

精通Java Web 动态图表编程

钟京馗 唐 桓 编著





Java 技术大系



精通 Java Web 动态图表编程

本书不仅详述了如何利用 Java 对图形的处理能力, 通过 Java Applet、Java Servlet、JSP、JavaBean 以及借助 Java 开放源代码组织发布的 JFreeChart 及 Cewolf 动态图表生成引擎等多种方式开发 Web 动态图表, 还详细了解了如何设计 Web 动态图表生成引擎, 并将其封装成 JavaBean, 以达到高效、重复使用的目的。

本书包含以下内容:

-  Java 概述
-  Java Applet 与绘图基础
-  Java Applet 图表绘制实例
-  JSP /Servlet 运行环境的搭建
-  基于 Servlet 的 Web 图表编程
-  JSP Web 图表编程基础
-  JSP 与 Java2D Web 图表编程
-  开放源代码与 Web 图表编程
-  Web 图表生成引擎的设计思路与实现

本书是国内第一部系统阐述如何开发基于 Java 的 Web 动态图表的专著。

本书面向各级在 Java 应用程序中实现 Web 动态图表功能的 Java 及 Web 程序设计人员。

图书分类: 编程语言 > Java 技术

ISBN 7-121-01488-2



9 787121 014888 >

网上订购: www.dearbook.com.cn
第二书店·第一服务



责任编辑: 李 冰
责任美编: 张子建
崔云霞

本书贴有激光防伪标志, 凡没有防伪标志者, 属盗版图书。

ISBN 7-121-01488-2 定价: 55.00 元

Java 技术大系

精通Java Web动态图表编程

钟京馗 唐 桓 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是国内第一部系统阐述如何开发基于 Java 的 Web 动态图表的专著。

本书不仅详述了如何利用 Java 对图形的处理能力,通过 Java Applet、Java Servlet、JSP、JavaBean 以及借助开源软件 JFreeChart 及 Cewolf 动态图表生成引擎等多种方式开发 Web 动态图表,还详解了如何设计 Web 动态图表生成引擎,并将其封装成 JavaBean,以达到高效、重复使用的目的。本书不但剖析理论,更强调实际运用;针对编程人员开发过程中遇到的技术难点,提供了典型的实例,逐一剖析,帮助读者快速掌握和精通 Java Web 动态图表编程思想和技巧,提高编程能力和水平。全书共分 9 章。

本书面向各级 Java 应用程序中实现 Web 动态图表功能的 Java 及 Web 程序设计人员。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

精通 Java Web 动态图表编程 / 钟京旭, 唐恒编著. —北京: 电子工业出版社, 2005.9

(Java 技术大系)

ISBN 7-121-01488-2

I. 精… II. ①钟… ②唐… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2005) 第 071158 号

责任编辑: 李 冰

印 刷: 北京智力达印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 32.75 字数: 681 千字

印 次: 2005 年 9 月第 1 次印刷

印 数: 5000 册 定价: 55.00 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。联系电话:(010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

本书向读者展示如何使用 Java Applet、Java Servlet、Java Server Pages (JSP)、JavaBean 以及开放源代码的 Web 图表生成引擎——JFreeChart 和 Cewolf 来开发奇妙的 Web 动态图表应用程序——以一种跨平台、小巧、结构清晰的模式在 Web 上生成动态图表。

随着计算机网络及编程技术的发展,使得越来越多的应用程序被移植成以 Web 应用程序(浏览器/服务器)的方式,工作在因特网/局域网环境之中。网络开发人员发现,某些传统 IDE 开发环境,例如:在 Dephi 和 VB 中可以轻而易举地实现的图形界面,而在 Web 程序中却很难实现,其中一个典型的例子就是 Web 图表(Chart)的生成与处理。即使 Web 程序(ASP、ASP.NET 等)能够处理一些图表,也基本是采用 ActiveX 或者 COM 组件的结构。这种方法有三大弊端:

- 开发 ASP 比较简单,但开发 ActiveX 和 COM 却很难;
- 基于 ActiveX 和 COM 构架的 Web 应用程序只能用于微软 Windows 的 Web 服务器 IIS 环境下,移植性差;
- 由于 ActiveX 和 COM 与 Windows 深度紧密结合,在带来强大功能的同时,也一并带来了 Windows 固有的缺陷——安全性和稳定性差。

PHP 可以用于 Web 图表处理,它具有跨平台、安全性能高、兼容性及扩展性强的优点,但也有其固有的缺陷,具体表现为以下几点:

