

高 | 等 | 学 | 校 | 教 | 材

Java 技术 及其应用 (第二版)

王克宏 主编 董 丽 王少锋 赵 仪 编著 马小军 审

01010010101010101000100101
0100100100010010100100100
10100010010001000100100
010001000010001000100100
001000100101010110
100100100010110



高等教育出版社

高等学校教材

Java 技术及其应用

(第二版)

王克宏 主编

董 丽 王少锋 赵 仪 编著

马小军 审

高等教育出版社

内容提要

本书较为全面地介绍了 Java 语言的特点和应用技术。全书分为三篇。第一篇 Java 语言规范, 介绍 Java 语言的基本特点, 且为了方便读者掌握 Java 语言的最新规范, 对 JDK 5.0 的新规范进行了全面介绍。第二篇编程技术及应用, 着重讲述如何利用 Java 语言的特性编写应用程序, 内容包括: 输入/输出处理; Java 数据结构; 线程; 图形用户界面设计 (AWT 和 Swing); 编写适合在因特网上发布的 applet 程序; 利用 Java 多线程机制提高程序的性能; 网络通信。第三篇高级应用技术, 内容包括: J2EE 平台技术; Java 安全框架; JDBC 技术; Java 的 XML 处理技术; Java Web 应用开发技术 (JSP 和 Servlet); Java 的分布式计算技术 (CORBA 与 Java 的集成, Java RMI 技术, Java 的 Web Service 开发技术)。

本书可作为普通高等学校本科程序设计类课程的教材, 也可作为广大计算机技术爱好者自学 Java 技术的参考书。对于了解和掌握高级程序设计技术的软件开发人员, 也可以从本书中获得所需的参考资料。

图书在版编目(CIP)数据

Java 技术及其应用 / 王克宏主编; 董丽, 王少锋, 赵仪编著. —2 版. —北京: 高等教育出版社, 2007.1
ISBN 978-7-04-020769-9

I. J… II. ①王… ②董… ③王… ④赵…
III. Java 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2007) 第 000451 号

策划编辑 倪文慧 责任编辑 康兆华 封面设计 刘晓翔 责任绘图 黄建英
版式设计 王艳红 责任校对 张颖 责任印制 毛斯璐

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100011
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 北京未来科学技术研究所
有限责任公司印刷厂

开 本 787×1092 1/16
印 张 32.25
字 数 730 000

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 1999 年 6 月第 1 版
2007 年 1 月第 2 版
印 次 2007 年 1 月第 1 次印刷
定 价 39.80 元

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换。

版权所有 侵权必究

物料号 20769-00

前 言

Java 编程技术经过十余年的发展,已经从最初的跨平台编程语言,发展成为当今网络计算平台的标准。Java 也逐步成为全球程序员的首选开发平台。全世界 Java 程序开发人员的人数已经超过 450 万,掌握 Java 编程技术的人士就业前景一般比较乐观。

目前,国内各类高等学校普遍开设了 Java 语言程序设计课程,成为各专业学生竞相选修的热门课程。本书第一版于 1999 年推出,作为面向高等学校计算机及相关专业的普及型教材,经过广大师生较广泛的使用,获得了一定的好评。随着近几年 Java 语言相关技术的飞速发展,笔者也陆续收到了来自任课教师和学生的一些建议,希望根据目前的技术发展状况,适当修改和补充一些内容,以满足广大读者的需要。

根据读者使用第一版的反馈意见,为了保持教材的完整性和系统性,本书第二版继续保留作为教学基本要求的一些核心内容,对书中的各个篇章做了适量的补充和修改。第一篇 Java 语言规范,根据 Java 语言的发展情况增加了一些新特性的介绍,如正则表达式、JDK 5.0 语言新特性。第二篇编程技术及应用,增加了 NIO、实用数据结构处理以及 Swing 图形用户界面设计等设计。第三篇高级应用技术,增加了对 J2EE 平台技术的概要介绍及安装使用说明,对新增的 Java 安全框架做了较详细的介绍;配合 XML 技术的流行,增加了 Java 语言的 XML 处理技术介绍;随着 Java 技术在 Web 应用方面的全面发展,也增加了适当的篇幅进行入门级介绍。另外,对书后习题提供了部分参考答案。

本书由王克宏主编,第 1 章~第 8 章、第 10 章~第 16 章、第 19 章~第 22 章由董丽编写,第 9 章、第 17 章、第 18 章由 Java 软件架构师赵仪先生编写。杨京峰、李健、杨林畅为本书再版提供了许多参考资料。本书在再版编写过程中得到了许斌博士、王少锋博士等提出的修改意见,马小军副教授对全书进行了审阅,在此对他们的辛勤劳动表示衷心的感谢。

