

北京科海培训中心

实
用
指
南

廖卫东 陈梅 编著
施振川 审校



JAVA

程序设计

机械工业出版社

北京科海培训中心

Java 程序设计实用指南

廖卫东 陈 梅 编著

施振川 审校



机械工业出版社

337506

这是一本向广大编程人员学习 Java 语言的教程。全书共分五部分,第一部分介绍 Java 基本概念和 JDK 环境建立;第二部分介绍 Java 语言;第三部分是 Java 的面向对象部分,介绍 Java 的类和类库的应用;第四部分介绍出错处理与多线程;第五部分介绍编程实践。

本书内容新颖,结合 JDK 最新 1.5.2 版本介绍,涵盖了 Java 语言的各个方面,特别是通过完整的 JDK 类库介绍及大量的程序实例讲解,使读者尽快掌握 Java 这一面向对象编程语言。

图书在版编目(CIP)数据

Java 程序设计实用指南/廖卫东等著. —北京:机械工业出版社,1995.9
ISBN 7-111-05439-3

Java 107

Ⅰ. [..] 廖... 陈... Ⅱ. Java 语言-程序设计-指南 Ⅲ. TP382

中国版本图书馆 CIP 数据核子(95)

出版人 马九荣 (北京市百万庄南街1号 邮政编码 100037)

责任编辑:科培 责任校对:成昊

门头沟胶印厂印刷,新华书店北京发行所发行

1996年10月第1版,1996年10月第1次印刷

187mm×1092mm 1/16·28.75印张·699千字

0 001—5 000 册

定价:48.00 元

目 录

第1章 Internet 基本概念 (1)

1.1 网络与网络协议 (1)

1.1.1 何谓网络 (1)

1.1.2 何谓网络协议 (2)

1.1.3 开放系统互连参考模型 (2)

1.2 TCP/IP 协议 (2)

1.2.1 TCP/IP 协议的起源 (2)

1.2.2 TCP/IP 协议族 (3)

1.2.3 TCP/IP 协议的层次 (4)

1.3 Internet 网络及其发展 (4)

1.3.1 Internet 的产生 (5)

1.3.2 Internet 的现状和发展 (6)

1.3.3 Internet 服务与应用 (7)

1.4 World Wide Web (7)

1.4.1 何谓 WWW (7)

1.4.2 WWW 服务的特点 (8)

1.4.3 WWW 的起源 (8)

1.4.4 客户机/服务器工作模式 (8)

1.4.5 WWW 浏览器 (WWW Browser) (9)

1.4.6 统一资源定位器 (URL: Uniform Resource Location) (9)

1.4.7 超文本传输协议 (HTTP: HyperText Transfer Protocol) (10)

1.4.8 WWW 主页 (Home Page) (10)

1.5 超文本标识语言 (HTML) (10)

1.6 Java 与 HTML 标记 (13)

1.7 小结 (15)

第2章 Java 及 HotJava 概述 (16)

2.1 何谓 Java (16)

2.1.1 Java 语言 (16)

2.1.2 Java 开发及执行环境 (17)

2.2 Java 的起源 (18)

2.2.1 Oak——Java 原型的问世 (18)

2.2.2 FirstPerson——几经周折 (18)

2.2.3 WWW 与 Mosaic——它山之石 (19)

2.2.4 Internet Play——改变 Java 命运的策略 (19)

2.3 Java 的语言特性 (20)

2.3.1 面向对象 (20)

2.3.2 可移植性 (21)

2.3.3 稳定性和安全性 (23)

2.3.4 简单性 (23)

2.3.5 高性能 (24)

2.3.6 动态特性 (25)

2.3.7 分布式 (25)

2.4 HotJava 浏览器 (26)

2.4.1 何谓 HotJava (26)

2.4.2 HotJava 的功能 (26)

2.4.3 HotJava 的主要特征 (26)

2.4.4 HotJava 的安全性 (27)

2.5 小结 (28)

第3章 建立 JAVA 开发环境 (29)

3.1 下载 JDK (29)

3.1.1 利用 WWW 服务下载 JDK (29)

3.1.2 利用匿名 FTP 服务下载 JDK (30)

3.2 安装 JDK (32)

3.2.1 在 Windows 95/NT 上安装 JDK (32)

3.2.2 在 Sun Solaris 上安装 JDK (35)

3.3 获取并安装 HotJava 浏览器 (35)

3.4 JDK 环境工具及其使用 (36)

3.4.1 小应用程序观察器 (36)

3.4.2 Java 编译器 (37)

3.4.3 Java 解释器 (39)

3.4.4 Java 文档生成器 (42)

3.4.5 类分解器 (42)

3.4.6 Java 调试器 (43)

3.4.7 C 文件生成器 javah (44)

3.5 小结 (44)

