

精通

JBUILDER

飞思科技产品研发中心 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

精通系列

精通 JBuilder 9

飞思科技产品研发中心 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书在当前 J2EE 项目的红火之际, 针对最新版的 JBuilder 9 的特点, 用深刻的理论和详实的例子展示了 Java 语言的巨大威力、JBuilder 9 的高效快捷, 以及 J2EE 的高级架构。全书共分为 15 章, 包括 JBuilder 9 的新特点、Java 语言的基本特点、JBuilder 9 的可视化组件开发、Java 语言的数据库开发、JSP 及 Servlet 的开发、Web Service 的开发、Struts 框架的 Web 应用开发, 以及 J2EE 项目的开发, 使读者对于 Java 语言本身有一个比较深刻的理解和掌握, 并且能够运用 JBuilder 9 开发工具熟练高效地进行项目开发, 以及应用 J2EE 设计模式进行 J2EE 项目架构的搭建。书中带有很多实例, 这无论是对于初学者的学习还是熟练者的提高都具有很大的帮助。

本书适合于专业和准专业的 Java 程序员阅读, 也可作为正在转向进行 Java 开发的各类程序员的必备 Java 参考书。

未经许可, 不得以任何方式复制或抄袭本书的部分或全部内容。

版权所有, 侵权必究。

图书在版编目 (CIP) 数据

精通 JBuilder 9 / 飞思科技产品研发中心编著. —北京: 电子工业出版社, 2003.8
(精通系列)

ISBN 7-5053-8963-7

I. 精... II. 飞... III. Java 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 065735 号

责任编辑: 郭 晶 王 蒙

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 35.75 字数: 915.2 千字

版 次: 2003 年 8 月第 1 版 2003 年 8 月第 1 次印刷

印 数: 6 000 册 定价: 49.00 元

凡购买电子工业出版社的图书, 如有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系调换。联系电话: (010) 68279077

前 言

关于本丛书

精通系列是电子工业出版社经典的技术精品丛书，一直受到广大读者，特别是计算机专业技术人员的关注。在这些专业人士的支持和鼓励下，精通系列已经成为一个标杆，反映出目前国内外最新计算机技术的动态和发展方向。

精通系列中的每部著作完全是作者的呕心沥血之作，代表了作者在该领域的最高成就，集成了作者多年的理论和实践经验，突出了作者为计算机技术的发展做总结和展望的写作初衷。丛书的作者由著名高校的学科带头人、国际上知名的教授学者、权威的计算机专业人士和业界的集大成者组成。他们的知识结构、理论和实践体系有着突出的卓越之处：

- 站在技术的最前沿。
- 有最深刻的理论基础。
- 实践环境具有广泛的代表性和权威性。
- 结论的指导价值。

因此，这样雄厚的写作班子保证了本系列丛书的高层次、高质量和高品质，也足以满足国内读者的高品位、高需求和高要求。除了作者之外，审校者同样实力超群，他们从理论的角度、读者需求的角度、技术成熟度的角度等不同的侧面，为作者提出了大量的参考意见和修改建议，使得每部著作的结构更坚实，体系更完整，理论更完善，针对性更强。

电子工业出版社计算机研发部本着服务于读者、服务于科技的精神，在选题上精益求精，综合考虑和平衡了目前技术上的热点、未来发展的重点及潜在读者需求的卖点等多方面因素，精心推出此精通系列，并将不断进行补充。

关于本书

随着软件构架日益庞大且呈现分布化，软件的组件化和 Web 服务尤显重要，B/S 构架带来的基于 Web 的组件技术，如 Sun 的 EJB、Microsoft 的 DCOM 等技术也迅速发展起来。Java 语言的应用越来越广泛，J2EE 开发平台在众多项目中的采用和成功，促使很多人疯狂地学习 Java 语言，同时很多 Java 应用程序开发者也在不断地提高自己的语言水平及实践经验。在实际的软件开发过程中，熟练地使用一个优秀的集成开发工具将会极大地提高软件开发质量和软件开发效率。为了给大家展现一个更好的学习和提高的途径，我们编写了这本书。

本书针对最新版的 JBuilder 9 的特点，在当前 J2EE 项目的红火之际，用深刻的理论和详实的例子展示了 Java 语言的巨大威力、JBuilder 9 的高效快捷，以及 J2EE 的高级架构。全书共分为 15 章，包括 JBuilder 9 的新特点、Java 语言的基本特点、JBuilder 9 的可视化组件开发、Java 语言的数据库开发、JSP 及 Servlet 的开发、Web Service 的开发、Struts 框

架的 Web 应用开发, 以及 J2EE 项目的开发, 使读者对于 Java 语言本身有一个比较深刻的理解和掌握, 并且能够运用 JBuilder 9 开发工具熟练高效地进行项目开发, 以及应用 J2EE 设计模式进行 J2EE 项目架构的搭建。书中带有很多实例, 这无论是对于初学者的学习还是熟练者的提高都具有很大的帮助。

本书具有如下特点:

- 理论细致深刻
- 实例丰富详实
- 涉及知识面广
- 知识层次具有阶梯性, 循序渐进
- 排版清晰, 便于阅读

本书适合于专业和准专业的 Java 程序员阅读, 也是正在转向进行 Java 开发的各类程序员的必备 Java 参考书。

本书由飞思科技产品研发中心策划并组织编写, 由刘晓华先生对全书进行统稿。李建平、左学明参与了本书的具体写作, 李文强、顾玉春、陈量负责收集相关资料, 吴宾、林涛、王嘉兴、黄晓东、高俊敏编写了本书中的部分示例, 李金永、阎旁勇、陈宏伟、张金旗、张明对书中的代码进行了测试, 范瑾出色的文字润色工作也使本书增色不少。特别感谢欧阳晃兵、何青、黄常珍、李娜, 他们对本选题提出了许多建设性意见。

我们的联系方式:

咨询电话: (010) 68134545 68131648

答疑邮件: support@fecit.com.cn

网 址: <http://www.fecit.com.cn> <http://www.fecit.net>

答 疑: <http://www.fecit.com.cn> 的“问题解答”专区

下 载: <http://www.fecit.com.cn> 的“下载专区”

通用网址: 计算机图书、FECIT、飞思教育、飞思科技、飞思

飞思科技产品研发中心

目 录

第1章 Java 与 Java 开发工具.....1	
1.1 Java 简介.....1	
1.1.1 Java 语言.....1	
1.1.2 Java 程序.....2	
1.1.3 Java 运行环境.....4	
1.1.4 Java 技术的应用状况.....7	
1.2 Java 开发工具概述.....8	
1.2.1 文本编辑器.....8	
1.2.2 Web 开发工具.....9	
1.2.3 集成开发环境.....9	
1.3 搭建基本的 Java 开发环境.....10	
1.3.1 安装 JDK.....10	
1.3.2 设置环境变量.....12	
1.3.3 JDK 开发工具包.....13	
1.4 编写简单的 Java 程序.....18	
1.4.1 基本开发流程.....18	
1.4.2 命令行应用程序.....19	
1.4.3 Java 小程序.....20	
1.4.4 GUI 应用程序.....22	
1.5 本章小结.....22	
第2章 JBuilder 9 入门.....23	
2.1 JBuilder 简介.....23	
2.1.1 JBuilder 的主要特性.....23	
2.1.2 JBuilder 9 安装指南.....24	
2.2 JBuilder 9 集成开发环境.....25	
2.2.1 应用浏览器.....25	
2.2.2 主菜单与工具栏.....26	
2.2.3 工程面板.....27	
2.2.4 结构面板.....27	
2.2.5 内容面板.....27	
2.2.6 消息面板.....28	
2.2.7 调试器.....28	
2.2.8 状态栏.....29	
2.2.9 查询窗体.....29	

2.3 从 JDK 过渡到 JBuilder.....30	
2.3.1 了解 JBuilder 工程.....30	
2.3.2 新建命令行应用程序.....32	
2.3.3 移植 Java 小程序.....35	
2.3.4 体验代码预示功能.....38	
2.4 本章小结.....40	
第3章 Java 语言基础.....41	
3.1 Java 语言中的数据类型.....41	
3.1.1 标识符.....41	
3.1.2 关键字.....42	
3.1.3 数据类型概述.....43	
3.1.4 整型数据.....46	
3.1.5 浮点型(实型)数据.....46	
3.1.6 字符型数据.....47	
3.1.7 布尔型数据.....48	
3.1.8 举例.....48	
3.1.9 变量类型间的相互转换及 混合运算.....49	
3.1.10 常量.....51	
3.1.11 变量.....51	
3.2 Java 语言中的运算符和表达式.....52	
3.2.1 运算符和表达式概述.....52	
3.2.2 算术运算符和表达式.....53	
3.2.3 关系运算符和表达式.....56	
3.2.4 布尔运算符和表达式.....57	
3.2.5 移位运算符和表达式.....58	
3.3 Java 语言中的逻辑和流程控制.....59	
3.3.1 条件分支语句.....59	
3.3.2 循环语句.....65	
3.3.3 注释.....69	
3.4 Java 语言中的数组.....69	
3.4.1 一维数组.....70	
3.4.2 二维数组和多维数组.....72	
3.5 Java 语言中的面向对象思想.....74	

