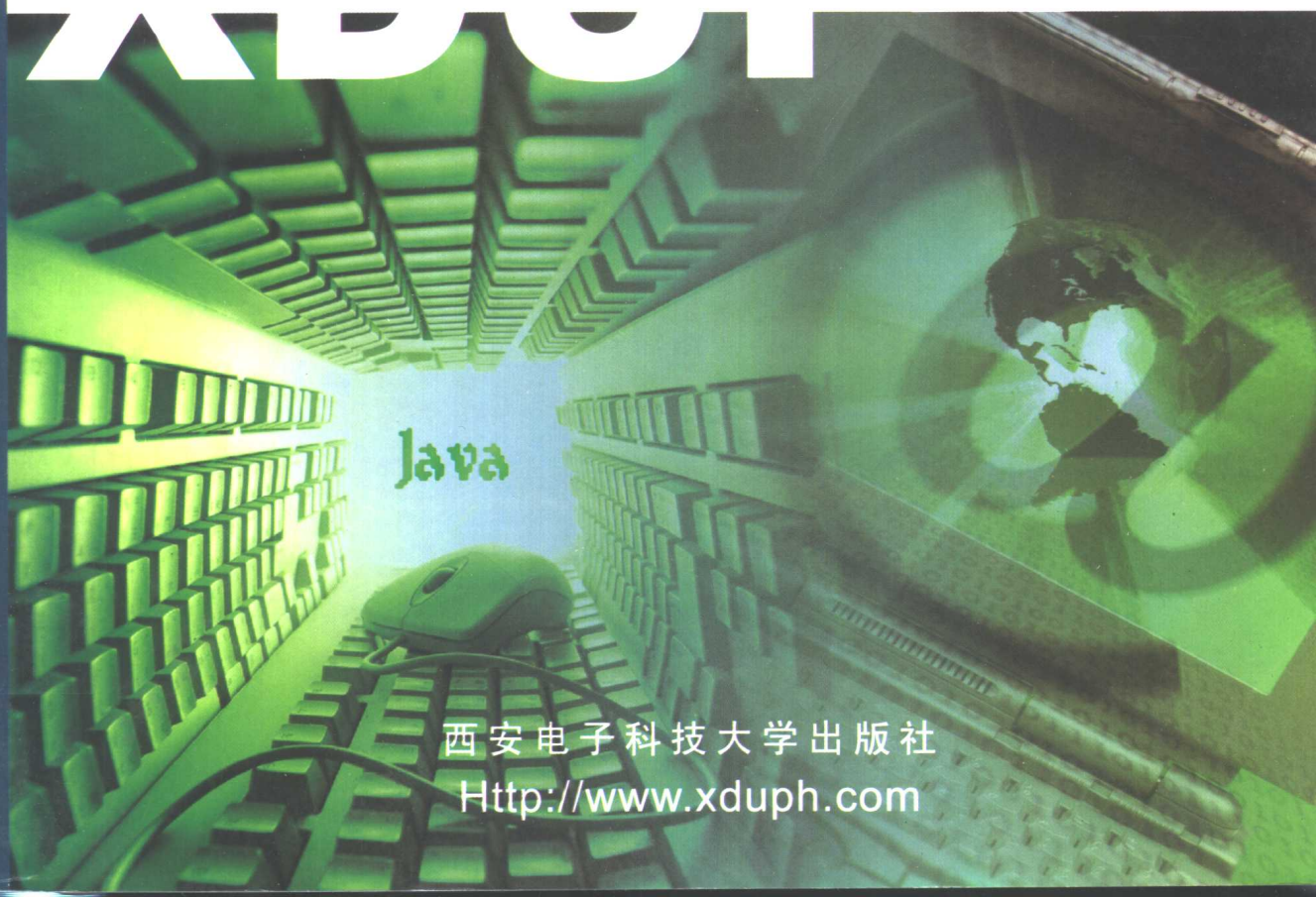
面向21世纪

高等学校计算机类专业系列教材

Java语言程序设计教程

A Course in Java Language Program Design

张 席 戴 劲 编著



西安电子科技大学出版社
Http://www.xduph.com

面向21世纪高等学校计算机类专业系列教材

Java 语言程序设计教程

A Course in Java Language Program Design

张席　戴劲　编著

西安电子科技大学出版社

2003

内 容 简 介

Sun 公司这样形容自己的 Java 语言：它是一种简单、面向对象、分布式、解释型、稳定、安全、结构中立、易移植、高性能、多线程的动态语言。这段长长的定语准确地描述了 Java 语言的基本特征，也道出了 Java 为何流行的秘密。当前，在网络的程序设计开发过程中，Java 已成为网络上的世界语，为 Internet 和 WWW 开辟了一个崭新的时代。