第 4 章 数据类型、变量和表达式 (46)	第 6 章 类、接口和包 (77)
4.1 概述 (46)	6.1 面向对象编程基础 (77)
4.1.1 本章主要内容 (46)	6.1.1 过程式编程与面向对象编程 (77)
4.1.2 Java 程序的基本结构 (47)	6.1.2 面向对象编程的基本概念 (77)
4.2 Java 符号集 (48)	6.2 Java 语言中的对象和类 (78)
4.2.1 注释 (48)	6.2.1 对象和类层次 (78)
4.2.2 标识符 (49)	6.2.2 类声明 (79)
4.2.3 关键字 (50)	6.2.3 实例变量 (79)
4.3 常量(Literals) (50)	6.3 方法 (80)
4.3.1 整数 (50)	6.3.1 方法声明 (80)
4.3.2 浮点数 (51)	6.3.2 对象的可达性 (82)
4.3.3 布尔型常量 (51)	6.3.3 方法重载 (83)
4.3.4 字符型常量 (51)	6.4 静态成员 (83)
4.3.5 字符串常量 (52)	6.5 对象的创建和销毁 (84)
4.4 运算符和分隔符 (52)	6.5.1 创建对象 (84)
4.5 变量声明 (53)	6.5.2 new 操作符 (85)
4.5.1 声明整数类型 (53)	6.5.3 撤销对象 (85)
4.5.2 声明浮点类型 (54)	6.6 main()方法 (86)
4.5.3 声明字符类型 (54)	6.7 包(Package) (87)
4.5.4 声明布尔类型 (54)	6.7.1 包声明 (87)
4.5.5 声明数组 (54)	6.7.2 装载其他包 (88)
4.6 变量的作用域 (54)	6.8 继承 (88)
4.7 运算符及其优先级 (55)	6.8.1 extends 修饰符 (88)
4.7.1 整数运算符 (56)	6.8.2 方法重载(overriding) (89)
4.7.2 布尔运算符 (59)	6.8.3 抽象方法和类 (90)
4.7.3 浮点数运算符 (61)	6.8.4 接口 (90)
4.7.4 数组运算符 (61)	6.8.5 类转换(casting) (92)
4.7.5 字符串运算符 (62)	6.9 null, this 和 super (92)
4.8 类型转换 (65)	6.10 final 修饰符 (93)
4.9 小结 (65)	6.10.1 类的 final 修饰符 (93)
第 5 章 流程控制 (65)	6.10.2 变量的 final 修饰符 (93)
5.1 概述 (65)	6.10.3 方法的 final 修饰符 (94)
5.2 条件语句 (66)	6.11 小结 (94)
5.2.1 简单条件语句 (66)	第 7 章 Java 类库构成与语言基础类库 (95)
5.2.2 复合条件语句 (67)	7.1 Java 类库的整体结构 (95)
5.2.3 条件语句的使用 (68)	7.1.1 类库 (95)
5.3 循环语句 (68)	
5.3.1 while 语句 (68)	
5.3.2 do 语句 (69)	
5.3.3 for 语句 (70)	

7.1.2	HorJava 类库	(96)	9.5.2	使用堆栈类	(128)
7.2	语言基础类库	(98)	9.6	小结	(139)
7.2.1	Object 类	(99)	第 10 章 浏览器类库和 Applet 编程	(140)	
7.2.2	数据类型的对象包装类	(99)	10.1	浏览器类库的构成	(140)
7.2.3	Math 类	(100)	10.2	Applet 类的继承层次	(140)
7.2.4	字符串类	(103)	10.3	Applet 类的构造函数、实例变量和方法	(141)
7.2.5	System 类	(104)	10.4	Applet 程序的结构	(143)
7.3	小结	(106)	10.5	Applet 编程技巧	(144)
第 8 章 I/O 类库及其使用	(107)		10.5.1	扩充 Applet 基类	(144)
8.1	概述	(107)	10.5.2	获取属性	(145)
8.2	InputStream 和 PrintStream	(108)	10.5.3	装载图象	(146)
8.2.1	InputStream	(108)	10.5.4	显示图象	(146)
8.2.2	read() 方法	(108)	10.5.5	鼠标事件	(148)
8.2.3	PrintStream	(110)	10.5.6	键盘事件	(149)
8.3	文件访问	(110)	10.5.7	声音播放	(149)
8.3.1	File 类	(111)	10.6	小结	(151)
8.3.2	RandomAccessFile 类	(113)	第 11 章 AWT 类库及其使用	(152)	
8.4	其他输入输出操作	(117)	11.1	AWT 类库的继承层次	(152)
8.4.1	文件拷贝	(117)	11.2	AWT 的显示机制	(153)
8.4.2	pipe	(120)	11.3	AWT 构件类及其使用	(154)
8.4.3	管道	(121)	11.4	构件在独立应用程序中的使用	(157)
8.5	小结	(127)	11.4.1	Graphics 类	(157)
第 9 章 Utility 类库及其使用	(128)		11.4.2	Color 类	(158)
9.1	Utility 类库的大致构成	(128)	11.4.3	扩充 Button 类	(158)
9.2	日期时间类	(128)	11.4.4	程序总体框架	(159)
9.3	向量类及其使用	(130)	11.5	小结	(162)
9.3.1	向量和数组的异同	(130)	第 12 章 网络类库	(163)	
9.3.2	向量的构造函数、实例变量和方法	(131)	12.1	Java 与网络编程	(163)
9.3.3	建立向量实例	(132)	12.2	Java 网络基础类库	(164)
9.3.4	向量维护	(133)	12.2.1	Java 网络基础类库的构成	(164)
9.3.5	对象查找	(134)	12.2.2	InetAddress	(164)
9.4	哈希表类及其使用	(135)	12.2.3	URLConnection 和 URLConnectionStream	(165)
9.4.1	哈希表类的构造函数、实例变量和成员方法	(135)	12.3	用 Java 语言建立 ftp 会话过程	(165)
9.4.2	创建、维护哈希表	(136)	12.3.1	类 FtpClient 的实例变量	(166)
9.4.3	查找关键字	(137)	12.3.2	创建类实例	(166)
9.5	堆栈类	(137)	12.3.3	打开连接	(166)
9.5.1	堆栈类的构造函数、实例变量和成员方法	(137)	12.3.4	注册到 FTP 服务器	(167)