- 效率不高,基于解释型,而非编译型,这一点与 ASP 类似。ASP.NET 吸取了 JSP 的优点,属于编译型,大大提高了运行速度和效率;
- 安装复杂,每一种扩展模块不完全是由 PHP 自身来完成。当图形需要 GD 库,使用类似的扩展库时,安装和调试是 PHP 的一大问题;
- 因为是开发源代码产品,所以缺乏企业级的商业支持。

JSP 的出现,使得上述弊端不复存在。随着 Java 对二维图形及三维图像的处理能力越来越强大,利用 JSP 来简单、高效地开发一个 Web 图表应用程序已经不是一件难事了。为了展示如何编写 Web 图表应用程序,本书中,我们不仅提供了多个 JSP 图表生成的实例,而且还将从以下两个方面加以说明 Web 图表编程是如何实现的。

1. 利用 Java 自身对图形的处理能力,由开发者编写代码来生成 Web 动态图表。将以大量的例程,不同的方式(例如:Java Applet、Java Servlet、JSP、JavaBean),从不同的角度来展示如何编写 Web 图表的 Java 程序。

2. 借助第三方的图表生成引擎来完成图表,主要利用一些 Java 开放源代码组织开发的作品,例如:

- 由 www.jfree.org 推出的 JFreeChart;
- 由 cewolf.sourceforge.net 推出的 Cewolf。

Cewolf 是基于 JFreeChart 的二次开发。实质上，是基于 JFreeChart 的应用。JFreeChart 没有免费提供开发文档，只有英文版的相关资料，因此 JFreeChart 和 Cewolf 在国内的应用受到了一定的限制。

根据我们使用及研究 JFreeChart 和 Cewolf 的经验，将在本书中提供大量的例程和详尽的讲解，帮助读者轻松地掌握上述两种非常优秀的、基于 Java 的 Web 图表生成引擎。相信阅读本书，并加以必要的练习，读者将会发现，基于 JSP 的 Web 图表编程是可以轻松完成的。

本书结构及内容

本书例程丰富、代码简洁、结构清晰、讲解准确、图文并茂。共分九章，各章具体内容如下：

第 1 章 Java 概述

本章主要概述 Java 历史及其发展、Java 开发环境的建立，以及编写一个简单的 Java 应用程序。

第 2 章 Java Applet 与绘图基础

本章简要介绍 Applet 绘图基础，例如：图形环境、图形对象、基本的字体及颜色控制，以及如何使用 Applet 绘制一些基本的几何图形。通过本章的学习，读者将掌握基本的 Java 图像编程方法。

第 3 章 Java Applet 图表绘制实例

本章综合利用绘制基本几何图形的方法来开发以下常见的图表：

➤ 垂直柱状图

➤ 饼图

探讨如何以图表的方式生成以下内容：

➤ 在 Web 动态图表中加载外部图像文件

➤ 支票

➤ 如何从 HTML 文档中获取参数生成动态图表

通过本章的学习，读者将会理解任何复杂的图表都可以分解成一些基本的几何图形；通过排列组合不同的外部图像文件，以及基本的几何图形，就可以生成复杂的图表。

第 4 章 JSP/Servlet 运行环境的搭建

本章主要介绍 JSP Web 服务器的安装。在运行 JSP/Servlet 之前，首先需要安装 JSP/Servlet 的运行环境，它就是我们平常所说的，支持 JSP/Servlet 的 Web 服务器（容器）。

目前有很多支持 JSP/Servlet 的 Web 服务器，我们先介绍两种免费的、高性能的、适合中小企业 JSP/Servlet 的 Web 服务器——Tomcat 和 Resin，并将详细阐述 Tomcat 实现

JSP/Servlet 的运行机制。此外，介绍一种企业级的 J2EE 服务器——Weblogic，学习如何在这些服务器中部署及发布基于 JSP/Servlet 的 Web 图表应用程序。

第 5 章 基于 Servlet 的 Web 图表编程

本章讲述 Web 请求/响应机制（get 和 post）、如何部署 Servlet、如何利用 Servlet 生成 Web 动态图表，并且提供了模拟网站投票统计、生成登录验证码、普通/3D 甘特图等 Servlet 实例。

第 6 章 JSP Web 图表编程基础

本章讲述 JSP 绘图基础。在 JSP 的环境下，如何使用 Java.awt.Graphics 类的各种方法，包括绘制直线、文本（字符串）、矩形、椭圆和圆、圆弧、多边形、折线，以及如何加载外部图像文件，等等。

