

万水计算机编程技术与应用系列

JBuilder 9.0

——Java程序员成功之路

冯军 程超 卫欣 等编著

程序设计



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

JBuilder 是目前应用最广、用户最多且广受用户好评的 Java 集成开发环境，众多的软件公司都在使用 JBuilder 进行项目的研发。本书依托具体开发环境和平台进行讲解。学生学完这门课程之后，就可以知道如何入手编写程序。全书共 11 章，详细介绍了具体实例的开发过程。本书适合于软件开发工程师自学如何使用 JBuilder 9.0 开发 Java 应用程序，也适合作大专院校 Java 课程的教材或教辅材料。

图书在版编目 (CIP) 数据

JBuilder 9.0 程序设计: Java 程序员成功之路 / 冯军等编著. —北京: 中国水利水电出版社, 2004

(万水计算机编程技术与应用系列)

ISBN 7-5084-2082-9

I J… II.冯… III.JAVA 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2004) 第 034166 号

书 名	JBuilder 9.0 程序设计——Java 程序员成功之路
作 者	冯军 程超 卫欣 等编著
出版 发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 63202266 (总机) 68331835 (营销中心) 82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787mm×1092mm 16 开本 21.75 印张 493 千字
版 次	2004 年 5 月第 1 版 2004 年 5 月第 1 次印刷
印 数	0001—5000 册
定 价	32.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社营销中心负责调换

版权所有·侵权必究

前 言

JBuilder 是目前应用最广、用户最多且广受用户好评的 Java 集成开发环境，众多的软件公司都在使用 JBuilder 进行项目的研发。如何使用这个功能强大的 IDE 进行 Java 程序设计是现在众多 Java 程序员急切盼望的事情。大中专院校和培训班在讲授 Java 课程时，也依托具体开发环境和平台进行教学，从而保证学生学完这门课程之后，不会变得虽然熟背 Java 语法，但不知道如何入手编程。

本书的组织编排充分体现了教学的特点。每章前面的知识点给出了本章的核心知识点；每章的各个小节分别对各个知识点进行了深入地阐述，部分小节还给出了相关的示例；每章的最后还对本章内容进行总结，有利于读者回顾本章所学内容；整体而言，每章各节任务量安排适当。每章都提供不同数量的练习，练习针对性强，读者做练习的同时又温习了本章介绍的主要技术方法。在介绍具体知识点时都提供很多小例子，使读者可以快速掌握所介绍知识点的用法和注意事项。同时每章还提供 1~2 个大实例，分成三部分“实例说明”、“实例实现”和“实例总结”来讲解，详细陈述实例实现步骤，并对实例进行透彻分析，非常适合于培养初学者或自学者进行实际编程的能力。

全书共分 11 章。第 1 章总体概述了 Java 语言和 JBuilder 9.0 的特性；第 2 章讲述了 Java 语言的基本语法，包括：Java 数据类型、流程控制语句以及 Java 语言面向对象概念；第 3 章详述 Java Applet 程序的架构、实施以及相关的实例；第 4 章详述了 Java Swing 图形界面的设计、步骤以及相关实例；第 5 章详述了如何开发 Java 数据库应用，包括数据库的连接、操作和管理等；第 6 章讨论了 JMF 以及 Java 多媒体应用的开发，并提供了相关实例；第 7 章详细介绍了 Java I/O 流的概念及相关编程技巧，并使用一个综合实例来具体讲解主要对象和方法；第 8 章介绍了多线程的概念和 Java 多线程的实现，并提供了一个综合实例；第 9 章讨论了网络通信的概念，以及如何用 Java 语言实现网络通信的方法；第 10 章详细介绍了 JavaBean 的概念和编程思想并给出了相关实例；第 11 章讨论了如何开发 Java Web 应用，包括 JSP 和 Servlet 的语法及编程思想。

本书的组织编排特别适合于软件开发工程师自学如何使用 JBuilder 9.0 开发 Java 应用程序，也适合作大专院校 Java 课程的教材或教辅材料。

南京财经大学电子商务系老师对本书的构思和出版给予了大力支持，很多老师根据自己的教学和开发经验，提出了很多很好的意见和建议，在此对他们表示由衷的谢意。本书由冯军、程超和卫欣组织编写，其他对本书的编写和完稿作出贡献的还有：龚波、葛丽、田军、韩存兵、赵欣胜、张溟、龚志翔、李红玲、白红利、陈磊、冯军、刘砚、李志、张巧莉、刘卫宏、姜南、赵军锁、罗贤锋等。由于编者水平有限，书中难免有错误与不足之处，欢迎读者批评指正。

