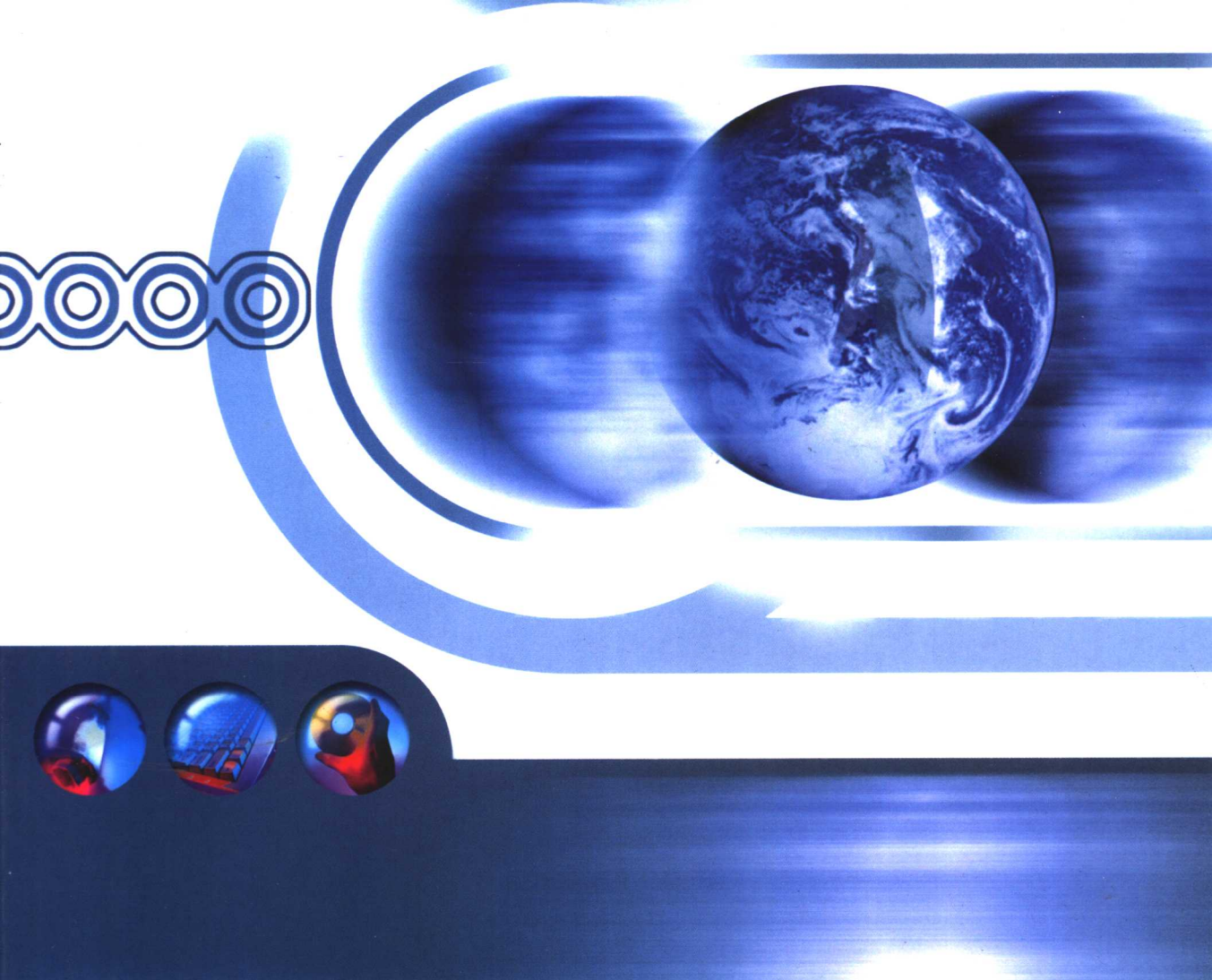




JBUILDER 9

基础教程

张大治 王辉 等编著



清华大学出版社

JBuilder 9 基础教程

张大治 王辉 等编著

清华大学出版社

北 京

内 容 简 介

Java 是目前最流行的编程语言之一, SUN 基于 Java 语言推出的 SunOne 和微软基于 VB/VC++/C# 语言推出的 .NET 是当今世界软件领域的两大阵营, 所以, 掌握 Java 就等于掌握了软件开发的半壁河山。JBuilder 是 Borland 公司出品的开发 Java 应用的集成开发环境, 它是业界最流行的, 也是最好的集成开发环境之一, 本书介绍的即是 JBuilder 的最新版本 JBuilder 9 —— 一个开发 Java 的利器。

