2	编写代码	137
5.5.3	DurationUpdateEvent 事件	138
5.5.4	Time 类	138
5.5.5	DurationUpdateEvent.getDuration 方法	139
5.5.6	Time.getSeconds 方法	139
5.5.7	运行分析	139
5.6	根据状态响应 TransitionEvent 事件	140
5.6.1	问题分析	140
5.6.2	编写代码	141
5.6.3	TransitionEvent 事件	143
5.6.4	TransitionEvent.getPreviousEvent 方法和 getCurrentState 方法	143
5.6.5	Player 的状态	143
5.6.6	运行分析	144
5.7	关闭窗口	146
5.7.1	问题分析	146
5.7.2	编写代码	147
5.7.3	Player.close()方法	148
5.7.4	运行分析	148
5.8	打开文件	150
5.8.1	问题分析	150
5.8.2	设计界面	150
5.8.3	编写代码	152
5.8.4	运行分析	154
5.9	多次打开文件	155
5.9.1	问题分析	155
5.9.2	编写代码	156
5.9.3	ControllerClosedEvent 事件	157
5.9.4	运行分析	157
5.10	控制面板	158
5.10.1	问题分析	158
5.10.2	编写代码	158
5.10.3	player.getControlPanelComponent 方法	160

5.10.4 运行分析	160
5.11 循环放映.....	162
5.11.1 问题分析	162
5.11.2 设计界面	163
5.11.3 编写代码	164
5.11.4 EndOfMediaEvent 事件.....	166
5.11.5 player.setMediaTime 方法.....	166
5.11.6 运行分析	166
5.12 本章小结	167
5.12.1 TransitionEvent 事件.....	167
5.12.2 Controller.realize 方法	167
第 6 章 Java 网络编程	169
6.1 SOAP 基础.....	169
6.1.1 什么是 SOAP	169
6.1.2 构思 SOAP	170
6.1.3 HTTP 上的 SOAP	173
6.2 Web Service Kit for Java 开发包	174
6.2.1 安装 Web Service Kit for Java 开发包	174
6.2.2 开发包对 JBuilder 8 的更新.....	175
6.3 开发 Web Service 程序	177
6.3.1 创建 SOAP Server.....	178
6.3.2 创建 Bean	181
6.3.3 把 Bean 导出成 Web Service.....	182
6.3.4 创建客户端程序	188
6.3.5 调试 Web 服务程序	188
6.4 理解 WSDL	190
6.4.1 什么是 WSDL	190
6.4.2 wsdl:definitions 标记	191
6.4.3 wsdl:message 标记	192
6.4.4 wsdl:part 标记	192
6.4.5 wsdl:portType 标记	193
6.4.6 wsdl:binding 标记	194
6.4.7 wsdl:port 标记	195
6.4.8 wsdl:service 标记	195
6.4.9 理解 WSDL	196
6.5 从 wsdl 生成代码.....	196
6.5.1 [TimeService] deploy.wsdd 文件	196
6.5.2 EchoTimeServicePortType.java 文件	197
6.5.3 EchoTimeService.java 文件	197

6.5.4	EchoTimeServicePortSoapBindingStub.java 文件.....	198
6.5.5	EchoTimeServiceLocator.java 文件.....	198
6.5.6	执行过程分析.....	199
6.6	Axis 管理.....	201
6.6.1	管理 Axis 模块.....	202
6.6.2	访问 Axis Servlet.....	203
6.6.3	查询部署信息.....	203
6.7	Web 服务浏览器.....	204
6.7.1	连接到 Axis 服务器.....	205
6.7.2	查看服务信息.....	206
6.7.3	导入 WSDL 文件.....	208
6.8	TCP 监视器.....	208
6.8.1	创建一个 TCP 监听配置.....	209
6.8.2	图形界面的客户端程序.....	210
6.8.3	使用 TCP 监视器监听.....	212
6.9	相关说明.....	213
6.9.1	XML 名称空间和架构的位置.....	213
6.9.2	相对 URI.....	214
6.9.3	生成 URI.....	214
第 7 章	JBuilder 数据库编程.....	215
7.1	JDBC 简介.....	215
7.1.1	什么是 JDBC.....	215
7.1.2	JDBC 和 ODBC 比较.....	215
7.1.3	JDBC 驱动程序的类型.....	216
7.2	JDBC 数据库应用程序.....	217
7.2.1	如何使用 JDBC.....	217
7.2.2	java.sql.DriverManager.....	217
7.2.3	java.sql.Connection.....	218
7.2.4	java.sql.Statement.....	218
7.2.5	java.sql.ResultSet.....	219
7.2.6	一个 JDBC 程序的实例.....	219
7.2.7	JDBC2.0 扩展 API.....	224
7.3	用 JBuilder 组件创建数据库程序.....	227
7.3.1	添加 JDBC 驱动至 JBuilder.....	227
7.3.2	创建一个数据库应用程序.....	228
7.3.3	增加数据库组件.....	229
7.3.4	增加 dbSwing 用户界面组件.....	231
7.3.5	QueryDataSet 组件.....	231
第 8 章	服务器端小程序 Servlet 的开发.....	237