12.3.5 执行 FTP 命令	(167)
12.3.6 传输数据	(167)
12.3.7 ftp 程序实例	(167)
12.4 Java 与 NNTP 协议	(168)
12.5 Java 与 WWW 协议	(169)
12.5.1 Java WWW 类库的构成	(169)
12.5.2 net.www.html	(169)
12.5.3 net.www.http	(171)
12.5.4 net.www.content.image	(171)
12.6 小结	(171)

第 13 章 出错与异常情况处理

13.1 出错处理的两种方式	(173)
13.2 Throw、Catch 和 Try	(173)
13.2.1 Throw	(174)
13.2.2 try	(174)
13.2.3 catch	(174)
13.2.4 嵌套的异常情况处理	(176)
13.3 finally 语句	(176)
13.4 运行时异常情况	(177)
13.4.1 ArithmeticException	(178)
13.4.2 ArrayIndexOutOfBoundsException	(178)
13.4.3 ArrayStoreException	(178)
13.4.4 ClassCastException	(178)
13.4.5 IllegalArgumentException	(179)
13.4.6 IllegalThreadStateException	(179)
13.4.7 IndexOutOfBoundsException	(179)
13.4.8 NegativeArraySizeException	(179)
13.4.9 NullPointerException	(179)
13.4.10 NumberFormatException	(180)
13.4.11 SecurityException	(180)
13.4.12 UnsupportedOperationException	(180)
13.4.13 OutOfMemoryException	(180)
13.4.14 NoClassDefFoundException	(181)
13.4.15 IncompatibleTypeException	(181)
13.4.16 UnsatisfiedLinkException	(181)
13.4.17 InterruptedException	(182)
13.5 小结	(182)

第 14 章 Java 的多线程机制

14.1 理解多线程	(183)
14.1.1 “线程”与“进程”	(183)
14.1.2 线程的执行与并发性	(183)
14.1.3 优先级	(184)
14.1.4 调度	(184)
14.1.5 同步	(184)
14.1.6 线程组	(184)
14.2 线程的生命周期	(185)
14.2.1 Newborn	(185)
14.2.2 Runnable	(185)
14.2.3 Running	(186)
14.2.4 Blocked	(186)
14.2.5 Dead	(186)
14.3 Java 的多线程实现机制	(186)
14.3.1 创建类 Thread 的子类	(186)
14.3.2 实现 Runnable 接口	(186)
14.4 Thread 类	(190)
14.4.1 构造函数	(190)
14.4.2 实例变量	(191)
14.4.3 方法	(191)
14.5 线程同步	(192)
14.5.1 synchronized 关键字	(192)
14.5.2 wait() 和 notify()	(192)
14.6 编程实例	(195)
14.7 小结	(195)

第 15 章 Java 编程实例详解

15.1 Java 程序的基本类型	(196)
15.2 RenderImageApp 功能简介	(196)
15.3 建立 RenderImageApp 应用程序	(197)
15.3.1 装载所需的包	(197)
15.3.2 声明类 RenderImageApp	(197)
15.3.3 声明类变量和实例变量	(198)
15.3.4 构造构造函数	(198)
15.3.5 创建 main() 方法	(200)
15.3.6 实现 main() 方法	(200)
15.3.7 声明类 renderWindow	(202)
15.3.8 声明类 inputWindow	(203)
15.3.9 声明类 urlField	(205)
15.3.10 保存并编译 RenderImageApp.java	(207)

15.5.11 运行 RenderImageApp	(207)
15.4 小结	(208)

第 16 章 编写协议句柄和内容句柄

(209)

16.1 协议句柄概述	(209)
16.2 协议句柄的编写过程	(210)
16.2.1 准备工作——创建目录结构 ..	(210)
16.2.2 创建文件 Handler.java	(210)
16.2.3 编写 net.www.protocol.run 包	(210)
16.2.4 装入所需的系统包	(210)
16.2.5 声明 Handler 类	(210)
16.2.6 声明并实现 openStream 方法	(211)
16.2.7 实现 openStream	(211)
16.2.8 编译 Handler 类	(213)
16.2.9 修改环境变量 CLASSPATH	(213)
16.2.10 测试 Protocol handler	(213)
16.3 协议句柄的程序清单	(213)
16.4 内容句柄概述	(214)
16.5 编写内容句柄	(215)
16.5.1 创建目录	(215)
16.5.2 编写句柄程序	(215)
16.5.3 更改 HoiJava 环境变量 CLASSPATH	(216)
16.5.4 测试内容句柄	(216)
16.6 小结	(217)

