

程序员书库

冯博 应群 编著

面向对象的Java网络编程



清华大学出版社

程序员书库

面向对象的 Java 网络编程

冯博 应群 编著

清华大学出版社

北 京

内 容 简 介

Java 是美国 SUN 公司推出的编程语言, 由于其跨平台的特性而得到了广泛的应用。随着互联网技术的发展, Java 在网络编程方面的优势更加突出。全书共分 13 章, 在简单介绍 Java 编程和面向对象软件开发的基础知识后, 围绕用户日常接触最广泛的网络应用软件给出了详细的讲解, 为普通用户搭起了学与用的平台, 并展示了 Java 网络编程的强大功能。

本书使用通俗而简洁的语言对各个实例进行描述, 图文并茂, 方便读者在比较短的时间内认识并掌握 Java 网络编程方法。

本书具有很强的实践性和操作性, 内容紧凑、结构严谨。无论是对于 Java 的初学者, 还是已经接触过 Java 的网络编程用户, 都有很高的参考价值, 同时也是大专院校学生学习 Java 网络编程的理想教材。

本书中的实例源代码可以到 <http://www.tupwk.com.cn> 网站下载。

版权所有, 翻印必究。举报电话: 010-62782989 13901104297 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

面向对象的 Java 网络编程/冯博, 应群编著. —北京: 清华大学出版社, 2004.11
(程序员书库)

ISBN 7-302-09304-0

I. 面… II. ①冯… ②应… III. JAVA 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2004)第 086086 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: 胡辰浩

文稿编辑: 袁建华

封面设计: 王 永

版式设计: 康 博

印 刷 者: 北京市昌平环球印刷厂

装 订 者: 三河市李旗庄少明装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印 张: 25.5 字 数: 589 千字

版 次: 2004 年 11 月第 1 版 2004 年 11 月第 1 次印刷

书 号: ISBN 7-302-09304-0/TP·6525

印 数: 1~4000

定 价: 38.00 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 编程语言是由美国 SUN 公司研发的第一种在国际互联网络(Internet)上具有“硬件/软件中立性”交互能力的编程语言。Java 一经推出即引起世界各大媒体的注意,可见其功能及魅力。随着互联网的迅速发展,Java 在网络编程方面势必会掀起一番高潮。

全书共分 13 章,按照由浅入深的方式进行安排。第 1 章主要介绍 Java 语言和软件工程的基础知识,以及如何配置相关的编程环境和必要的工具。使读者对 Java 网络编程有一个基本的认识,为后面的网络软件开发实例做好铺垫。

第 2 章主要介绍计算机网络、Java 网络编程的基础知识。通过第 2 章的学习,读者能够掌握一些 Java 常用编程模式和方法,并了解网络程序的结构。

从第 3 章到第 13 章,每章讲述一个用 Java 语言编写的网络应用软件实例,分别为二人聊天室、多线程下载、telnet 远程登录、邮件服务器和客户端、Web 服务器和浏览器、FTP 服务器和客户端、网络会议室、流媒体播放器、安全信息的获取、电子白板、黑白棋游戏。这些实例由简单到复杂,符合大部分读者的认知习惯。在讲述实例的过程中,重点向读者介绍 Java 网络编程思想。读者可以一边学习实例,一边在本书的指导下培养自己面向对象软件开发的思维。在每章的最后,编者对本章所讲述的重点做了小结。由于 Java 编程语言应用十分广泛,考虑到读者的实际需要,故本书通过这些实用的小例子,讲述了 Java 在网络编程上的应用,而更重要的是向读者阐述了面向对象的网络编程思想。

全书内容覆盖 Java 网络软件开发的各个要点,知识面广泛,注重条理性,编程步骤清晰、完善,而且易于操作。读者只要按照书中的讲解一步步操作,即能完成对网络应用程序的学习与实践。

本书对基础概念的讲解比较全面,在编写过程中由浅入深,既照顾到了初学者,也为了解面向对象程序开发基础知识的读者提供了有益的参考。建议读者在学习过程中适当配合上机实践,这样会达到更好的学习效果。