8.1 Servlet 简介	237
8.1.1 什么是 Servlet	237
8.1.2 Servlet 的工作原理	237
8.1.3 Servlet 的生命周期	238
8.1.4 Servlet 与 CGI 程序相比的优点	239
8.2 Java Servlet API 编程实例解析	239
8.2.1 HttpServlet 类	239
8.2.2 第一个 Servlet 程序	241
8.2.3 HttpServletRequest 和 HttpServletResponse	243
8.2.4 HttpSession 接口和 Cookie 类	251
8.2.5 HttpSessionBindingListener 和 HttpSessionBindingEvent	258
8.3 建设一个自己的搜索引擎	260
第 9 章 JBuilder 开发 EJB	265
9.1 EJB 简介	265
9.1.1 什么是 EJB	265
9.1.2 Web 编程的三层体系结构	265
9.1.3 EJB 技术的设计目标	266
9.1.4 使用 EJB 技术的优点	266
9.1.5 EJB 和其他技术的关系	267
9.2 EJB 规范	268
9.2.1 EJB 容器	268
9.2.2 会话 bean	269
9.2.3 实体 bean	269
9.2.4 消息驱动 bean	270
9.2.5 EJB 的部署	271
9.3 创建会话 Bean	272
9.3.1 JBuilder 下设置 Weblogic Server	273
9.3.2 建立测试客户程序	279
9.4 实体 Bean	284
9.4.1 用 JBuilder8 创建 CMP 实体 Bean	285
9.4.2 本地接口	287
9.4.3 远程接口	287
9.4.4 回调方法	288
9.4.5 运行与测试	289
第 10 章 用 JBuilder 开发 JSP	293
10.1 JSP 技术概述	293
10.1.1 JSP 的技术特点	293
10.1.2 JSP 与 Asp 的比较	294
10.1.3 JSP 是如何运作的	295

10.2 JSP 语法详解	295
10.2.1 Page 指令	295
10.2.2 表达式指令	296
10.2.3 注释指令	297
10.2.4 Scriptlet 指令	297
10.2.5 声明指令	298
10.2.6 JSP Action 指令	298
10.3 构建一个购物网站	304
10.3.1 欢迎页面	305
10.3.2 商品列表页面	306
10.3.3 购物页面	310
10.3.4 计费页面	316
10.3.5 欢送页面	319

第 1 章 走进 JBuilder 8

本章第 1 节主要介绍了 Java 的产生、历史以及 Java 的一些特性, 比如安全问题; 在第 2 节对 JBuilder 的产生和历史作了简要的介绍并对如何安装也作了详细的说明。

1.1 Java 介绍

Java 来自于 Sun 公司的一个叫 Green 的项目, 其原先的目的是为家用消费电子产品开发一个软件产品, 这样人们就可以发指令给电冰箱、电视机等家用电器, 对它们进行控制, 并进行信息交流。在规划这个产品时, 他们首先想到这个产品必须具有平台无关性, 因而很自然地把扩展 C++ 编译器作为最初的考虑, 但很快他们便意识到这个产品还必须具有高度的简洁性和安全性, 而 C++ 在这方面显得无法胜任, 于是他们决定自行开发一种新的语言 Oak。这就是 Java 语言的前身。