尽管本书经过了编者的反复讨论和推敲,并经历了教学实践的检验,但限于个人水平,难免存在不妥之处,希望读者不吝赐教。

王克宏

2006 年 10 月

第一版前言

计算机系统的计算模式经历了从集中计算模式（1946—1983）到分散计算模式（1984—1993）的发展。而自从1993年WWW推出以来，世界已进入了以网络为中心的计算时代，即网络计算模式。在这种发展趋势下，计算机处理信息的方法与过程都发生了根本的演变：

1. 处理的对象从各种类型的数据向具有丰富内容的知识的发展；
2. 处理的形式和方法从数据结构向知识表示的发展；
3. 处理的过程从算法指导向推理指导与问题求解的发展；
4. 处理的结果从静态的文本格式发展为动态的多模式的链接，包括声音、图形、图像、文字、影视、卡通、电子表格，等等。

更重要的是在知识的内涵“质”的变化：知识的内涵不仅仅只是语义的表达，而是可以有多种模式的表现形式，即声音、图形、图像、文字、影视、卡通、电子表格，等等。而知识的传播形式则借助于WWW。用户使用的环境千差万别，而要都能实现充分地从中获取、利用信息与知识，Java则是最理想的工具之一。

Java从1995年5月23日正式发布以来，发展势头异常迅猛，如今它已不仅仅是一门语言，而已逐渐成长为一门技术，包括Java的芯片技术、Java的编译技术、Java的数据库连接技术、基于Java的信息家电的连网技术（Jini技术）、企业信息服务的综合求解方案技术（Enterprise JavaBeans, EJB），等等。这些技术都是与人类生活息息相关的，对国民经济的发展，对提高我们民族的科学技术和文化教育水平都具有重要的意义。

在西文发达国家里，几乎所有著名的高等学校都先后开设了有关Java技术的课程（尽管所使用的名称不一样），甚至一些文科专业都争相开设。这是因为，“Java技术的应用与发展是未来知识经济与信息技术发展的一个晴雨表”，甚至掌握Java技术就抓住了“找工作与就业的好机会！”事实上，具有“Java程序员”（Java Programmer Certification）和“Java开发员”（Java Developer Certification）“资格证书”的人，在美国各地区很容易找到工作，而且其薪金比一般的软件开发人员要高。

我们国家也正在掀起使用Java的热潮，从网上购物、网上银行、远程教学、虚拟课堂、远程医疗和诊断，直到“电子政府”，都在纷纷使用Java技术。

因此，“学会Java！掌握Java！熟练地使用Java！”就是本书的初衷。

在本书的编写过程中，得到了清华大学计算机系知识工程组全体成员的大力支持和帮助，特别是刘波硕士、姜河硕士和王少锋博士为本书的编写提供了许多宝贵的资料和经验，朱家维教授审阅了全书，在此对他们表示衷心的感谢。

王克宏

1999年4月

目 录

第一篇 Java 语言规范

第 1 章 Java 语言概述3	5.3 面向对象程序设计方法 39
1.1 Java 语言发展简史 3	5.4 面向对象程序设计语言 39
1.2 Java 语言的特点 4	第 6 章 Java 语言中的面向
1.3 Java 程序举例 6	对象特性42
第 2 章 简单数据类型 10	6.1 类 42
2.1 标识符和保留字 10	6.2 对象 49
2.2 数据类型概述 11	6.3 包 53
2.3 简单数据类型 12	6.4 面向对象特性 55
2.4 各类型数据间的优先关系及相互转换 16	6.5 抽象类和接口 68
第 3 章 运算符和表达式 17	6.6 Inner Class 72
3.1 运算符 17	第 7 章 数组和向量75
3.2 表达式 22	7.1 一维数组 75
第 4 章 控制语句 24	7.2 多维数组 79
4.1 分支语句 24	7.3 向量 82
4.2 循环语句 27	第 8 章 字符串处理85
4.3 与程序转移有关的其他语句 29	8.1 字符串的表示 85
4.4 例外处理语句 32	8.2 访问字符串 86
4.5 注释语句 32	8.3 修改字符串 89
第 5 章 面向对象技术基础 33	8.4 正则表达式 91
5.1 面向对象技术的形成与发展 33	8.5 其他操作 98
5.2 面向对象的基本概念与特征 34	第 9 章 J2SE 5.0 的语言新特性 100
	9.1 泛型编程 100