编者

2004 年 2 月

目 录

前言

第 1 章 JBuilder 入门 1

1.1 Java 语言简介 1

1.1.1 Java 虚拟机 1

1.1.2 Java 语言特点 2

1.1.3 Java 语言与互联网 4

1.2 JBuilder 9.0 简介 6

1.2.1 JBuilder 9.0 功能简介 6

1.2.2 安装 JBuilder 9.0 8

1.2.3 JBuilder 9.0 应用浏览器环境 10

1.3 本章小结 12

1.4 练习 12

第 2 章 Java 语言基础 13

2.1 一个 Java 应用程序 13

2.1.1 编写第一个 Java Application .. 13

2.1.2 编译第一个 Java Application .. 15

2.1.3 运行第一个 Java Application .. 16

2.1.4 用 JBuilder 9.0 构建具有基本 输入/输出的 Java 程序 17

2.2 数据类型与表达式 24

2.2.1 Java Application 的构成 24

2.2.2 数据类型、变量与常量 26

2.2.3 表达式 29

2.2.4 数组 33

2.2.5 计算器程序 34

2.3 流程控制语句 36

2.3.1 选择语句 36

2.3.2 循环语句 39

2.3.3 跳转语句 44

2.4 类、对象和继承 47

2.4.1 类基础 47

2.4.2 声明对象 49

2.4.3 类的方法 50

2.4.4 方法的重载 53

2.4.5 Java 的参数传递 55

2.4.6 Java 类的访问控制 56

2.4.7 继承 58

2.4.8 Object 类 61

2.5 包和接口 62

2.5.1 包 62

2.5.2 类路径 63

2.5.3 包内的访问控制 63

2.5.4 引入包 64

2.5.5 接口 65

2.6 本章小结 66

2.7 练习 67

第 3 章 Java Applet 68

3.1 第一个 Java Applet 程序 68

3.2 Applet 类 70

3.3 Applet 程序机理 72

3.3.1 Applet 主框架 72

3.3.2 Applet 的生命周期 73

3.3.3 Applet 浏览器 74

3.3.4 Applet 的安全限制 75

3.4 Applet 编程 75

3.4.1 Applet 绘图 75

3.4.2 请求重画 77

3.4.3 HTML APPLET 标记 78

3.5 实例：计算器 Java 小程序 79

3.5.1 实例说明 79

3.5.2 实例实现 79

3.5.3 实例总结 87

3.6 本章小结 87

3.7 练习	87
第4章 图形界面设计	88
4.1 基本组件	88
4.1.1 按钮	88
4.1.2 标签	89
4.1.3 文本框	89
4.1.4 复选框	90
4.1.5 单选按钮	91
4.1.6 选择框	91
4.1.7 滚动条	92
4.2 高级组件	92
4.2.1 表格	93
4.2.2 工具栏	94
4.2.3 树型控件	96
4.3 面板和布局控制	100
4.3.1 面板	100
4.3.2 实例：面板实例	100
4.3.3 布局控制	106
4.4 字体、颜色、图形的绘制	109
4.4.1 颜色	109
4.4.2 字体	110
4.4.3 图形的绘制	111
4.5 窗口、菜单和对话框	113
4.5.1 窗口	113
4.5.2 菜单	114
4.5.3 对话框	116
4.6 实例：图形界面设计综合实例	116
4.6.1 实例说明	116
4.6.2 实例实现	117
4.6.3 实例总结	120
4.7 本章小结	121
4.8 练习	121
第5章 数据库	122
5.1 连接数据库	122
5.1.1 JDBC 基本概念	122
5.1.2 DataExpress 体系结构	127
5.1.3 实例：数据库连接	127

5.2 操作数据库	134
5.2.1 数据库操作组件	134
5.2.2 实例：数据库查询实例	135
5.3 管理数据库	140
5.3.1 数据库备份和恢复	140
5.3.2 导入和导出数据	141
5.3.3 实例：数据库备份和恢复	143
5.4 本章小结	149
5.5 练习	150
第6章 多媒体	151
6.1 Java 多媒体框架	151
6.1.1 JMF 历史	151
6.1.2 下载和安装 JMF	153
6.1.3 JMF 体系结构	156
6.2 音频开发	159
6.2.1 声音文件	159
6.2.2 Java Sound 结构	160
6.2.3 音频文件的播放	162
6.3 实例：MP3 播放器	165
6.3.1 实例说明	165
6.3.2 实例实现	165
6.3.3 实例总结	168
6.4 视频开发	169
6.4.1 视频文件	169
6.4.2 视频播放	171
6.5 实例：视频播放器	171
6.5.1 实例说明	171
6.5.2 实例实现	172
6.5.3 实例总结	177
6.6 本章小结	179
6.7 练习	179
第7章 输入输出	180
7.1 理解 Java I/O	180
7.1.1 流的概念	180
7.1.2 字节流和字符流	180
7.1.3 预定义流	182
7.2 基本 I/O	183

7.2.1	向控制台输出.....	183
7.2.2	读取控制台输入.....	183
7.3	文件 I/O	185
7.3.1	File 类	185
7.3.2	目录	187
7.3.3	FilenameFilter 接口	188
7.3.4	listFiles()方法	189
7.3.5	创建目录	190
7.4	字节流	190
7.4.1	InputStream 类.....	190
7.4.2	OutputStream 类	190
7.4.3	FileInputStream 类.....	191
7.4.4	FileOutputStream 类.....	192
7.4.5	ByteArrayInputStream 类.....	194
7.4.6	ByteArrayOutputStream 类.....	194
7.4.7	过滤字节流.....	195
7.4.8	缓冲字节流.....	196
7.4.9	PrintStream 类	197
7.4.10	RandomAccessFile 类.....	197
7.5	字符流	198
7.5.1	Reader 类	198
7.5.2	Writer 类.....	199
7.5.3	FileReader 类	199
7.5.4	FileWriter 类.....	200
7.5.5	CharArrayReader 类	201
7.5.6	CharArrayWriter 类	202
7.5.7	BufferedReader 类	203
7.5.8	BufferedWriter 类.....	204
7.5.9	PushbackReader 类.....	204
7.5.10	PrintWriter 类	205
7.6	实例: I/O 应用实例	205
7.6.1	实例说明	205
7.6.2	实例实现	205
7.6.3	实例总结	214
7.7	本章小结.....	215
7.8	练习	215
第 8 章	多线程	216

