

全面突破网络编程技术丛书



JSP 网站开发技术

徐进明 主编
但正刚
谢锋波 等 编著



清华大学出版社

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



JSP 网站开发技术

徐进明 主编
但正刚 谢锋波 等编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

JSP(JavaServer Pages)是由 Sun Microsystems 公司倡导,许多公司参与一起建立的一种动态网页技术标准。相对于传统的网页制作技术而言,它有着明显的优点。JSP 不像 CGI、ISAPI 和 NSAPI 一样难于编写和维护,同时不像 PHP 一样只能适应中小流量的网站,而且具有良好的扩充性,也不像 ASP 一样受到跨平台的限制(只能运行于 Microsoft 公司开发的 IIS 和 PWS 上)。JSP 体现了当今最先进的网站开发思想。

本书由浅入深地介绍了 JSP 的基本原理与相关技术,并根据作者自己的开发经验,在书中给出了两个较大的开发实例。在本书中,作者给出了实例中所有的源代码,并对源代码进行了详细的解说。如果需要电子版的源代码,读者可以到如下站点下载:<http://ftp.tup.tsinghua.edu.cn/Jspnetdesign>。

本书内容详尽、图文并茂,适合学习 JSP 的初中级读者,对高级用户也有很好的参考作用。

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: JSP 网站开发技术

作 者: 但正刚 谢锋波 等

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>

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 20 字数: 472 千字

版 次: 2001 年 5 月第 1 版 2001 年 5 月第 1 次印刷

书 号: ISBN 7-302-04382-5/TP·2574

印 数: 0001~5000

定 价: 28.00 元

清华大学出版社

前 言

在短短的时间里，Internet 从简单的信息提供发展到了今天的提供各种各样的交互式服务。目前有很多方法和技术可以实现交互式服务，如 CGI、ASP、PHP 等。但是，他们都不能很好地满足网络时代的需要。

CGI (Common GatewayInterface, 通用网关接口) 程序有编译性和解释性两种。编译性 CGI 程序需要编译成二进制代码方可运行，这无疑延长了开发周期。解释性 CGI 程序不需事先编译，速度上有所提高。但是二者都难于编写、维护和扩展，并且每个 CGI 程序在运行时都要在服务器上新增一个进程。如果多个用户并发地访问该程序，这些进程将消耗该 Web 服务器所有的可用资源，并且系统性能降低到极其低下的地步。

针对 CGI 程序的弱点，很多公司、团体或个人开发了嵌入式的开发技术，如 ASP 和 ISAPI 技术、PHP 技术、NSAPI 技术等。ASP 是有 Microsoft 开发的，它具有良好的扩充性，但同时它是和平台相关的，只能运行在 Windows 平台的 IIS 服务器和 PWS 服务器上。但是，由于 Unix 的健壮性和 Linux 的源码开放性，它们都广泛地应用在网站服务器中，因而，ASP 的平台相关性大大地限制了它的应用。

PHP (Personal Home Page) 程序可以运行于多种平台，但它的扩展性较差。PHP，顾名思义，只能适应中小流量的网站，目前国内的一些用 PHP 制作的较大型网站，都已显露出比较严重的问题。ISAPI 和 NSAPI 技术性能比较优越，但它们的学习门槛较高，并且也是和平台相关的。

另外的一些网站开发技术，也都有着先天的不足，从来没能成为网站开发的主流。

针对上面这些网站开发平台的弱点，Sun 公司推出了自己的 JSP (JavaServer Pages)，可以说它是真正的后起之秀。JSP 技术综合了以上各种网站开发工具的优点和长处，而避开了它们的弱点。正受到越来越多人的欢迎。

JSP(JavaServer Pages)是由 Sun Microsystems 公司倡导，许多公司参与一起建立的一种动态网页技术标准，其网址为 <http://www.javasoft.com/products/jsp>。在传统的网页 HTML 文件 (*.htm,*.html) 中加入 Java 程序片段 (Scriptlet) 和 JSP 标签 (Tag)，就构成了 JSP 网页 (*.jsp)。Web 服务器在遇到访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。程序片段可以操作数据库、重新定向网页以及发送 Email 等等，这都是建立动态网站所需要的功能。利用 JSP，所有程序操作都在服务器端执行，网络上传送给客户端的仅是得到的结果，对客户浏览器的要求很低，在客户端可以实现无 Plug-in，无 ActiveX，无 JavaApplet，甚至无 Frame。