9.2 增强的 for 循环表达式	115	9.5 变长参数方法	120
9.3 自动装箱与自动拆箱	115	9.6 静态引用	122
9.4 枚举类型	117	9.7 元数据及注解	123

第二篇 编程技术及应用

第 10 章 异常处理	131	第 13 章 线程	189
10.1 异常的概念	131	13.1 线程的概念	189
10.2 异常的处理	134	13.2 线程的使用	190
10.3 异常类的使用	138	13.3 多线程的互斥与同步	201
第 11 章 输入/输出处理	140	第 14 章 图形用户界面设计	207
11.1 I/O 流概述	140	14.1 生成 Java 图形用户界面	207
11.2 文件处理	144	14.2 AWT 事件处理机制	222
11.3 过滤流	153	14.3 AWT 组件库	234
11.4 字符流处理	155	14.4 Swing 组件介绍	245
11.5 对象的串行化	159	第 15 章 Java Applet 技术基础	267
11.6 其他常用流	162	15.1 Applet 简介	267
11.7 NIO	164	15.2 Applet 的执行	268
第 12 章 Java 数据结构	177	15.3 Applet 的 AWT 绘制	275
12.1 数据结构概述	177	15.4 Java 对多媒体的支持	284
12.2 链表	182	15.5 Applet 通信	308
12.3 堆栈	184	第 16 章 网络通信	320
12.4 队列	185	16.1 URL	320
12.5 树	186	16.2 Socket 通信	325
		16.3 数据报通信	332

第三篇 高级应用技术

第 17 章 J2EE 平台技术	343	17.2 J2EE 1.4 中的应用构件	344
17.1 概述	343	17.3 J2EE 1.4 中的标准服务	345
		17.4 J2EE 1.4 平台的安装与运行	349

第 18 章 Java 安全框架	375	20.2 JAXP 简介.....	424
18.1 概要	375	20.3 SAX API.....	425
18.2 Java 加密扩展	376	20.4 DOM API.....	431
18.3 Java 安全套接字扩展	390	第 21 章 Java Web 应用开发技术 ...	444
第 19 章 JDBC 技术	401	21.1 Servlet.....	444
19.1 JDBC 概述.....	401	21.2 JSP	453
19.2 通过 JDBC 访问数据库.....	408	第 22 章 Java 的分布式计算技术 ...	465
第 20 章 Java 的 XML 处理技术	423	22.1 CORBA 与 Java 的集成—— Java IDL 技术.....	465
20.1 概述	423	22.2 Java RMI 技术.....	473
		22.3 Java 的 Web Service 开发技术.....	480
附录 习题与解析	490		
习题一	490		
习题二	490		
习题三	494		
习题四	494		
习题解析	494		
参考文献	506		

第一篇

Java 语言规范

第 1 章 Java 语言概述



1.1 Java 语言发展简史

1.1.1 Java 语言出现的背景

近年来,计算机界最热门的话题大概非 Internet 莫属,而当前 Internet 上最吸引公众注意力的就是 Java 了。自从 1995 年 5 月在 Sun World 大会上 Java 受到万众瞩目以来,便掀起了一股对于 Internet 和 Java 的热潮。Java 语言以其强大的生命力吸引着软件开发人员,而 Internet 上蕴藏了 Java 语言各种模式的无尽的信息资源,一时间众人竞相评说这一新一代网络计算机语言。比尔·盖茨称 Java 为“长时间以来最卓越的程序设计语言”,万维网创始人 Berners-Lee 说:“计算机事业发展的下一个浪潮就是 Java,并且很快会发生。”因此可以说,使用 Java 已成为大势所趋!

我们知道,计算机技术及其应用正在朝着网络化的方向发展,WWW (World Wide Web, 万维网)正在成为世界上最大的信息中心。但目前 WWW 上仍多为文本、图像、声音等静态信息,Java 为其提供了极为简便且功能强大的编程接口,使得软件开发人员可以向 Web 增加动态的、交互性的内容,这也使 Web 页面设计翻开了崭新的一页。Java 的诸多新特性使得网络开发变得简单易行,完善的可移植性及可靠的安全性更为其开辟了极为广阔的前景。Java 同时具有其他许多特性,因此对于那些对生成可在 Internet 上运行的应用程序感兴趣的开发人员来说,有着极大的吸引力。