在绘制基本几何图形的基础上，我们将以圆柱体和立方体为例，展示如何通过绘制多个多边形并将其组合成复杂几何图形的方法；利用 Java.imageio 包中 ImageIO 类的支持，调用 ImageIO 来执行诸如加载图像，以及对图像进行编码输出等工作。

第 7 章 JSP 与 Java2D Web 图表编程

本章阐述如何利用 Java2D API 对于高级二维图形的处理能力，例如：笔划属性的设定、图形填充、渐变色、图像的透明度、字体处理、图形边缘的反锯齿、图形对象的转换以及变形，等等，利用 B/S 应用程序中生成 Web 图表的方法，并以二维及三维图表的形式，包括折线图、水平直方图、垂直柱状图、堆栈图、饼图，以及二次曲线来进行讲解和说明。

提供了一个绘制复杂图表的例程——股市大盘指数走势图。通过本章的学习，读者可以不借助任何第三方的程序，编写生成各种风格、复杂的 Web 图表应用程序。

第 8 章 开放源代码与 Web 图表编程

本章详细讲述了 JFreeChart 和 Cewolf 这两个 Web 图表生成引擎的安装、配置及使用方法。提供了在 JFreeChart 和 Cewolf 的环境下生成以下类型图表的完整例程及讲解：

- 普通/3D 水平及垂直直方图、普通/3D 饼图、普通/3D 堆栈图
- 线段（曲线）图、区域图、时序图、蜡烛图、移动平均线图
- 罗盘图、温度计图、速度表图、信号图
- 甘特图、多轴图表、组合图表

此外，还包括如何在 Cewolf 的环境中，创建自定义绘制属性的图表。

第 9 章 Web 图表生成引擎的设计思路与实现

本章讨论如何开发一个图表生成引擎；如何设计图表生成引擎（封装成 JavaBean 的形式）以达到高效、重复使用的目的。图表生成引擎的设计难点是什么？该如何处理？如何优化引擎？我们将提供具体的实例加以讲解。通过本章的学习，读者将会感受到，图表引擎的设计原来可以这么轻松地实现！

本书的适用对象

本书适合从事 Java 及 Web 编程的开发人员，在 JSP/Servlet 应用程序中需要提供图表显示及处理功能的 JSP/Servlet 程序开发者，以及编程爱好者阅读。对于初学入门的应用开发人员，可以作为自学的实践指导书；对于已经从事 Web 编程工作的人员，可以作为一本供随时翻阅查询的案头参考书。

建议

我们建议读者在学习期间避免使用可视化的开发工具，例如：JBuilder、JCreator、KAWA、Visual Cafe、Eclipse、IntelliJ IDEA、BEA WebLogic Workshop、Oracle JDeveloper 等，所有工作都可以使用 JSP 服务器（Tomcat/Resin）+ WWW 浏览器（IE/Mozilla/Firefox）+ 文本编辑器（EditPlus）来完成。附录中，我们向读者介绍了一种优秀的 Java IDE 开发工具——Gel。利用 Gel，可以方便地创建、编辑、编译及运行 Java/JSP/Servlet 应用程序。

为了达到最佳的学习效果，我们建议读者在阅读本书的同时，亲自动手按照本书的示例去操作。所有的源程序都可以在 www.broadview.com.cn 网站上下载，但建议读者最好对照源程序自己动手录入一次，这样可以加深理解和记忆。

因本书的例程非常丰富，为节省篇幅，故基本上不列出每个例程的完整清单。本书详尽地讲解了每个源程序的设计思路、运行机制及相对应的核心代码段。建议读者在阅读本书时，首先用 EditPlus 之类的文本编辑器来打开相应例程的源程序，其次阅读其完整的源程序，最后再阅读本书的讲解，相信这样会让读者取得事半功倍的效果。

致谢

本书由钟京旭（City University of New York）执笔主编和最后定稿。唐桓（University of Bridgeport）参与本书例程的编写与调试工作。

本书出版之际，特别鸣谢钟志祥、李美行、朱晓蕾、朱正才、胡元妹、黄桂玉、武冠军、唐光富、彭燕给予的大力支持与协助。

在编写本书的过程中，我们尽力保持内容的科学性、先进性和实用性，同时力求做到讲解原理，深入浅出，语言通俗易懂。但鉴于作者学识有限，不足之处在所难免，见仁见智，恳请广大读者指正。我们将在本书的再版中使其更臻完善。