本书由浅入深地介绍了 JBuilder 9 所集成的 Java 技术, 这些技术包括: 使用 JBuilder 9 进行 Web 开发、数据库开发、XML 开发、JavaBean 开发以及 EJB 的开发。而且, 本书还详细介绍了一些实际开发项目中的经验和所涉及的技术, 例如 Struts 技术、MVC 设计思想、团队开发、版本控制、Ant 编译打包部署工具, 以及 Web 服务的理念。

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。本书既可作为 JBuilder 9 的入门教材, 也可供有一定经验的开发人员查阅。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

JBuilder 9 基础教程/张大治, 王辉等编著. —北京: 清华大学出版社, 2004
ISBN 7-302-08289-8

I. J… II. ①张… ②王… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2004)第 019133 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦
http://www.tup.com.cn 邮 编: 100084
社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 曹 康

文稿编辑: 胡辰浩

封面设计: 康 博

版式设计: 郑国强

印 刷 者: 北京密云胶印厂

装 订 者: 北京鑫海金澳胶印有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185 × 260 印张: 23 字数: 545 千字

版 次: 2004 年 4 月第 1 版 2004 年 4 月第 1 次印刷

书 号: ISBN 7-302-08289-8/TP · 5976

印 数: 1 ~ 5000

定 价: 33.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或(010)62795704

前 言

Java 是目前最流行的编程语言之一，基于 Java 语言产生的 Java 体系是世界上很多开发人员用其开发企业应用的首选。JBuilder 作为 Java 的一个工具，是最优秀的 Java 集成开发环境之一。目前它的最新版本是 JBuilder 9，本书也是以 JBuilder 9 进行介绍的。

本书共分为 9 章。第 1 章简单介绍了 Java 语言的产生和发展，以及 JBuilder 9 的应用浏览器；第 2 章使用 JBuilder 9 做了几个简单应用，在这几个应用中读者将了解工程、应用程序和 JavaBean 的概念；第 3 章介绍了 JBuilder 9 的可视化编程环境，并且使用两个实例来介绍 Java 用户界面中的组件、容器、菜单以及布局管理等概念；第 4 章介绍了使用 JBuilder 9 开发 Web 应用，并且通过 Struts 技术的使用说明了 Web 应用的设计理念和 MVC 的设计思想；第 5 章介绍了使用 JBuilder 9 开发数据库应用；第 6 章是 JBuilder 9 的高级应用之一，详细介绍了团队开发和 CVS 的概念和应用，以及 JavaDoc 的制作；第 7 章是 JBuilder 9 的高级应用之二，详细介绍了单元测试的概念和应用，Ant 编译部署技术以及 Web 服务技术；第 8 章是 XML 的内容，详细介绍了 XML 的基本语法和设计思路、XML 的转换和验证以及数据绑定；第 9 章是 J2EE 的内容，详细介绍了 J2EE 的基本概念、EJB 技术以及 JBuilder 9 对这些技术应用的集成。

本书的两位主编人员是具有多年 Java 开发经验的高级程序员，在本书中融入了他们的很多经验和设计思想，相信读者能从本书中学到很多经验。本书内容丰富，结构清晰，语言简练，实例众多。对于入门级读者来说，本书将是带领其进入 Java 领域的首选；对于有一定经验的开发人员来说，本书也是深化编程思想和解决困惑的好帮手。

本书是多人智慧的结晶，除封面署名的作者外，参与整理资料 and 制作的人员还有王强、史新霞、周惠、杨玉敏、张凤霞、买芳芳、贺宝江、宋军山、亢喜成、龙红霞、龙海、王燕、陈丙离、张挂云、张极超和赵永魁等。由于作者水平有限，书中难免有不足之处，欢迎广大读者批评指正。

