免费提供电子教案

高等院校规划教材
计算机基础教育系列

Java 语言实用教程

常 亮　张智勇　赵福军　编著

机械工业出版社
CHINA MACHINE PRESS

高等院校规划教材·计算机基础教育系列

Java 语言实用教程

常 亮　张智勇　赵福军　编著

机 械 工 业 出 版 社

Java语言以其完全面向对象、简单高效、与平台无关等突出的特点，已经逐渐成为程序设计的首选语言。

本书共12章，从程序设计方法讲起，详细介绍了Java开发环境、语法知识、数组、类、对象、继承、接口等面向对象程序设计和开发的知识及应用，同时包括异常处理、输入输出方法、图形图像、多媒体、Applet等方面的内容。

本书每章都有相当数量的精选例题和习题，还附有两章完整的综合练习及答案。通过学习本书的内容，将为读者以后学习Java语言的各种高级应用打下坚实的基础。

本书实用性强，由浅入深，适合作为高等院校计算机及相关专业的教材和培训用书。同时，本书也是广大Java爱好者学习Java程序设计，Java语言等级考试的参考书。

图书在版编目(CIP)数据

Java语言实用教程/常亮，张智勇，赵福军编著. —北京：机械工业出版社，2007.7
(高等院校规划教材·计算机基础教育系列)
ISBN 978-7-111-21809-8

Ⅰ.J… Ⅱ.①常… ②张… ③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098598号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)
策　　划：赵　慧
责任编辑：韩　菲
责任印制：洪汉军

三河市宏达印刷有限公司印刷

2007年7月第1版·第1次印刷
184mm×260mm·14.75印张·359千字
0001－5000册
标准书号：ISBN 978-7-111-21809-8
定价：24.00元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：(010)68326294
购书热线电话：(010)88379639　88379641　88379643
编辑热线电话：(010)88379739
封面无防伪标均为盗版

出版说明

计算机技术的发展极大地促进了现代科学技术的发展，明显地加快了社会发展的进程。因此，各国都非常重视计算机教育。

近年来，随着我国信息化建设的全面推进和高等教育的蓬勃发展，高等院校的计算机教育模式也在不断改革，计算机学科的课程体系和教学内容趋于更加科学和合理，计算机教材建设逐渐成熟。在“十五”期间，机械工业出版社组织出版了大量计算机教材，包括“21世纪高等院校计算机教材系列”、“21世纪重点大学规划教材”、“高等院校计算机科学与技术‘十五’规划教材”、“21世纪高等院校应用型规划教材”等，均取得了可喜成果，其中多个品种的教材被评为国家级、省部级的精品教材。

为了进一步满足计算机教育的需求，机械工业出版社策划开发了“高等院校规划教材”。这套教材是在总结我社以往计算机教材出版经验的基础上策划的，同时借鉴了其他出版社同类教材的优点，对我社已有的计算机教材资源进行整合，旨在大幅提高教材质量。我们邀请多所高校的计算机专家、教师及教务部门针对此次计算机教材建设进行了充分的研讨，达成了许多共识，并由此形成了“高等院校规划教材”的体系架构与编写原则，以保证本套教材与各高等院校的办学层次、学科设置和人才培养模式等相匹配，满足其计算机教学的需要。

本套教材包括计算机科学与技术、软件工程、网络工程、信息管理与信息系统、计算机应用技术以及计算机基础教育等教材系列。其中，计算机科学与技术系列、软件工程系列、网络工程系列和信息管理与信息系统系列是针对高校相应专业方向的课程设置而组织编写的，体系完整，讲解透彻；计算机应用技术系列是针对计算机应用类课程而组织编写的，着重培养学生利用计算机技术解决实际问题的能力；计算机基础教育系列是为大学公共基础课层面的计算机基础教学而设计的，采用通俗易懂的方法讲解计算机的基础理论、常用技术及应用。

本套教材的内容源自致力于教学与科研一线的骨干教师与资深专家的实践经验和研究成果，融合了先进的教学理念，涵盖了计算机领域的核心理论和最新的应用技术，真正在教材体系、内容和方法上做到了创新。而且本套教材根据实际需要配有电子教案、实验指导或多媒体光盘等教学资源，实现了教材的“立体化”建设。本套教材将随着计算机技术的进步和计算机应用领域的扩展而及时改版，并及时吸纳新兴课程和特色课程的教材。我们将努力把这套教材打造成为国家级或省部级精品教材，为高等院校的计算机教育提供更好的服务。