鉴于 ASP 和 PHP 目前在国内的广泛使用，下面我们将 JSP 和 ASP、PHP 来做一个比较详细的对比。来看一下，相对于 ASP 和 PHP，JSP 技术究竟有什么优势。

- JSP 技术以 Java 语言为基础，因此天生具有平台无关性。
- 以强大的 Java API 为后盾，功能十分强大。

- 可以使用 JavaBeans 组件和自定义标签，因此具有良好的扩展性。
- 使用嵌入式的开发技术，简单易学。
- 通过视图分离技术，强化可重用性组件，大大加快了网站的开发过程。
- 不像 ASP 或 PHP 是纯粹的解释性语言，JSP 是半编译半解释语言。每个 JSP 脚本在第一次被调用时将被编译为一个 Servlet，并常驻于内存中，这显然将大大提高它的执行效率。
- JSP 技术用 Java 语言作脚本。Java 语言使开发者的工作比用其他方式更容易。Java 语言拥有内存区域管理，可以防止内存漏洞和提供对难于发现的降低应用效率的指针 Bugs 的保护。此外，JSP 还为现实世界的应用提供了强有力的例外处理能力。
- 面向对象的方法使基于 JSP 的应用程序要比基于 ASP 和 PHP 的应用程序更易于维护。
- 脚本语言对于小应用是有很多优点，但是不适合于管理大型的、复杂的应用。相反 Java 语言是结构化的，它易于建造和维护大规模的、复杂的应用。
- JSP 技术强调的组件比脚本实现更加容易，它在设计网页内容时，不会影响程序逻辑。在设计程序逻辑时，而不需改变网页内容。
- JavaBeans 模式包括像数据库访问、安全、事务完整性等完整的企业模式，并且它们与应用本身相分离。
- JSP 技术是一种开放的、跨平台的结构，所以 Web 服务器、平台和其他组件能够很容易转换，而不需影响基于 JSP 的应用程序。这使 JSP 更适合现实世界的 Web 应用程序，无论在哪儿都可以迅速转化、成长，这是它的准则。
- JSP 技术是通过 Java 开放模式开发的，它可以广泛地被其他开发工具、Web 服务器和应用服务开发商所支持。这使得它比其他技术的发展更具多元化、融入更多的智慧。

综上所述，JSP 技术有着较为明显的优势，这使得它成为未来的发展方向。正是由于这些优点，JSP 正在取代 ASP、PHP，日渐成为网站开发的主流。目前，新的网站公司都使用 JSP 作为网站开发的工具，以前的一些公司也都在使用 JSP 重新改造自己的网站。

本书是为所有对 JSP 和网站建设感兴趣的读者编写的。在阅读本书之前，要求读者有一定的 Java 基础但不要求精通，对动态网页和 Internet 也应该有所了解。如果缺乏这些知识，应该先自行参看一下有关的资料。

希望本书能为 JSP 在中国的普及尽一份力量！

全书由徐进明主编，但正刚和谢锋波执笔编写。另外，参加本书编写工作的人员还有刘涛、施红芹、王政伟、孙剑、刘成、郭秀英、林培红、张双泉、郑易、施红芳、徐泽民、蒋东等。由于编者水平有限，错误之处在所难免，欢迎批评指正，以便在下次再版时进行改进。对本书的意见和建议请发到如下的电子信箱：xujinming@263.net。