作 者

2004.1.28

目 录

第 1 章 JBuilder 9 简介	1
1.1 Java 简介	1
1.1.1 Java 的产生和发展	1
1.1.2 Java 的特点	3
1.1.3 Java 与 C 和 C++ 语言的异同	7
1.1.4 Java 的几个重要概念	7
1.1.5 Java 应用领域	8
1.2 JBuilder 9 的安装	9
1.3 JBuilder 9 的新特性	12
1.4 JBuilder 9 的应用浏览器	18
第 2 章 使用 JBuilder 9 开发简单应用	26
2.1 创建一个 JBuilder 工程	26
2.2 使用应用程序向导创建一个 HelloWorld 程序	28
2.3 使用可视化工具创建开发人员界面	32
2.3.1 JBuilder 9 可视化编程环境简单介绍	32
2.3.2 添加 JLabel 组件并设置组件属性	33
2.3.3 添加 JButton 组件并为组件添加事件	35
2.3.4 完善开发人员界面	37
2.4 打包应用程序	39
2.5 使用 BeanExpress 创建 JavaBeans	41
2.5.1 JavaBeans 的基本概念	41
2.5.2 BeansExpress 介绍	42
2.5.3 创建 JavaBeans	42
2.5.4 为新创建的 JavaBeans 添加属性	46
2.5.5 为添加过新属性的 JavaBeans 添加事件	49
2.5.6 创建 BeanInfo 类	50
2.5.7 验证 JavaBeans 的合法性	51
2.5.8 使用 JavaBeans	53
2.5.9 将 JavaBeans 安装到组件选项板	55
第 3 章 JBuilder 9 可视化编程	59
3.1 组件和容器介绍	59

3.1.1	组件介绍	59
3.1.2	容器介绍	61
3.2	布局管理器介绍	61
3.3	改变用户界面的外观	65
3.3.1	在程序代码中指定程序的外观	66
3.3.2	改变 JBuilder 开发环境的外观	68
3.4	菜单设计器的使用	69
3.4.1	菜单设计器介绍	69
3.4.2	下拉菜单	70
3.4.3	弹出式菜单	73
3.5	制作按钮式文本编辑器实例	75
3.5.1	样本分析	75
3.5.2	创建按钮式文本编辑器	76
3.6	制作菜单式文本编辑器	82
3.6.1	创建文本区	82
3.6.2	使用菜单设计器修改菜单	83
3.6.3	给菜单项添加事件	83
第 4 章	使用 JBuilder 9 开发 Web 应用	93
4.1	WebApp 和 WAR 文件	93
4.1.1	创建 WebApp 和 WAR 文件	93
4.1.2	设置 WebApp 的属性	96
4.2	使用 Servlet 向导创建小服务程序	99
4.2.1	Servlet 介绍	99
4.2.2	创建 Servlet	101
4.3	创建 JSP 程序	105
4.3.1	JSP 介绍	105
4.3.2	使用 JSP 向导创建 JSP 程序	109
4.4	在 Servlet 和 JSP 中使用 InternetBeans	113
4.4.1	InternetBeans 介绍	113
4.4.2	在 Servlet 中使用 InternetBeans	115
4.4.3	在 JSP 中使用 InternetBeans	125
4.5	使用 Struts framework 开发 Web 应用	133
4.5.1	Struts framework 中的相关知识	133
4.5.2	Struts framework 中的 3 层结构	134
4.5.3	业务数据层组件	136
4.5.4	业务功能实现层组件	137

4.5.5 表现层组件	139
4.5.6 控制层组件	140
4.5.7 在 JBuilder 9 中使用 Struts 开发 Web 应用	143
第 5 章 使用 JBuilder 9 开发数据库应用	149
5.1 JBuilder 9 数据库基础知识	149
5.1.1 理解数据库应用程序	149
5.1.2 DataExpress 组件	150
5.1.3 支持 EJB 组件开发的 DataExpress	152
5.1.4 XML 数据库组件	152
5.1.5 dbSwing 组件	153
5.1.6 数据模块和模块生成器	154
5.1.7 数据库导航器	154
5.1.8 JDBC 监听器	154
5.1.9 JDataStore 组件	155
5.1.10 JBuilder 9 数据库应用结构	155
5.2 访问数据文件	156
5.2.1 从数据文件中取得数据	157
5.2.2 修改数据并保存到数据文件中	160
5.2.3 使用数据样式向文本文件中输出数据	161
5.3 访问数据库	163
5.3.1 连接数据库	163
5.3.2 从数据源接收数据	165
5.3.3 根据参数从数据源接收数据	168
5.4 数据库的关联关系	171
5.4.1 数据库关联关系介绍	171
5.4.2 JBuilder 9 对数据库关联关系的支持	171
5.4.3 使用 MasterLink 创建数据库的主从关系	172
5.5 数据的过滤、排序和查找	174
5.5.1 获得数据库中的数据	175
5.5.2 数据的过滤	176
5.5.3 排序	178
5.5.4 查找	180
5.6 使用 Database Pilot 管理数据库	182
5.6.1 使用 Database Pilot 管理数据库表和元数据	183
5.6.2 使用 JDBC Monitor 监听数据库连接	187