对于本套教材的组织出版工作，希望计算机教育界的专家和老师能提出宝贵的意见和建议。衷心感谢计算机教育工作者和广大读者的支持与帮助！

机械工业出版社

前　言

Java语言是当前最为流行的程序设计语言之一，诸多优秀的特性使其成为业界广泛认可和采用的对象。同时，越来越多的高校也将其作为程序设计教学中主要的编程工具。目前，社会上需要大量的Java开发人员，随处可见的高薪诚聘Java开发人员的信息让很多刚开始学习编程的朋友心动不已。然而，Java作为一种跨平台的程序设计语言，其版本涵盖的范围较广，从定位于嵌入式系统应用的J2ME、定位于客户端程序应用的J2SE，到定位于服务器端程序应用的J2EE，Java均完整地提供了相关的解决方案。初学者很难在众多的Java图书中找到适合自己的入门教材。本书的编写目的就在于帮助Java初学者，力求以最简单、最实用的文字帮助初学者甚至是毫无编程基础的读者们快速走进Java程序的精彩世界。

读者只需具有最基本的计算机知识，按照本书的体系便能循序渐进地学会使用Java语言进行程序设计。每章的结尾都配有精心挑选的练习题，通过它们，读者能将每章节的知识掌握得更为扎实和牢固。学习编程一定要多操作多练习，为此，本书为读者精心挑选了很多精彩的Java应用范例，相信通过自己亲手调试，结合所学习的理论知识，每运行一个Java范例程序，读者都会有新的启示和提高。这些丰富的范例能够帮助读者迅速地掌握Java语言和面向对象程序设计的精髓。

附录部分包括教学计划参考以及两套完整的综合练习题，难度与Java等级考试相仿，作为读者检验学习效果所用。

本书的读者对象包括：

- 学习程序设计的大学生。以本书作为程序设计的基础教材，可以为他们学习和掌握面向对象编程语言及深入学习Java的其他高级应用奠定基础。
- 因工作或科研需要，希望迅速掌握Java编程，以完成不太复杂的编程任务的人员。
- 有一些实际经验但没有系统学习过相关专业知识的Java编程爱好者，或一直使用其他开发工具并希望了解Java语言的程序开发人员。他们通过本书能够系统学习Java程序设计，快速掌握具体概念和理论。

本书是在作者多年实际教学和软件开发经验的基础上编写的。是一本实用性很强的教科书。作者对书中内容进行了精心的设计和安排，按照由浅入深、循序渐进的原则进行组织，程序样例大多简短实用，易于教学和读者学习。

程序源代码和教学课件，读者可以在http://www.cmpbook.com上下载。

本书的出版离不开家人和朋友的支持。在本书的编写过程中，董春游教授、姜成志教授、谢子殿教授、麻秀丽教授为本书的编写提出了许多宝贵的建议。陈静老师为本书做了大量外文资料的整理工作，赵宇宁老师为本书进行了细致的校正和修改，同时要感谢郭继坤和周波的鼎力帮助。

由于作者水平有限加之时间仓促，书中难免有错误、疏漏和不妥之处，恳请各位专家、同仁和读者不吝赐教。

编　者

目　录

第 1 章　程序设计与 Java 语言概述

开始学习程序设计开发前，首先要了解什么是程序、算法以及如何使用流程图对算法进行描述等基础知识和方法。本章旨在使读者对程序设计和 Java 语言有一个初步的认识，为以后的学习打下基础。其内容主要包括程序设计中涉及的一些基本概念、方法，以及面向对象程序设计的相关特点。另外，本章也对 Java 语言的发展、主要特点以及应用进行了介绍。

1.1　程序与算法

程序是操作计算机完成特定任务的指令的集合，程序由程序设计语言来实现。算法用来描述程序的实现步骤。

1.1.1　程序设计语言

人类的语言是一个渐变发展的过程，正像人与人之间的交流是从手势逐渐进化到语言一样，人们与计算机之间的交流也是从简单的机械开关开始，逐渐发展到程序设计语言(Programming Language)——计算机语言。

计算机语言的发展从面向机器到面向结构和过程再到今天的面向对象，根据其所提供的指令能否被计算机直接执行，可将其分为机器语言、汇编语言和高级语言。

1. 机器语言

仅由硬件组成的计算机只能接受由“0”和“1”组成的二进制信息，要操作计算机就要编写一系列的二进制代码。这些由“0”和“1”组成的、能够让计算机直接执行的指令叫做机器指令，由机器指令组成的集合称为机器语言(Machine Language)。