1.1.2 Java 语言的深远影响

Java 语言的出现,已经引起一场软件革命。

Java 是一种面向对象语言,它能产生小的轻量代码,这些代码从业已存在于客户计算机的类库中获取其多种功能。因为 Java 是一种解释性的跨平台语言,所以用它编写的程序可在许多硬件平台上运行而无需重新编译。它同时还有一个强大的安全模块,可以生成用户可下载的 applet 而无需担忧任何安全性问题。对于那些要在其 Web 站点上增加高级交互性的开发商来说,这些特性使得 Java 成为炙手可热的工具。

大家也许已经发现, 现在的编程工具越来越复杂。Borland C++、Visual C++动辄占用几百兆字节, 性能的确是非常强大了, 但要掌握它们则非得花上一番功夫不可, 令人望而却步。而 Java 的基本解释器只有 40 KB 左右, 加上标准类库和线程的支持也不过 215 KB 左右, 可谓短小精悍, 而功能毫不逊色, 对于多线程到面向对象程序设计以至于多媒体均提供全面支持。

另外, 传统的编程软件往往依赖于具体的实现环境, 一旦环境更换则可能有大批代码需要重写或重新编译。Java 在二进制码上的兼容性使得同一软件能不加修改地用于不同的系统之上, 只需提供相应的 Java 解释器即可。因此, Java 程序可以非常容易地移植到网络上的不同机器上, 而只有这样的程序设计语言才能满足网络上形形色色的系统的不同要求。



1.2 Java 语言的特点

Java 之所以如此受人推崇, 是因其确实堪称新一代编程语言, 具有众多优点, 它简单、面向对象、可移植、与硬件无关且其系统强健安全、具有很高的性能, 并提供了分布性、多线程、动态性的支持, 它还提供了在因特网上加入应用程序的简便方法(Java applet, Java 小应用程序)。下面就具体介绍 Java 的特点, 并将其与 C、C++进行比较。

1.2.1 Java 语言的特点

1. 简单性

Java 的简单性首先体现在其精简的系统, 力图用最小的系统实现足够多的功能。无需强大的硬件环境, Java 在小型机或微型机上便可良好地运行。

对于编程人员来说, Java 也是非常友善的, 无需进行艰难的钻研, 只要理解一些基本概念, 就能编写出合适的应用程序。为了使大多数程序员能够迅速掌握 Java, 它采取了类似于 C++的语法。Java 语言虽然基于 C++, 但它删除了 C++中许多难得一用而又复杂的内容以及可能威胁 Java 安全性的模块, 因为它们往往是弊大于利。其中包括运算符重载、多重继承等, 使人不必纠缠于这些原本就显模糊的概念。所保留的只是一种简单的、面向对象的多线程语言, 其特性包括一个单继承的类系、强类型、支持多线程执行、自动垃圾收集, 并且只采用引用(非指针)。

Java 实现了自动垃圾收集, 使程序设计者不必费心管理内存, 一切工作皆由系统完成。这不但使程序设计变得更加简便, 同时也大大减少了出错的可能。

2. 面向对象技术

同所有新一代的程序设计语言一样, Java 也采用了面向对象技术并更为彻底, 所有的 Java 程序和 applet 均是对象, 通过封装性实现了模块化和信息隐藏, 通过继承性实现了代码的复用, 使用户可以建立自己的类库。

3. 分布性

通过 Java 所提供的类库可以方便地处理 TCP/IP 协议(Transmission Control Protocol/Internet

Protocol, 传输控制协议/网际协议), 用户的应用程序可以通过 URL 地址访问网络上的其他对象, 如同访问本地文件一样简单。

4. 鲁棒性

Java 在编译及运行程序时, 都要进行严格的检查, 以消除发生错误的可能性。作为一种强类型语言, Java 在编译和连接时都进行大量的类型检查, 以防止不匹配问题的发生。如果编程人员引用一种非法类型或执行一个非法类型操作, Java 将在解释时指出该错误。

在 Java 程序中不能用地址计算的方法通过指针访问内存单元, 这样就大大减少了错误发生的可能性。Java 的数组并非用指针实现, 这样就可以避免数组越界情况的发生。Java 的自动垃圾收集功能避免了管理内存时容易出现的许多错误。