1.1.1 Java 的历史

在 Green 小组开发 Oak 的同时, 在 Internet 上对 WWW(Wide World Web)的研究也在独立地进行中。WWW 是建立在 Internet 和客户/服务器模型上, 以超文本传输协议(HTTP)、超文本标记语言(HTML)和统一资源定位器(URL)为基础, 能够提供各种 Internet 服务, 并且用户界面一致的信息浏览系统。WWW 的出现使得 Internet 的表现力及魅力陡增, 它的影响力已远远超出专业技术人员的应用范畴, 而进入到广告、新闻、销售和服务等各个领域, 使 Internet 迅速地走入千家万户。WWW 页面虽然很丰富, 可以实现声、图、文并茂, 但它是静态的。需要通过一种机制来使它具有动感, 嵌入语言是一种很自然的解决方案, 但这种语言必须简练、小巧、与平台无关, Oak 正好满足这些要求。于是 Oak 语言被重新定位于 WWW 浏览器的应用上。

1995 年 1 月, 更为完善的 Oak 版本被重新起了一个更响亮的名字: Java。

1995 年春季, Sun 公司公布了 Java 的完整技术规范, 立即得到包括 Netscape 公司在内的各 WWW 厂商的广泛支持。不久, 许多著名的大公司, 如 IBM、Novell、Oracle 和 Borland 公司都相继购买了 Java 的使用许可, Java 的地位得到了广泛的支持。Microsoft 公司一开始对 Java 无动于衷, Bill Gates 在反复仔细研究了 Java 的技术规范之后, 不得不承认“Java 确定是有史以来最伟大的程序设计语言”, 并确定微软公司整个软件开发的战略从 PC 单机时代向着以网络为中心的计算时代转移, 而购买 Java 则是他的重大战略决策。

Java 的取名也有一个趣闻。有一天, 几位 Java 成员组的会员正在讨论给这个新的语言取什么名字, 当时他们正在咖啡馆喝着 Java(爪哇)咖啡, 有人灵机一动说“就叫 Java 怎样”, 得到了其他人的赞赏后, Java 这个名字就传开了。

1.1.2 Java 的特点

Java 是一种简单、动态、面向对象、分布式、解释执行、健壮、安全、结构中立、可移植、高效能、具有多线程编程能力的新一代语言。

1. 简单性

Java 删除了许多极少被使用、不容易理解和容易令人混淆的 C++ 功能。剔除的功能主要包括运算符重载、多重继承。与 C++ 相比, Java 语言也提供重载函数, 不过它重载的对象是函数而非变量或是运算符。Java 还增加了自动内存收集功能, 从而简化了 Java 程序的工作。存储管理(storage management)是使 C 与 C++ 应用程序变得复杂的一项常见的原因, 即关于内存的分配与释放。Java 语言的自动垃圾收集功能(周期性地释放未被使用的内存)不仅简化了程序设计工作, 而且能大幅度减少小错误(bug)数量。Java 的目标之一是要协助开发能独立地在小型机器上顺利执行的软件。Java 的基本解释器和对象类支持约占 40KB 空间, 而若增加基本的标准程序库和线程支持则需增加约 175KB 的空间。体积小对于在嵌入式系统中的使用是很重要的, 正因为这样, Java 才能够通过网络轻松下载。

2. 面向对象

面向对象是在业界中一度流行的术语之一。它可以促成明确的接口定义, 并允许开发人员建立可重复使用的软件部件。

3. 分布式

Java 拥有广泛的能轻易地处理 TCP/IP 的运行库, 例如 HTTP 与 FTP 类库等。这使得在 Java 中比在 C 或 C++ 中更容易建立网络连接。Java 应用程序可以借助 URL 通过网络开启和存取对象, 就如同存取本地文件系统一样简单。

4. 健壮性

Java 的目标是要协助开发人员建立各方面可靠的程序, Java 强调: 开发人员可在设计初期即检查可能存在的问题, 之后则进行动态地检查, 并排除容易出现的错误。C++ 语言的优点是允许在编译时进行深入的检查, 以便提前发现错误。

Java 与 C/C++ 之间最大的不同点之一在于 Java 拥有一种模型, 能排除内存被覆盖和毁损数据的可能。Java 不采用指针计算法, 而是提供真正的数组阵列。这允许程序执行下标检查, 但不允许将一个整数转成指针的情形发生。