编著于清华园

2000.11

目 录

第一部分 JSP 基础知识	1
第 1 章 走近 JSP 的世界	3
1.1 JSP 的开发背景及发展历史	3
1.1.1 JSP 的开发背景	3
1.1.2 JSP 的发展历史	4
1.2 JSP 与 Java	5
1.3 JSP 与其他的动态网页技术	7
1.4 JSP 的运行方式和优点	10
1.4.1 JSP 的运行方式	10
1.4.2 JSP 的优点	11
1.5 JSP 与 Web 应用程序	12
1.6 JSP 与 JavaBeans 的结合	13
1.7 运行一个 JSP 程序	13
1.7.1 一个 JSP 应用程序	13
1.7.2 程序的剖析	15
1.8 小结	16
第 2 章 配置服务器	17
2.1 JSP 运行对系统的基本要求	17
2.2 常见的几种支持 JSP 的应用服务器的介绍	17
2.3 几个典型的配置例子	19
2.3.1 在 Windows NT 4.0 上安装 Apache+Tomcat	19
2.3.2 在 Windows2000 上安装 Apache+ApacheJserv+gnujsp	21
2.3.3 在 Linux 上安装 Apache+ApacheJserv	21
2.4 Resin 介绍	22
2.4.1 Resin 1.1 的特性	23
2.4.2 安装与启动	23
2.4.3 Resin 的运行与配置	23
2.4.4 对于 Resin 的配置	24
2.5 JSWDK 的配置和使用	25
2.5.1 JSWDK 的获取和安装	25
2.5.2 启动和关闭 JSWDK	25
2.5.3 安装自己的 Web Application	26

2.6 小结	27
第3章 JSP 的语法	28
3.1 JSP 语法概述	28
3.2 JSP 的脚本要素	30
3.2.1 注释	30
3.2.2 表达式	31
3.2.3 声明	32
3.2.4 Scriptlet (小脚本)	32
3.3 JSP 标签	33
3.3.1 Include 标签	33
3.3.2 Page 标签	35
3.3.3 Taglib 标签	37
3.4 与 JavaBeans 相关的语法标签	38
3.4.1 <jsp:useBean>标签	38
3.4.2 <jsp:setProperty>标签	40
3.4.3 <jsp:getProperty>标签	42
3.4.4 <jsp:plugin>标签	43
3.5 JSP 的动作语法	46
3.5.1 <jsp:forward>标签	46
3.5.2 <jsp:include>标签	48
3.6 JSP 的内部对象	49
3.7 小结	51
第4章 JSP 与 JavaBeans	53
4.1 组件技术简介	53
4.2 JavaBeans 的属性	54
4.2.1 Simple 属性	54
4.2.2 Indexed 属性	55
4.2.3 Bound 属性	56
4.2.4 Constrained 属性	57
4.3 JavaBeans 的事件	59
4.3.1 概述	60
4.3.2 事件状态对象 (Event State Object)	60
4.3.3 事件监听者接口 (Event Listener Interface) 与事件监听者	61
4.3.4 事件监听者的注册与注销	62
4.3.5 适配类	63
4.4 用户化	64

4.4.1	用户化器接口 (Customizer Interface)	64
4.4.2	属性编辑器接口 (Property Editor Interface)	65
4.4.3	BeanInfo 接口	65
4.5	JavaBeans 的持久化和存储格式	66
4.5.1	JavaBeans 的持久化	66
4.5.2	JavaBeans 的存储格式	66
4.6	在 JSP 中使用 JavaBeans	66
4.7	一个实例	68
4.8	小结	71
第 5 章	JSP 与其 Servlet	72
5.1	Servlet 简介	72
5.1.1	Servlet 概述	72
5.1.2	Servlet 和 JSP 的区别	73
5.1.3	Servlet 的应用范围	73
5.1.4	Servlet 的属性简介	74
5.2	Servlet 结构和生命周期	75
5.2.1	Servlet 的结构	75
5.2.2	Servlet 的生命周期	76
5.3	Servlet 的 Init 和 Destroy 方法	77
5.3.1	Init(初始化)方法	77
5.3.2	destroy 方法	78
5.3.3	Servlet 中断涉及的多线程	79
5.3.4	提供关于 Servlet 的信息	80
5.4	Servlet 简单应用示例	81
5.4.1	与客户端的交互性	81
5.4.2	一个 HTTPServlet 的简单例子	82
5.4.3	一个 HTTPServlet 处理 POST 表单的例子	83
5.4.4	在 Servlet 中显示 CGI 变量	84
5.5	Servlet 调用 JavaBeans 输出结果集	85
5.6	Servlet 和 JSP 的通信	89
5.6.1	Servlet 访问 JSP	89
5.6.2	JSP 使用 Servlet	91
5.7	Servlet 类介绍	96
5.7.1	javax.servlet 包	96
5.7.2	javax.servlet.http 包	98
5.7.3	javax.servlet.jsp 包	100
5.8	小结	102