5.7 使用 JBuilder 9 创建数据库应用实例	189
5.7.1 为实例创建一个工程和应用程序	190
5.7.2 给应用程序添加 DataExpress 组件	190
5.7.3 设计应用程序的列字段	192
5.7.4 给应用程序添加 dbSwing 组件创建开发人员界面	194
5.7.5 使用有计算能力的域来统计数据	195
第 6 章 JBuilder 9 的高级应用之一	197
6.1 在 JBuilder 中使用 CVS 进行版本控制	197
6.1.1 团队开发介绍	197
6.1.2 版本控制介绍	198
6.1.3 JBuilder 9 中的 CVS	198
6.1.4 在 JBuilder 中使用本地 CVS 服务器进行工作	199
6.2 使用 JavaDoc 生成器创建 Doc 文档	214
6.2.1 JavaDoc 介绍	215
6.2.2 创建 JavaDoc 文档	217
第 7 章 JBuilder 9 的高级应用之二	227
7.1 使用 Ant 编译工程	227
7.1.1 Ant 介绍	227
7.1.2 Ant 使用	229
7.2 创建并运行 test case 和 test suites	237
7.2.1 单元测试介绍	237
7.2.2 Junit 介绍	238
7.2.3 使用 JBuilder 进行 Junit 测试	240
7.3 Web 服务技术介绍	244
7.3.1 Web 服务框架	244
7.3.2 Web 服务标准	244
7.3.3 JAX-RPC 介绍	247
第 8 章 使用 JBuilder 9 开发 XML 应用	248
8.1 XML 介绍	248
8.1.1 XML 的基础知识	248
8.1.2 文档类型定义	254
8.1.3 模式	267
8.1.4 命名空间	274
8.1.5 XSL 转换	275
8.1.6 链接	278

8.1.7 级联样式表	283
8.1.8 JBuilder 9 对 XML 开发的支持	289
8.2 创建和验证 XML 文档	289
8.2.1 XML 文档的格式正确和有效性	289
8.2.2 创建 XML 文档	290
8.2.3 验证 XML 文档的有效性	293
8.2.4 查看 XML 文档	294
8.3 转换 XML 文档	296
8.3.1 视图转换窗口	296
8.3.2 使用样式表转换 XML 视图	297
8.4 数据绑定	299
8.4.1 数据绑定介绍	299
8.4.2 使用 BorlandXML 实现 DTD 数据绑定	299
8.4.3 使用 Castor 实现 Schema 数据绑定	304
8.5 创建 SAX 处理器解析 XML 文档	309
8.5.1 DOM 介绍	310
8.5.2 SAX 介绍	310
8.5.3 创建 SAX 解析器	312
第 9 章 使用 JBuilder 9 开发 J2EE 应用	320
9.1 J2EE 介绍	320
9.1.1 多层框架结构的优点	320
9.1.2 多层结构模型的多样性	323
9.2 在 JBuilder 9 中为开发 EJB 配置应用服务器	325
9.2.1 WebLogic 应用服务器的安装	326
9.2.2 配置 WebLogic 的 Domain	329
9.2.3 把 WebLogic 配置成 JBuilder 9 使用的应用服务器	333
9.3 EJB 介绍	336
9.3.1 J2EE Server、EJB Container 和 Web Container	337
9.3.2 EJB 的类型	337
9.3.3 远程(remote)和本地(local)接口	338
9.3.4 会话 bean 的创建过程和生命周期	339
9.3.5 实体 bean 的创建过程和生命周期	341
9.3.6 home 和 remote/local 接口的创建过程	343
9.3.7 EJB 的设计约束	345
9.3.8 数据库访问的几点考虑	345

9.4 使用 EJB 设计器开发会话 bean 实例	346
9.4.1 创建会话 bean 组件	346
9.4.2 启动 WebLogic 服务器	350
9.4.3 部署 EJB 应用	351
9.4.4 创建客户端程序	352

第1章 JBuilder 9简介

Borland JBuilder 是世界上领先的 Java 开发环境。JBuilder 9 是一个用于开发企业级 Java 应用的综合性跨平台环境；它建立在 JBuilder 屡获殊荣的历史上，并增加了崭新的特性，从而成为目前最强大的可视化 Java 开发环境。