3.5.1 面向对象程序设计的基本概念和特征	74	5.2.3 画布 (Canvas)	119
3.5.2 类	75	5.2.4 复选框 (Checkbox)	119
3.5.3 构造方法和 finalizer	78	5.2.5 单选按钮 (Radio Buttons)	120
3.5.4 重写和重载	80	5.2.6 选择菜单 (Choice Menu)	121
3.5.5 Java 的包	81	5.2.7 列表框 (Scrolling List)	122
3.5.6 Java 的接口	83	5.2.8 单行文本输入框 (TextField)	122
3.6 本章小结	86	5.2.9 多行文本输入框 (TextArea)	123
第 4 章 JBuilder 应用开发基础	87	5.2.10 滚动条 (Scrollbar)	124
4.1 工程与工程组	87	5.3 布局管理	124
4.1.1 JBuilder 的工程管理机制	87	5.3.1 FlowLayout	125
4.1.2 类库路径配置	91	5.3.2 GridLayout	125
4.1.3 Java 应用程序实例	94	5.3.3 GridBagLayout	126
4.1.4 JBuilder 工程组	103	5.3.4 BorderLayout	128
4.2 JBuilder 代码编辑器	103	5.3.5 CardLayout	129
4.2.1 JBuilder 编辑器视图	103	5.3.6 综合例子	131
4.2.2 源代码视图	104	5.4 JBuilder 的可视化设计	133
4.2.3 查询与替换	105	5.4.1 UI 设计器	133
4.2.4 视图分隔	105	5.4.2 UI 设计	138
4.2.5 代码格式化	107	5.5 Swing 组件概述	148
4.2.6 JavaDoc 快捷键	108	5.5.1 轻量组件与重量组件	148
4.2.7 配置编辑器	109	5.5.2 Swing 组件	149
4.3 编译、运行、调试与部署	110	5.6 Swing 组件的体系结构	151
4.3.1 编译程序	110	5.6.1 典型的“模型-视图-控制器”体系结构	151
4.3.2 创建系统	111	5.6.2 Swing MVC	152
4.3.3 运行程序	112	5.7 Swing 可视化设计	153
4.3.4 调试程序	113	5.7.1 框架 (JFrame)	153
4.3.5 部署程序	115	5.7.2 菜单条 (JMenuBar)	153
4.4 本章小结	116	5.7.3 菜单 (JMenu)	154
第 5 章 UI 设计与 Swing	117	5.7.4 菜单项 (JMenuItem)	155
5.1 UI 组件	117	5.7.5 弹出式菜单 (JPopupMenu)	156
5.1.1 容器	117	5.7.6 对话框 (JDialog)	158
5.1.2 控制组件	118	5.7.7 文件选择框 (JFileChooser)	159
5.1.3 用户自定义成分	118		
5.2 UI 基本元素	118		
5.2.1 标志 (Label)	118		
5.2.2 按钮 (Button)	119		

5.7.8 滑杆和进度指示条 (JSlider 和 JProgressBar)	160
5.7.9 树 (JTree)	161
5.7.10 表格 (JTable)	164
5.8 Swing 应用设计	166
5.8.1 制作椭圆按钮	166
5.8.2 创建 2D 图形	168
5.9 本章小结	173
第 6 章 JavaBean 的开发	175
6.1 JavaBean 的基本概念	175
6.1.1 什么是 JavaBean 组件	176
6.1.2 JavaBean 和 Java	176
6.2 JavaBean 的设计目标	177
6.3 JavaBean 的属性	178
6.3.1 Simple 属性	178
6.3.2 Indexed 属性	179
6.3.3 Bound 属性	180
6.3.4 Constrained 属性	181
6.4 JavaBean 的事件	182
6.4.1 JavaBean 事件概述	182
6.4.2 事件状态对象	183
6.4.3 事件监听者接口与事件 监听者	183
6.4.4 事件监听者的注册与 注销	184
6.4.5 适配类	185
6.5 JavaBean 的用户化	185
6.5.1 用户化器接口 (Customizer Interface)	186
6.5.2 属性编辑器接口 (Property Editor Interface)	186
6.5.3 BeanInfo 接口	187
6.6 JavaBean 的持久化	187
6.7 JavaBean 的编写准则	188
6.8 JavaBean 的 JBuilder 可视化 开发	191
6.8.1 创建一个 JavaBean 类	191
6.8.2 为 JavaBean 进行 UI 设计	193

6.8.3 为 JavaBean 添加简单 属性	193
6.8.4 修改和删除 JavaBean 属性	196
6.8.5 为 JavaBean 添加 Bound 或 Constrained 属性	196
6.8.6 BeanInfo 设计器	197
6.8.7 为 JavaBean 添加事件	199
6.9 本章小结	204
第 7 章 数据库应用开发	205
7.1 JDBC 简介	205
7.1.1 什么是 JDBC	205
7.1.2 JDBC 驱动程序的类型	206
7.1.3 安装 JDBC - ODBC 桥 驱动	206
7.1.4 JDBC API 接口	206
7.2 数据库简介	208
7.2.1 数据类型	208
7.2.2 数据定义语言 (DDL)	209
7.2.3 数据操作语言 (DML)	210
7.3 与数据库的连接步骤	212
7.3.1 装载驱动程序	213
7.3.2 建立连接	213
7.4 连接数据库参考	214
7.4.1 连接 Oracle 8/8i/9i 数据库	214
7.4.2 连接 SQL Server 7.0/2000 数据库	214
7.4.3 连接 DB2 数据库	215
7.4.4 连接 Informix 数据库	215
7.4.5 连接 Sybase 数据库	216
7.4.6 连接 MySQL 数据库	216
7.4.7 连接 PostgreSQL 数据库	217
7.5 JDBC 访问数据库	218
7.5.1 用 JDBC 访问 Access 数据库的例子	218
7.5.2 用 JDBC 访问 SQL Server 2000 数据库的例子	219
7.6 JDBC 的 JBuilder 开发	221