第 17 章 Java 与 Internet 网络程序设计

(218)

17.1 网络资源的下载与分析	(218)
17.1.1 下载 URL 所指向的资源	(218)
17.1.2 使用 URLConnection 类对内容进行简单分析	(219)
17.2 编写数据报应用程序	(220)
17.2.1 发送数据报	(220)
17.2.2 接收数据报	(221)
17.3 编写客户机/服务器应用程序	(222)
17.3.1 客户机/服务器的基本概念	(222)
17.3.2 客户机/服务器的工作模式	(222)
17.3.3 客户机/服务器编程实例	(223)
17.3.4 利用回调技术编写客户机/服务器应用	

程序	(229)
----------	-------

17.4 小结	(231)
---------------	-------

附录 A 按字母顺序索引的 JDK 类库清单

(232)

A	
java.lang.AbstractMethodException	(232)
java.applet.Applet	(233)
java.applet.AppletContext	(234)
java.applet.AppletStub	(235)
java.lang.ArithmeticException	(235)
java.lang.ArrayIndexOutOfBoundsException	(236)
java.lang.ArrayStoreException	(236)
java.applet.AudioClip	(237)
java.awt.AWTError	(237)
java.awt.AWTException	(238)
B	
java.util.BitSet	(239)
java.lang.Boolean	(239)
java.awt.BorderLayout	(240)
java.io.BufferedInputStream	(241)
java.io.BufferedOutputStream	(242)
java.awt.Button	(243)
java.awt.peer.ButtonPeer	(243)
java.io.ByteArrayInputStream	(244)
java.io.ByteArrayOutputStream	(245)
C	
java.awt.Canvas	(246)
java.awt.peer.CanvasPeer	(246)
java.awt.CardLayout	(247)
java.lang.Character	(248)
java.awt.Checkbox	(249)
java.awt.CheckboxGroup	(250)
java.awt.CheckboxMenuItem	(251)
java.awt.peer.CheckboxMenuItemPeer	(251)
java.awt.peer.CheckboxPeer	(252)
java.awt.Choice	(252)
java.awt.peer.ChoicePeer	(253)
java.lang.Class	(253)
java.lang.ClassCastException	(254)
java.lang.ClassCircularError	(255)
java.lang.ClassFormatError	(255)
java.lang.ClassLoader	(256)

java.lang.ClassNotFoundException	(257)	java.awt.image.FilteredImageSource	(299)
java.lang.Closable	(258)	java.io.FilterInputStream	(300)
java.lang.CloneNotSupportedException	(258)	java.io.FilterOutputStream	(301)
java.awt.Color	(259)	java.lang.Float	(302)
java.awt.image.ColorModel	(260)	java.awt.FlowLayout	(303)
java.lang.Compiler	(261)	java.awt.Font	(304)
java.awt.Component	(262)	java.awt.FontMetrics	(306)
java.awt.peer.ComponentPeer	(266)	java.awt.Frame	(307)
java.awt.Container	(267)	java.awt.peer.FramePeer	(308)
java.awt.peer.ContainerPeer	(268)		
java.net.ContentHandler	(269)	java.awt.Graphics	(309)
java.net.ContentHandlerFactory	(269)	java.awt.GridBagConstraints	(311)
java.awt.image.CropImageFilter	(270)	java.awt.GridBagLayout	(312)
		java.awt.GridLayout	(314)
D			
java.net.DatagramPacket	(270)	H	
java.net.DatagramSocket	(271)	java.util.Hashtable	(314)
java.io.DataInput	(272)	I	
java.io.DataInputStream	(273)	java.lang.IllegalAccessException	(316)
java.io.DataOutput	(275)	java.lang.IllegalArgumentException	(317)
java.io.DataOutputStream	(276)	java.lang.IllegalMonitorStateException	(317)
java.util.Date	(278)	java.lang.IllegalThreadStateException	(318)
java.awt.Dialog	(280)	java.lang.IllegalAccessError	(318)
java.awt.peer.DialogPeer	(281)	java.awt.Image	(319)
java.util.Dictionary	(282)	java.awt.image.ImageConsumer	(319)
java.awt.Dimension	(282)	java.awt.image.ImageFilter	(321)
java.awt.image.DirectColorModel	(283)	java.awt.image.ImageObserver	(322)
java.lang.Double	(284)	java.awt.image.ImageProducer	(323)
		java.lang.IncompatibleClassChangeError	(323)
E			
java.util.EmptyStackException	(286)	java.awt.image.IndexColorModel	(324)
java.util Enumeration	(286)	java.lang.IndexOutOfBoundsException	(325)
java.io.EOFException	(287)	java.net.InetAddress	(325)
java.lang.Error	(287)	java.io.InputStream	(326)
java.awt.Event	(287)	java.awt Insets	(327)
java.lang.Exception	(291)	java.lang.InstantiationException	(328)
		java.lang.InstanceOfError	(328)
F		java.lang.Integer	(329)
java.io.File	(292)	java.lang.InternalError	(330)
java.io.FileDescriptor	(294)	java.io.InterruptedIOException	(331)
java.awt.FileDialog	(294)	java.lang.InterruptedError	(331)
java.awt.peer.FileDialogPeer	(296)	java.io.IOException	(332)
java.io.FileInputStream	(296)		
java.io.FileNameFilter	(297)	L	
java.io.FileNotFoundException	(297)	java.awt.Label	(332)
java.io.FileOutputStream	(298)		