第二部分 JSP 的提高	103
第 6 章 JSP 处理 HTML 表单及 JSP 中管理会话状态	105
6.1 HTML 表单简介	105
6.1.1 text 和 password 标签	105
6.1.2 submit 和 reset 标签	106
6.1.3 radio 标签	106
6.1.4 checkbox 标签	107
6.1.5 textarea 标签	107
6.1.6 菜单项	108
6.2 用 Servlet 操纵表单数据	108
6.3 JSP+JavaBeans 处理 HTML 表单数据	113
6.4 管理会话状态	117
6.4.1 Cookies 简介	117
6.4.2 Servlet 控制 Cookies	119
6.4.3 JavaBeans 控制 Cookies	123
6.4.4 JSP 在浏览器关闭 cookies 情况下的会话管理	129
6.4.5 Session 对象的应用	130
6.5 小结	133
第 7 章 JSP 的文件操作	134
7.1 JSP 读取文件操作	134
7.2 JSP 写入文件操作	137
7.3 JSP 追加文件操作	142
7.4 网站计数器的例子	146
7.5 小结	151
第 8 章 JSP 进行网络程序开发	152
8.1 用 JSP 编写 Internet 应用程序的要点	152
8.2 简单“日历”示例	153
8.3 发送 E-mail 的例子	156
8.4 上传文件	165
8.4.1 编写自己的上传文件程序	165
8.4.2 使用第三方组件上传文件	174
8.5 与 Java Applet 的交互	175
8.5.1 通过 HTML 页面传递 Applet 参数	175
8.5.2 用 java.net 包建立双向通讯	176
8.5.3 远程方法调用 (RMI)	177
8.5.4 使用 CORBA	179

8.6 小结	180
第 9 章 数据库知识及 JDBC 简介	181
9.1 关系数据库及 SQL 语言	181
9.1.1 数据库	181
9.1.2 关系数据库	181
9.1.3 ODBC 介绍	182
9.1.4 关系数据库与 JDBC 的联系	183
9.1.5 SQL 语言简介	184
9.2 JDBC 概述及特点	186
9.2.1 JDBC 的简介	186
9.2.2 JDBC 的目标	187
9.3 JDBC 的主要接口	189
9.3.1 连接数据库所用的 JDBC 接口	189
9.3.2 其他 JDBC 接口	190
9.4 JDBC 的使用场合	192
9.4.1 Applet (小应用程序)	192
9.4.2 Application (应用程序)	193
9.4.3 其他应用场合	193
9.5 JDBC 的使用方法和安全性	193
9.5.1 ODBC 设置	194
9.5.2 JDBC 的使用	196
9.5.3 JDBC 的安全性考虑	197
9.6 小结	199
第 10 章 用 JSP 访问数据库	200
10.1 JSP 访问数据库概述	200
10.2 用 JavaBeans 连接数据库	201
10.3 JSP 使用 JavaBeans 来操作数据库	204
10.4 用 Servlet 来操作数据库	208
10.4.1 Servlet 使用 JavaBeans 组件操作数据库	208
10.4.2 Servlet 直接连接到数据库	211
10.5 数据库访问的几个技术问题	212
10.5.1 JSP 页面中实现检索数据的分页显示	212
10.5.2 用连接池提高访问数据库的效率	215
10.5.3 用哈希表对数据库进行操作	221
10.6 留言簿 (GuestBook)	225
10.6.1 guestbook.java 和数据库表格	226