8.1	理解 Java 多线程	216
8.1.1	线程与进程	216
8.1.2	Java 线程模型	217
8.1.3	线程优先级	217
8.1.4	线程间的同步	218
8.1.5	消息传递.....	218
8.1.6	Thread 类和 Runnable 接口.....	218
8.2	线程的生命周期	219
8.2.1	创建线程	219
8.2.2	启动线程	220
8.2.3	线程调度	220
8.2.4	终止线程	221
8.2.5	延迟线程	222
8.3	实现多线程	222
8.3.1	主线程	222
8.3.2	创建线程	224
8.3.3	创建多线程	227
8.3.4	使用 isAlive()和 join().....	228
8.3.5	线程优先级	231
8.3.6	线程同步	233
8.3.7	挂起、恢复和终止线程	237
8.3.8	使用多线程机制	241
8.4	实例: 多线程应用实例	241
8.4.1	实例说明	241
8.4.2	实例实现.....	241
8.4.3	实例总结	250
8.5	本章小结	250
8.6	练习	250
第 9 章	网络通信.....	251
9.1	理解 TCP/IP 网络通信	251
9.1.1	套接字	251
9.1.2	网络地址与端口号	252
9.1.3	客户机/服务器模式	252
9.1.4	Internet 编址.....	253
9.2	Java 网络类和接口	254
9.2.1	InetAddress 类.....	254
9.2.2	URL 类.....	256

9.2.3	URLConnection 类	258	10.3	JavaBean 事件	294
9.2.4	Java 网络模型	259	10.4	实例: JavaBean 实例	295
9.3	面向连接的套接字通信	259	10.4.1	实例说明	295
9.3.1	TCP 客户套接字	259	10.4.2	实例实现	296
9.3.2	TCP 服务器套接字	260	10.4.3	实例总结	312
9.3.3	面向连接的 Socket 通信 过程	261	10.5	本章小结	313
9.3.4	一个简单的 TCP 服务器/ 客户机程序	262	10.6	练习	313
9.3.5	实例: TCP 通信应用	267	第 11 章 Web 应用	314	
9.4	无连接的数据报通信	280	11.1	用 J2EE 开发 Web 应用的几种 主要方式	314
9.4.1	DatagramPacket 类与 DatagramSocket 类	281	11.1.1	直接使用 JSP	314
9.4.2	无连接的 Datagram 通信过程	281	11.1.2	JSP + JavaBeans	315
9.4.3	一个简单的 UDP 服务器/ 客户机程序	282	11.1.3	JSP+JavaBeans+Servlet	315
9.4.4	实例: UDP 通信应用	284	11.1.4	J2EE 开发模型	315
9.5	本章小结	291	11.2	Servlet 开发	316
9.6	练习	291	11.2.1	理解 Servlet	316
第 10 章 JavaBean	292		11.2.2	Servlet 编程	318
10.1	理解 JavaBean	292	11.2.3	实例: Servlet 实例	320
10.2	JavaBean 属性	293	11.3	JSP 开发	328
10.2.1	简单属性	293	11.3.1	理解 Java Server Page	328
10.2.2	可索引属性	294	11.3.2	JSP 工作原理	328
10.2.3	绑定性属性	294	11.3.3	JSP 编程	329
10.2.4	限制性属性	294	11.3.4	实例: JSP 实例	333
			11.4	本章小结	337
			11.5	练习	338
			参考文献	339	

第 1 章 JBuilder 入门

知识点

- Java 语言的特点
- 认识 JBuilder 9.0 的特性
- JBuilder 9.0 的安装

JBuilder 是 Borland 公司开发的一个功能强大的跨平台的可视化 Java 集成开发环境。它提供的强大功能和友好的用户界面使 Java 程序员从烦琐的界面设计和重复的代码输入中解脱出来,能够快速高效地构建符合业界标准的 Java 应用系统,包括:Java 小应用程序、EJB、Web、XML 以及数据库等各类应用程序。借助 JBuilder 的双向、可视化设计工具,Java 程序员可以快速地开发各种 J2EE 应用程序,并部署到多种应用服务器。JBuilder 9.0 是其目前最新发布的版本。

1.1 Java 语言简介

Java 是 Sun 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,是当今风靡全球的网络编程语言。在介绍 Java 编程语言的有关概念之前,有必要先介绍一下 Java 虚拟机。

1.1.1 Java 虚拟机

Java 虚拟机 (JVM, Java Virtual Machine) 是一个抽象的概念,主要是为开发者提供一个与众不同的独立的 Java 平台。JVM 可以看作是一台计算机,而这台计算机具有特殊的机器语言可以运行 Java 程序。它是一个以堆栈为基础的 32 位机。机器语言指令称为字节码 (bytecodes)。这些字节码包括数值进栈指令、数值运算指令及数值出栈指令等。JVM 还有一个特殊的内存区域,相当于微处理器的寄存器区。这种方法的优点是多数计算机都可以被模拟为 JVM 并实现其功能。

字节码主要是由用 Java 语言编写的源程序经过编译而得到的,除此之外,还有其他高级语言及汇编语言的编译器。图 1-1 描述了高级源代码、编译器、JVM 之间的关系。

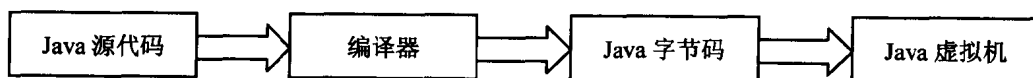


图 1-1 Java 虚拟机与 Java 语言

JVM 为用户提供 4 种主要功能:

1. 字节码的装载

当某 Java 程序需要调用其他类或者库中的代码时, JVM 负责从磁盘或者通过网络加载该程序需要的字节码。如果其他类需要调用刚才所访问类中的某函数时, JVM 能够知道需要的代码已经加载了。JVM 还可以有选择地卸载一些代码, 从而节省内存空间。

2. 内存垃圾收集

编程中所遇到的最常见的问题就是对内存的管理和分配。程序员经常在使用了内存之后忘记或改变了该块内存的指针, 从而不知不觉地生成内存漏洞。

使用了 Java 问题就会有所改变。JVM 会自动地跟踪每一个对象的标识, 当发现已经没有任何标识指向某个对象时, JVM 就会自动地释放该对象所使用的内存。

3. 安全性

Java 最引人入胜的地方也许就是它的内置安全特性。当字节码加载进 JVM 后, 代码将经过原理校验器的检测, 这个校验器将检测代码的安全性。尤其要检测以下几点内容:

(1) 代码是否访问未初始化的内存。

(2) 是否执行了非法的数据类型转换。举例来说, JVM 不允许将整型数据用作指针变量。

(3) 程序是否存在堆栈溢出(包括上溢出和下溢出)。

该校验器不仅使 Java 程序提高了安全性, 而且使程序更加健壮, 并大大减小了系统崩溃的可能性。总之, 一个程序如果不满足以上提到的三点检测要求, 那么它要么是违反安全限制的程序, 要么就是存在许多毛病(bug)的程序。

Java 的安全性还可以阻止一些不确定的程序访问本地存储器, 如硬盘, 而且可以阻止该程序任意访问网络。

4. 本地代码访问

许多操作系统拥有一些在 Java 标准中并未定义的性能。举例来说, Java 并不直接支持一些硬件设备, 如串行端口、音频数字转换器及特殊的 I/O 设备。为解决这一问题, JVM 可以调用本地代码, 即这些代码充当适用于本系统的微处理器。有一点开发者应当注意, 应当尽可能少地使用本地代码。因为如果一个应用程序使用了本地代码, 那么该应用程序将失去 Java 程序的相对于平台的独立性。

1.1.2 Java 语言特点

Java 是定位于网络计算的计算机语言, 它几乎所有的特点都是围绕着这一中心展开并为之服务的, 这些特点使得 Java 语言特别适合于用来开发网络上的应用程序; 另外, 作为一种面世较晚的语言, Java 也集中体现和充分利用了若干当代软件技术新成果, 如面向对象、多线程等, 这些也都在它的特点中有所反映。

1. 简单易学

Java 的设计目的是让专业程序员觉得既易学又好用。假设您有编程经历, 您将不觉得 Java 难掌握。如果您已经理解面向对象编程的基本概念, 学习 Java 将更容易。如果您是一名经验丰富的 C++ 程序员, 那就更好了, 学习 Java 简直不费吹灰之力。因为 Java 继承 C/C++

语法和许多 C++ 面向对象的特性, 大多数程序员在学习 Java 时都不会觉得太难。另外, C++ 中许多容易混淆的概念, 或者被 Java 弃之不用了, 或者以一种更清楚、更易理解的方式实现。除了和 C/C++ 类似以外, Java 的另外一个属性也使它更容易学习, 即设计人员努力使 Java 中不出现显得让人吃惊的特性。在 Java 中, 很少明确地告诉您如何才能完成一项特定的任务。

2. 面向对象

Java 是面向对象的编程语言。面向对象技术较好地适应了当今软件开发过程中新出现的种种传统面向过程语言所不能处理的问题, 包括软件开发的规模扩大、升级加快、维护量增大以及开发分工日趋细化、专业化和标准化等, 它是一种迅速成熟、推广的软件开发方法。面向对象技术的核心是以更接近于人类思维的方式建立计算机逻辑模型, 它利用类和对象的机制将数据与其上的操作封装在一起, 并通过统一的接口与外界交互, 使反映现实世界实体的各个类在程序中能够独立、自治、继承; 这种方法非常有利于提高程序的可维护性和可重用性, 大大提高了开发效率和程序的可管理性, 从而使得面向过程语言难于操纵的大规模软件可以很方便地创建、使用和维护。C++ 也是面向对象的语言, 但是为了与 C 语言兼容, 其中还包含了一些面向过程的成分; Java 去除了 C++ 中非面向过程的部分, 其程序编写过程就是设计、实现类, 定义其属性、行为的过程。

3. 结构中立

Java 设计者考虑的一个主要问题是程序代码的持久性和可移植性。程序员面临的一个主要问题是, 不能保证今天编写的程序明天能否在同一台机器上顺利运行。操作系统升级、处理器升级以及核心系统资源的变化, 都可能导致程序无法继续运行。Java 设计者对这个问题做过多种尝试, JVM 就是试图解决这个问题的。他们的目标是“只要写一次程序, 在任何地方、任何时间该程序永远都能运行”。在很大程度上, Java 实现了这个目标。

4. 安全稳定

对网络上应用程序的另一个需求是较高的安全可靠。用户通过网络获取并在本地运行的应用程序必须是可信赖的, 不会充当病毒或其他恶意操作的传播者去攻击用户本地的资源; 同时它还应该是稳定的, 轻易不会产生死机现象或其他错误, 使得用户乐于使用。Java 特有的“沙箱”机制是其安全性的保障, 同时它去除了 C++ 中易造成错误的指针, 增加了自动内存管理等措施, 保证了 Java 程序运行的可靠性。

万维网上多平台的环境使得它对程序有特别的要求, 因为程序必须在许多系统上可靠地执行。这样, 在设计 Java 时, 创建健壮的程序被放到了高度优先考虑的地位。为了获得可靠性, Java 在一些关键的地方加以限制, 强迫您在程序开发过程中及早发现错误。同时, Java 使您不必担心引起编程错误的许多最常见的问题。因为 Java 是一种严格的类型语言, 它不但在编译时检查代码, 而且在运行时也检查代码。事实上, 在运行时经常碰到的难以重现的、难以跟踪的许多错误在 Java 中几乎是不可能产生的。要知道, 使程序在不同的运行环境中以可预见的方式运行是 Java 的关键特性。

为了更好地理解 Java 是如何具有健壮性的, 让我们考虑使程序失败的两个主要原因: 内存管理错误和误操作引起的异常情况 (也就是运行时错误)。在传统的编程环境下, 内存