本章包括以下内容：

- Java 介绍
- JBuilder 9 的安装
- JBuilder 9 的新特性
- JBuilder 9 的应用浏览器

1.1 Java 简介

Java 是 Sun Microsystem 公司开发的一种面向对象的程序设计语言。它由于具有小巧、稳定以及与平台无关等特点，在 Internet 上得到了广泛的应用，并成为一种风靡世界的编程语言。其实，现在将 Java 仅定义为一门编程语言已经不适当了，Java 应该是一个基于 Java 语言的平台，这个建立在操作系统之上的平台能够提供企业级服务、通用服务、无线服务以及智能卡服务。在软件领域尤其是比较高层的企业级应用软件领域，Java 绝对是个超重量级的平台。下面我们就带领您进入这个充满光明前景和诱惑力的 Java 世界。

1.1.1 Java 的产生和发展

1991 年初，位于美国加利福尼亚的 Sun Microsystem 公司(以下简称 Sun 公司)成立了由 James Gosling 领导的研发小组。这个小组成立的目的是开发跟小型或便携式的消费性电子产品(如 PDA 等)配套的独立平台软件环境，并建立分布式的系统结构，同时将现代软件技术移植到消费性电子产品上，即随后提出的 Green Project(绿色计划)的主要内容，该小组也因此被命名为 Green 小组。

与 Sun 公司以往所擅长的工作站及微处理器技术不同的是，Green 小组所开发的消费性电子产品技术的顾客是一般消费者。这些用户不愿意在这种产品上花太多的钱，也不愿意花很多的时间来学习如何操作它，并且他们重视的是产品的稳定性，而并不在意产品内部使用的是什么 CPU，或者 CPU 的速度有多快。因此，开发一种小巧、易用、安全稳定且与平台无关的系统就成为了 Green 小组研究的目标。

最初, Green 小组试图通过扩充 C++编译器的功能来开发这个系统。经过一段时间的努力后, 该小组发现, C++离其他的目标差距太大。这时 Gosling 等人认识到, 必须开发一套符合自己目标的新语言。于是, 1991 年 Java 的前身——Oak 诞生了。Oak 这个名字来自于 Sun 公司 Gosling 办公室外的一棵橡树, Gosling 为他们的语言起这个名字, 是希望它能够具有像橡树一般坚强的生命力。

之后, Green 小组在 Oak 的基础上开发了“*7”系统, 它包括 Oak、Green OS(一种操作系统)、用户界面以及硬件部分。在小组向 Sun 公司内部展示产品时, “*7”成功地表现了高效率的小程序代码技术, 给公司的高层管理人员留下了深刻印象, 也增强了小组全力开拓家用电子市场的信心。他们认为需要有一个真正的公司能与电子制造商们打交道。因此, 1993 年 5 月, 在原 Green 小组的基础上成立了 FirstPerson 公司。

FirstPerson 公司选择了一种新型电子产品——电视机机顶盒来推广其技术。当时, Time Warner 公司正寻求开发电视机机顶盒的操作系统和视频技术的合作伙伴, FirstPerson 公司很希望能够得到这份合同。但最后因为商业上的某种原因, Time Warner 公司选择了另外一家公司作为合作伙伴。接着, FirstPerson 公司与 3DO 公司商谈了合作事宜, 同样是由于商业上的原因, 合作也没能成功。FirstPerson 公司以后又提出了一些产品建议, 但都没有得到 Sun 公司的批准。由于没有找到合作伙伴, 1994 年, FirstPerson 公司宣布解散, 研发小组中的许多人也退出了, 只留下了几个人。

在经历了挫折之后, Gosling 等人并没有放弃他们原来的计划。此时, WWW(World Wide Web, 万维网)正在 Internet 上以迅猛的势头发展。在分析了整个 WWW 的结构以后, 研发小组认为可以把他们的技术应用于这个领域。于是开始开发第一个可以在 WWW 上执行 Java 程序的 WWW 浏览器——WebRunner。在此期间, 由于其名字与另一个已注册的计算机语言相冲突, Oak 被迫更名为 Java。几个月后, 使用了 Java 语言和原有的 WWW 浏览器所没用过的新技术编制成的新的 WWW 浏览器——HotJava 诞生了。1995 年 5 月 23 日, Sun 公司在 SunWorld 上正式发布了 Java 和 HotJava, 从此引发了至今不衰的 Java 热。