10.6.2	index.jsp	227
10.6.3	insert.jsp	229
10.6.4	lookup.jsp	231
10.7	小结	233
第三部分 JSP 编写实例		235
第 11 章 JSP 编程实例一——聊天室		237
11.1	数据库表格及 Beans 组件	237
11.1.1	数据库表格	237
11.1.2	Beans 组件	238
11.2	聊天室首页	246
11.3	注册页面	247
11.4	聊天主页面	257
11.5	管理员工作的流程	266
11.6	用户退出	274
11.7	小结	276
第 12 章 JSP 编程实例二——企业应用		277
12.1	Java 企业应用概述	277
12.2	EJB (Enterprise JavaBeans)	278
12.2.1	简介	278
12.2.2	EJB 与 JavaBeans 的区别	278
12.2.3	EJB 的角色(Role)	279
12.2.4	EJB 的三个重要接口	280
12.2.5	EJB 的类型	282
12.3	J2EE (Java2 Platform Enterprise Edition)	284
12.3.1	J2EE 技术	284
12.3.2	核心 API 与组件	287
12.3.3	J2EE 中的 EJB	288
12.4	实现一个简单例子——购物车	288
12.4.1	Beans 组件	288
12.4.2	购物车程序使用的数据库表格	294
12.4.3	购物车首页	295
12.4.4	购物车程序的演示	301
12.6	小结	303
附录 1 JSP 的语法表格		304
附录 2 HTML 参考		306

第一部分 JSP 基础知识

JSP(JavaServer Pages)是一门新兴的网络编程语言，其诞生的历史才一年多，同当初 Java 语言的诞生一样，在短短的时间内，它已经在业界得到了广泛的应用，而且获得了一致好评。

本书将本着详细准确介绍 JSP 基础的目的来编写，所以在第一部分就会对 JSP 进行一个总的介绍。主要包括以下几个方面的内容：

- JSP 概述
- 应用服务器的配置
- JSP 的基本语法
- JavaBeans 组件的介绍
- Servlet(服务器小应用程序)的介绍

第 1 章 走近 JSP 的世界

欢迎大家学习 JSP，它即将成为一门最热门的网络编程语言。JSP 全称 JavaServer Pages，它是由 Sun Microsystems 公司提出、许多公司参与建立的一种动态网页技术标准。其主要特点是在传统的网页 HTML 文件 (*.htm, *.html) 中加入 Java 程序片段 (Scriptlet) 和使用各种各样的 JSP 标签 (Tag)，构成 JSP 网页 (*.jsp)。Web 服务器在接收到客户端访问 JSP 网页的请求时，首先执行其中的程序片段，然后将执行结果以 HTML 格式返回给客户。嵌入 HTML 页面的程序片段可以完成操作数据库、重新定向网页、文件上传以及发送 E-mail 等等功能，这些正是建立动态网站所必需的功能，而且所有程序操作都在服务器端执行，网络上传送给客户端的仅是程序运行后得到的结果。通过这种方式，极大地减轻了网络负担。另外，程序对客户浏览器的要求也比较低，可以实现无 Plug-in、无 ActiveX、无 Java Applet、甚至无 Frame。

看了上面这一大段介绍，不知读者是否能够明白其中的意思。不过不要紧，通过本书的介绍，你一定会慢慢地对 JSP 有清晰的了解的。

作为本书的基础，本章主要介绍以下内容：

- JSP 的开发背景、发展历史
- JSP 与 Java
- JSP 和以前一些网络开发技术在整体上的比较
- 运行第一个比较大的 JSP 程序

1.1 JSP 的开发背景及发展历史

JSP 目前在国外已经得到了广泛的应用，在国内，因为 Java 语言普及的不足，所以尚未得到重用，但是越来越多的公司和个人开始关注 JSP 的应用前景，国内也已经出现了许多 JSP 做的网页和网站。JSP 正日益火爆，大有取代其他网页编程语言的趋势。

下面介绍一下 JSP 的一些基础知识。

1.1.1 JSP 的开发背景

在万维网 (World Wide Web) 发展的短暂历史中，它已经从一个只能显示静态信息的网络发展到能进行网上股票交易和能进行网上购书等动态操作的一个基础设施。在各种各样的应用程序中，对于可能使用的基于 Web 的客户端，看上去没有任何限制。