钟京旭

2005 年 9 月于纽约

目 录

第 1 章 Java 概述.....	1	2.8.1 绘制普通（空心）椭圆	46
1.1 Java 简介	1	2.8.2 绘制实心椭圆	48
1.1.1 Java 发展简史	1	2.8.3 绘制 3D 椭圆	49
1.1.2 Java 的体系	2	2.8.4 绘制圆柱体	50
1.1.3 Java 的优点	3	2.9 绘制圆弧	54
1.2 Java 开发环境的搭建	4	2.9.1 绘制普通（空心）圆弧	55
1.2.1 Java 运行环境的要求	4	2.9.2 绘制实心圆弧	56
1.2.2 Java 的安装和配置	6	2.9.3 绘制 3D 圆弧	57
1.3 Java/JSP 开发工具	8	2.10 绘制多边形	59
1.3.1 EditPlus 简介	9	2.10.1 绘制空心多边形	59
1.3.2 UltraEdit 简介	11	2.10.2 绘制实心多边形	61
1.3.3 其他 Java/JSP 开发工具	13	2.10.3 绘制折线	61
1.4 第一个 Java 程序	15	2.10.4 绘制三角形（箭头）	62
1.5 本章小结	17	2.10.5 绘制平行四边形及立方体	69
第 2 章 Java Applet 与绘图基础	18	2.11 图像的载入与显示	78
2.1 Java Applet 概述	18	2.11.1 在 Applet 中加载和 显示图像	78
2.2 Java Applet 工作流程	20	2.11.2 使用 MediaTracker 加载并 显示图像	84
2.2.1 组件与容器	20	2.11.3 使用双缓冲技术绘制图像	89
2.2.2 Applet 的编译与执行	22	2.12 本章小结	93
2.2.3 包含 Applet 的 HTML	23	第 3 章 Java Applet 图表绘制实例	94
2.3 绘制文本（字符串）	23	3.1 Java Applet 生成 Web 动态图表	94
2.4 绘制线段	29	3.1.1 垂直柱状图	95
2.5 色彩管理	31	3.1.2 饼图	102
2.5.1 色彩基础	31	3.2 Java Applet 生成单据	109
2.5.2 Java 的色彩管理	34	3.2.1 带徽标的 Web 动态图表	110
2.6 字体控制	36	3.2.2 支票的生成	111
2.7 绘制矩形	40	3.3 从 HTML 文档获取参数 生成动态图表	123
2.7.1 绘制普通矩形	40		
2.7.2 绘制 3D 矩形	43		
2.8 绘制椭圆	46		

3.3.1 传递参数的 HTML 文档	123	5.5 Servlet 生成 Web 投票统计图	191
3.3.2 获取参数并生成图表	124	5.6 Servlet 生成登录验证码	198
3.4 本章小结	126	5.6.1 Servlet 生成登录验证码	
第 4 章 JSP/Servlet 运行环境		实例 1	199
的搭建	128	5.6.2 Servlet 生成登录验证码	
4.1 Tomcat 的安装和配置	128	实例 2	209
4.1.1 Tomcat 的安装	129	5.7 Servlet 高级设置	214
4.1.2 测试第一个 JSP 程序	132	5.7.1 Servlet 初始化参数	214
4.1.3 配置 Tomcat	132	5.7.2 Servlet 加载优先级	216
4.1.4 转换后的 JSP 页面文件	145	5.7.3 Servlet 映射	217
4.2 Resin 的安装和配置	147	5.8 Servlet 绘制甘特图	218
4.2.1 Resin 的安装	147	5.9 Servlet 绘制 3D 甘特图	222
4.2.2 Resin 的配置	149	5.10 本章小结	228
4.3 BEA Weblogic 的安装和配置	150	第 6 章 JSP Web 图表编程基础	229
4.3.1 BEA Weblogic 的安装	150	6.1 JSP 概述	230
4.3.2 BEA Weblogic 的配置	152	6.1.1 JSP 运行机制	231
4.3.3 测试 BEA Weblogic		6.1.2 JSP 的优点	233
的配置	155	6.2 JSP 语法简介	234
4.3.4 部署第一个 Web		6.2.1 JSP 文件结构	234
应用程序	156	6.2.2 JSP 文件中的元素简介	240
4.4 本章小结	157	6.3 JSP 调用 Servlet 生成	
第 5 章 基于 Servlet 的 Web		动态图表	257
图表编程	158	6.3.1 JSP 生成验证码	257
5.1 Servlet 简介及其构架	158	6.3.2 JSP 生成甘特图	258
5.1.1 Servlet 的特点	158	6.3.3 JSP 其他相关知识	258
5.1.2 Servlet 的接口	159	6.4 JSP 生成基本动态图表	260
5.1.3 HttpServlet 类简介	160	6.4.1 JSP 绘制文本和线段	260
5.1.4 HttpServletRequest 接口	161	6.4.2 JSP 与字体控制	266
5.1.5 HttpServletResponse 接口	162	6.4.3 JSP 绘制矩形	273
5.2 Servlet 处理 HTTP get 请求	163	6.4.4 JSP 绘制椭圆	275
5.3 Servlet 处理包含数据的 HTTP		6.4.5 JSP 绘制圆弧	276
get 请求及解决中文乱码	177	6.4.6 JSP 绘制多边形和折线	277
5.4 Servlet 处理 HTTP post 及包含		6.4.7 JSP 绘制三角形	277
数据的 post 请求	187	6.4.8 JSP 绘制平行四边形和	
		立方体	280