每条机器指令都是一串二进制代码。虽然对于人类来说要记住每一条机器指令及其含义是非常困难的，但是计算机却可以直接识别并执行。因此，计算机执行机器语言的效率是最高的。

例如，在某计算机上要完成 1 + 2 的加法运算，要用 10111000 命令将加数 1(二进制为 00000001)保存，用 00000100 命令完成 1 + 2(二进制为 00000010)的运算。机器语言代码如下：

```
10111000
00000001
00000100
00000010
```

可以看到，使用机器语言编写出来的程序很难阅读。同时，机器语言与计算机硬件有关，不同系列计算机上运行的机器指令各不相同，一台计算机上编制的程序在另一台计算机上可

能根本无法运行，一个问题要在多个计算机上求解，就必须重复编写多个应用程序。因此，机器语言开发程序的缺点是：难以编写、调试、移植和维护。

2. 汇编语言

在机器语言的基础上，利用一些英文缩写的助记符表示机器语言中的指令，通过一个汇编程序翻译成机器语言后再执行，这种语言称为汇编语言(Assembly Language)。汇编语言也是一种面向机器的程序设计语言。汇编语言使用符号来表示指令，例如要完成 1 + 2 的加法运算，需要使用 MOV 命令将加数 1 保存在累加器 AL 中，然后使用 ADD 命令完成 1 + 2 的运算。汇编语言代码如下：

```
MOV AL,1
ADD AL,2
```

一般来说，汇编语言指令与机器语言指令之间是一一对应的。汇编语言通常都为特定的计算机系统而设计。虽然汇编语言比机器语言容易理解，但即使实现简单的功能，其编写的代码仍然很长，也没有解除编程语言对计算机硬件的依赖关系。

3. 高级语言

为了提高编程的效率，在汇编语言的基础上，人们开发了高级语言(High-level Language)。高级语言更加接近自然语言，所以它的代码简短、易学易用，用一条语句就能够完成大量的任务。高级语言使程序开发者能够编写更像英语的指令，可以包含常用的数学符号，例如，用 Java 语言设计 1 + 2 的加法运算，只需 x = 1 + 2 即可。

常见的面向结构的高级语言有：FORTRAN、ALGOL、COBOL、Pascal、Basic、C 等。

常见的面向对象的高级语言有：C++ 、Java、Visual Basic、Delphi 等。

高级语言同样不能被计算机直接执行，而是需要通过语言翻译程序将其转换为计算机可以执行的代码。按照程序翻译方式的不同，可分为解释型翻译程序和编译型翻译程序。

解释型翻译程序(解释器)在工作时读入一句源程序，就翻译一句，执行一句，这样反复操作直到最终完成。编译型翻译程序(编译器)则将高级语言写的源程序转变为计算机可以识别的机器语言，然后交给计算机执行。编译方式速度较快，而解释方式交互性更强。

1.1.2 算法的说明

程序是用来解决特定问题的，而算法是对解决问题步骤的描述。算法本身也可以采取不同的方式进行描述，常用的有：自然语言描述、伪代码描述和程序流程图描述。

1. 自然语言描述

自然语言描述就是通过人类常用的语言对解决问题的过程进行描述。例如，要进行两个数据的加法运算，利用自然语言描述如下：

- 第 1 步：输入两个数据，分别存储在变量 x 和 y 中。
- 第 2 步：完成 x + y 的运算，将结果保存在变量 z 中。
- 第 3 步：输出变量 z 的值。

2. 伪代码描述

伪代码(Pseudo Code)是人为设计的非正式语言，类似于日常的英语，方便而且容易掌握，

但不是实际的计算机编程语言，只是帮助程序员开发算法而用。例如下面的一段伪代码：

x←a
y←b
z←0
z←x + y;

3. 程序流程图描述

程序流程图描述就是使用图形对算法进行描述，这些图形有固定的含义。用图形表示算法，直观、形象便于理解。目前常用的有两种流程图：标准流程图和 N-S 流程图。

标准流程图主要涉及的图形及其含义如表 1-1 所示。例如，用标准流程图描述两个数据的加法运算，如图 1-1 所示。

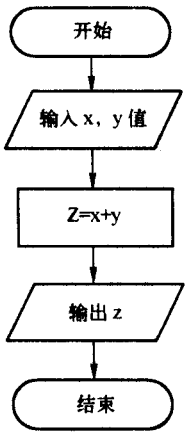
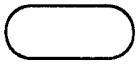
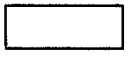
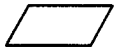
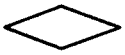