管理是一项困难、乏味的任务。例如，在 C/C++ 中，程序员必须手工地分配并且释放所有的动态内存。这有时会导致问题，因为程序员可能忘记释放原来分配的内存，或者释放了其他部分程序正在使用的内存。Java 通过替您管理内存分配和释放，可以从根本上消除这些问题（事实上，释放内存是完全自动的，因为 Java 为闲置的对象提供内存垃圾自动收集）。

在传统的环境下，异常情况可能经常是由“被零除”或“文件未找到”这样的情况引起的，而我们又必须用既繁多又难以理解的一大堆指令来对它们进行管理。Java 通过提供面向对象的异常处理机制来解决这个问题。一个写得很好的 Java 程序，所有的运行时错误都可以并且应该被您的程序自己管理。

5. 支持多线程

多线程是当今软件技术的又一重要成果，已成功地应用在操作系统、应用开发等多个领域。多线程技术允许同一个程序有两个执行线索，即同时做两件事情，满足了一些复杂软件的需求。Java 不但内置多线程功能，而且提供语言级的多线程支持，即定义了一些用于建立、管理多线程的类和方法，使得开发具有多线程功能的程序变得简单、容易和有效。

6. 解释性和高性能

通过把程序编译为 Java 字节码这样一个中间过程，Java 可以产生跨平台运行的程序。字节码可以在提供 JVM 的任何一种系统上被解释执行。早先的许多尝试解决跨平台的方案对性能要求都很高。其他解释执行的语言系统，如 Basic、Tcl、Perl 都有无法克服的性能缺陷。然而，Java 却可以在非常低档的 CPU 上顺利运行。前面已解释过，Java 确实是一种解释性语言，Java 的字节码经过仔细设计，因而很容易便能使用 JIT 编译技术将字节码直接转换成高性能的本机代码。Java 运行时系统在提供这个特性的同时仍具有平台独立性，因而“高效且跨平台”对 Java 来说不再矛盾。

7. 分布式

Java 为 Internet 的分布式环境而设计，因为它处理 TCP/IP 协议。事实上，通过 URL 地址存取资源与直接存取一个文件的差别是不太大的。Java 原来的版本（Oak）包括了内置的地址空格消息传递（intra-address-space）特性。这允许位于两台不同的计算机上的对象可以远程地执行过程。Java 最近发布了叫做远程方法调用（Remote Method Invocation, RMI）的软件包，这个特性使客户机/服务器编程达到了无与伦比的抽象级。

8. 动态

Java 程序带有多种的运行时类型信息，用于在运行时校验和解决对象访问问题。这使得在一种安全、有效的方式下动态地连接代码成为可能，对小应用程序环境的健壮性也十分重要，因为在运行时系统中，字节码内的小段程序可以动态地被更新。

Java 的上述特性不但适应网络应用开发的需求，而且体现了当今软件开发方法的若干新成果和新趋势。

1.1.3 Java 语言与互联网

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。虽然在 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，要求程序的运行与平台无关；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。如果您想播放一种声音或运行一个演示程序，您不得不下载那个文件并用您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 提供了可让您的浏览器运行程序的方法。您能从您的浏览器里直接播放声音，您还能播放页面里的动画。Java 还能告诉您的浏览器怎样处理新的类型文件。当我们能以 2400Baud 传输视频图像时，HotJava 将能显示这个视频。

Internet 使 Java 成为网上最流行的编程语言，同时 Java 对 Internet 的影响也意义深远。原因相当简单，Java 扩展了可以在赛博空间自由流动的对象的世界。在网络中，有两大类对象在服务器和个人计算机之间传输——被动的信息和动态的、主动的程序。例如，当阅读电子邮件时，您在看被动的数据。甚至当您下载一个程序时，该程序的代码也是被动的数据，直到您执行它为止。但是，可以传输到个人计算机的另一类对象却是动态的、可自运行的程序，虽然这类程序是客户机上的活动代理，但却是由服务器来初始化的。例如，被服务器用来正确地显示服务器传送数据的程序。网上程序在动态性上是令人满意的，但它们在安全性和可移植性方面也显示出严重的缺陷。在 Java 产生以前，当前赛博空间有一半的对象实体无法进入网络世界，是 Java 为它们打开了便利之门，而且在这个过程中定义了一种全新的程序形式——Applet（小应用程序）。

1. Java 小应用程序和应用程序

Java 可用来生成两类程序：应用程序（Applications）和小应用程序（Java Applet）。应用程序是可以在您的计算机的操作系统中运行的程序，从这一方面来说，用 Java 编制的应用程序多多少少与使用 C 或 C++ 编制的应用程序有些类似。在创建应用程序时，Java 与其他计算机语言没有大的区别。而 Java 的重要性就在于它具有编制小应用程序的功能。小应用程序是可以在 Internet 中传输并在兼容 Java 的 Web 浏览器中运行的应用程序。小应用程序实际上就是小型的 Java 程序，能像图像文件、声音文件和视频片段那样通过网络动态下载，它与其他文件的重要差别是，小应用程序是一个智能的程序，能对用户的输入作出反应，并且能动态变化，而不是一遍又一遍地播放同一动画或声音。如果 Java 不能解决两个关于小应用程序的最棘手的问题——安全性和可移植性，那么小应用程序就不会如此令人激动。在继续下一个话题之前，先说明一下安全性和可移植性对 Internet 的意义。