6.4.9 JSP 加载并显示图像	281	7.8.3 AlphaComposite 应用实例	352
6.5 本章小结	282	7.9 图形对象的转换	357
第 7 章 JSP 与 Java2D Web		7.9.1 图形对象转换 (transformation)	
图表编程	283	概述	357
7.1 Java2D 概述	283	7.9.2 平移 (translation)	359
7.2 Java AWT 与 Java2D	285	7.9.3 旋转 (rotation)	360
7.3 Java2D 与填充属性	287	7.9.4 缩放 (scale)	362
7.3.1 设置填充属性	287	7.9.5 扭曲 (shear)	364
7.3.2 填充属性的用法	287	7.10 图形渲染线索	366
7.4 Java2D 与笔划属性	291	7.11 Java2D 与高级文本处理	368
7.4.1 线段端点的形状风格	292	7.11.1 空心文本	368
7.4.2 线段转折处的形状风格	292	7.11.2 弯曲文本	369
7.4.3 虚线风格	293	7.11.3 单行长文本自动分行	371
7.4.4 BasicStroke 构造器	294	7.12 Java2D 创建复杂图形	373
7.4.5 Java2D Web 图表实例		7.12.1 Area	374
之折线图	294	7.12.2 曲线	375
7.5 创建基本 Java2D 图形	309	7.12.3 通用路径	377
7.5.1 Java2D 图形 (Shape)		7.13 Web 图表实例解析	379
接口概述	309	7.13.1 透明 3D 饼图	379
7.5.2 Point2D	310	7.13.2 股市指数走势图	381
7.5.3 Line2D	311	7.14 本章小结	391
7.5.4 Rectangle2D	312	第 8 章 开放源代码作品与 Web	
7.5.5 RoundRectangle2D	314	图表编程	392
7.5.6 Java2D Web 图表实例之		8.1 开放源代码作品简介	392
柱状图	315	8.2 JFreeChart 与 JSP 图表编程	394
7.5.7 Ellipse2D	333	8.2.1 JFreeChart 简介	394
7.5.8 Arc2D	334	8.2.2 JFreeChart 的安装及其	
7.6 Java2D 实例饼图类图表	337	核心类	395
7.6.1 普通饼图	338	8.2.3 JFreeChart 生成直方图表	398
7.6.2 圆圈图	342	8.2.4 JFreeChart 生成饼型图表	411
7.6.3 3D 饼图	344	8.2.5 JFreeChart 生成线段图表	416
7.7 图形重叠	346	8.2.6 JFreeChart 生成区域图表	420
7.8 alpha 复合	348	8.2.7 JFreeChart 生成时序	
7.8.1 alpha 复合概述	348	(Time Series) 图表	424
7.8.2 AlphaComposite 类的使用	350	8.2.8 JFreeChart 生成甘特图表	430