图 1-1　用标准流程图描述

表 1-1　标准流程图图形含义

图　形	含　义
	起止框(圆弧形)——表示程序的开始或终止
	处理框(矩形)——表示一般的处理功能
	输入输出框(平行四边形)——数据的输入和输出
	判断框(菱形)——表示对一个给定的条件进行判断，根据条件是否成立决定如何执行其后的操作
	流程线——表示程序流程的路径和方向
	连接点(圆圈)——用于将画在不同地方的流程线连接起来，以避免流程线的交叉或过长

1.2　程序设计方法

20 世纪 60 年代，由于程序代码不存在基本的组成结构，无条件的跳转命令使程序流程异常混乱，导致大型软件开发进度被严重拖延，开发成本大大超出预算，且最终产品不可靠，维护起来异常困难。种种因素使人们意识到软件开发是一项非常复杂的工程，需要采取规范化的程序设计方法或者说是程序设计规范(Paradigm)来进行代码的编写。

目前程序设计的方法主要有两类：采用结构化的程序设计思想和面向对象的程序设计思想。

1.2.1　结构化的程序设计

按照结构化方法编写程序，比用非结构化编程方法编写的程序结构更清楚，更容易调试和

测试及修改。它将程序划分为3种基本结构,即顺序结构、选择结构和循环结构。每种结构支持的程序流程不同。

1. 顺序结构

当需要按照语句的先后次序从上到下依次执行每条语句时,采用顺序结构。

顺序结构是最简单的一种结构,在流程图中表示为任务框一个个地串行连接。在计算机执行程序时表现为从头至尾严格按照次序逐条语句地执行,并且每一条语句均被执行一遍。顺序程序流程图如图1-2所示。图中的A、B和C分别代表的可以是一条语句,也可以是一段程序。

2. 选择结构

当需要根据某个条件执行程序的不同部分时,采用选择结构。

选择结构程序流程图都包含一个判断框,这个判断框具有一个入口和两个出口,从而形成程序的两个分支,如图1-3所示。在程序运行时究竟是执行B还是C,要由判断框内的条件判为"是"或"否"来决定。语句A执行完之后通常产生一个条件码CC,当条件CC判断为"是"(Yes或Y)时进入B分支;当条件CC判断为"否"(No或N)时进入C分支。由此可见,只有一个分支中的程序被执行了一遍,而另一分支中的程序没有得到执行。在实际编程时,不仅会用到上述的二分支程序结构,还会用到分支数多于两个的多分支程序结构。不过,多分支结构可以看作由二分支结构嵌套而成,即分支中又包含分支。

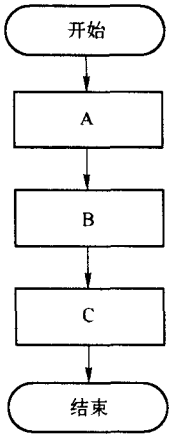


图1-2 顺序结构流程图

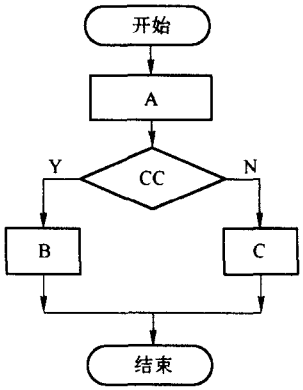


图1-3 选择结构流程图

3. 循环结构

当需要根据某个条件是否成立决定是否反复执行某段程序时,采用循环结构。

一个循环结构包含循环初态设置、循环体和循环控制3部分。循环初态设置就是在循环开始时,指定或定义一个循环变量CC(可以是循环次数计数器、地址指针等),并且给它设置一个初始值。循环体就是需要重复执行的程序段。循环控制就是根据循环结束条件,判断是否结束循环。在循环程序中必须给出循环结束条件,否则就成为"死循环"。

如图1-4所示为循环结构流程图。该图有两种画法。图1-4a中的循环体至少执行一次,这是因为先执行循环体,后判断循环结束条件。而在图1-4b中,先判断结束条件,再执行循环体,如果一开始就满足结束条件,则循环体将一次也不被执行。在实际编程时,不仅会用到上