本书由冯博、应群执笔编写。此外,刘庆生、徐因林、胡春、张宗训、罗模林、赵晓娟、钱玉莲、徐孝萍、杨国淑、张苏川、陈晖、邹丽、苏学远、李华富、刘熙、唐中国、许巧、王进华、杜洪芳、要飞、何颖、林果、龚世虎、胡晓霞、夏利、熊珍斌、包键、陈炜、李永光和赵鸿明等同志在整理材料方面给予了很大的帮助,在此,编者对他们表示衷心的感谢。

本书所选实例内容翔实、结构紧凑、条理清晰、覆盖知识点全面。但由于编写时间较为仓促,书中难免会有疏漏和不足之处,恳请广大读者提出宝贵意见。

编 者

2004 年 4 月

目 录

第 1 章	Java 编程基础	1
1.1	Java、Application 和 Applet	1
1.1.1	Java 简介	1
1.1.2	应用程序(Application)	3
1.1.3	小应用程序(Applet)	3
1.2	软件工程和 UML 简介	4
1.2.1	软件工程简介	4
1.2.2	统一建模语言(UML)简介	5
1.3	构建单机网络运行环境	6
1.3.1	Tomcat 的安装和配置	6
1.3.2	其他软件的安装和配置	9
1.4	Java 开发环境简介	9
1.4.1	JCreator 的安装和使用	9
1.4.2	其他开发环境的安装	12
1.5	UML 相关工具简介	12
1.5.1	Jude 的安装和使用	12
1.5.2	其他工具的安装	13
1.6	本章小结	14
第 2 章	计算机网络概述	15
2.1	计算机网络的发展历史	15
2.2	网络硬件	16
2.2.1	服务器和工作站	17
2.2.2	网络互联的硬件设备	17
2.3	网络软件网络协议	19
2.3.1	网络操作系统(NOS)	19
2.3.2	开放系统互连参考模型 OSI	20
2.3.3	TCP/IP 协议	24
2.4	常见的应用模式	27
2.4.1	客户端/服务器(Client/Server)模式	28
2.4.2	浏览器/服务器(Browser/Server)模式	29
2.5	Java 网络编程简介	30
2.5.1	流套接字(Stream Socket)	31
2.5.2	数据报套接字(Datagram Socket)	38
2.6	本章小结	39

第 3 章 二人聊天室	40
3.1 系统开发背景	40
3.1.1 业务背景	40
3.1.2 技术背景	41
3.2 需求描述	41
3.2.1 用例描述	42
3.2.2 活动图	45
3.3 分析和设计	46
3.3.1 获取对象	46
3.3.2 对象的静态关系	47
3.3.3 对象的动态关系	49
3.3.4 类的设计	52
3.4 编程和实现	55
3.4.1 主要代码分析	55
3.4.2 调试运行	62
3.5 本章小结	63
第 4 章 多线程下载	64
4.1 系统开发背景	64
4.1.1 业务背景	64
4.1.2 技术背景	65
4.2 需求描述	67
4.2.1 用例描述	67
4.2.2 活动图	70
4.3 分析和设计	71
4.3.1 获取对象	71
4.3.2 对象的静态关系	72
4.3.3 对象的动态关系	73
4.3.4 类的设计	74
4.4 编程和实现	76
4.4.1 主要代码分析	76
4.4.2 调试运行	84
4.5 本章小结	86
第 5 章 telnet 远程登录	87
5.1 系统开发背景	87
5.1.1 业务背景	87
5.1.2 技术背景	88
5.2 需求描述	89

5.2.1	用例描述	90
5.2.2	活动图	93
5.3	分析和设计	94
5.3.1	获取对象	94
5.3.2	对象的静态关系	94
5.3.3	对象的动态关系	96
5.3.4	类的设计	97
5.4	编程和实现	98
5.4.1	主要代码分析	99
5.4.2	调试运行	104
5.5	本章小结	104
第 6 章	邮件服务器和客户端	105
6.1	系统开发背景	105
6.1.1	业务背景	105
6.1.2	技术背景	107
6.2	需求描述	108
6.2.1	客户端的用例描述	108
6.2.2	客户端的活动图	113
6.2.3	服务器端的用例描述	114
6.2.4	服务器端的活动图	119
6.3	分析和设计	120
6.3.1	获取客户端对象	120
6.3.2	客户端对象的静态关系	121
6.3.3	客户端对象的动态关系	123
6.3.4	客户端类的设计	124
6.3.5	获取服务器端的对象	126
6.3.6	服务器端对象的静态关系	126
6.3.7	服务器端对象的动态关系	128
6.3.8	服务器端类的设计	129
6.4	编程和实现	131
6.4.1	客户端主要代码分析	131
6.4.2	服务器端主要代码分析	134
6.4.3	调试运行	146
6.5	本章小结	148
第 7 章	Web 服务器和浏览器	149
7.1	系统开发背景	149
7.1.1	业务背景	149