8.2.9 JFreeChart 生成多轴 (Multiple Axis) 图表	432	8.5 本章小结	473
8.2.10 JFreeChart 生成组合 (Combined Axis) 图表	435	第 9 章 Web 图表生成引擎的设计 思路与实现	475
8.2.11 JFreeChart 生成其他类型 的图表	441	9.1 Web 动态图表生成引擎的 设计思路	475
8.3 JFreeChart 与 Servlet 图表编程	444	9.2 Web 动态图表生成引擎的 设计模型	480
8.3.1 简单的 Servlet 图表编程	444	9.2.1 生成普通线段图的 JavaBean	480
8.3.2 交互式 Servlet 图表编程	446	9.2.2 生成 3D 线段图的 JavaBean	483
8.4 Cewolf 与 JSP 图表编程	448	9.2.3 生成普通直方图的 JavaBean	484
8.4.1 Cewolf 简介	448	9.2.4 生成 3D 直方图的 JavaBean	487
8.4.2 Cewolf 的下载与安装	448	9.2.5 生成普通饼图的 JavaBean	488
8.4.3 Cewolf 生成直方图表	450	9.2.6 生成 3D 饼图的 JavaBean	490
8.4.4 Cewolf 生成基于 DefaultCategory Dataset 数据集的图表	454	9.3 数据分离	491
8.4.5 Cewolf 生成饼图	460	9.3.1 创建及调用 CategoryDataset 类数据集对象	491
8.4.6 Cewolf 生成基于 XYDataset 数据集的图表	462	9.3.2 创建及调用 PieDataset 类 数据集对象	495
8.4.7 Cewolf 生成基于 OHLCDataset 数据集的图表	465	9.4 引擎的优化概述	496
8.4.8 Cewolf 生成甘特图表	466	9.5 本章小结	498
8.4.9 Cewolf 生成信号图表	467	附录 Gel 使用指南	499
8.4.10 Cewolf 生成速度图表	468		
8.4.11 Cewolf 生成 OverLay 类型 的图表	468		
8.4.12 Cewolf 生成组合图表	470		
8.4.13 生成自定义风格的 Cewolf 图表	472		

第 1 章 Java 概述

Java 是美国 Sun 微系统公司 (Sun Microsystems, Inc.) 开发, 近年来飞速发展的一项崭新的计算机技术。

1.1 Java 简介

Java 既是一种程序设计语言, 也是一个完整的平台。作为一种程序语言, 它简洁、面向对象、安全、健壮, 以及适用于 Internet 技术; 而作为一个平台 (JRE, Java Runtime Environment, Java 运行环境或者说是 Java 虚拟机), 对于符合 Sun 公司 Java 标准的应用程序, 都可以在 Java 平台上正确运行, 与程序运行的操作系统无关。

Java 为什么能够成为目前编写 Web 应用程序的首选设计语言呢? 并且具备跨平台的能力呢? 我们先简单地了解一下 Java 的发展历史和体系结构。

1.1.1 Java 发展简史

Java 起初并非叫做 Java, 而是叫做 Oak。早期是为了嵌入式系统而设计的一项产品。20 世纪 90 年代初期, Sun 公司预测微处理器的发展将会对家用电器以及智能化电子消费设备起到决定性的作用, 于是, 在 1990 年 12 月, Sun 公司就由 Patrick Naughton、Mike Sheridan 和 James Gosling 成立了一个叫做 Green Team 的小组。其主要目标就是开发一种分布式系统架构, 使其能够在智能化电子消费设备作业平台上执行。例如, PDA、手机、信息家电 (IA, Internet/Information Appliance) 等。一开始, 准备采用 C++, 但 C++ 太过复杂, 安全性差。最后, 基于 C++ 开发出一种新的语言 Oak, Oak 是一种用于网络、精巧而安全的语言。

1992 年, Green Team 发表了一款名叫 Star Seven (*7) 的机器, 它有点像现在我们熟悉的 PDA, 但市面上的 PDA 几乎都不是它的对手, 更不要说是早在 10 年前那个计算机还不普及的时代了。Java 语言的前身 Oak 就这样诞生了, 主要目的是用来编写在 Star Seven 上的应用程序。为什么叫 Oak 呢? 原因是 James Gosling 办公室的窗外, 正好有一棵橡树 (Oak), 就取了这个名字。至于为什么 Oak 又会改名为 Java 呢? 这是因为在 Oak 注册商标时, 发现已经被另外一家公司使用了。那要取什么新名字呢? 工程师们边喝着

咖啡边讨论着，看着手中的咖啡，突然灵机一动，就叫 Java 好了。就这样它就变成了大名鼎鼎、如日中天的 Java 了。

但是，Green Team 的项目发展却不尽如人意。智能化电子消费设备的市场发展远远低于 Sun 的预期设想。Sun 公司曾依此投标一个交互式电视项目，结果被 SGI 打败了。可怜的 Oak 几乎无家可归，就在举步维艰，随时会被取消的时刻，情况却发生了巨大变化。1993 年，因特网在美国开始大规模的发展，基于因特网的 WWW 也爆炸性地流行起来。Sun 公司发现，Green Team 的项目研究成果——Java 似乎天生就是为因特网而诞生的，因为恰巧就是 Mark Andreessen 开发的 Mosaic 和 Netscape 启发了 Oak，他们利用 Java 开发了 HotJava 浏览器，得到了 Sun 公司首席执行官 ScottMcNealy 的支持，触发了 Java 进军 Internet。