述的单一循环结构，还会用到多重循环结构。多重循环结构可以看作是由单一循环结构嵌套而成，即循环体中又包含有一个或多个循环。

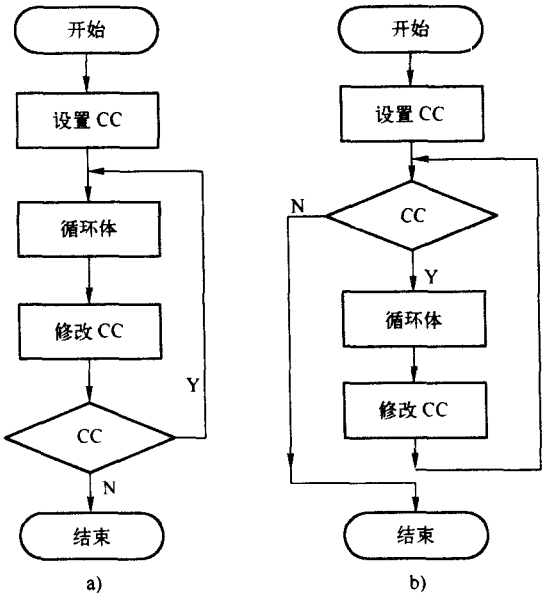


图 1-4　循环结构流程图

1.2.2　面向对象的概念

面向结构化的编程方法，其编程的主要思路专注于算法的实现。明显的特点就是数据与程序的分开，数据是静止的东西，不会自行变化，必须通过过程操作来改变数据，因此函数调用在面向过程编程中被大量使用。

然而随着计算机软件的发展，程序越来越大，后期维护的工作就越发艰难，经常出现因大型软件的程序结构不清楚而无法对软件进行修改的情况。结构化方法已经无法承担快速高效开发复杂软件系统的重任。于 20 世纪 80 年代逐渐成熟的面向对象的程序设计方法，使软件开发者对软件的分析、设计和编程等方面有了一种全新的认识。由于“对象”概念的引入，更大程度上让计算机语言结构与人类的思维方式保持一致，简单而清晰。目前，面向对象的程序设计方法已成为软件开发者的第一选择。

面向对象的基本思想是使用对象、类、继承、封装、消息等基本概念来进行程序设计。从现实世界中客观存在的事物（即对象）出发来构造软件系统，并且在系统构造中尽可能运用人类的自然思维方式。开发一个软件是为了解决某些问题，这些问题所涉及的业务范围称作该软件的问题域。面向对象的应用领域不仅仅是软件，还有计算机体系结构和人工智能等方面。

1. 类的基本概念

把众多的事物归纳、划分成一些类是人类在认识客观世界时经常采用的思维方法。分类的原则是抽象。类（Class）是具有相同属性和服务的一组对象的集合，它为属于该类的所有对象提供了统一的抽象描述。类可以有多个抽象层次，较高抽象层次的类称为“一般类（父类或

基类)",较低抽象层次的类称为"特殊类(或称子类或派生类)"。在 Java 语言中,通常称一般类为父类(Super Class),特殊类为子类(Sub Class)。例如,"汽车"可以抽象成一个父类,在其中又可以划分出"轿车"、"卡车"、"客车"等子类。

在面向对象的编程语言中,类是一个独立的程序单位,它有一个类名并包括属性说明和服务说明两个主要部分。类与对象的关系就如模具和铸件的关系,类的实例化结果就是对象,而对一类对象的抽象就是类。

2. 对象的基本概念

对象(Object)是系统中用来描述客观事物的一个实体,是构成系统的一个基本单位。一个对象由一组属性和对这组属性进行操作的一组服务组成。从更抽象的角度来说,对象是问题域或实现域中某些事物的一个抽象,它反映该事物在系统中需要保存的信息和发挥的作用。是一组属性和有权对这些属性进行操作的一组服务的封装体。客观世界便是由对象和对象之间的联系组成的。

对象的三要素是:对象标识、属性和服务。

1) 对象标识(Object Identifier)是对象的名字,供系统内部惟一地表示对象。定义或使用对象时均应指定其标识。

2) 对象属性(Object Attribute)也称为状态或数据,用来描述对象的静态特征。

3) 对象服务(Object Service)也称为操作、行为或方法,用来描述对象的动态特征。

3. 消息

消息(Message)是向对象发出的服务请求,它应该包含下述信息:提供服务的对象标识、服务标识、输入信息和回答信息。对象与传统的数据有本质的区别,它不是被动地等待外界对其施加的操作。相反,进行处理的主体必须发送消息请求对象执行它的某个操作或处理它的某个私有数据,而不能从外界直接对对象的私有数据进行操作。