5. 安全性

网络上最重要的是安全问题。作为网络语言, Java 必须提供足够的安全保障, 并且要防止病毒的侵蚀。Java 在运行应用程序时, 严格检查其访问数据的权限, 例如不允许网络上的应用程序修改本地的数据。下载到用户计算机中的字节代码在其被执行前要经过核实工具审核, 一旦字节代码被核实, 它便由 Java 运行时解释器来执行, 该解释器通过阻止对内存的直接访问来进一步提高 Java 的安全性。同时, 由于 Java 具有很强的鲁棒性, 也使它具有更好的安全性, 不能使用指针, 应用程序便不能通过计算来非法地访问对象的私有成员。

6. 体系结构中立

Java 是一种网络语言, 而网络上充满了各种不同类型的机器和操作系统。为了使 Java 程序能在网络上的任何地方运行, Java 解释器生成与体系结构无关的字节码结构的文件格式, 它们能在任何种类的处理器的上运行, 只需提供相应的 Java 运行系统即可。

在单一系统上, 体系结构中立同样具有非常大的作用。若不使用 Java, 编程人员不得不编写各种版本的程序, 以在不同的处理器上运行, 而处理器的结构又在不断地发生变化, 这使得开发出能在所有平台上工作的软件几乎是不可能的。使用 Java, 则同一版本的应用程序就可以运行在所有的平台之上。

7. 可移植性

体系结构中立使得 Java 程序具有很好的可移植性。同时, Java 的类库也具有可移植性, 可以在不同的平台上使用。另外, Java 的编译器由 Java 语言实现, 运行器由标准 C 实现, 因此 Java 本身也具有可移植性。

8. 解释执行

Java 解释器直接对 Java 字节码解释执行, 字节码本身包含了许多编译时所产生的信息, 使连接过程更加简单。

9. 高性能

尽管 Java 是解释执行的, 但它具有非常高的性能, 在一些特定的 CPU (Central Processing Unit, 中央处理器) 上, Java 字节码可以飞速地转换成机器码, 让人几乎无法感觉到它是在解释执行的。

由于 Java 字节码格式的设计是针对于机器码的转换进行的, 实际转换时便相当简便, 自动的寄存器分配及编译器对字节码的一些优化使之能够生成高质量的代码。

10. 多线程

多线程使得应用程序可以同时进行不同的操作, 处理不同的事件。例如, 网络连接需要时间, 如果采用事件循环机制, 则会造成长时间的等待。在多线程机制中, 不同的线程处理不同的任务, 彼此互不干涉, 不会由于某处的等待而影响其他部分, 这样就可以容易地实现网络上的实时交互性操作。

11. 动态性

在许多方面, Java 较之 C、C++更具有动态性, 它被设计为适应于不断发展的环境。在类库中可以自由地加入新的方法或实例量而不会影响使用这一类库的应用程序的执行。

1.2.2 Java applet 的特点

Java applet 是一类特殊的 Java 程序, 它嵌入在 HTML 语言中, 随主页发布到 Internet 上。用户利用 Java applet, 可以非常方便地实现 Internet 上的程序编写, 实现多媒体用户界面或进行复杂的计算。Java 程序的简单性使其与其他语言相比大大缩短了下载时间, 而 Java applet 的结构更加简单, 代码长度更短。Java 的可移植性保证了 Java applet 可以在网络的任意地方运行, 只要具备支持 Java 的浏览器即可。Java 的安全检查机制保证用户不会受到病毒的入侵。Java applet 可以使 Internet 上的信息真正具有动态性和交互性。

所有 Java applet 都是 Applet 类的子类。Applet 类为用户做好了程序的框架, 用户只需填入相应的函数实现具体功能即可, 无需考虑诸如窗口创建、事件处理之类的问题, 大大简化了编程的复杂性。



1.3 Java 程序举例

Java 程序分为两种: Java application (Java 应用程序) 和 Java applet (Java 小应用程序), 下面通过两个简单的例子分别说明这两种程序的应用方法。

1.3.1 Java 应用程序

例 1.1

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

这个程序只实现了一个功能，即输出一行信息：

Hello World!

通过这个简单的应用程序，我们可以粗略地了解 Java 应用程序的基本结构。

1. 类定义

所有的 Java 应用程序都是类，本例中即为 HelloWorldApp 类。用保留字 `class` 来声明新类，`public` 指明这是一个公共类。Java 程序中可以定义多个类，但最多只能有一个公共类。

2. main()方法

例 1.1 中定义了一个方法，这就是 `main()` 方法。事实上，所有应用程序都必须有且仅有一个 `main()` 方法；而且必须用 `public`、`void` 和 `static` 指定。`public` 指明所有的类都可以使用这一方法，`void` 指明本方法不返回任何值；`static` 则指明本方法是一个类方法，可以通过类名直接调用。Java 解释器在未生成任何实例的情况下，以 `main()` 方法作为入口来执行程序。