7.6.1	JBuilder 数据库应用	
	程序简介	221
7.6.2	JBuilder 数据库常用	
	组件或类简介	222
7.6.3	向 JBuilder 中添加	
	JDBC 驱动	223
7.6.4	JBuilder 的 DataExpress	
	体系结构	226
7.6.5	JBuilder 数据库开发	
	实例	227
7.7	本章小结	236
第 8 章	Java 应用程序开发实例	237
8.1	图形编辑器整体设计	237
8.1.1	功能分析	237
8.1.2	类图设计	238
8.2	图形编辑器界面设计	238
8.2.1	添加菜单	239
8.2.2	快捷按钮设计	239
8.2.3	元素选择按钮的设计	242
8.2.4	前景及背景按钮	243
8.3	图形编辑器绘图元素的设计	245
8.4	图形编辑器中的对象	
	串行化及保存	248
8.5	图形编辑器的其他操作	250
8.5.1	前景色和背景色的选择	250
8.5.2	Redo 和 Undo 操作	251
8.6	图形编辑器的文档生成	252
8.7	本章小结	254
第 9 章	Web 应用开发概述	255
9.1	Java Web 应用程序简介	255
9.1.1	程序组成	255
9.1.2	程序运行环境	257
9.1.3	Web 应用框架	259
9.2	创建 Web 应用程序	260
9.2.1	WebApp 向导	261
9.2.2	Web 应用程序的属性	268
9.3	本章小结	271

第 10 章	开发 Applet	273
10.1	Applet 简介	273
10.1.1	Applet 的运行原理	273
10.1.2	Applet 的功能特性	274
10.1.3	HelloWorld 示例	274
10.1.4	生命周期	276
10.2	java.applet 包	277
10.2.1	Applet 类	277
10.2.2	接口	279
10.2.3	javax.swing.JApplet 类	279
10.3	使用 JBuilder 开发 Applet	280
10.3.1	Applet 向导	280
10.3.2	创建新的 Applet	280
10.3.3	添加代码	283
10.3.4	运行程序	284
10.4	Applet 编程技巧	284
10.4.1	使用图片	284
10.4.2	使用声音	288
10.4.3	使用字体和参数	289
10.5	Applet 编程实例	294
10.5.1	Applet 小时钟	294
10.5.2	Applet 计算器	297
10.6	本章小结	300
第 11 章	开发 Servlet	301
11.1	Servlet 简介	301
11.1.1	Servlet 工作流程	301
11.1.2	Servlet 的特点	302
11.1.3	Servlet 的应用范围	302
11.1.4	开发环境	303
11.2	Servlet 编程基础	303
11.2.1	Servlet API	303
11.2.2	HttpServlet 类	304
11.2.3	Servlet 示例	305
11.2.4	Servlet 的生命周期	307
11.3	使用 JBuilder 开发 Servlet	308
11.3.1	Servlet 向导	308
11.3.2	使用向导	308
11.3.3	添加代码	311

11.3.4 编译 Servlet 代码	312	12.4.2 Response 对象	379
11.3.5 配置服务器	313	12.4.3 Out 对象	380
11.3.6 测试 HelloServlet	314	12.4.4 Session 对象	381
11.4 Server-Side Includes	314	12.4.5 PageContext 对象	382
11.4.1 什么是 Server-Side Includes	314	12.4.6 Application 对象	382
11.4.2 Servlet 标记的语法	315	12.4.7 Config 对象	383
11.4.3 Server-Side Includes 实例	315	12.4.8 Page 对象	383
11.5 会话管理	319	12.5 使用 JavaBeans	383
11.5.1 会话跟踪	320	12.5.1 什么是 JavaBean	383
11.5.2 管理会话数据	322	12.5.2 非可视化的 JavaBean	384
11.5.3 浏览会话	323	12.5.3 JavaBean 的简单例子	384
11.5.4 Cookies	327	12.5.4 在 JSP 中使用 JavaBeans	385
11.5.5 URL Rewriting	329	12.6 本章小结	388
11.5.6 会话事件	332	第 13 章 应用 Struts 框架	389
11.6 HTML 表单	334	13.1 Struts 简介	389
11.6.1 两种数据传递方式	335	13.1.1 Model 1 与 Model 2	389
11.6.2 表单基础	335	13.1.2 MVC 在 Struts 框架中的 实现	390
11.6.3 表单应用示例	341	13.1.3 Model: 系统状态和 商业逻辑 JavaBeans	392
11.7 本章小结	345	13.1.4 View: JSP 页面和 表示组件	393
第 12 章 开发 JSP	347	13.1.5 Controller: ActionServlet 和 ActionMapping	393
12.1 JSP 简介	347	13.2 Struts 开发环境	393
12.1.1 B/S 应用开发技术	347	13.2.1 基本配置	394
12.1.2 JSP 技术	348	13.2.2 Struts 开发流程	397
12.1.3 JSP 应用	349	13.2.3 使用 JBuilder 9	398
12.1.4 JSP 与 XML	350	13.3 创建 Model 组件	409
12.1.5 JSP 与 ASP、PHP	351	13.3.1 JavaBeans 和范围	409
12.2 JSP 的语法	352	13.3.2 ActionForm Beans	410
12.2.1 JSP 示例	352	13.3.3 系统状态 Beans	411
12.2.2 注释	353	13.3.4 商业逻辑 Beans	411
12.2.3 JSP 指令	354	13.3.5 访问关系数据库	412
12.2.4 JSP 动作标记	357	13.4 创建 View 组件	413
12.2.5 脚本元素	366	13.4.1 国际化消息	413
12.3 使用 JBuilder 开发 JSP	368	13.4.2 表单和 FormBean 的 交互	414
12.3.1 JSP 向导	368		
12.3.2 创建第一个 JSP 页面	371		
12.4 JSP 内置对象	374		
12.4.1 Request 对象	375		