2. 安全性

每次下载一个“正常”的程序时，您都要冒着被病毒感染的危险。在 Java 出现以前，大多数用户并不经常下载可执行的程序文件；即使下载了程序，在运行它们以前也都要进

行病毒检查。尽管如此,大多数用户还是担心他们的系统可能被病毒感染。除了病毒,另一种恶意的程序也必须警惕。这种恶意的程序可通过搜索计算机本地文件系统的内容来收集您的私人信息,例如信用卡号码、银行账户账号和口令。Java 在网络应用程序和计算机之间提供了一道防火墙,消除了用户的这些顾虑。当使用 Java 兼容的 Web 浏览器时,您可以安全地下载 Java 小应用程序,不必担心病毒的感染或恶意的企图。Java 实现这种保护功能的方式是,将 Java 程序限制在 Java 运行环境中,不允许它访问计算机的其他部分。下载小应用程序并能确保它对客户机的安全性不会造成危害是 Java 的一个最重要的方面。

3. 可移植性

正如前面所讨论的,许多类型的计算机和操作系统都连接到 Internet 上。要使连接到 Internet 上的各种各样的平台都能动态下载同一个程序,就需要有能够生成可移植性执行代码的方法。很快就可以看到,有助于保证安全性的机制同样也有助于建立可移植性。实际上,Java 对这两个问题的解决方案是优美的也是高效的。

1.2 JBuilder 9.0 简介

如果您选择 JBuilder 作为 Java 程序的开发环境,那么几乎可以说您已经选择了一个功能最为强大的、最容易使用的、程序开发速度最快的 Java 开发工具。在以往,许多软件开发人员在使用快速程序开发工具(RAD)的时候犹豫不决,因为他们担心这样的开发环境及语言会使他们失去程序灵活性及可控性。JBuilder 改变了这些观点,它将 Java 语言的强大功能与带有用户图形接口的快速程序开发环境很好地结合在了一起。

JBuilder 9.0 是 Borland 公司最新版本的可视化 Java 开发工具。它提供了 4 种版本: Personal (个人版)、SE (专业版)、Enterprise (企业版) 以及 Performance Bundle。它们可以满足不同层次的 Java 开发需求。

1.2.1 JBuilder 9.0 功能简介

JBuilder 不仅可以协助您快速开发与部署高品质的 J2EE 应用程序,也可以整合复杂的企业开发流程。最新版的 JBuilder 9.0 包含许多重要的特性。

1. 优秀的 Java 开发解决方案

借助于双向的可视化开发工具, JBuilder 可以提升 EJB、XML、Web Services 以及数据库应用程序的开发效率,并支持绝大多数 J2EE 应用服务器。JBuilder 强化了团队开发效率,让 Java 开发团队能够更快速、更有效地研发 Java 应用程序。JBuilder 允许自由选择开发平台、版本控制系统,以及应用服务器,避免受限于特定厂商或产品。JBuilder 开发人员可以从广大的开发者讨论组中获得许多宝贵资源,定制与扩充 JBuilder 开发环境,并利用 JBuilder 高度弹性的 Open Tools 架构。

该解决方案可以大幅提升 Java 开发团队的生产力,降低开发成本,开发高品质的应用程序,并且快速进入市场。在 JBuilder 集成开发环境中,无论是从设计阶段到部署过程,您都可以完整地掌握应用程序的开发生命周期。新加入的 JBuilder Performance Bundle 完美地结合了最佳的

Java 效率管理工具 Borland Optimizeit Suite, 提供了一个最佳化且易于使用的企业级应用程序解决方案。

2. 跨平台的 Java 开发环境

JBuilder 支持 Windows、Linux 和 Solaris 操作系统, 提供真正的跨平台支持能力。JBuilder 9.0 可安装于下列操作系统:

- Windows 2000、Windows XP、Windows 98 与 Windows NT 4.0
- RedHat 6.2 以上
- Solaris 7 以上

3. 开发符合业界标准的 J2EE 应用程序

JBuilder 9.0 提供完整的 EJB1.1 和 EJB2.0 支持能力。可视化的 EJB Designer 工具能够快速创建可重用的 EJB, 包括: Session Bean、Entity Bean、Message-Driven Bean。JBuilder 提供的可视化 Deployment Descriptor 编辑器, 能够自动建立各种应用服务器所需要的 EJB 及环境参数设定。Entity Bean Modeler 可建立对象与关系型数据库之间的对应关系, 包括数据来源、数据表格、目录、属性和 Entity Bean 的关联性。

4. 支持业界主流 J2EE 应用程序

JBuilder 9.0 支持多种 J2EE 应用服务器, 包括: Borland Enterprise Server、BEA WebLogic Application Server、IBM WespHERE Application Server、Oracle9i Application Server、iPlanet Application Server、Sybase Enterprise Server 和 Sun ONE Application Server。可以在本机或远程执行并调试 EJB, 不需要停止应用服务器即可部署 EJB 组件。

5. 高效率的 Web Services 开发环境

JBuilder 9.0 可协助您快速开发、搜寻、使用与发布 Web Services。JBuilder 9.0 支持最新的 Web Services 技术, 包括 SOAP、WSDL、UDDI 和 WSIL。对于 Web Services 的支持则包含 SOAP 服务器向导、TCP Monitor、WSDL 输入向导、支持 Web Services 的 EJB 向导以及 Web Services Explorer (UDDI 浏览器)。

6. 具有 UML 可视化开发过程与程序代码重构功能

JBuilder 9.0 可以由程序代码自动产生 UML 模型。通过 UML 类图来呈现程序代码, 开发人员更能够掌握应用程序的架构与对象的关联性。您只要单击 UML 类图, 即可检视程序代码的关联性或依存性。如果您在 UML 类别图内变更类别、方法、属性或套件名称, JBuilder 9.0 的程序代码重构功能会自动重建相对应的程序代码。新的程序代码格式工具可以在开发团队中维持一致的程序代码格式。JBuilder 9.0 也允许定制程序开发环境, 例如缩排、Tab 键间隔、大括号位置、事件处理函数, 以及其他项目。

