



全面掌握在JBuilder环境中开发Java应用程序的方法和技巧

JBuilder 5

实例教程

张焱 编著



清华大学出版社

[Http://www.tup.tsinghua.edu.cn](http://www.tup.tsinghua.edu.cn)



JBuilder 5 实例教程

张焱 编著

清华大学出版社

(京) 新登字 158 号

内 容 简 介

JBuilder 是 Borland 公司推出的可视化 Java 开发工具。JBuilder 5 是其最新版本, 增加了网络开发、团队开发、企业版 JavaBeans 等新特性, 扩展了对应用程序服务器的支持, 改进了 JDBC 和 JDataStore, 使数据处理更为简便。

本书以 JBuilder 5 专业版为基础全面阐述了 Java 编程方法。本书含有大量的范例程序, 相信读者通过本书, 可以迅速掌握在 JBuilder 5 环境下开发 Java 程序的方法。

在编写本书时, 考虑了不同的读者层次。因此, 对没有接触过 Java 及 JBuilder 的新手来说, 可以从本书中学到实际编程中实用软件的基本开发方法。对有开发经验的人员来说, 可以迅速掌握 JBuilder 5 的编程技巧。

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。

书 名: JBuilder 5 实例教程

作 者: 张焱 编著

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 张秋香

印 刷 者: 世界知识印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 18.25 字数: 433 千字

版 次: 2002 年 2 月第 1 版 2002 年 4 月第 2 次印刷

书 号: ISBN 7-302-05207-7/TP·3058

印 数: 4001~6000

定 价: 26.00 元

前 言

随着 Internet 的普及, 21 世纪将是网络的世纪。而 Java 语言作为 Internet 编程的重要工具, 具有其他编程语言无可比拟的优点。首先, Java 语言具有完全面向对象的特点, 使其可以具有极大的灵活性。Java 严格的语言定义、安全性、与操作平台无关, 以及多线程技术都是使其广泛流行的重要特点。可以预见, Java 技术的出现将对未来生活产生重大影响。

然而, 早期的 Java 程序是在 JDK 环境下开发的, 其编译及运行采取命令行的方式, 使开发过程很不方便, 从而限制了 Java 的流行。

JBuilder 是 Borland 公司推出的可视化 Java 开发工具。JBuilder 5 是其最新版本, 与以往版本相比, 它具有以下特点:

- 增加了网络开发、团队开发、企业版 JavaBeans 等新特性;
- 扩充了 OpenTools API 并使其更易于操作;
- 扩展了对应用程序服务器的支持;
- 用户界面更为友好;
- 增加了调试器的功能配置;
- 改进了 JDBC 和 JDataStore, 使数据处理更为简便。

本书以 JBuilder 5 企业版为基础阐述了在 JBuilder 环境中开发 Java 应用程序的方法。本书的内容主要包括 4 个方面, 即基本应用程序的开发、数据库应用程序的开发、Web 应用程序的开发和 XML 应用程序的开发。

本书服务的对象是对 Java 编程语言已有一定了解的程序开发人员, 因此本书中不再讲述有关 Java 语言的基本知识。在学习数据库应用程序开发部分时, 用户应已掌握 SQL 语言的基本知识。在学习 XML 应用程序的开发时, 用户应已掌握 XML 的相关知识。