java.awt.peer.LabelPeer	(333)	java.awt.Point	(382)
java.awt.LayoutManager	(334)	java.awt.Polygon	(383)
java.io.LineNumberInputStream	(334)	java.io.PrintStream	(384)
java.lang.LinkageError	(335)	java.lang.Process	(386)
java.awt.List	(336)	java.util.Properties	(386)
java.awt.peer.ListPeer	(337)	java.net.ProtocolException	(387)
java.lang.Long	(338)	java.io.PushbackInputStream	(388)
M		R	
java.net.MalformedURLException	(340)	java.util.Random	(369)
java.lang.Math	(340)	java.io.RandomAccessFile	(369)
java.awt.MediaTracker	(342)	java.awt.Rectangle	(372)
java.awt.image.MemoryImageSource	(345)	java.awt.image.RGBImageFilter	(374)
java.awt.Menu	(345)	java.lang.Runnable	(375)
java.awt.peer.MenuBarPeer	(348)	java.lang.RuntimeException	(375)
java.awt.MenuBar	(347)	java.lang.Runtime	(376)
java.awt.MenuContainer	(348)	S	
java.awt.peer.MenuComponentPeer	(348)	java.awt.peer.ScrollbarPeer	(377)
java.awt.MenuComponent	(348)	java.awt.Scrollbar	(377)
java.awt.peer.MenuItemsPeer	(349)	java.lang.SecurityManager	(378)
java.awt.MenuItem	(350)	java.lang.SecurityException	(380)
java.awt.peer.MenuPeer	(350)	java.io.SequenceInputStream	(381)
N		java.net.ServerSocket	(381)
java.lang.NegativeArraySizeException	(351)	java.net.Socket	(382)
java.lang.NoClassDefFoundError	(351)	java.net.SocketImplFactory	(383)
java.util.NoSuchElementException	(352)	java.net.SocketImpl	(384)
java.lang.NoSuchMethodException	(352)	java.net.SocketException	(385)
java.lang.NoSuchFieldError	(353)	java.util.Stack	(385)
java.lang.NoSuchMethodError	(353)	java.lang.StackOverflowError	(386)
java.lang.NullPointerException	(354)	java.io.StreamTokenizer	(386)
java.lang.Number	(354)	java.lang.String	(388)
java.lang.NumberFormatException	(355)	java.io.StringBufferInputStream	(388)
O		java.lang.StringBuffer	(392)
java.lang.Object	(355)	java.util.StringTokenizer	(395)
java.util.Observable	(357)	java.lang.StringIndexOutOfBoundsException	(396)
java.util.Observer	(357)	java.lang.System	(397)
java.lang.OutOfMemoryError	(358)	T	
java.io.OutputStream	(358)	java.awt.peer.TextAreaPeer	(398)
P		java.awt.TextArea	(399)
java.awt.Panel	(359)	java.awt.peer.TextComponentPeer	(400)
java.awt.peer.PanelPeer	(359)	java.awt.TextComponent	(400)
java.io.PipedInputStream	(360)	java.awt.peer.TextFieldPeer	(401)
java.io.PipedOutputStream	(361)	java.awt.TextField	(402)
java.awt.image.PixelGrabber	(361)		

java.lang.Thread	(403)
java.lang.ThreadGroup	(406)
java.lang.ThreadDeath	(407)
java.lang.Throwable	(408)
java.swing.Toolkit	(409)

U

java.lang.UNIXProcess	(410)
java.net.UnknownHostException	(411)
java.net.UnknownServiceException	(411)
java.lang.UnknownError	(412)
java.lang.UnsupportedLinkError	(412)
java.net.URL	(413)
java.net.URLConnection	(415)
java.net.URLStreamHandlerFactory	(417)
java.net.URLStreamHandler	(418)
java.io.UTFDataFormatException	(418)

V

java.util.Vector	(419)
java.lang.VerifyError	(421)
java.lang.VirtualMachineError	(422)

W

java.lang.Win32Process	(422)
java.awt.Window	(423)
java.awt.peer.WindowPeer	(424)

附录 B JDK 类库清单 (425)

B.1 java.applet	(425)
B.1.1 接口	(425)
B.1.2 类	(425)
B.2 java.awt	(425)
B.2.1 接口	(425)
B.2.2 抽象类	(425)
B.2.3 类	(426)
B.2.4 java.awt 类库的结构	(427)
B.3 java.awt.image	(428)
B.3.1 接口	(428)
B.4 java.awt.peer	(428)
B.4.1 接口	(428)
B.5 java.io	(429)
B.5.1 接口	(429)
B.5.2 抽象类	(429)
B.5.3 类	(429)
B.6 java.lang	(431)