7.1.2	技术背景	151
7.2	需求描述	151
7.2.1	客户端的用例描述	151
7.2.2	客户端的活动图	154
7.2.3	服务器端的用例描述	155
7.2.4	服务器端的活动图	159
7.3	分析和设计	160
7.3.1	获取客户端对象	160
7.3.2	客户端对象的静态关系	160
7.3.3	客户端对象的动态关系	162
7.3.4	客户端类的设计	163
7.3.5	获取服务器端对象	165
7.3.6	服务器端对象的静态关系	166
7.3.7	服务器端对象的动态关系	167
7.3.8	服务器端类的设计	168
7.4	编程和实现	169
7.4.1	客户端主要代码分析	169
7.4.2	服务器端主要代码分析	173
7.4.3	调试运行	179
7.5	本章小结	181
第 8 章	FTP 服务器和客户端	182
8.1	系统开发背景	182
8.1.1	业务背景	182
8.1.2	技术背景	184
8.2	需求描述	184
8.2.1	客户端的用例描述	185
8.2.2	客户端的活动图	189
8.2.3	服务器端的用例描述	191
8.2.4	服务器端的活动图	195
8.3	分析和设计	197
8.3.1	获取客户端对象	197
8.3.2	客户端对象的静态关系	197
8.3.3	客户端对象的动态关系	198
8.3.4	客户端类的设计	199
8.3.5	获取服务器端对象	201
8.3.6	服务器端对象的静态关系	202
8.3.7	服务器端对象的动态关系	203
8.3.8	服务器端类的设计	204

8.4	编程和实现	205
8.4.1	客户端主要代码分析	206
8.4.2	服务器端主要代码分析	213
8.4.3	调试运行	228
8.5	本章小结	230
第 9 章	网络会议室	231
9.1	系统开发背景	231
9.1.1	业务背景	231
9.1.2	技术背景	232
9.2	需求描述	233
9.2.1	用例描述	233
9.2.2	活动图	237
9.3	分析和设计	239
9.3.1	获取对象	239
9.3.2	对象的静态关系	240
9.3.3	对象的动态关系	242
9.3.4	类的设计	243
9.4	编程和实现	246
9.4.1	主要代码分析	246
9.4.2	调试运行	252
9.5	本章小结	253
第 10 章	流媒体播放器	254
10.1	系统开发背景	254
10.1.1	业务背景	254
10.1.2	技术背景	256
10.2	需求描述	257
10.2.1	用例描述	257
10.2.2	活动图	260
10.3	分析和设计	261
10.3.1	获取对象	261
10.3.2	对象的静态关系	262
10.3.3	对象的动态关系	263
10.3.4	类的设计	264
10.4	编程和实现	266
10.4.1	主要代码分析	266
10.4.2	调试运行	270
10.5	本章小结	271

第 11 章	安全信息的获取	272
11.1	系统开发背景	272
11.1.1	业务背景	272
11.1.2	技术背景	273
11.2	部署 SSL	275
11.3	需求描述	275
11.3.1	用例描述	275
11.3.2	活动图	279
11.4	分析和设计	280
11.4.1	获取服务器端对象	280
11.4.2	服务器端对象的静态关系	280
11.4.3	服务器端对象的动态关系	282
11.4.4	服务器端类的设计	283
11.5	编程和实现	284
11.5.1	服务器端主要代码分析	284
11.5.2	调试运行	290
11.6	本章小结	292
第 12 章	电子白板	293
12.1	系统开发背景	293
12.1.1	业务背景	293
12.1.2	技术背景	294
12.2	需求描述	294
12.2.1	用例描述	294
12.2.2	活动图	297
12.3	分析与设计	298
12.3.1	获取对象	298
12.3.2	对象的静态关系	298
12.3.3	对象的动态关系	299
12.3.4	类的设计	300
12.4	编程与实现	301
12.4.1	主要代码分析	301
12.4.2	调试运行	312
12.5	本章小结	314
第 13 章	网络黑白棋游戏	315
13.1	背景介绍	315
13.1.1	业务背景	315
13.1.2	技术背景	316