本书共分 21 章, 每章均有针对性的练习和示范程序。通过本书, 可以迅速掌握在 JBuilder 5 环境下开发 Java 程序的方法。

本书由张焱、韩江、仇爽、黄志明、李立华和刘晓亮编写, 周予滨审校。

由于作者水平有限, 加之时间仓促, 书中不妥之处在所难免, 恳请读者批评指正。

作者
2001 年 9 月

— c /

目 录

第 1 章	JBuilder 5 —— 优秀的 Java 开发工具	1
1.1	认识 Java	1
1.1.1	Java 语言的产生与发展	1
1.1.2	Java 和 Internet	3
1.1.3	Java 语言的特点	3
1.2	JBuilder 5 简介	8
1.2.1	JBuilder 5 的特性	8
1.2.2	安装 JBuilder 5	9
第 2 章	JBuilder 5 编程环境	13
2.1	应用浏览器简介	13
2.2	使用联机帮助	23
2.2.1	获得特定主题的帮助	23
2.2.2	在应用浏览器界面中直接使用帮助功能	23
第 3 章	创建和部署应用程序	25
3.1	Hello World: 第一个 JBuilder 程序	25
3.1.1	创建新工程	26
3.1.2	创建应用程序	28
3.1.3	创建用户界面	31
3.1.4	为程序添加事件	34
3.2	部署应用程序	35
3.2.1	部署概述	35
3.2.2	创建 JAR 文件	36
第 4 章	设计用户界面	39
4.1	组件和容器	39
4.1.1	组件概述	39
4.1.2	容器	40
4.2	改变用户界面的外观	41
4.2.1	在程序代码中指定程序的外观	41
4.2.2	改变 JBuilder 5 开发环境中的用户界面外观	42



4.3	布局管理器	43
4.4	使用菜单设计器	46
4.4.1	菜单设计器	46
4.4.2	创建下拉菜单	47
4.4.3	创建弹出式菜单	50
4.5	编程实例——文本编辑器	51
4.5.1	创建用户界面	52
4.5.2	添加菜单	56
4.5.3	为用户界面添加事件	57
第 5 章	在 JBuilder 中调试程序	68
5.1	处理语法错误和编译器错误	68
5.1.1	错误类型	68
5.1.2	处理语法错误	69
5.1.3	编译器错误	71
5.2	JBuilder 5 调试环境	74
5.2.1	调试器的启动和终止	74
5.2.2	视图	75
5.2.3	工具栏	82
5.2.4	执行点	83
5.2.5	Tool tip 功能和 ExpressionInsight 功能	84
5.2.6	快捷键	85
5.3	使用断点	85
5.3.1	断点的类型	85
5.3.2	设置和删除断点	86
5.3.3	设置断点属性	88
5.3.4	使断点失效和重新生效	90
5.4	使用调试器调试程序	90
5.4.1	控制程序的运行	91
5.4.2	观察变量或对象的值	92
5.4.3	为变量添加监视	93
5.4.4	改变变量的值	95
第 6 章	创建 JavaBeans	98
6.1	概述	98

6.1.1	JavaBeans 的概念及特点	98
6.1.2	使用 BeansExpress 的原因	99
6.2	创建 JavaBeans	99
6.2.1	创建 Bean 类	99
6.2.2	设计用户界面	101
6.2.3	添加属性	102
6.2.4	为 Bean 添加事件	106
6.2.5	创建 BeanInfo 类	107
6.2.6	检验 JavaBeans 的合法性	108
6.3	JavaBeans 的使用和安装	110
6.3.1	使用 JavaBeans	110
6.3.2	将 JavaBeans 安装到组件选项板	111
第 7 章	JBuilder 5 数据库编程基础	116
7.1	理解 JBuilder 5 的数据库应用程序	116
7.2	JBuilder 5 数据库编程工具	117
7.2.1	JBuilder 5 中与数据库相关的类	117
7.2.2	dbSwing 组件	119
7.3	使用 DatabasePilot 浏览数据库表和元数据	120
7.3.1	浏览数据库模式对象	121
7.3.2	设置驱动程序来存取远程与本地数据库	121
7.3.3	执行 SQL 语句	122
7.3.4	查看并编辑表数据	123
第 8 章	数据文件的访问与修改	125
8.1	从文本文件中存取数据	125
8.1.1	取得文本文件中的数据	125
8.1.2	保存修改结果	128
8.1.3	向文本文件中输出数据	130
8.2	查询数据库	131
8.2.1	执行无参数查询	132
8.2.2	执行带参数的查询	136
第 9 章	数据的过滤、排序和查找	140
9.1	过滤	140
9.2	排序	146
9.3	查找	148