Java 发布的当天, Netscape 公司便宣布打算在 Netscape 浏览器上获得使用 Java 的授权。从 Netscape 2.0 版开始, 所有 Netscape 浏览器都提供了对 Java 的支持。随后, IBM、Symantec、Inprise 以及 Microsoft 等许多软件公司都注册并宣布支持 Java。Java 语言获得了巨大的成功。

在获得 Netscape Navigator 的支持后, 研发小组便不再开发新的 HotJava 浏览器, 这个 WWW 浏览器一直停留在 Alpha 3 版本上。

Sun 公司于 1996 年发布了 Java 1.02, 这个版本存在着许多问题, 甚至无法用它来进行打印。但在很短的时间以后, Sun 公司发布了 Java 1.1, 成功地解决了上一版本的一些问题。随后发布的一些新版本虽然作出了改进, 但总的说来改变都不很大。

在 1998 年召开的 JavaOne 大会上, Java 1.2 又问世了。3 天后, Java 1.2 被改成了 Java 2。它增加了全新的 GUI 界面和图形工具包, 离当初许诺的“一次编写, 到处运行”的目标又前进了一大步。

1.1.2 Java 的特点

在 Sun 公司发布的 Java 白皮书中, Java 是这样被定义的: Java 是一种简单的、面向对象的、分布式的、稳健的、解释型的、安全的、中性结构(体系中立)的、可移植的、高性能的、多线程的以及动态的语言。以上的定义概括了 Java 语言的特点, 下面将对这些特点简单地作出解释。

1. 简单性

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几个方面:

- Java 的风格类似于 C++。从某种意义上讲, Java 语言是 C 和 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。
- 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。
- Java 提供了丰富的类库。
- Java 不再使用预处理器和头文件, 减少了程序或软件在开发和维护时的难度, 也方便了程序的阅读。
- Java 中不再有多重继承。多重继承是一项很强的功能, 但它增加了程序的复杂性, 一般人难以掌握。Java 使用接口——一种抽象类, 它能够达到部分多重继承的作用, 而且容易掌握。类似地, Java 还取消了操作符重载, 使它更简单易学。
- Java 使用“引用”来代替指针, 通过引用来访问申请到的内存里的数据, 以确保不会访问不属于自己的内存空间, 大大减少了编程中出错的可能。Java 还可以自动完成“垃圾收集”工作, 将不再使用的内存回收, 使用户无须担心内存管理之类的事情。

2. 面向对象性

面向对象可以说是 Java 最重要的特性。Java 语言的设计完全是面向对象的, 它不支持类似 C 语言那样的面向过程的程序设计技术。从比较严格的意义上来说, 一种面向对象的编程语言至少应包括以下 4 个特点:

- 封装性。系指模块化的性质和信息隐藏的能力。
- 多态性。不同的对象可以按照其本身的性质对同一种信息进行响应。
- 继承性。继承性定义了一种层次关系。下层的对象继承上层对象的特性, 可以减少程序代码的重复编写, 并有效地组织整个程序。
- 动态绑定。允许在运行时而非编译时按具体数据类型和参数来确定选用哪个函数。

如果按照以上的标准来衡量现有的一些程序设计语言, 有许多都不能算是严格定义上的面向对象的语言。例如, Visual Basic 中就没有实现封装性, C++ 中也没有实现动态绑定。而 Java 则很好地体现了这 4 个特点。

3. 分布式

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 打开并访问网络上的对象, 其访问方式与访问本地文件系统几乎完全相同。为分布式环境尤其是 Internet 提供动态内容无疑是一项非常宏伟的任务, 但 Java 的语法特性却使大家很容易地实现这项目标。Java 是一种分布式的程序设计语言, 它提供了 URL 对象来使数据分布得以实现。利用 URL 对象来打开及存取分布于 Internet 上的远程对象, 就像访问本地文件系统一样简单。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(RMI API), 可以调用远程对象的方法。Java 还可以通过 Socket 类来建立低级别的网络连接。而另一种被称为小服务程序(Servlet)的机制, 使得用 Java 进行服务器编程获得非常高的效率。因此, 可以把运算从服务器端分散到客户端, 从而提高整个系统的执行效率, 避免了瓶颈制约, 最终增加了动态可扩充性。

4. 稳健性

考虑到 Gosling 等人组成的研发小组最初开发的技术需要面对的顾客是一般消费者, Java 设计人员在开发这种语言时的目标之一就是不管在何种条件下, 都能保证编程绝对可靠。为此, Java 将大量重点放在早期的潜在问题检查、以后的动态(运行时)检查以及避免出现潜在错误的情况上面。