13.2	需求描述.....	320
13.2.1	用例描述.....	320
13.2.2	活动图.....	323
13.3	分析与设计.....	324
13.3.1	获取对象.....	324
13.3.2	对象的静态关系.....	325
13.3.3	对象的动态关系.....	326
13.3.4	类的设计.....	327
13.4	编程与实现.....	328
13.4.1	游戏代码.....	328
13.4.2	客户端代码.....	345
13.4.3	服务器端代码.....	357
13.4.4	调试运行.....	391
13.5	本章小结.....	392

第 1 章 Java 编程基础

本章重点

本章将为读者介绍一些必要的基础知识，包括 Java 语言和软件工程的基本知识，以及如何配置相关的编程环境和必要的工具。通过本章的学习，读者能够掌握 Java 的基本概念，了解开发环境的配置和使用。

本章主要内容

- Java 的基本知识
- 软件工程和 UML 的基本知识
- 单机网络运行环境的配置
- Java 的集成开发环境(IDE)
- UML 辅助工具

1.1 Java、Application 和 Applet

本节将介绍 Java 的基本知识，主要是 Java 语言以及运行机制、应用程序 Application 和小应用程序 Applet。

1.1.1 Java 简介

从 Java 的产生历史来看，Java 和网络有密不可分的关系。1991 年 4 月份，SUN 公司的一个叫 Green 的项目试图为家用消费电子产品开发一个分布式代码系统，用来控制电冰箱、电视机等家用电器并进行信息交流。由于采用 C++ 进行开发较为复杂且安全性差，所以项目组基于 C++ 开发了一种新的语言 Oak，这就是 Java 的前身。Oak 是一种用于网络的精巧而安全的语言，项目组成员将它重新命名为 Java，并编制了 HotJava 浏览器，从此 Java 进入 Internet 的应用领域。Java 作为一种编程语言具有其独有特征，如图 1-1 所示。



图 1-1 Java 语言的特点

这些特点决定了 Java 的强大生命力并且能在网络应用领域得到广泛的应用。

使用 Java 语言写成的代码可以被编译并解释执行。通过编译器，可以把 Java 程序翻译成一种中间代码，这种中间代码可以被 Java 解释器解释并独立于平台；通过解释器，每条 Java 字节指令被分析，然后在计算机上运行。整个过程如图 1-2 所示。

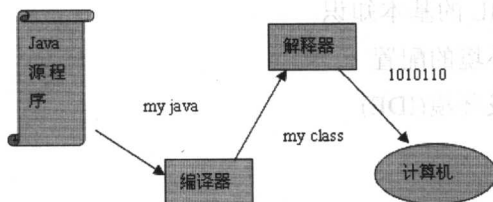


图 1-2 Java 的运行机制

但是，不能仅仅把 Java 看成是一种编程语言，Java 同时还是一个平台。作为平台，Java 是程序运行的硬件或软件环境。Java 平台与大多数其他平台的不同之处在于它是运行于其他基于硬件平台的纯软件平台，而大多数其他平台是硬件和操作系统的结合。

Java 平台由两部分组成：Java 虚拟机(Java VM)和 Java 应用程序接口(Java API)，简要介绍如下：

(1) Java 虚拟机(Java VM)

Java 虚拟机是一个想象中的机器,在实际的计算机上通过软件模拟来实现，它有自己的想象中的硬件。Java 虚拟机是 Java 平台的基础，可以移植到各种基于硬件的平台上，也可以用硬件来实现。

(2) Java 应用程序接口(Java API)

Java API 是软件组件的集合，提供了很多有用的功能，例如图形用户界面(GUI)等，Java API 被分组为相关组件的包，编程人员很重要的一个工作就是了解现有的 API 并使用这些包。