于是，Java 第一次以 Applet 的形式在因特网上出现了。Applet 不但使 WWW 页面显示静态的内容，而且可以显示动态的内容和动画。同时实行了本地计算机，从远程联网的服务器上下载资料并正确地显示出来。这些技术在当时引起了巨大的震撼，促使 Java 在因特网上得以飞速地发展。1995 年 5 月 23 日，Sun 在 SunWorld'95 上正式发布 Java 和 HotJava 浏览器。

1.1.2 Java 的体系

Java 发展到今天，已从编程语言发展成为全球第一大通用开发平台。Java 技术已被计算机行业主要公司所采纳。1999 年，Sun 公司推出了以 Java2 平台为核心的 J2EE、J2SE 和 J2ME 三大平台。随着三大平台的迅速推进，全球形成了一股巨大的 Java 应用浪潮。

1. Java 2 Platform, Micro Edition (J2ME)

Java 2 平台微型版。Sun 公司将 J2ME 定义为“一种以广泛的消费性产品为目标、高度优化的 Java 运行环境”。自 1999 年 6 月在 JavaOne Developer Conference 上声明之后，J2ME 进入了小型设备开发的行列。通过 Java 的特性，遵循 J2ME 规范开发的 Java 程序可以运行在各种不同的小型设备上。

2. Java 2 Platform, Standard Edition (J2SE)

Java 2 平台标准版，适用于桌面系统应用程序的开发。本书例程就是利用 J2SE 5.0 版的相关图形 API 包来开发的。

3. Java 2 Platform, Enterprise Edition (J2EE)

J2EE 是一种利用 Java 2 平台来简化企业解决方案的开发、部署和管理等相关复杂问题的体系结构。J2EE 技术的核心就是 Java 平台或 Java 2 平台的标准版，J2EE 不仅巩固了标准版的许多优点，例如：“一次编写、随处运行”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，同时还提供了对 EJB（Enterprise JavaBeans）、Java Servlets API、JSP（Java Server Pages），以及 XML 技术的全面支持。

本书第 4 章，阐述了如何将我们开发的 JSP 图表应用程序，在 J2EE 平台上进行部署和管理。

1.1.3 Java 的优点

Java 是一种面向对象、分布式、解释、健壮、安全、可移植、性能优异，以及多线程的语言。下面简单介绍其中的几个优点。

1. Write Once, Run Anywhere

“一次编写，随处运行”。这是程序设计师们喜爱 Java 的原因之一，核心就是 JVM（Java 虚拟机）技术。

编写好一个 Java 程序，首先，要通过一段翻译程序，编译成一种叫做字节码的中间代码。然后经 Java 平台的解释器，翻译成机器语言来执行——平台的核心叫做 JVM。Java 的编译过程与其他语言不同。例如，C++在编译的时候，是与机器的硬件平台信息密不可分的。编译程序通过查表将所有指令操作数和操作码等，转换成内存的偏移量，即程序运行时的内存分配方式，以保证程序运行。而 Java 却是将指令转换成为一种扩展名为 class 的文件，这种文件不包含硬件的信息。只要安装了 JVM，创立内存布局后，通过查表来确定一条指令所在的地址，这就保证了 Java 的可移植性和安全性。

上述 Java 程序的编译和运行流程，如图 1.1 所示。

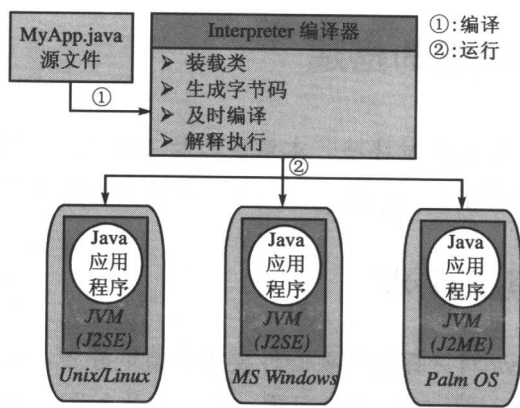


图 1.1 Java 的编译和运行流程

2. 简单

纯粹的面向对象，加上数量巨大的类所提供的方法（函数）库的支持，使得利用 Java 开发各种应用程序，可以说是易如反掌。此外，在程序除错、修改、升级和增加新功能等方面，因其面向对象的特性，使得这些维护也变得非常容易。