经过研发小组的努力, Java 从根本上杜绝了一些错误出现的源头。在前面有关 Java 简单性的描述中, 已经介绍了 Java 不支持指针的特性。异常处理是 Java 的另外一个特性, 它可以使程序长时间地可靠运行而不容易发生致命的错误。另外, Java 是一种强类型的语言, 即需要显式的类型声明, 不支持隐式声明以及 C 和 C++中的自动类型转换, 这种严格要求保证了编译器可以俘获方法调用时的错误, 从而生成更可靠的程序。

5. 解释型

Java 解释器(运行系统)能直接运行目标代码指令。链接程序通常比编译程序所需资源少, 所以程序员可以在创建源程序上花费更多的时间。必须承认的是, 由于 Java 程序是被解释执行的, 其运行速度仅能达到与它同级的被编译执行的程序运行速度的 1/6~1/2。但是, 编译和解释结合在一起给 Java 程序提供了安全性和稳固性, 最重要的是消除了平台之间的差别。

6. 安全性

在 Java 语言里, 像指针和释放内存等 C++功能被删除了, 避免了非法内存操作。Java 语言在开发人员的机器上执行前, 要经过很多次的测试。由于 Java 是一种分布式的语言, 必须保证从别的地方下载的程序代码在运行时不会对本地的机器和文件系统造成破坏。因此, 在设计 Java 时就把安全性放到了一个十分重要的地位。一般来说, 一个安全的网络程序, 至少要防止以下几种被破坏的可能性:

- 恶意使用系统资源。如未经许可便修改或删除文件或数据, 改变现在使用的内存内

容, 终止系统某个进程或线程的任务, 不断地向系统索取内存空间, 影响其他程序的运行等。

- 窃取系统或个人机密。如窃取个人密码及私人资料等。
- 骚扰正常工作的进行。如在屏幕上产生奇怪的图案, 或发出奇怪的声音等。

Java 提供了几个层次的安全性控制机制。首先, 在 Java 的语言定义中, 对与系统有关的对象和变量, 可以加以不同的使用限制, 以避免被其他对象所修改, 或以继承的方式重新编写。Java 取消了指针, 使用户无法直接管理内存, 避免了恶意使用系统资源。其次, 在 Java 程序的执行过程中, 字节码载入器要对字节码做一次安全检查, 包括确定程序没有违反对象的存取权限, 使用正确的参数类型来调用对象的方法, 以及系统堆栈没有溢出等。另外, Java 程序的执行系统, 也就是支持 Java 的 WWW 浏览器可以阻止 Java 小应用程序对系统资源的不正当应用, 以及调整对小应用程序的处理方式, 这样, 本地系统就可以受到保护。最极端的情况就是完全禁止 Java 小应用程序的执行, 在受到骚扰时, 用户只要关闭浏览器就可以终止恶意程序的运行。

由上述可以看出, Java 语言在设计时便对安全性作了充分的考虑。尽管不能说 Java 是一种绝对安全的语言, 但可以说它是目前最安全的程序设计语言之一。

7. 中性结构(体系中立)

为了建立 Java 作为网络的一个整体, Java 将它的程序编译成一种结构中立的中间文件格式。Java 提供了一种特殊的字节码指令, 同任何一种特定的计算机体系都无关, 它不仅在任何机器上都易于解释, 也易于动态翻译成本地代码。这一机制的最大好处就是使 Java 程序可以在任何系统上运行, 但前提条件是该系统能支持 Java 虚拟机。换句话说, 如果用户使用 Java 语言来编写程序, 那么该程序可以在所有的平台上运行; 如果用户在程序中使用 JFC(Java Foundation Classes), 那么该程序在每一种平台上都会具有相似的外观和行为。

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中运行, 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。Java 具有严格的语言定义, 使它不存在任何取决于工作平台或编译器的功能或特性。或者说, 不管用户使用的机器及使用的编译器如何不同, 最后产生的目标码(Object code)都相同。这是因为, 与 C 和 C++语言不同, Java 中的每一种数据类型的长度及表达方式都是确定的, 这样就避开了移植时可能出现的大量问题。另外, 在 Java 语言中, 对各种数据类型的变量初值也有完整明确的定义。因此, Java 程序在不同系统上运行不会产生不可预料的结果。这就是说, Java 具有良好的可移植性。