13.4.3 其他的表示技术.....	418	15.2 JNDI.....	469
13.5 创建 Controller 组件.....	419	15.2.1 JNDI 基本概念.....	469
13.5.1 Action 类.....	419	15.2.2 JNDI 架构.....	470
13.5.2 ActionMapping 实现.....	421	15.2.3 JNDI 的技术特点.....	471
13.5.3 Action 映射配置文件.....	421	15.2.4 JNDI 使用范例.....	473
13.5.4 Web 应用程序展开 描述符.....	422	15.3 EJB.....	474
13.6 本章小结.....	424	15.3.1 EJB 类型.....	474
第 14 章 Web Service 应用开发.....	425	15.3.2 EJB 的基本结构.....	476
14.1 Web Service 概述.....	426	15.3.3 EJB 开发者的角色.....	478
14.1.1 什么是 Web Service.....	426	15.3.4 EJB 开发简介.....	479
14.1.2 Web Service 的技术特点.....	429	15.3.5 用 JBuilder 开发 EJB.....	482
14.1.3 UDDI.....	433	15.3.6 Stateless Bean 的开发.....	489
14.1.4 Web 服务描述语言 WSDL.....	433	15.3.7 Stateful Bean 的开发.....	491
14.2 简单对象访问协议 (SOAP).....	434	15.3.8 Entity Bean 的开发.....	494
14.2.1 SOAP 简介.....	434	15.3.9 Message Driven Bean 的 开发.....	504
14.2.2 SOAP 封装.....	437	15.4 JMS.....	507
14.2.3 SOAP 编码.....	438	15.4.1 关于消息系统.....	508
14.2.4 在 HTTP 中使用 SOAP.....	440	15.4.2 JMS 系统简介.....	509
14.2.5 在 RPC 中使用 SOAP.....	441	15.4.3 消息驱动 Bean 简介.....	512
14.3 JBuilder 9 和 WebLogic 的 集成.....	443	15.4.4 创建 JMS 客户端程序.....	514
14.3.1 WebLogic 开发 Domain 的配置.....	443	15.4.5 Pub/Sub.....	516
14.3.2 在 JBuilder 中集成 WebLogic.....	448	15.4.6 PTP.....	517
14.4 Web Service 和 J2EE.....	451	15.5 JavaMail.....	518
14.4.1 用 J2EE 开发 Web Service.....	451	15.5.1 JavaMail 邮件协议类型.....	518
14.4.2 Web Service 的远端 调用.....	455	15.5.2 JavaMail 管理对象.....	518
14.5 本章小结.....	459	15.5.3 JavaMail 的属性和方法.....	520
第 15 章 J2EE 介绍.....	461	15.5.4 JavaMail 的应用实例.....	525
15.1 J2EE 概述.....	461	15.6 J2EE 的设计模式简介.....	534
15.1.1 J2EE 组件概述.....	462	15.6.1 创建型.....	536
15.1.2 J2EE 组件.....	463	15.6.2 结构型.....	538
15.1.3 J2EE 模型.....	466	15.6.3 行为型.....	543
		15.7 设计模式应用实例.....	550
		15.7.1 Command 设计模式 应用.....	550
		15.7.2 Observer 设计模式应用.....	553
		15.8 本章小结.....	561

第 1 章 Java 与 Java 开发工具

Java 是新型的网络编程语言, JBuilder 是优秀的 Java 集成开发环境 (Integrated Development Environment, 简称 IDE)。Java 语言的卓越特性为无数开发人员有所推崇, 越来越多的应用开发采用基于 Java 技术的解决方案。而 JBuilder 作为业界领先的 IDE 工具生产商 Borland 公司的又一力作, 也日益成为广大 Java 开发人员的必备工具软件。

本章作为本书的开篇, 首先简要介绍 Java 语言的基本概念、运行环境和开发工具等基础知识, 为后面用 JBuilder 开发 Java 程序打下基础。

本章的主要内容包括:

- Java 语言的基本概念及其特性
- Java 程序分类及其运行机制
- Java 开发工具概述
- 如何开始编写简单的 Java 程序
- 使用 JBuilder 的预备工作——安装 JBuilder 9

1.1 Java 简介

Java 是一种程序设计语言, 也是一门软件开发技术。Java 的出现导致了程序设计领域的一场革命, 它被誉为 20 世纪出现的最重要的技术之一。本节概述 Java 语言的特性、Java 程序的类别及 Java 平台的运行机制, 帮助读者快速了解 Java 技术体系。