1.2.3 面向对象的特征

1. 封装性

面向对象的第一个原则是把数据和对该数据的操作都封装在一个类中。就是把对象的属性和服务结合成一个独立的单位,并尽可能隐蔽对象的内部细节,包含两个含义:

- 把对象的全部属性和全部服务结合在一起,形成一个不可分割的整体。
- 信息隐蔽,即尽可能隐蔽对象的内部细节,对外形成一个边界(或者说形成一道屏障),只保留有限的对外接口使之与外部发生联系。

封装的原则在软件上的反映是:要求使对象以外的部分不能随意存取对象的内部属性,从而有效地避免了外部错误对它的"交叉感染",使软件错误能够局部化,大大减少了查错和排错的难度。

2. 继承性

子类的对象拥有其父类的全部属性与服务,称作特殊类对一般类的继承。例如,轮船、客轮;人、大人。一个类可以是多个父类的子类,它从多个父类中继承了属性与服务,这称为多继承。例如,客轮是轮船和客运工具的子类。在面向对象的设计方法中,继承是提高软件开发效率的重要手段之一。

3. 多态性

对象的多态性是指在父类中定义的属性或服务被子类继承之后，可以具有不同的数据类型或表现出不同的行为。这使得同一个属性或服务在父类及其各个子类中具有不同的语义。例如："几何图形"的"绘图"方法，"椭圆"和"多边形"都是"几何图"的子类，但是其"绘图"方法功能不同。多态性不仅提高了软件开发的灵活性，进一步减少了信息冗余，而且显著提高了软件的可重用性和可扩充性。

1.3 Java 语言概述

1.3.1 Java 语言的产生

Java 语言来自于 Sun 公司的一个由 James Gosling 负责、名字叫 Green 的项目小组。这个小组最初的目标是能够在诸如电视机顶盒、烤面包机、PDA 这样的数字控制的电子消费产品上开发应用程序。然而消费电子产品种类繁多，即使是同一类消费电子产品所采用的处理芯片和操作系统也不相同，也存在着跨平台的问题。当时最流行的编程语言是 C 和 C++ 语言，开发人员就考虑是否可以采用 C++ 语言来编写消费电子产品的应用程序，但是研究结果表明，对于消费电子产品而言，C++ 语言过于复杂和庞大，并不适用，安全性也并不令人满意。于是，该小组就力图设计一种独立于硬件平台的计算机语言来解决这个问题，最终设计出了一种以 C++ 为基础，融合了 C 和 C++ 等传统语言优点的一种面向对象的程序设计语言，并起名为 Oak(Java 语言的前身，据传是以 Gosling 窗外的一颗橡树为名)。但是 Oak 语言在商业应用上并未获得成功，直到 1995 年，互联网在世界上蓬勃发展，Sun 公司发现 Oak 语言所具有的跨平台、面向对象、安全性高等特点非常符合互联网的需要，于是决定改进 Oak 语言的设计，将其应用于 WWW 开发中，并正式注册命名为 Java。Sun 公司用 Java 开发了一个 Web 浏览器(Hot Java)，通过这个完全用 Java 语言设计的浏览器在互联网上展示了 Java 的风采并决定让程序开发者免费使用 Java，这才真正地将 Java 推向了全世界。

Java 名字的来历也有一段趣闻，据传在申请注册商标时，由于 Oak(橡树)这个商标已经有人使用了，所以 Sun 公司必须重新为这个语言起一个名字。有一天，项目组几位成员在咖啡馆中边喝咖啡边讨论新名称的问题，其中一人突然发现他们喝的是一种 Java(爪哇)咖啡，于是灵机一动就提议叫 Java，马上得到了其他人的赞同，于是 Java 就成为了这个语言的新名字。这也就是为什么我们经常会在与 Java 相关的各种文档中看到一杯冒着热气的咖啡的原因了。

1.3.2 Java 语言的发展

1995 年，当时以 Web 为主要形式的互联网正在迅猛发展，Java 语言的出现，以其安全、跨平台、面向对象、简单、适用于网络等显著特点迅速引起所有程序员和软件公司的极大关注。程序员们纷纷尝试用 Java 语言编写网络应用程序，并利用网络把程序发布到世界各地。包括 IBM、Oracle、微软、Netscape、Apple、SGI 等 IT 业界大公司纷纷与 Sun 公司签订合同，要求被授权使用 Java 平台技术。微软公司总裁比尔·盖茨先生也曾经过研究后认为"Java 语言是长时间以来最卓越的程序设计语言之一"。

到目前为止，Java 语言已经成为最流行的网络编程语言，全球大约有 400 多万软件开发