5. 安全性

Java 设计的目的是要能够使用于网络/分布式运算环境。为此, Java 非常强调安全性, 以确保建立无病毒且不会被侵入的系统。Java 的验证技术是以公钥加密法为基础的。健壮性与安全性之间存在一种很强的相互作用关系。例如, 指针语意的改变, 让应用程序不可能伪造对象结构存取权, 或存取它们在对象中没有存取权的私有数据。这等于关上大门, 杜绝大多数病毒的侵入。

6. 中立性

Java 的设计目标是要支持网络应用程序。一般而言,网络由许多不同的平台系统构成,包括各种 CPU 与操作系统结构。为了让 Java 应用程序能够在网络上任何地方执行,其编译器会生成一种具备结构中立性的目标文件格式。编译后的程序码可以在提供 Java 运行系统的多种不同处理器上执行。

Java 的这种中立性结构不仅对网络应用很有帮助,而且也很适合单一的系统软件流通。Java 编译器是通过生成与某一特定电脑结构无关的字节代码指令,达到上述功能的。这些字节代码指令能很容易在任何机器上解释执行。

结构中立性是确保程序可移植的最重要部份,不过除此之外还需很多必须配合的条件。和 C 与 C++ 不同的是,Java 规范中并无任何与机器结构相依存的陈述存在,它指定了基本数据类型的大小,及其算术运算元的执行行为。例如:“int”代表一个有符号的二进制补码 32Bit 整数,而“float”代表一个 32Bit IEEE754 浮点数。这些选择在今天的环境很适用,因为几乎所有 CPU 都具备这些特性。

程序运行库属于系统的一部份,它定义了一些可移植的程序接口。例如,它包括一个抽象的 Windows 类,并且提供了该类在 UNIX、Windows 和 Macintosh 平台上的实现。

7. 解释执行

Java 解决器可以直接在任何已移植该解释器的机器上解释、执行 Java 字节代码。再者,由于其连接过程比较倾向于逐步增量与轻量过程,因此程序开发更快、更精密。

8. 高效能的 Java 程序

虽然解决过的字节代码性能已相当不错,不过有些情形下还是要求程序达到更高执行效能。字节代码可以动态地解释为执行应用程序特定 CPU 的机器码。这对于习惯使用一般编译器与动态载入器的程序设计者而言,有点类似将最终的机器码生成器放到动态载入器之内。字节代码格式在设计上即已考虑了机器码的产生,因此实际的机器码生成程序相当简单。其生成的机器码是有效的,编译器自动分配寄存器,而在生成字节代码期间也会进行一些优化。解决码在 Sun Microsystem SPARC Station10 上执行时,可以达到每秒三十万个函数调用速率。字节代码转换至机器码的速度性能,几乎和直接生成机器目标代码的 C 或 C++ 没有什么两样。

9. 多线程的支持

Java 拥有一组复杂的同步化基本单元,它们是以广泛使用的 C.A.R.Hoare 监视器与条件变量图为基础的。将这些概念融合到语言中之后,它们就变得更加容易使用且更为健壮。这种融合方式大部分来自 Xerox 的 Cedar/Mesa 系统。

多线程所带来的其他好处包括更好的交互式回应能力与实时运行能力。然而这会受到底层平台的限制:独立执行的 Java 运行环境有着很好的实时执行能力,而若在其他系统例如 UNIX、Windows、Macintosh 或 Windows NT 等平台上执行时,则会由于底层平台的原因而使实时响应能力受到影响。

10. 动态特性

就各方面而言, Java 是一种比 C 或 C++ 更具动态特性的语言, 在设计上强调为不断发展的运算环境提供支持。Java 能了解由 Objective C 引用过来的接口概念。简单地说, 接口就是规范一组与对象相对应的成员函数, 但对象如何实现这些成员函数则亟待解决。一个类实现一个接口是要提供这个接口所包含的所有成员函数的实现, 与此相反, 派生子类则从父类继承了一组成员函数以及它们的实现。一个 Java 类可以实行多个接口, 但只能从一个父类继承。接口告诉连接对象它可以做什么而不是怎么做, 使得它在代码上更具有灵活性和可复用性。在 C 或 C++ 程序中, 如果有一个指针指向一个对象, 但不知道该对象的类型是什么, 那么将没有办法找出它。然而, 在 Java 中根据运行类型信息进行查找是很直截了当的, 因为在编译阶段和运行阶段都会检查数据类型转换。所以在 Java 中完全可以信任这种转换, 而 C 与 C++ 的编译器则只是相信用户自己已做了正确的处理。