1.1.1 Java 语言

Java 语言诞生于 1991 年, 起初被称为 OAK 语言, 是 Sun 公司为一些消费性电子产品设计的一个通用环境。最初 Java 并不为人们所知, 然而 Internet 的普及彻底改变了 Java 的命运。

和 C、C++、PASCAL、BASIC 等一样, Java 是一门程序设计语言, 人们用它来和计算机交流, 描述自然界的逻辑规律, 指导计算机实现复杂的运算功能。从基本的语法和程序结构来看, Java 与其他程序设计语言 (尤其是 C++) 并无多大区别。Java 语言之所以在如此短的时间内被广大软件开发人员所接受, 是因为它具有诸多其他编程语言所无可比拟的特性。

概括地说, Java 是一种性能优异的、简单的、面向对象的、分布式的、解释的、健壮的、安全的、结构的、中立的、可移植的、多线程的和动态的语言。

Java 语言主要具有如下特性。

1. 平台无关性

平台无关性是指 Java 能运行于不同的平台上。Java 引进虚拟机原理, 并运行于虚拟机, 实现不同平台之间的 Java 接口。使用 Java 编写的程序能在世界范围内共享。Java 的数据类型与机器无关, Java 虚拟机 (Java Virtual Machine) 是建立在硬件和操作系统之上的, 它实现 Java 二进制代码的解释执行功能, 提供不同平台的接口。

2. 安全性

Java 的编程类似 C++, 学习过 C++ 的读者将很快掌握 Java 的精髓。Java 舍弃了 C++ 的指针对存储器地址的直接操作, 程序运行时, 内存由操作系统分配, 这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器, 防止程序的非法访问。

3. 面向对象

Java 吸取了 C++ 面向对象的概念, 将数据封装于类中, 利用类的优点, 实现了程序的简洁性和便于维护性。类的封装性、继承性等有关对象的特性, 使程序代码可以只需一次编译, 然后通过上述特性反复利用。程序员只需把主要精力集中在类和接口的设计上。Java 提供了众多的一般对象的类, 子类可以继承父类的方法。在 Java 中, 类的继承关系是单一的非多重的, 一个子类只有一个父类。Java 提供的 Object 类及其子类的继承关系如同一棵倒立的树形, 根类为 Object 类。Object 类功能强大, 经常会使用到它及其派生的子类。

4. 分布式

Java 建立在 TCP/IP 网络平台上。Java 库函数提供了用 HTTP 和 FTP 协议传送和接收信息的方法, 这使得程序员使用网络上的文件和使用本机文件一样容易。使用 Java 语言和相关技术可以十分方便地构建分布式应用系统。

5. 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 自动回收内存, 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。这些功能特征大大提高了开发 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、字节码校验。

此外, Java 取消了 C 语言的结构、指针、#define 语句、多重继承、goto 语句、操作符重载等不易被掌握的特性, 提供垃圾收集器自动回收不用的内存空间, 这使得程序员学习掌握 Java 语言也相对容易得多。

1.1.2 Java 程序

使用 Java 可以开发从命令行应用程序到图形用户界面应用程序、从桌面应用程序到 Web 应用程序、从小型嵌入式系统到大型分布式企业级应用等各种各样的程序。通常所指的 Java 程序可以分为以下几类:

- 命令行应用程序 (Command-line Application)
- 图形用户界面应用程序 (GUI Application)
- 小程序 (Applet)
- 服务端小程序 (Servlet)

- 服务器页面 (JavaServer Pages, 简称 JSP)
- Web 应用程序
- 嵌入式应用程序
- 企业级应用程序

命令行应用程序是指通过字符输入输出终端控制和显示的应用程序。命令行应用程序集中处理后台运算,对程序的用户界面友好性不做要求,类似于早期的程序员在 DOS 下用 PASCAL、C 语言编写的程序。在 Java 中,只需调用标准系统输入输出类 (System.out, System.out),即可实现一些简单的命令行应用程序。

图形用户界面设计技术大大的增强了软件的界面友好性,使得不懂计算机技术的人们也能十分方便地使用软件。Java 的 AWT (java.awt.*)、Swing (javax.swing.*) 软件包提供了强大的图形用户界面开发功能,使用它们和其他一些辅助的可视化设计软件 (例如 JBuilder),可以十分方便地编写界面友好的应用程序。

Java 小程序 (又称 Applet),它嵌在网页当中运行,能够表现出与桌面 GUI 应用程序一样的效果。Applet 的出现使得早期静态的 HTML 页面具有了动态交互功能,也使得 Java 技术得以迅速传播并为广大程序员所知。现在,虽然已出现 DHTML、XML、ASP、JSP、Flash 等众多动态 Web 开发技术,但是,Applet 的许多优点仍然不为其他技术所具备。在一些场合,程序员还是要使用 Applet 技术实现特定的功能。

与 Applet 相对应的是 Servlet 技术,Servlet 是运行在 Web 服务器端的小程序。Servlet 与 Applet 的不同之处在于:Applet 运行在客户端,Servlet 运行在服务器端。Applet 嵌在网页当中运行,通常以图形用户界面的形式表现;Servlet 在后台执行应用逻辑,动态生成页面并返回给客户端,通常没有图形用户界面。Servlet 技术大大增强了 Web 页面的可交互性。