第 10 章	建立主从关系	151
10.1	主从关系	151
10.2	在 JBuilder 5 中建立主从关系	152
第 11 章	为数据库应用程序添加功能	156
11.1	使用计算列	156
11.1.1	创建一个计算列	157
11.1.2	使用计算列合计数据	158
11.2	创建查找	160
11.2.1	使用参数选用表输入数据	160
11.2.2	使用计算列创建查找	161
11.3	使用 DataSetView 观察数据	163
第 12 章	使用数据模块	166
12.1	数据模块的概念	166
12.2	创建数据模块	167
12.2.1	使用 Data Modeler 创建数据模块	167
12.2.2	为数据模块创建库文件	171
12.2.3	在应用程序中使用数据模块	172
第 13 章	使用 WebApp 和 WAR 文件	175
13.1	WebApp	175
13.1.1	创建 WebApp	175
13.1.2	设置 WebApp 的属性	177
13.2	Web 存档文件	180
第 14 章	创建小应用程序	181
14.1	小应用程序	181
14.1.1	小应用程序概述	181
14.1.2	小应用程序的生命周期	181
14.2	创建第一个小应用程序	182
14.2.1	创建新工程	183
14.2.2	创建小应用程序	183
14.2.3	建立用户界面	185
14.2.4	添加程序代码	187
14.2.5	部署小应用程序	188
14.2.6	修改 HTML 文件	190

14.2.7 在浏览器中查看修改后的 HTML 文件	191
第 15 章 创建小服务程序	192
15.1 小服务程序概述	192
15.1.1 小服务程序的概念及特点	192
15.1.2 小服务程序的应用范围	193
15.1.3 小服务程序的基本框架	193
15.1.4 小服务程序的生命周期	194
15.2 使用 JBuilder 5 开发小服务程序	194
15.2.1 创建小服务程序	195
15.2.2 与数据库相连接	200
第 16 章 创建 JSP 文件	209
16.1 JSP 概述	209
16.1.1 什么是 JSP	209
16.1.2 JSP 的特点	210
16.1.3 JSP 的基本语法	210
16.2 创建 JSP 文件	211
第 17 章 使用 InternetBeans	214
17.1 InternetBeans Express 概述	214
17.2 在小服务程序中使用 InternetBeans	215
17.3 在 JSP 中使用 InternetBeans	223
第 18 章 验证和转换 XML 文件	231
18.1 XML	231
18.1.1 XML 概述	231
18.1.2 JBuilder 5 对 XML 的支持	232
18.2 XML 文件的验证	232
18.2.1 格式正确和有效	232
18.2.2 验证 XML 文件	233
18.3 XML 文件的转换	237
18.3.1 转换视图窗口	237
18.3.2 使用默认的样式表	238
18.3.3 使用样式表转换 XML 外观	240
18.3.4 设置转换跟踪选项	242



第 19 章	创建 SAX 处理器	243
19.1	SAX	243
19.1.1	SAX 概述	243
19.1.2	SAX 的特点	243
19.2	使用 SAX 处理器解析 XML 文件	245
19.2.1	创建 SAX 解析器	245
19.2.2	编辑 SAX 解析器代码	247
19.2.3	解析 XML 文件	249
第 20 章	数据绑定	253
20.1	数据绑定简介	253
20.2	使用 BorlandXML 实现 DTD 数据绑定	253
20.2.1	由 DTD 文件创建 Java 类	254
20.2.2	将数据解包	256
20.2.3	将数据打包	257
20.3	使用 Castor 实现模式数据绑定	259
20.3.1	由模式创建 Java 类	260
20.3.2	将数据解包	261
20.3.3	将数据打包	262
第 21 章	用 XML 数据库组件进行数据传输	266
21.1	使用 XML 数据库组件进行数据传输简介	266
21.2	使用基于模型的 XML 数据库组件	266
21.2.1	创建映射文件和 SQL 脚本文件	267
21.2.2	创建数据库表	269
21.2.3	使用 XML-DBMS 进行数据传输	270
21.3	使用基于模板的 XML 数据库组件	276
21.3.1	使用 xTable 组件进行数据传输	276
21.3.2	使用 xQuery 组件进行数据传输	278