基于浏览器客户端的应用程序比传统的基于客户机/服务器的应用程序有几个好处。这些好处包括几乎没有限制的客户端访问和极其简化的应用程序部署和管理（要更新一

个应用程序，管理人员只需要更改一个基于服务器的程序，而不是成千上万的安装在客户端的应用程序)。这样，软件工业正迅速地向建造基于浏览器客户端的多层次应用程序迈进。

这些快速增长的基于 Web 的应用程序，要求开发技术上有大的改进。静态 HTML 对于显示相对静态的内容是一个不错的选择；新的挑战在于创建交互的基于 Web 的应用程序，在这些程序中，页面的内容是基于用户的请求或者系统的状态，而不是预先定义的文字，也就是我们常说的良好的交互性。

对于这个问题，早期的一个解决方案是使用 CGI-BIN 接口。开发人员编写与接口相关的单独的程序，以及基于 Web 的应用程序，后者通过 Web 服务器来调用前者。这个方案有着严重的扩展性问题——每个新的 CGI 要求在服务器上新增一个进程。如果多个用户并发地访问该程序，这些进程将消耗该 Web 服务器所有的可用资源，并且使系统性能降低到极其低下的地步。

接下来，某些 Web 服务器供应商尝试通过为他们的服务器提供“插件”和 API 来简化 Web 应用程序的开发。这些解决方案是与特定的 Web 服务器相关的，不能解决跨多个服务器供应商的解决方案的问题，也即我们常说的跨平台性。例如，微软的 ASP (Active Server Pages) 技术，它就只能工作在微软的 IIS 和 PWS 上。

还存在其他的解决方案，例如，Java Servlet (下文将做详细介绍，在此不赘述)。但这些都不是一个普通的程序员所能掌握的。很显然，目前所需要的是一个业界范围内的创建动态内容页面的解决方案。这个方案将解决当前方案所受到的限制，即：

- 能运行在任何 Web 或应用程序服务器上。
- 能分离应用程序的逻辑和页面显示。
- 能快速地开发和测试。
- 能简化开发基于 Web 的交互式应用程序的过程。

JavaServer Pages (JSP) 技术就是被设计来满足这些要求的。JSP 规范是 Web 服务器、应用服务器开发者、交易系统、以及开发工具供应商等各方广泛合作的结果。Sun Microsystems 开发出这个规范来整合和平衡已经存在的对 Java 编程环境(例如, Java Servlet 和 JavaBeans) 进行支持的技术和工具。其结果是产生了一种新的、开发基于 Web 应用程序的方法，给予使用基于组件应用逻辑的页面设计者以强大的功能。

1.1.2 JSP 的发展历史

JSP 从诞生直至发展到现在的历史并不长，它是由 Sun Microsystems 公司倡导、许多公司参与建立的一种动态网页技术标准。读者可以通过访问下面的网址来获取更多有用的信息：<http://www.javasoft.com/product>。

JSP 1.0 规范的诞生日期是 1999 年 6 月 1 日，离现在只有一年多的时间。此前 Sun 公司曾经也推出过 JSP 0.91 和 JSP 0.92 的规范，但作为一种产品从 1.0 版开始才能算做是合理的。更何况 1.0 版本也是现在各个厂商支持最多、而且最为完善的版本。尽管现在 JSP 的最新版本是 1.1，但是由于受到业界的支持较少，因此本书不做太多介绍。本书将以 JSP

1.0 版本作为蓝本来展开。

1.2 JSP 与 Java

本节将简要叙述一下 JSP 和 Java 的关系。Java 可以算做 JSP 的老祖宗了。Java 刚推出的时候, 虽然被吹得神乎其神, 其实却很不完善, 没有得到很多的应用。早期的 Java 都是在客户端运行的, 网络的功能并不是很强。Java 2 以及 Java Servlet 的诞生改变了这一切。Servlet 是小的服务器端 Java 应用程序, 它的出现使得 Java 的一系列优点得到了充分的发挥, 使 Java 的应用进入一个新的时代。同时, 由于 EJB(Enterprise JavaBeans, 企业级 JavaBeans 组件)、J2EE 和应用服务器技术的成熟以及电子商务热浪的涌现也把 Java 应用推向了一个新的高峰。