7. 利用企业级的单元测试工具开发高品质的应用程序

JBuilder 9.0 紧密整合了 JUnit, 可针对程序单元进行分析与测试。图形化调试器让本地和远程的调试过程变得较为简单与直观。JBuilder 9.0 远程调试器可以同时多个虚拟机和平台上调式。您可以自由设定断点, 查看并更改变量值, 观察多个线程, 或是使用智能型单步调试功能。借助 HotSwap, 您可以在调试时修改程序代码, 甚至重新定义已经加载内存的类。JBuilder 9.0 甚至可以在本地和远程针对非 Java 语言所编写的程序进行调试。

8. 提升团队开发效率

JBuilder 9.0 提供具备整合性、延展性及扩充性的团队开发环境, 支持 Borland TeamSource DSP (Developer Services Platform)、Rational ClearCase、CVS 与 Microsoft Visual Sourcesafe 等程序代码及文件版本管理系统, 有效简化开发团队管理程序代码的复杂度。JBuilder 9.0 的历史记录浏览器可以动态比对目前项目文件与储存库内程序代码是否冲突, 并突显出程序代码更新部分、相异处, 或是互相冲突的地方。

9. 快速构建和调试 Web 应用程序

JBuilder 9.0 内建许多辅助向导及可视化工具, 可协助您快速开发以 Web 为基础、精简型客户端的 Java 应用程序。JBuilder 9.0 的 Archive Builder 向导可为您建立 WAR 文件格式, 轻松部署 Web 应用程序。图形化调试器可以针对 JSP 进行调试, 包括本地或远程的 Servlets。JBuilder 9.0 支持 Apache Struts, 提供 HTML 和 Java 程序代码清楚的隔离功能, 而且明确定义了 Web 可视化功能, 以及企业逻辑程序代码之间的架构。JBuilder 9.0 允许您转换现有的 HTML 或是 JSP, 成为特定 Struts 标记, 并提供 JSP 标记库的功能。

10. 创建专业级数据库应用程序

JBuilder 9.0 内置具备数据感知能力的 dbSwing 组件, 包含了树状的结构、清单、表格等多种进阶功能组件, 可协助您轻易构建 Java 数据库应用程序。数据库 dbSwing 组件完整支持 Swing 的数据模型。此外, DataExpress 与 InternetBeans 组件搭配最新的 EJB 技术, 可以快速建构面向数据的动态 Web 应用程序, 让您更容易地结合企业逻辑与 Web 应用程序。JBuilder 9.0 可以在 XML 和数据库文件之间轻易转移数据。JBuilder 9.0 内置的 JdataStore 是完全以 Java 编写的关系型数据库, 拥有以下特性: 执行效率高、占用空间小、不需组态管理, 可运用于 Web 应用程序、移动式或嵌入式数据库应用程序。Database Pilot 可用来管理数据库对象, 例如表格或存储过程。

11. 创建专业级数据库应用程序

JBuilder 9.0 支持最新的 Java 标准, 包括 Java 2、Java 2 Swing/JFC、XML、Java2D、Java Collections、Message Queue、Accessibility API、JavaBean、JDBC、Enterprise JavaBean、JSP/Servlet、Serialization、Inner Class、RMI、JNI、Java Archives 等。JBuilder 9.0 完整支持 JDK1.4, 其 JDK 切换功能允许开发人员自行选择各项目所使用的 JDK 版本。在编译 Java 项目时, 您也可以选择外部 Java 编译器或是 JBuilder 9.0 内置的编译器。

12. JBuilder MobileSet 支持移动计算

JBuilder 9.0 和 JBuilder MobileSet 可用来开发移动计算应用程序, 并提供仿真器与调试功能。MobileSet 完全整合于 JBuilder 开发环境, 您可以利用 MIDP 和 CLDC 来开发和部署 J2ME 应用程序。最新的 MobileSet 包含 J2ME Wireiess Toolkit, 并支持 Nokia Developer's Suite、Siemens Mobility Toolkit 和 Sprint PCS Wireless Toolkit。JBuilder MobileSet 还支持 Over the Air (OTA), 以及更多功能。

1.2.2 安装 JBuilder 9.0

在 Windows 下安装 JBuilder 9.0, 系统最低要求如下:

- Intel Pentium II 233MHz 或兼容处理器。
- Microsoft Windows 2000 (SP2) /XP, 或 NT4.0 (SP6)。
- 内存最少 128MB, 推荐使用 256MB。
- 700MB 可用硬盘空间 (取决于安装多少特性)。

本书只介绍 Windows 操作系统下 JBuilder 9.0 的安装和使用, 至于其他工具的安装请参阅相关的帮助文件。

JBuilder 9.0 企业版的安装比较简单, 步骤如下:

(1) 将 JBuilder 9.0 的安装盘放入光驱, 随即安装程序会自动运行, 出现如图 1-2 所示的安装界面。如果安装程序未能自动运行, 可以直接双击执行 JBuilder 9.0 安装盘根目录下的安装文件。