9. 高性能

尽管没有编译运行的程序运行速度快, Java 程序与其他解释执行的程序相比, 运行速

度已经快了很多。除了前面提到的半编译、半解释的执行方式以外,Java 还提供了一种被称为 JIT(Just In Time,即时编译)的编译器。JIT 首先将字节码编译成固有代码(即某种平台专用的机器码),并将结果保存下来,然后在需要时调用它们。这种方法可以明显提高速度,因为只需进行一次解释就够了。实际上, Sun 公司声称字节码经 JIT 转换生成的机器码几乎与原有的 C 和 C++语言一样优秀,而且 Java 2 与以前的版本相比性能上又有了大幅度的提高。

10. 多线程性

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,有时也称为小进程,是一个大进程里分出来的小的独立的进程。因为 Java 能实现多线程技术,所以比 C 和 C++语言更健壮。例如一个用户可以在他接受声音文件时将一个屏幕向上滚动一屏,同时浏览器在后台下载一幅图像。

Java 很好地支持了多线程性。Java 的基本函数库中已定义了 Thread 这个基本类,并内置了一组方法,使程序设计者在编写多线程程序时,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、执行一个线程、终止一个线程的任务,或查看执行状态。

为了解决数据同步化的问题,Java 采用了著名的 C.A.R.Hoare 所提出的监控器(monitor)和状态变量(condition variable)的技巧,这样对各个线程或进程共用的数据可以加以严格的控制,避免了因多个线程或进程同时修改相同的数据而造成数据的内容前后不一致以致出错。同时,也避免了发生死锁的可能。

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时所无法实现的。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 从下述几个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传递给解释器,再由解释器完成动态连接类后,将符号引用信息转换为数值偏移量。
- 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。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。如果开发人员的系统运行 Java 程序时遇到了不知怎样处理的程序,那么 Java 能自动下载开发人员所需要的功能程序。

1.1.3 与 C 和 C++语言的异同

本章在 1.1.1 节中提到过, Gosling 等人最初试图通过增加 C++编译器的功能来开发新的系统,但最终失败了,原因之一就是 C++语言的复杂性使得编写可靠程序的难度很高,不能实现计划的目标。所以, Gosling 等人设计 Java 时省略了 C++语言中所用的一些罕见的、难以理解的和极易混淆的特性。Java 提供了一个功能强大语言所具有的所有功能,但几乎没有含混特征。虽然 C++语言安全性不好,但 C 和 C++语言还是被大家所接受,所以 Java 设计成 C++形式,让大家很容易学习。Java 去掉了 C++语言的许多功能,让 Java 的语言功能很精炼。由于 Java 是面向对象的,而且不用像 C++语言那样考虑同 C 兼容的问题,所以结构(structure)和联合(union)等数据类型就没有存在的必要了。Java 中不再有函数的概念,代之以对象中的方法,再一次强化了面向对象的开发策略。

1.1.4 Java 的几个重要概念

1. Java 虚拟机(JVM)

Java 虚拟机是 Java 应用程序运行的平台,这个平台位于操作系统之上,屏蔽了与操作系统交互的复杂性,提供统一的应用程序接口,根据操作系统的不同,开发人员应该选用基于这个操作系统编写的 Java 虚拟机,但开发人员自己基于 Java 编写的程序对于任何虚拟机来说是一样的,所以可以实现 Java “一次编写,到处运行”的承诺。但这样的承诺也需要付出一定的代价,那就是会牺牲一部分性能,对于现在硬件集成度的提高来说,这种牺牲变得越来越不重要了。

2. Java 开发工具集(JDK)

JDK 是提供给开发人员使用的 Java 开发包,它提供了大量的工具类,接口定义和命令工具来让开发人员编写、调试、部署和运行应用程序。目前最新的 JDK 版本是 1.4。

3. Java 架构

Java 已经不再是以一种单纯的编程语言的面貌出现,而是一个 Java 体系,或者称为 Java 框架结构。目前针对不同的应用领域,Java 支持 4 种框架结构。

- J2EE 框架结构。主要应用领域是企业级应用,包括基于 Web 的 Servlet 和 JSP 的应用、EJB 应用和 Web 服务应用等。
- J2SE 框架结构。这是 Java 框架结构的基础,J2EE 结构也是建立在 J2SE 的基础之上的,它提供了 Java 基础应用,如 SwingUI 设计、XML 应用、访问数据库的 SQL 操作以及访问文件的 IO 操作等。
- J2ME 框架。主要应用在无线领域,针对嵌入式设备的 Java 平台,如 PDA、机顶盒以及手机等。