1.1.3 Java 语言的应用前景

Java 语言有着广泛的应用前景, 大体上可以从以下几个方面来考虑其应用:

- 所有面向对象的应用开发, 包括面向对象的事件描述、处理、综合等。
- 计算过程的可视化、可操作化的软件的开发。
- 动态画面的设计, 包括图形、图像的调用。
- 交互操作的设计, 包括选择交互、定向交互、控制流程等。
- Internet 的系统管理功能模块的设计, 包括 Web 页面的动态设计、管理和交互操作设计等。
- Intranet (企业内部网) 上的软件开发 (直接面向企业内部用户的软件)。
- 与各类数据库连接查询的 SQL 语句实现。

1.2 JBuilder 8 介绍

JBuilder 是目前最好的 Java 语言编程工具, 是由 Borland 公司开发的。

Borland 软件公司是全球一流的技术提供商, 其技术主要用于开发、分发和集成软件应用。Borland 公司提供的优秀技术解决方案是专门针对互操作性而设计的, 各种规模的企业都可以借此成功迈入基于 Web 的网络计算时代, 提升现有系统的应用价值。从《财富》1 000 强企业到由上百万开发人员组成的“Borland 一族”(Borland Nation), Borland 公司都使其客户能够自由地开发应用程序, 将应用分发到任何地方, 并在整个企业内对这些应用进行集成和管理。Borland 解决方案能够帮助企业提高效率, 并在降低用户总成本的情况下, 使客户更快速经济地完成更高性能的项目。

Borland 公司成立于 1983 年, 更多信息请访问 <http://www.borland.com> 或者 <http://community.borland.com>。

1.2.1 JBuilder 的历史

java 发布之初, Borland 公司是率先进入 java 工具市场的公司之一。那时的所谓进入,就是将 java 作为附属物,“粘贴”在他们的 C++ 产品,即 BorlandC++5.0 之上。这一战略决策,与 Symantec 公司所遵从的做法大致相同, Symantec 最初也是利用其已有的 IDE 编码基地,然后才转移到专用的编码基地之上的。但在对 java 付出最初努力之后, Borland 公司经历了一段时间的反思,它在 java 工具的前沿,从表面上看,几乎没有获得什么进展。现在,随着 JBuilder 的发布, Borland 公司将就 java 工具市场提出新的势力划分要求。

然而不幸的是,那时 Borland 公司的努力与所期望的还相差甚远。这一产品的最大缺点就是其对 java 的支持其实是将 java 以一种嵌入模式(add-onmodule)加入到 C++ 中,而不是开发一个有自身特色的工具。将 java 嵌入 C++ 中的做法所带来的一个难题,就是从编译部件、对象文件和编译目标等方面来看, java 都与 C++ 格格不入。在 java 环境中,编译了一个类文件成为对象之后,就可以实例化在系统已有的其他对象。在这种情况下,没有在继承式的 C++ IDE 中应用的,诸如“.exe”和“.dll”之类的目标文件。因此,利用这一工具,创建一个 Java 类是十分麻烦的,类文件几乎是不存在的。

在这个嵌入式的 C++ 产品推出之后,有关“Latte”的消息也开始传扬了出来。“Latte”是他们给将要开发的 IDE 环境所起的名称代码,这个 IDE 环境将由来自于 Delphi 开发组的工程师来开发,但它将全部采用 java 语言来编写。然而,这个雄心勃勃的开发项目却迟迟不能得到实现;1996 年,在 SanFrancisco 的第一次 JavaOne 开发者会议上,展出了它的演示版,接下来的 1997 年 JavaOne 会议上,只能又一次看到它的演示,直到 JBuilder 1 的正式发布。

1.2.2 JBuilder 特性

JBuilder 软件的目标定位是代码开发人员而不是高级设计人员,所以 JBuilder 中包含了大量的向导程序和其他针对中间层的快速开发工具。

Borland 公司的 JBuilder 是一款非常成功的软件。最近,该公司又推出了 JBuilder 的新版本 JBuilder 8。新版本将应用开发生命周期中的产品无缝集成为一体,这些产品包括在开发过程中起协调作用的 Borland TeamSource,以及用于性能测试的 Borland Optimizeit Suite,而它们共同的目的就是为了开发出高质量的应用程序。Borland 公司一直在致力提高 JBuilder 的性能和易用性,下面笔者就介绍一下 Jbuilder 4 至 JBuilder 8 的发展历程,虽然本书是介绍 JBuilder 8 的,但是以前版本的功能大多数现在还存在,所以了解一下对用户使用最新版本也是很有帮助的。