B.6.1 接口	(431)
B.6.2 抽象类	(431)
B.6.3 类	(431)
B.7 java.net	(433)
B.7.1 接口	(433)
B.7.2 抽象类	(434)
B.7.3 类	(434)
B.8 java.util	(434)
B.8.1 接口	(435)
B.8.2 抽象类	(435)
B.8.3 类	(435)

附录 C Java 资源 (436)

C.1 主要的 WWW 站点	(436)
C.2 支持 Java 的 WWW 浏览器	(436)
C.2.1 Netscape Navigator 2.0	(436)
C.2.2 Oracle 公司的 PowerBrowser	(436)
C.2.3 Microsoft Explorer	(436)
C.2.4 Liquid Reality	(437)
C.3 Java 开发工具	(437)
C.3.1 Sun 公司的 Java 开发工具	(437)
C.3.2 Symantec Espresso	(437)
C.3.3 Symantec Cafe	(437)
C.3.4 Hyperware Opus	(438)
C.3.5 ETL HORB	(438)
C.3.6 JSC	(438)
C.3.7 JavaMaker	(438)
C.3.8 Liquid Reality	(439)
C.3.9 ADONIS	(439)
C.3.10 JavaScript	(439)
C.3.11 Ruaster for Macintosh	(439)
C.3.12 Black Star 免费 Java 编译器	(439)
C.3.13 SunSoft 的 Java Workshop	(439)
C.3.14 Rogue Wave 的 JFactory	(439)
C.3.15 Holland C++	(440)

附录 D Java 和 C++ 之间的不同点 (441)

D.1 Java 不支持的 C++ 特性	(442)
D.2 Java 对 C++ 的扩充	(442)

附录 E Java 虚拟机简介 (444)

E.1 Java 源文件的编译、下载、解释和执行	(444)
--------------------------	-------

E.2 JVM 规格描述	(445)	E.2.4 JVM 设计回收堆	(446)
E.2.1 JVM 指令系统	(445)	E.2.5 JVM 存储区	(446)
E.2.2 JVM 寄存器	(445)		
E.2.3 JVM 栈结构	(446)	附录 F Unicode 编码标准简介	(447)

本书可提供随书软盘,内容如下:

1. 书中源程序清单;
2. JDK 1.0.2 安装程序及文档;
3. HotJava 1.0 prebeta 版;
4. 最新 Applet 编程竞赛获奖程序源码。

欲购者请与北京科海培训中心联系。联系电话:62562449 62589259 62569289
通信地址:北京海淀路 82 号科海培训中心

第1章 Internet 基本概念

本章是全书的预备章节,讲述 Internet 的一些基本概念。Internet 近年来已经成为一个广为流行的话题,各种报章杂志都纷纷开设专栏,对其专门予以介绍。国内的 Internet 使用也初具规模,形成了相当的用户群。与 Internet 相比,Java 的出现要晚得多,它的出现和流行与 Internet 密切相关。要想充分发挥 Java(以及 HotJava 浏览器)的潜力,必须对 Internet 有所了解。这就是本章的目的所在。

Java 和 Internet 的关系主要有三个方面:首先,Java 是随着 Internet 的应用普及而流行开来的,Internet 是 Java 的主要发行渠道;其次,从语言特性上讲,Java 是适合于 Internet 网络计算环境的编程语言;最后,Java 小应用程序(Applet)是专门基于 WWW 环境的一种应用程序。本书将在一些章节专门介绍 Java 小应用程序,本章在很大程度上也是专门为此作准备的。

本章内容是专门为一部分读者而安排的;由于各种主观或客观上的原因,这些读者对 Internet 和 WWW 等领域知之甚少。如果你对 Internet 已经很了解,熟悉这个遍布世界的网络的各个方面,你可以跳过本章,直接进入第2章。

1.1 网络与网络协议

本节首先阐述计算机网络的基本概念,然后介绍网络协议和 OSI 参考模型,最后介绍与 Internet 密切相关的 TCP/IP 协议。

1.1.1 何谓网络

计算机网络是利用通信线路连接起来的、相互独立的计算机的集合。这包含两个概念:首先,这些计算机之间必须存在通信线路;其次,这些计算机必须相互独立。从这个定义出发,传统的主机/终端式系统就不能算是完全意义上的计算机网络。

按地理位置划分,计算机网络可以分为四种:网际网、广域网(WAN)、城域网(MAN)和局域网(LAN)。其中,本书专门介绍的 Internet 就是世界上最大的广域网,另外还存在其他网际网类型,它们通常连接着处于同一大洲或同一地域范围的多个国家;广域网一般指连接一个国家的各个地区的网络,分布距离一般在 100 到 1000 公里之间;城域网又称都市网,它的覆盖范围一般为一个城市,方圆不超过 10 到几十公里;局域网的地理分布则相对较小,例如一幢建筑物或一所学校等。

计算机网络的传输速率通常以每秒钟传输的数据位来计算,从传输速度看,局域网的通信速率是最高的,例如以太网的速率一般为 10M bps,快速以太网为 100M bps;城域网的速度低于局域网;网际网的速度则相对较低。