JSP 技术与 ASP、CGI 技术类似,他们都用来编写动态的 Web 页面,开发 Web 应用程序。JSP 技术出现得最晚,但是受程序员关注的程度却最高。JSP 技术完全基于 Java,使用 JSP 无需掌握其他脚本语言即可开发出丰富多彩的 Web 应用,这是其他技术所做不到的。JSP 页面经过引擎转化以后实际上就是一个 Servlet。

Applet、Servlet、JSP 可以单独执行,也以结合起来开发功能强大、内容丰富的 Web 应用程序。随着网络的普及,越来越多的应用都是基于 Web 的。Java 作为一门网络编程语言,在这方面具有得天独厚的优势。

Java 最早为开发微波炉、PDA 等产品而设计,它在嵌入式应用开发方面自然具有相当的优势。Sun 公司制定的 J2ME 标准是专门针对于嵌入式系统的 Java 开发技术的,有关内容可以参见 Sun 公司的网站或其他书籍,本书对此不做讲解。

Java 最突出的应用领域是在其企业级应用开发方面,Sun 公司制定的 J2EE 标准已经被业界广为采纳,并成为当今主流的大型应用系统的开发技术之一。J2EE 体系包含了一系列技术,例如 EJB、JMS、JavaMail、JNDI、RMI 等。本书将在最后简要介绍这些技术和借助 JBuilder 开发相关程序的方法。

1.1.3 Java 运行环境

Java 程序运行于 Java 平台之上, Java 平台运行于 Windows、Linux、Solaris 等操作系统之上, Java 平台由 Java 虚拟机和 Java 应用编程接口 (API) 组成。Java 虚拟机屏蔽了不同操作系统的差异, Java API 为程序员提供统一的编程接口。程序员一次编写的 Java 程序可以在不同的操作系统上运行。

Java API 和 JVM 把 Java 程序从硬件依赖中分离出来, 从而实现了 Java 程序对操作系统和硬件平台的无关性。

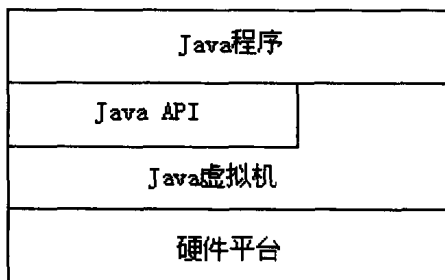


图 1-1 Java 平台的组成

图 1-1 描述了 Java 程序的运行环境。这样的执行架构, 大致上可以分成以下四层。

- 第一层: Java 字节码 (*.class 文件)
- 第二层: Java 虚拟机 (包括 Java 核心 API)
- 第三层: 操作系统
- 第四层: 芯片 (计算机硬件、嵌入式硬件系统等)

芯片执行其上层的操作系统, 操作系统执行再上一层的 Java 虚拟机, Java 虚拟机解释并执行 Java 字节码。Java 虚拟机被看做是 Java 程序的虚拟运行平台。

1.1.3.1 Java 平台

Java 平台由两部分组成:

- Java 虚拟机 (JVM)
- Java 应用编程接口 (Java API)

所谓 Java 虚拟机, 实际上也是运行在操作系统上的一种程序, 它可以解读 Java 字节码 (bytecode), 并调用操作系统的系统函数执行 Java 字节码。其原理如图 1-2 所示。

JVM 是 Java 平台的基础, 它可以移植到各种硬件平台上。Java 虚拟机是程序与底层操作系统和硬件无关的关键。它是一个平台, 将 Java Applets 和应用程序与底层操作系统相分离。通过虚拟机接入到浏览器或其他操作系统是非常容易的。而且, 虚拟机定义了一种与机器无关的被称为类的二进制文件格式。这种格式以字节码的形式存储着用于操作虚拟计算机的指令集。代表任何 Java 语言程序的字节码本质上是一个象征, 它表明方法的索引与偏移量并不是一个常数值, 相反, 是被作为一个字符名字象征性给出的。方法首次被调用

时,首先是在类文件中通过方法的名字寻找它,为了在以后的查找中更加快速找到该方法,它的偏移量的数值大小便在此时确定出来。因此,任何新的或基本的方法可以在运行的早些时候被引入到类框架中的任何地方。它将能够在不打乱原有编码的情况下,被象征性地正确引入。字节码是程序的高级表示形式,以至于程序优化和机器码的产生(通过即时编译器)都可以在这个水平上来完成。而且,由于在 Java 平台的寻址空间中虚拟机将变量存储在堆栈中,因此可以在虚拟机的内部进行垃圾回收处理。