(2) 在如图 1-2 所示的安装界面中, 单击 Borland JBuilder 9.0 Enterprise 按钮进入安装配置界面, 如图 1-3 所示。

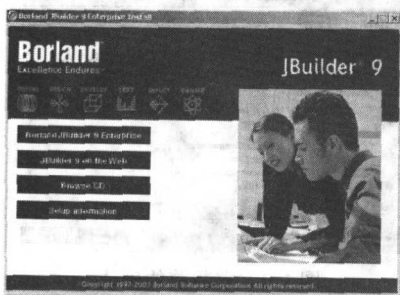


图 1-2 JBuilder 9.0 安装界面

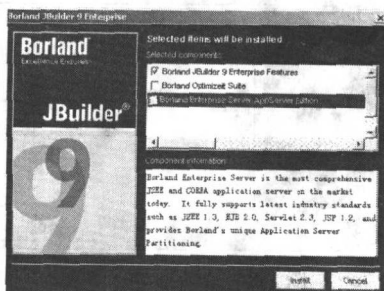


图 1-3 JBuilder 9.0 安装配置界面

(3) 选择所要安装的组件, 单击 Install 按钮开始安装前的初始化。初始化完成之后进入安装向导的第 1 步, 如图 1-4 所示。

(4) 在如图 1-4 所示的界面中单击 Next 按钮进入下一步, 如图 1-5 所示。

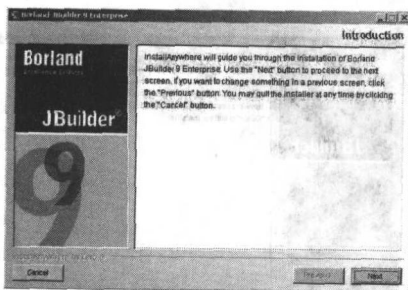


图 1-4 安装向导第 1 步

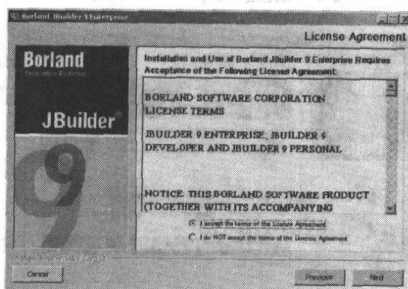


图 1-5 安装向导第 2 步

(5) 选择 I accept the license agreement 复选框, 然后单击 Next 按钮进入第 3 步, 如图 1-6 所示。

(6) 选择安装模式: 完全安装、最小安装和自定义安装。这里我们选择自定义安装。单击 Custom 前的图标, 并单击 Next 按钮, 进入安装向导的第 4 步, 如图 1-7 所示。

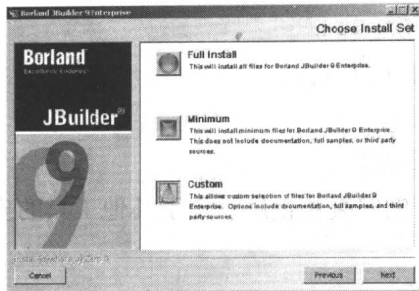


图 1-6 安装向导第 3 步

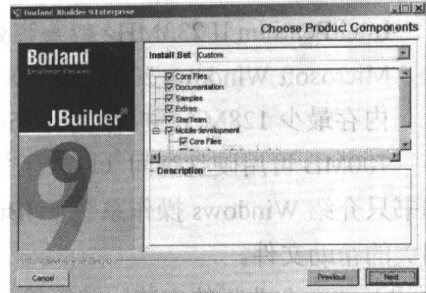


图 1-7 安装向导第 4 步

(7) 选择所要安装的内容。单击 Next 按钮，进入安装向导的第 5 步，如图 1-8 所示。

(8) 设置 JBuilder 9.0 的安装路径。单击 Next 按钮，进入安装确认界面，如图 1-9 所示。

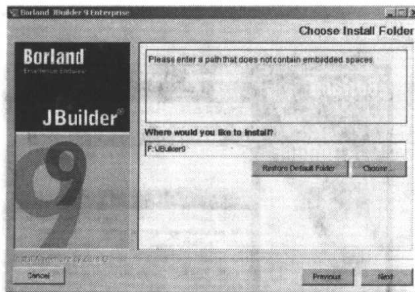


图 1-8 安装向导第 5 步

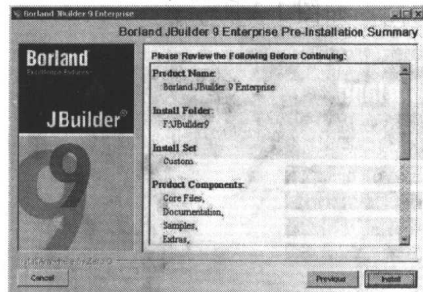


图 1-9 安装前的确认

(9) 单击 Install 按钮，开始安装，如图 1-10 所示。

(10) 安装结束时会出现如图 1-11 所示的界面，单击 Done 按钮，安装完毕。

安装完成后，在用户首次打开 JBuilder 9.0 时，会弹出用户注册窗口。用户可按照说明依次执行即可。这里就不再介绍了。

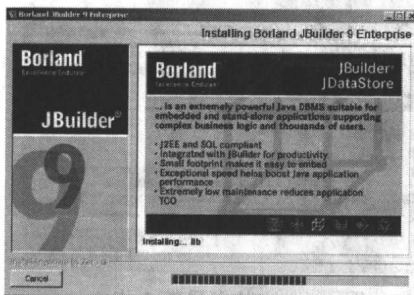


图 1-10 安装过程中

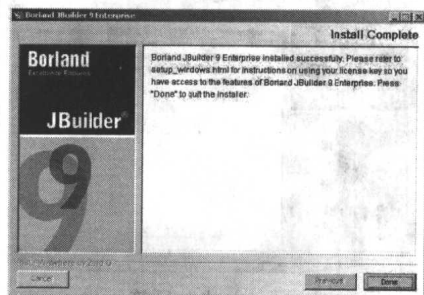


图 1-11 安装完毕

1.2.3 JBuilder 9.0 应用浏览器环境

在 JBuilder 集成开发环境中，可以用应用浏览器窗口来处理大多数的开发工作。应用浏览器提供了许多便利的功能，如建立、编辑、管理文件和项目。也可以通过按钮、工具栏、菜单以及其他工具来编辑、编译、调试和执行 Java 应用程序。图 1-12 就是 JBuilder 9.0