通信线路及联入网络的计算机结合在一起的总体布局模型称为网络拓扑。按网络拓扑划分,计算机网络可以分为总线网络、环网、星形网和混合网络等,局域网最常用的以太网是

典型的总线网络。

联入网络的各计算机称为主机(host)或节点(node)。在 Internet 发展初期,人们把每台计算机称为主机。节点是一个地理上的概念,一个节点可以有多个主机。现在这两个名称已经可以互换使用。如不特别说明,它们都指一台计算机,Internet 上所有节点的地位都是平等的。

1.1.2 何谓网络协议

计算机网络利用通信线路将计算机连接在一起,网络用户在计算机之间通过网络互相交换信息,就如人与人之间相互交流时需要遵循一定的规矩一样,计算机之间的相互通信也需要共同遵守一定的规则,这些规则称为计算机网络协议。

简单地说,计算机网络协议就是针对计算机之间相互交换信息的方式、秩序以及参数所作出的规定。一台计算机只有在遵守某个协议的前提下,才能在计算机网络上与其他计算机进行正常的通信。

网络协议通常被分为几个层次,每层完成自己单独的功能。通信双方只有在相同层之间才需要相互联系。低层协议为高层协议提供服务。这种工作方式可以使计算机之间的信息交换变得简单而易于理解。下节介绍的开放系统互连参考模型就是最典型的分层模型。

1.1.3 开放系统互连参考模型

开放系统互连参考模型(OSI)是国际标准化组织(ISO)推出的网络互连的标准模型。整个模型共分 7 层,如图 1.1 所示。

OSI 参考模型作为阐述计算机网络工作原理的一个理论模型是很合适的,但要投入实际应用,这个模型却存在一些缺陷,其中主要有两个原因,首先是这个模型太过详细,七层的层次数目显然多了一些;其次是各层次所包括的功能不均衡,其数据链路层功能过多,而应用层、传送层等的功能又太少。另外,当 OSI 问世时,许多计算机网络系统已经具有一定的规模,这些系统所遵循的协议也已经接近成熟。这种情形下,要使这些网络完全转到 OSI 模型显然不太现实。

1.2 TCP/IP 协议

OSI 分层协议模型更多的是被作为一个理论模型,与该模型相对应,一个非常实用的网络协议是 TCP/IP 协议。TCP/IP 协议的问世要早于 OSI 参考模型,它是 Internet 网络所遵循的协议。TCP/IP 可以说是一个“既成事实”的标准,起源于 Internet 的前身 ARPANET。

1.2.1 TCP/IP 协议的起源

TCP 协议最早由斯坦福大学的 Vinton Cerf 和 Bob Kahn 于 1973 年提出。1974 年,他们向“IEEE Transaction of Communications”杂志提交了一篇有关 TCP 协议的文章,详细介绍了 TCP 协议的设计思想。1983 年,TCP/IP 被 Unix 4.2BSD 系统采纳,成为该 Unix 版本的一个组成部分。随着 Unix 的成功,TCP/IP 逐步成为 Unix 机器的标准网络协议。目前不仅 Unix 系统,其他很多操作系统也实现了 TCP/IP 协议。



图 1.1 ISO 开放系统互连参考模型

ARPANET 最初使用 NCP 协议,即 Network Control Protocol,但 ARPANET 的研究人员一直在试验新的协议。由于 TCP/IP 逐渐成了一种工业标准,加入 ARPANET 的所有子网几乎都采用了 TCP/IP,因此,美国国防部希望也在 ARPANET 上尝试 TCP/IP。在试验过程中,ARPANET 的技术人员对 TCP/IP 作了许多改进。1983 年,NCP 终于被这种技术上更为先进的 TCP/IP 协议集所代替。与此同时,美国国防部部长办公室下令加入 ARPANET 的机器均必须采用 TCP/IP 协议。

1.2.2 TCP/IP 协议族

TCP/IP 这个名称可分为两个部分,TCP 和 IP,TCP 其实是“传输控制协议”(Transport Control Protocol)的缩写,IP 表示“Internet 协议”或称 IP 协议(Internet Protocol)。不过,TCP/IP 却并不仅仅代表这两个协议,它表示的是一个协议集合,称为 TCP/IP 协议族(protocol suite)。一个协议集就是指由一系列相互补充且互相协作的协议所构成的集合。除传输控制协议和 IP 协议外,TCP/IP 协议集还包括其他一些协议,下表给出了一些常用的 TCP/IP 协议。

表 1.1 常用 TCP/IP 协议及其用途

协议	用途
IP	用于在主机之间传递数据,相当于 ISO/OSI 中的网络层协议
TCP	用于在应用程序之间传递数据,相当于 ISO/OSI 中的传输层协议
UDP	用户数据报协议,一种比 TCP 更为简单的传输层协议
ICMP	Internet 控制报文协议,发送出错消息及异常情况报告

可以看出,TCP/IP 这个用词其实并不贴切,因为 TCP 和 IP 分别是两个协议,而整个

TCP/IP 协议集又还包括多个协议。这很容易使人产生疑惑。因此,有人提出用“Internet 协议集”来表示所有各种与 Internet 相关的协议,也即原来的 TCP/IP 协议集。这种方法有其可取之处,但许多专业人员仍然更愿意采用原来的称呼,甚至分别称呼其中的各种协议。