第1章 JBuilder 5——优秀的Java开发工具

本章包括以下内容：

- Java 语言的产生与发展
- Java 语言的特点
- JBuilder 5 的特点与安装

Java 是 Sun Microsystem 公司开发的一种面向对象的程序设计语言。由于它具有小巧、稳定、与平台无关等特点，使其在 Internet 上得到了广泛的应用，已成为一种风靡世界的编程语言。JBuilder 5 作为可视化的 Java 程序开发工具，它的出现极大地方便了 Java 程序的编写，也必将推动 Java 语言的进一步发展。显然，JBuilder 5 和 Java 的关系是密不可分的。为此，在学习使用 JBuilder 5 开发 Java 程序之前，有必要先对 Java 语言做一下简单介绍。

1.1 认识 Java

1.1.1 Java 语言的产生与发展

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 James Gosling 领导的研发小组。这个小组实际上与它的母公司独立，其成立目的是开发与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，即随即提出的 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，Green 小组所开发的消费性电子产品技术的顾客是一般消费大众。这些用户不愿意花太多的钱在这种产品上，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，而不很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，他们发现，C++离他们的目标差距太大。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 年中，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，FirstPerson 公司在原 Green 小组的基础上成立了。

FirstPerson 公司选择了一种新型电子产品——电视机机顶盒来推广他们的技术。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。由于没有签订到一份合同，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。于是他们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，由于名字与另一个已注册的计算机语言相冲突，Oak 被更名为 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, 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，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 和 Internet

Java 的成功与 Internet 的发展有极大的关系。甚至可以说，二者是相辅相成的。那么，Internet 的发展是怎样的呢？

Internet 的前身是美国国防部高级研究计划局(Advanced Research Project Agency, 简称 ARPA)1969 年建立的计算机网络 ARPAnet。刚建立时，ARPAnet 只有 4 个节点。但是到了 1984 年，该网络发展到了 1000 多个节点。1986 年，美国国家科学基金会(NSF)资助成立了美国科学基金会网(NSFnet)，把全美的 5 个超级计算中心连接起来。1990 年，NSFnet 取代了 ARPAnet 成为 Internet 的主干网。到目前为止，Internet 已深入到了世界上的每一个角落。

早期的 Internet 只有简单的命令行界面，使用起来很不方便，通常只有一些专家和科研人员使用，并未得到普及。1989 年，位于日内瓦的欧洲基本粒子物理实验室(CERN)的科学家发明了 WWW 网络传输协议，这种协议可以在文本中插入图片和声音，从而使原本单调的 Internet 世界变得图文并茂。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。WWW 和 Mosaic 的出现，使 Internet 进入了普通百姓的家庭。

WWW 页面虽然已经拥有了图文和声音，但仍然是静态的，也不具备交互性，而且不断改变的 HTML 的种类和过多的浏览器也给 WWW 造成了很大的混乱。要让页面拥有动态的画面，并能进行交互，一种很自然的想法就是在 WWW 页面中嵌入一段程序。考虑到 Internet 是由数以千计不同种类的计算机组成的，编制该程序的语言必须具有平台无关性。由于网络传输带宽的限制，这种语言必须简练且其支撑环境必须很小。另外，由于是在网络环境中使用，对安全性也提出了很高的要求。而 Java 语言恰好能满足这些要求。看准了这一点后，Sun 公司决定把 Java 语言引入 Internet。

Java 被引入到 Internet 后，Internet 的面貌起了很大的变化。首先，Web 页面不再是静止和呆板的了。用户现在可以在页面上看到动画按钮、滚动的和移来移去的大标题、闪闪发光的霓虹灯等。而且，Java 语言使 Web 实现了真正的交互，例如联机小游戏等。最重要的是，利用 Java 编制成的小程序可以在全世界几乎任意一台与 Internet 相连的计算机上运行，而不用考虑这台计算机上运行的是什么操作系统。Java 使 Internet 有了崭新的活力。