一个 Java 程序是通过 Java 平台来实现硬件跨平台机制的，其运行机制如图 1-3 所示。

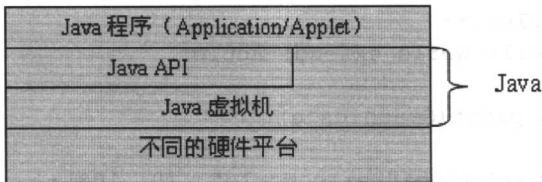


图 1-3 Java 的跨平台机制

上面介绍了 Java 和 Java 平台，下面介绍运行在 Java 平台上的两种 Java 程序：应用程序(Application)和小应用程序(Applet)。

1.1.2 应用程序(Application)

Java 的应用程序类似用 VB、VC 等编写的桌面程序，多以控制台(Console)方式运行，很容易实现窗口应用。看看用 Java 应用程序实现经典的 HelloWorld 的实例，这个实例在控制台上输出“Hello World!”，其源代码如下：

```

public class Hello World
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}
  
```

以上代码段显示了 Java 应用程序的如下特点：需要用保留字 class 来声明一个类名为 HelloWorld 的新类；类的定义是公共的(Public)，名字必须和文件名一致；在类中定义一个 main()方法。

对于一个 Java 应用程序，main()方法十分重要。Java 解释器在没有生成任何实例的情况下，以 main()作为入口来执行程序。一个 Java 应用程序中可以定义多个类,每个类中可以定义多个方法,但是最多只能有一个公共类，也只能有一个 main()方法作为程序的入口。在 main()方法中，括号()中的 args[]是传递给 main()方法的参数；名为 args 的参数是类 String 的一个实例，参数可以为 0 个或多个，多个参数间用逗号分隔。

1.1.3 小应用程序(Applet)

相对于应用程序，小应用程序不能独立运行，需要嵌入 Web 页面中，或者在支持 Java 虚拟机的浏览器上运行。这里也给出 HelloWorld 的实例，使用的是小应用程序。这个程序在网页中坐标(10,100)的地方输出一行文字“Hello World”。其源代码如下：

```
import java.awt.*;
import java.applet.*;
public class Hello World extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello World", 10, 100);
    }
}
```

上面的代码段中, Hello World 类继承了 Applet 类, 故使用 import 输入 java.applet 包, 使得包中定义的 Applet 类可用; 要使用图形界面, 输入了 java.awt 来支持使用图形界面类; 在 Hello World 类中, 重写了父类 Applet 的 paint() 方法; 参数 g 为 Graphics 类, 表明的是当前作画的上下文, 通过调用它的 drawString() 方法来实现图形的输出。

可以看到, 小应用程序有如下一些特点与应用程序不同: 必须要声明一个 Applet 类的子类, 本实例中是 HelloWorld 类; 这个类中没有 main() 方法, 这是与应用程序的本质区别。

由于 Applet 中没有 main() 方法作为 Java 解释器的入口, 因此必须编写 HTML 文件, 再把该 Applet 嵌入其中, 然后用 appletviewer 来运行, 或在支持 Java 的浏览器上运行。

1.2 软件工程和 UML 简介

软件工程提供了保证软件开发质量和效率的一系列工程方法, 而面向对象的软件开发是现在软件工程的主流。

1.2.1 软件工程简介

软件工程的定义, 并没有统一的说法。IEEE 给出的定义是: 软件工程是开发、运行、维护和修复软件的系统方法。

在软件工程中要注意使用 3 种基本策略: 复用、分而治之和优化与折衷。

(1) 复用(Reuse)

复用就是在构造新的软件系统时, 直接使用已有的软构件组装成新的系统。复用的优点是: 合理并简化了软件的开发过程, 减少了总的开发工作量与维护代价, 既降低了软件的成本又提高了生产率; 多数的软构件是经过反复使用验证的, 自身具有较高的可靠性。

(2) 分而治之(Divide and Conquer)

分而治之是把一个复杂的问题分解成若干个简单的问题, 然后逐个解决的过程。在执行分而治之的时候, 应该着重考虑两个问题, 一是复杂问题分解后, 每个问题能

否用程序实现；二是所有程序最终能否集成为一个软件系统并有效解决原始的复杂问题。

(3) 优化与折衷(Optimize and Compromise)