除上面的 4 个协议外,TCP/IP 协议集还包括了其他一些协议,其中较为重要的包括:

表 1.2 其他 TCP/IP 协议

协议	全名称
UUCP	Unix—Unix Copy Protocol
简单网络管理协议(SNMP)	Simple Network Management Protocol
简单邮件传输协议(SMTP)	Simple Mail Transmission Protocol
网络新闻传送协议(NNTP)	NetNews Transfer Protocol
文件传输协议(FTP)	File Transfer Protocol
点对点通信协议(PPP)	Point to Point Protocol
串行线路 IP 协议(SLIP)	Serial Line Internet Protocol
超文本传送协议(HTTP)	HyperText Transport Protocol

随着 Internet 服务的不断增加,TCP/IP 协议集又不断扩充了一些新的协议。比方说,超文本传送协议就是最近引入到 TCP/IP 协议集的,该协议的出现与 WWW 服务的出现、发展密切相关。

1.2.3 TCP/IP 协议的层次

TCP/IP 协议分为如下 4 层:

网络接口层:负责接收和发送物理帧。

网络层:负责相邻节点之间的通信。

传输层:负责起点到终点的通信。

应用层:提供诸如文件传输、电子邮件等应用程序。

TCP/IP 协议与 OSI 参考模型是相互兼容的。也就是说,我们可以画一个从 TCP/IP 协议栈到 OSI 模型的一个对应图,从中可以清楚地看出 TCP/IP 协议集中各协议所处的层次。TCP/IP 协议栈和 OSI 模型的对应关系如图 1.2 所示。

1.3 Internet 网络及其发展

Internet 是当今最大的国际性计算机互连网络,是美国正在建设中的“信息高速公路”的支柱网络之一。本节将简单介绍 Internet 网络的产生、发展及现状。

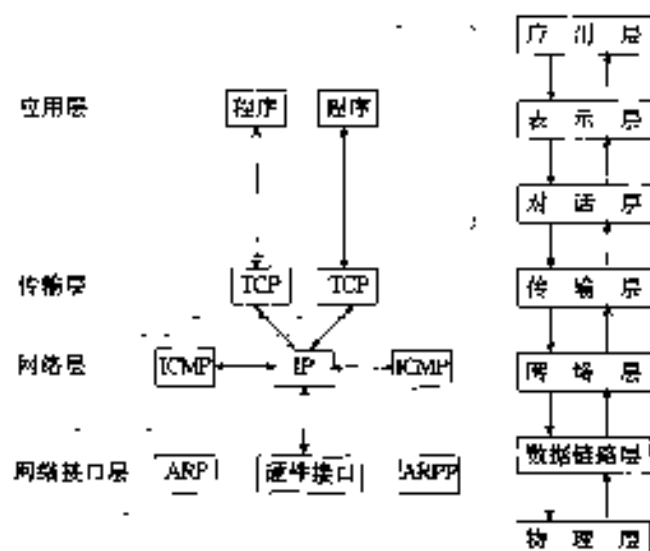


图 1.2 TCP/IP 协议层次与 ISO/OSI 模型的对应关系

1.3.1 Internet 的产生

Internet 的出现可以追溯到本世纪 60 年代初。当时正处于冷战时期,受美国国防部委托,Rand 公司开展了一项研究,研究的内容是一种受到核打击后仍能有效地实施控制和指挥的网络结构。在 Rand 公司于 1964 年提交的研究报告中指出:这样的网络必须是分布式的,也就是说,网络上各节点必须具有平等的地位,每个节点都必须能够生成、接收和传送信息;网上的信息首先被分解为“分组”(packet),并加以编号。这些分组从源节点传送到目的节点后,在目的节点进行组装,恢复为原始信息。用户只需要关心信息是否能正确地由源节点传送到目的节点,而不用关心每个分组的具体传输路径,这意味着每个分组所经过的传输路径可以不同。这样,即使某些节点或通信线路由于出现故障而无法使用,分组也仍然可以通过其他的路径传输,从而保证信息正确地传送到目的地。

1968 年,英国的图象物理实验室率先建立了一个基于上述思想的计算机网络。同一年,美国国防部下属的高级研究计划局(Advanced Research Project Agency,简称 ARPA)也决定建立一个类似的分布式计算机网络,这个网络就是后来的 ARPANET,ARPANET 的网络功能通过一种称为接口消息处理器(Interface Message Processor,简称 IMP)的部件实现。总部设在马萨诸塞州的 BBN 公司赢得了研制 IMP 的合同。

1969 年 9 月,BBN 公司提交了首批的 4 个 IMP,它们分别送到了斯坦福研究所(Stanford Research Institute)、加州大学圣巴巴拉分校、洛杉矶分校和犹他分校。9 月 2 日,这 4 个 IMP 正式开通,各节点之间开始交换信息。这就是 ARPANET 的原型,上面共有 4 个节点。

ARPANET 最初只是一个实验,其目的是想看看哪种网络最好并检验此种网络的稳定性。它的研究目标有二,首先是设计这样一种网络,这种网络即使损坏甚至丢失了某些物理部件也不会影响整个网络的通信;其次是该网络应该具有好的拓扑结构,使其在增加或撤消