3. 网络功能

Java 可以说是借助因特网而重获新生的，自然具备编写网络功能的程序。不论是一般因特网 / 局域网的程序，如 Socket、Email、基于 Web 服务器的 Servlet、JSP 程序，甚至连分布式网络程序，如 CORBA、RMI 等的支持也是非常丰富的，使用起来也很方便。

4. 资源回收处理（Garbage Collection）

Garbage Collection 是由 JVM 对内存实行动态管理的。程序需要多少内存、哪些程序的内存已经不使用了，需要释放归还给系统，这些烦琐且危险的操作全部交由 JVM 去管理。让我们能够更专心地编写程序，而不需要担心内存的问题。内存的统一管理，对于跨平台也有相当大的帮助。

5. 异常处理（Exception）

为了使 Java 程式更稳定、更安全，Java 引入了异常处理机制。能够在程序中产生异常情况的地方，执行相对应的处理，不至于因突发或意外的错误造成执行中断或是死机。通过这种异常处理，不仅能够清晰地掌握整个程序执行的流程，也使得程序的设计更为严谨。

1.2 Java 开发环境的搭建

现在我们开始搭建 Java 开发环境。本章仅涉及 J2SDK 开发环境的安装与调试，并不涉及 JSP 及 J2EE 服务器的安装与调试，这部分内容将在第 4 章进行说明。

1.2.1 Java 运行环境的要求

首先，需要了解 Java 对计算机主要硬件要求的最低配置，如表 1.1 所示。该表右边部分为笔者写作本书时所使用的计算机配置。因为涉及大量在计算机内存中进行绘制各种图表的运算与操作，建议使用较高配置的计算机以及尽可能多的内存，以便流畅地运行本书中的程序。

表 1.1 Java 的计算机硬件配置

硬 件	最低要求	笔者的配置
CPU	Intel 或者兼容微处理器, 奔腾 166MHz 及其以上	AMD Athlon 3200+
内存	最低 32MB, 可以运行图形界面的 Java Application; 最低 48MB, 可以运行 Java Applet; 内存不足, 将会 导致系统 (尤其是服务器) 的性能大幅下降	512 MB DDR, Kingston 内存 (如果读者希望把本书中的例程运行在 J2EE 的服务器, 如 WebLogic, 最好不低于 256 MB)
硬盘	242 MB	80 GB
图形显示卡	无要求	Nvidia Ti 4200 64 MB DDR
网卡	普通网卡	3Com OfficeConnect 10/100 兆位网卡

其他部件, 如图形显示卡、光驱及显示器等, Java 并没有特别的要求。因为本书涉及图形的实时生成, 而 Java 的图形生成是通过图形显示卡调用 OpenGL 功能来加速图像的渲染和处理的。因此, 如果系统拥有一块高质量的图形显示卡, 将会大大地提高系统的性能。目前市面上的图形显示卡, 无论是在计算机主板上集成的显卡, 还是普通的独立图形显示卡, 都具有强大的图形处理功能, 足以满足流畅地运行本书中的所有例程。

Java 具有跨平台的特点, 支持所有的主流操作系统, 如下所示。

1. 微软公司 Windows 系列

- Windows 98 / Me
- Windows NT
- Windows 2000 / Windows 2000 Server
- Windows XP
- Windows 2003 Server

2. UNIX 系列

- AIX, IBM 的 UNIX, 是根据 SVR2 以及一部分 BSD 延伸而来, 加上各种硬件的支持。具备特有的系统管理功能 (SMIT, 系统管理接口工具)。
- SunOS (680x0, Sparc, i386): 基于 4.3BSD, 包含许多来自 System V 的东西。Sun 公司开发的 UNIX 操作系统, 该系统对 UNIX 的贡献是: NFS, OpenLook GUI 标准, 现演变为 Solaris。
- HP-UX (HP): 惠普公司的 UNIX。
- SCO UNIX。

3. 类 UNIX 系列

- Linux: 包括各种版本的 Linux, 如 RedHat、Turbo 和 Mandrake Linux。
- FreeBSD: 由美国加州大学伯克利分校计算机系统研究小组设计和维护。