优化是指提高软件的各个质量因素。优化工作中很多目标存在复杂的联系，不可能所有的目标同时得到优化，这就需要折衷。折衷是指通过协调各个质量因素，实现整体质量的最优。

软件工程的整个开发过程，即从希望开发一个软件系统到该软件系统的正常运行主要有4个环节：可行性与需求分析、系统分析、程序设计与实现以及测试和维护，其关系如图1-4所示。



图 1-4 软件工程的主要环节

上述从用户需求出发，最终转化为软件系统需要的一系列活动总和称为软件过程(Software Process)。本书使用的方法即是基于统一软件开发过程(Rational Unified Process, 简称 RUP)。

统一软件开发过程是一个通用过程框架，可以应付种类广泛的软件系统、不同的应用领域、不同的组织类型、不同的性能水平和不同的项目规模。统一软件开发过程是基于构件的，用这个过程开发的软件系统是由构件构成的，构件之间通过定义良好的接口相互联系。在构建软件系统的时候，统一软件开发过程使用的是统一建模语言(Unified Modeling Language, 简称 UML)。统一建模语言是统一软件开发过程的有机组成部分，它们是被同步开发的。

统一软件开发过程的特点表现在3个方面：用例驱动的、以基本架构为中心、迭代式和增量性的，这些特点需要读者在用例中体会。

正如前面提到的，统一软件开发过程离不开统一建模语言(UML)，下面简单介绍统一建模语言。

1.2.2 统一建模语言(UML)简介

UML(Unified Modeling Language)是一种构建软件系统和文档的通用可视化建模语言，它有如下3个特点：

- UML 能与所有的开发方法一同使用，可用于软件开发的整个生命周期。
- UML 能表达系统的静态结构和动态信息，并能管理复杂的系统模型，便于软件团队之间的合作开发。

- UML 不是编程语言,但支持 UML 语言的工具可以提供从 UML 到各种编程语言的代码生成,也可以提供从现有程序逆向构建 UML 模型。

UML 由 3 种基本构造块组成,它们分别是:事物(Entity)、关系(Relationship)和图(Diagram),简要介绍如下:

(1) 事物(Entity)

UML 中的事物分为 3 种:

- 结构事物。包括类、接口、协作、用例、主动类、构件和节点。
- 行为事物。包括交互和状态机。
- 分组事物和注释事物。包括包和注解。

(2) 关系(Relationship)

UML 中的关系分为 4 种:依赖(Dependency)关系、关联(Association)关系、泛化(Generalization)关系和实现(Realization)关系。

(3) 图(Diagram)

UML 中的图有 8 种:类图、用例图、序列图、协作图、状态图、活动图、组件图和部署图。

1.3 构建单机网络运行环境

在本地构建一个支持 Java 的单机网络运行环境的原因有两点:Java 程序在服务器端的运行需要 Java 虚拟机的支持,但不是所有的 Web 服务器都支持 Java;构建本地网络环境之后,在开发 Java 网络应用程序的过程中,调试工作将变得更容易。

这里就简单地介绍 Windows 下的单机网络运行环境。Windows 下常见的 Web 服务器产品有 JSWDK(Sun)、Tomcat(Apache)和 Weblogic(BEA)等。本书开发环境使用的是 Apache Tomcat,下面简要介绍 Tomcat 的安装和配置。

1.3.1 Tomcat 的安装和配置

Tomcat 是 SUN 的 JSWDK(Java Server Web Development Kit)中 Servlet 的运行环境。Tomcat 的源代码被提供给 Jakarta 项目,在开放源码的模型下进行进一步的开发。当在一台机器上配置好 Apache 服务器,可利用它响应对 HTML 页面的访问请求。虽然 Tomcat 部分是 Apache 服务器的扩展,但它是独立运行的。当配置正确时,Apache 为 HTML 页面服务,而 Tomcat 实际上运行 jsp 页面和 Servlet。

安装 Tomcat 之前,先要安装 JDK 或 J2SDK。二者是运行 Java 应用程序所必需的,可以在 SUN 的主页免费下载。

在下载后按照提示安装即可,但是要让它正常运行还需要设置系统运行的环境变量。J2SDK 的配置方法如下。