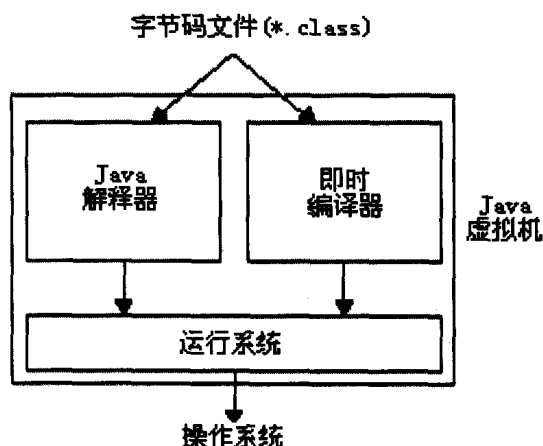


图 1-2 Java 虚拟机

Java API 是软件组件的集合,它们提供了很多有用的功能,如图形用户界面(GUI)、输入输出(IO)、异常处理等。Java API 按相关组件分组为库(Library),或称为包(Package)。

根据应用领域的不同,Java 有许多 API(Application Programming Interface),这些 API 可分成三大类。

- Java Core API: 由 Sun 公司制定的基本 API,任何 Java 平台都必须提供。
- Java Standard Extension API (javax): 由 Sun 公司制定的扩充 API,Java 平台可以选择性地提供或加装。
- 厂商或组织所提供的 API: 由各家公司或组织所提供。

其中 Core API 和 Standard Extension API 已经逐渐涵盖了大部分的信息应用领域,例如多媒体、数据库、Web、企业运算、语音、实时系统、网络、电话、影像处理、加解密、GUI、分布式运算,等等。如果你有某项需求尚未有标准的 Java API 可遵循,你可以向 Sun 公司提出制定新 API 的请求。经过审核之后,你的要求可能会通过或被驳回。如果通过,就会开始制定 API 的程序。Java API 的制定过程因为公开,且经过许多业界技术领先公司的共同参与,所以相当完善且优异。

作为一种独立于平台的环境,Java 比本地代码慢一些。但是,好的编译器、设计优良的解释器和即时字节码编译器可以在不牺牲可移植性的条件下使 Java 程序的执行效率接近于本地代码。

有关 Java 虚拟机的详细信息参见 Sun 公司的 Java 网站: <http://www.javasoft.com> 或 <http://java.sun.com>。

1.1.3.2 Java 平台的分类

目前, Java 2 平台有 3 个版本, 它们是适用于小型设备和智能卡的 Java 2 平台微型版 (Java 2 Platform Micro Edition, 简称 J2ME)、适用于桌面系统的 Java 2 平台标准版 (Java 2 Platform Standard Edition, 简称 J2SE)、适用于创建服务器应用程序和服务的 Java 2 平台企业版 (Java 2 Platform Enterprise Edition, 简称 J2EE)。

1. J2ME

Java 2 平台微型版 (J2ME) 定义了一组针对嵌入式设备和消费电子设备的 API 和运行环境, 这些设备包括无线手持设备、PDA、电视机顶盒, 以及其他缺乏足够的资源去支持 J2SE 的设备。

2. J2SE

Java 2 平台标准版 (J2SE) 是一组针对传统桌面应用的 API 和运行环境。Java 2 平台标准版是多种不同风格软件的开发基础, 包括客户方 Java 小程序和应用程序, 以及独立的服务器应用程序等。Java 2 平台标准版还是 Java 2 平台微型版 (J2METM) 的基础, 又是 Java 2 平台企业版 (J2EETM) 执行程序的推动力。

3. J2EE

J2EE 是一种利用 Java 2 平台来简化与企业解决方案的开发、部署和管理相关的复杂问题的体系结构。J2EE 技术的基础就是核心 Java 平台或 Java 2 平台的标准版, J2EE 不仅巩固了标准版中的许多优点, 例如“编写一次、到处运行”的特性, 方便存取数据库的 JDBC API、CORBA 技术, 以及能够在 Internet 应用中保护数据的安全模式, 等等, 同时还提供了对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 及 XML 技术的全面支持。其最终目的就是成为一个能够使企业开发者大幅缩短投放市场时间的体系结构。

J2EE 体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性及可扩展性的应用的需求。通过提供统一的开发平台, J2EE 降低了开发多层应用的费用和复杂性, 同时提供对现有应用程序集成强有力支持, 完全支持 Enterprise JavaBeans, 有良好的向导支持打包和部署应用, 添加目录支持, 增强了安全机制, 提高了性能。

把 Java 2 平台分成 3 种版本使得它能够更好地满足不同目标领域中开发者的需求, 同时也使 Java 技术在保持其“编写一次, 到处运行”精神的同时在不同领域得到继续发展。

1.1.3.3 Java 程序的运行机制

Java 既可以被编译, 也可以被解释。通过编译器, 可以把 Java 程序翻译成一种中间代码 (称为字节码), Java 字节码可以被 Java 解释器解释执行。通过解释器, 每条 Java 字节指令被分析, 然后在计算机上运行。只需编译一次, 程序运行时解释执行。图 1-3 说明了它是如何工作的。

可以把 Java 字节码看做运行在 Java 虚拟机 (JVM) 上的机器代码指令。每种 Java 解释器, 不管是 Java 开发工具还是可以运行 Java 小应用程序的 Web 浏览器, 都是一种 JVM 的实例。JVM 也可以由硬件实现。