Servlet 完全是靠书写 HTML 页面来工作的, 要想掌握它难度很大。JSP 的诞生则极大地减少了 Servlet 的编写难度。

由于 JSP 是用 Java 语言编写的, 它有着 Java 所有的特点, 提到 JSP 我们就不能不提到 Java, 这门曾经风风火火过一阵后来又在一段时间内受到冷落的语言。

Java 是一个由 Sun 公司开发的编程语言。使用它可在各式各样不同种机器、不同种操作平台的网络环境中开发软件。不论使用的是哪一种 WWW 浏览器、哪一种计算机、哪一种操作系统, 只要 WWW 浏览器注明“支持 Java 语言”, 就可以看到生动的页面。Java 正在逐步成为 Internet 应用的主要开发语言。它彻底改变了应用软件的开发模式, 带来了 PC 机的一次技术革命, 为迅速发展的信息世界增添了新的活力。

Sun 公司的 Java 语言开发小组成立于 1991 年, 其最初目的是开拓消费类电子产品市场, 例如, 交互式电视、烤面包箱等。Sun 公司内部人员把这个项目称为 Green, 那时万维网还只是在图纸上的构想。该小组的领导人 James Gosling 是一位非常杰出的程序员。他是 Sun News 窗口系统的总设计师, 也是第一个用 C 实现 EMACS 文本编辑器 COSMACS 的开发者的。

在研究开发过程中, Gosling 深刻体会到消费类电子产品和 workstation 产品在开发思路上的差异: 消费类电子产品要求可靠性高、费用低、标准化、使用简单, 用户并不关心 CPU 的型号, 也不欣赏专用昂贵的 RISC 处理器, 他们需要建立在一个标准基础之上、具有一系列可选的方案, 从 8086 到 80586 都可以选取。

Gosling 就是在上述思想指导下设计 Java 的, 因此跨平台成为 Java 的首要设计目标。

Java 语言有以下一些特点:

1. 简单性

Java 语言是一种面向对象的语言, 它通过提供最基本的方法来完成指定的任务, 只需理解一些基本的概念, 就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念, 并且通过实现自动垃圾收集, 大大简化了程序设计者的内

存管理工作。另外, Java 也适合于在小型机上运行, 它的基本解释器及类的支持只有 40KB 左右, 加上标准类库和线程的支持也只有 215KB 左右。

2. 面向对象的特性

Java 语言的设计集中于对象及其接口上, 它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法, 实现了模块化和信息隐藏; 而类则提供了一类对象的原型, 并且通过继承机制, 子类可以使用父类所提供的方法, 实现了代码的复用。

3. 分布性处理

Java 是面向网络的语言。通过 Java 提供的基础类库可以处理 TCP/IP 协议, 用户可以通过 URL 地址在网络上很方便地访问其他对象。

4. 健壮性

Java 在编译和运行程序时, 都要对可能出现的问题进行检查, 以消除错误的产生。它提供自动垃圾收集来进行内存管理, 防止程序员在管理内存时产生错误。使用集成的面向对象的例外处理机制, 在编译时, Java 会提示可能出现但未被处理的例外, 帮助程序员正确地进行选择以防止系统的崩溃。另外, Java 在编译时还可捕获类型声明中的许多常见错误, 防止动态运行时不匹配问题的出现。

5. 安全性

用于网络、分布式计算环境下的 Java 必须要能够防止病毒的入侵。Java 不支持指针, 一切对内存的访问都必须通过对象的实例变量来实现, 这样就防止程序员使用“特洛伊”木马等欺骗手段访问对象的私有成员, 同时也避免了指针操作中容易产生的错误。

6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令, 只要安装了 Java 运行时系统, Java 程序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示, Java 解释器得到字节码后, 对它进行转换, 使之能够在不同的平台运行。

7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时, Java 的类库中也实现了与不同平台的接口, 使这些类库可以从一个平台移植到另一个平台。另外, Java 编译器是由 Java 语言实现的, Java 运行时系统由标准 C 实现, 这使得 Java 系统本身也具有可移植性。

除了上文介绍的这些优点之外, Java 语言还具有解释执行、高性能、多线程、动态。