在 `main()` 方法定义中，括号中的 `String args[]` 是传递给 `main()` 方法的参数，参数名为 `args`，它是类 `String` 的一个实例。参数可以有一个或多个，每个参数用“类名 参数名”来指定，多个参数之间用逗号分隔。

3. 程序内容

在例 1.1 的 `main()` 方法中，只有一条语句

```
System.out.println("Hello World!");
```

用来实现字符串的输出，实现与 C 语言中的 `printf()` 和 C++ 语言中的 `cout<<` 相同的功能。

4. 注释

与 C++ 语言相同，一行中“//”后的内容为注释，“/*”和“*/”之间所包含的内容也是注释。

现在可以运行此程序。首先把它放在名为 `HelloWorldApp.java` 的文件中，注意：文件名必须与公共类名相同，因为 Java 解释器要求公共类必须放在与之同名的文件中。然后对文件进行编译：

```
javac HelloWorldApp.java
```

生成字节码文件 `HelloWorldApp.class`。最后用 Java 解释器运行它：

```
java HelloWorldApp
```

于是在屏幕上显示：

```
Hello World!
```

1.3.2 Java 小应用程序

例 1.2

```
import java.awt.Graphics;
import java.applet.Applet;
public class HelloWorldApplet extends Applet{
    public String s;
```

```

public void init() {
    s = new String("Hello World");
}

public void paint(Graphics g) {
    g.drawString(s, 25, 25);
}
}

```

通过例 1.2 可以了解到 Java applet 的一些基本结构。

1. 类定义

与 Java 应用程序相同，所有的 Java applet 也都是类，而且必须是 Applet 类的子类，用保留字 extends 说明。

2. 方法定义

在 Java applet 中没有 main() 方法，这是 Java applet 与 Java 应用程序之间的主要区别之一。Applet 类实现了许多方法，构造了一个窗口程序的主要框架，用户只需重写所需的方法即可，系统将对其自动调用。

3. 类库支持

这个程序中用到了类库中的一些方法，因此必须在程序中进行声明。在 Java 中，用 import 来输入包或类，例 1.2 中输入了 java.awt.Graphics 和 java.applet.Applet 类。

4. 运行 Java applet

首先要进行编译，同样也要放在与公共类同名的文件中，即为 HelloWorldApp.java，然后进行编译：

```
javac HelloWorldApp.java
```

得到字节码文件 HelloWorldApp.class。

与应用程序不同的是，Java applet 中没有 main() 方法作为程序的入口，不能用前面介绍的方法运行。必须编写 HTML 文件，把 applet 嵌入其中，然后用 appletviewer 来运行，或在支持 Java 的浏览器上运行。HTML 文件 helloworld.html 如下：

```

<HTML>
<applet code="HelloWorldApp.class">
</applet>
</HTML>

```

现在可以通过 appletviewer 来观看这个 applet 的运行效果了，如图 1.1 所示。

1.3.3 Java 程序结构

每种语言都有自己的程序书写规范，Java 语言也不

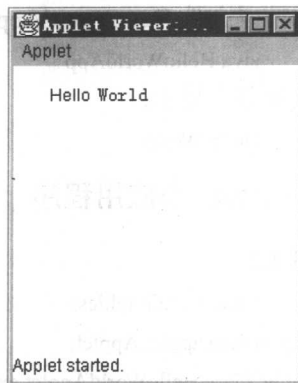


图 1.1 Java 小应用程序运行效果示例

例外。在编写 Java 程序时，必须遵从一定的规范，才能使程序顺利地通过编译器的编译。下面列出典型 Java 程序的书写规范，其中所涉及的某些概念将在后续章节中介绍。

1. 命名

如果.java 文件包含一个 **public** 类，它必须按该类名命名。如在例 1.1 中，包含一个 **public** 类型的类 **HelloWorldApp**，所以该文件名必须命名为 **HelloWorldApp.java**，否则编译器会指出错误。

2. 类的个数

一个源文件中最多只能有一个 **public** 类，其他类的个数不限。

3. 结构

<code>package 语句</code>	//无或 1 条，必须放在文件的开始处
<code>import 语句</code>	//无或多条，必须放在所有类的定义之前
<code>public classDefinition</code>	//无或 1 个 //文件名必须与该类的类名完全相同
<code>classDefinition</code>	//无或多个
<code>interfaceDefinition</code>	//无或多个