1.1.3 Java 语言的特点

在前面两节中，本书介绍了 Java 语言的产生和与 Internet 的关系，其中提到了 Java 语言的一些特点，如小巧、稳定、与平台无关等。而在 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、健壮的、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、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

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++ 的复杂性使得编写可靠程序的难度很高，不能实现他们的目标。所以，Gosling 等人设计 Java 时省略了 C++ 中所用的一些罕见的、难以理解的和极易混淆的特性。简要列举如下：

- 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，将没有被引用到的内存回收给系统使用，使用户无须担心内存管理之类的事情。

- 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。

- 由于 Java 是面向对象的，而且不用像 C++ 那样考虑和 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。

- Java 不支持 goto 语句，同时增加了 break 和 continue 的功能，可以允许多层循环的 break 或 continue，不但避免了滥用 goto 的可能性，也保存了 goto 的优点。

- 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。类似的，Java 还取消了操作符重载，使它更简单易学。

- Java 取消了自动类型转换，使编译器的检查更严格，可以防止一些人为的程序错误。

除了以上的一些特性以外，Java 和 C++ 很类似，因此，熟悉 C 和 C++ 的程序员可以很快掌握这门语言。

2. 面向对象

Java 是一种面向对象的语言。它的所有数据类型都有相应的类，整个语言都是基于对象的。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，即模块化的性质和信息隐藏的能力。
- 多态性，即不同的对象对同一种信息，可以按照对象本身的性质进行响应。
- 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，即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，有许多都不能算是完全的面向对象的语言。例如，Visual Basic 中没有实现封装性，C++中没有实现动态绑定。而 Java 则很好地实现了这 4 个特点。

3. 分布式

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 进行服务器编程时获得非常高的效率。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，增加了动态可扩充性。

4. 健壮

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任何条件下，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，即需要显示的类型声明，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 是一种半编译、半解释的语言。但严格地说，Java 是被解释的。Java 程序执行的流程如图 1-1 所示。

从图 1-1 中可以看出，一个 Java 程序的执行可分为两个过程，首先是 Java 源程序经 Java 编译器编译后生成字节码(Bytecode)，即.class 文件，然后再经 Java 解释器将字节码转换成实际使用的机器和操作系统上的机器码去执行。

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。

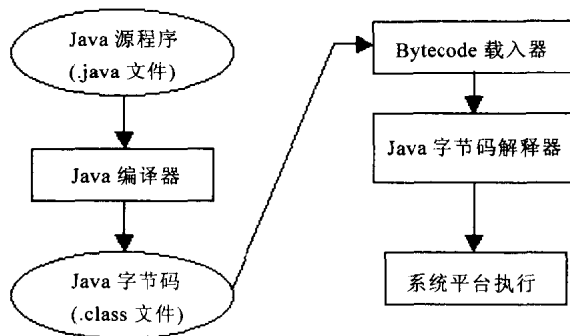


图 1-1 Java 程序执行的流程

6. 安全

由于 Java 是一种分布式的语言，必须保证从别的地方下载下来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。

- 偷窃系统或个人机密。如偷窃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，对于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，就可以终止了恶意程序的运行。

由上可以看出，Java 语言在设计时对安全性作了充分的考虑。尽管不能说 Java 是绝对安全的语言，但仍可以说它是目前最安全的程序设计语言。

7. 中性结构(体系中立)

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了 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

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++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。这种方法可以明显改善速度,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++一样优秀,而且 Java 2 与以前的版本相比又有了大幅度的提高。

10. 多线程

多线程的优点是具有更好的交互性和实时行为,这个优点在 Internet 上显得尤为重要。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,同时浏览器正在后台下载一幅图像。

Java 很好地支持了多线程。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,避免了多个线程或进程同时修改相同的数据,造成数据的内容前后不一致而导致错误。同时,也避免了发生死锁的可能。

11. 动态

在许多方面,Java 都比 C 或 C++更像一种动态的语言。经过设计,它与不断发展的工作环境有着很好的相容性。在库内,可以很容易地添加新方法以及新实例变量,