1. JBuilder 4

- 性能更稳定、使用更方便,特别适用于创建 Java 组件。此外该软件还提供了很多 JSP 功能。
- 内置的 Java 和 Swing (Java 的 GUI 程序框架) 技术支持跨平台的软件开发,它可以运行在 Windows、Linux 和 Solaris 操作系统之上,操作界面和工具完全一致。
- 和早期版本一样, JBuilder 4 IDE 设计得更富于效率性和功能性。用户可以在同一开发环境下很方便地查看 Java、HTML 和 JSP 文件、建立 Sun 公司推出的新型 JDK 1.3

HotSpot 虚拟机（这种新型虚拟机优化了 Java 代码）。JBuilder 具有了相当出色的响应界面。

- 在多种令人瞩目的特性之后，还有用户特别喜欢特别方便的自动补充功能（Auto-Completion Wizardry），该功能不仅支持即时帮助还支持键入 Java 代码的参数自动填充。
- JBuilder 4 Enterprise 版本更提供了对 Enterprise Java-Bean 组件开发的支持。JBuilder 支持 EJB 1.1，由于采用了灵活的 XML 解决方案，软件还支持其后出台的 EJB 2.0。目前用户可以自动地生成基于后端数据库表的 EJB Java 类。JBuilder 同时还简化了 EJB 的自动部署功能。此外它还支持 CORBA，相应的向导程序有助于用户全面地管理 IDL（分布应用程序所必需的接口定义语言 interface definition language）和控制远程对象。
- JBuilder 的本地开发依赖于软件中所包含的轻度 Java 数据库，这种数据库被叫做 JDataStore。
- JBuilder 4 Enterprise 版本包含了 300 多种组件，其中，大多数组件都属于 JavaBean 和 Swing 控件，这对建立客户端小程序和独立运行的 Java 应用程序是很有用的。此外软件还包括 16 种名为 InternetBeans 的特殊控件。不过，这些控件都是专用于 JSP 和 Servlet 的。这些组件封装了 Web 表单中采用的控件，其范围不仅包括了简单的按钮也包括表这样的高级特性。
- 提供了完整功能的调试器，用户可以通过它调试服务器端代码。该产品还针对团队开发推出了 CVS 版本控制功能，该技术已经被程序员用来建立测试应用程序了。采用 CVS，开发队伍就可以方便地跟踪项目开发全过程了。

2. JBuilder 5

JBuilder 5 新增了 100 多种功能，这将使 Java 开发更方便快捷。

- 提高了产品性能，达到了和大多数目前在 Java™ 2 平台上提供解决方案方面取得广泛声誉的可视化开发工具的产品相一致的性能。
- 提高了开发者生产效率。
- Java/XML 开发：充分利用跨平台 XML 的优势。可以使用源代码、XML 和 HTML 输出的浏览器显示转换 XML 文档。对 XML 文档进行合法性检查并跟踪错误。JBuilder 使用工具来定义结构和规则，数据绑定以及对 XML 文档可编程的操作，简化了 XML 和数据库文件之间的数据传输。
- 普通的 XML 支持：支持各种 XML 文件类型，诸如 XSD、XSP、DTD、XSL、MAP、（XML-DBMS）、WML、XHTML 等。
- 团队开发：JBuilder 5 简化了大型团队的开发协同管理，这是通过提供对 Rational ClearCase, CVS, 和 Microsoft Visual SourceSafe.™ 的支持来实现的。另外，还改进了源代码管理，可以使用带有高生产性工作流程模型的完整的同步版本知识库引擎，通过网络和 Internet 进行协作。使用历史修改浏览器来可视化地更新和解决版本冲突，协调工作站和知识库之间的改变，包括源码级别的差异显示、唯一性的源代码修改和在代码编辑器中的冲突导航等，增加 to-do 列表和 to-do Javadoc 注释来帮助在开发过程中保持对工作的跟踪。