本书对 Java 语言的内容、功能、特性和实际运用作了深入浅出的、系统的、全面的介绍，结构严谨、布局合理、重点突出、实例丰富，能够使读者很快地掌握 Java 语言程序设计的方法和技巧，同时对面向对象程序的设计也有较深入的了解。

本书既可以作为高等院校计算机、通信等相关专业的本科生和研究生学习 Java 语言的教材和参考书，也可作为其他工程技术人员，特别是软件开发者的自学用书。

★ 本书配有电子教案，有需要的老师可免费索取。

图书在版编目（CIP）数据

Java 语言程序设计教程=A Course in Java Language Program Design / 张席等编著.

—西安：西安电子科技大学出版社，2003.6

(面向 21 世纪高等学校计算机类专业系列教材)

ISBN 7-5606-1240-7

Ⅰ.J… Ⅱ.张… Ⅲ. JAVA 语言—程序设计—高等学校—教材 Ⅳ.P312

中国版本图书馆 CIP 数据核字（2003）第 034999 号

责任编辑 马晓娟

出版发行 西安电子科技大学出版社（西安市太白南路 2 号）

电 话 (029)8242885 8201467 邮 编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 16.5

字 数 386 千字

印 数 1～4 000 册

定 价 18.00 元

ISBN 7 – 5606 – 1240 – 7 / TP • 0651（课）

XDUP 1511001-1

*** * * 如有印装问题可调换 * * ***

前　言

我们在做任何事情之前，必须对它进行深入的理解，编程也一样，只有通晓了它的本质，才能在应用中有的放矢，对程序设计中各种情况进行分析，找出较好的解决方案。

正是因为 Java 的卓越和奇妙，才使得其开发人员的数量和薪资节节上扬：在北美地区，使用 Java 的软件开发人员比例在 2002 年上升到 61%；在美国，Java 程序开发人员的平均年薪最高，基本年薪为7万美元，拥有Java证书的程序员平均工资高于没有证书人员的37%；在印度和韩国等地，政府和企业出资鼓励人们去学习 Java；据权威部门统计，我国对 Java 技术人员的需求量将高达 20 万人。

Java 是简单、高效的语言。举个例子来说，在美国食品和药物管理局，他们用 C++和 Java 来完成一个同样的系统，发现 Java 写的程序要比 C++节省 60%的代码；也有一个专业的认证杂志以 C++和 Java 来做一个相同功能的项目，发现用 Java 来写要节省 66%的时间。正因为如此，Java 已经变成世界上最流行的编程语言之一。

本书的编写原则是由浅入深，使得一般的普通非计算机专业人员也能学懂 Java，同时，通过大量的实例向读者详细介绍 Java 语言及其编程设计方法。本书采用的编程环境为 JBuilder 3.0，相信读者在学习了本书之后，能够在一定程度上掌握面向对象的思维方式，并且有能力编写真正有实际意义的应用程序。

本书第 1 章介绍了 Java 语言的一些背景知识，同时与 C/C++作了一定的比较；第 2 章介绍了 Java 编程的基本知识与概念；第 3、4 章介绍数组与字符串的知识；第 5 章介绍 Java 例外处理的特点；第 6 章介绍类、接口与包的概念；第 7 章介绍了输入/输出流的基本内容；第 8、9 章介绍图形用户界面和 Java 多媒体处理的知识；第 10 章为线程处理；第 11、12 章分别介绍了 Java 中网络通信及小应用程序的知识。

本书第 1~7 章由张席老师编写，第 8~12 章由戴劲老师编写。在本书的编写过程中，深圳大学信息工程学院的张基宏教授及王小民副教授给予了大力的支持和帮助，在此表示最诚挚的谢意。

鉴于编者的水平有限，错误之处在所难免，敬请广大读者批评指正。

作　者

2003 年 4 月

目　录

第1章 预备知识

随着信息技术的快速发展与进步，电脑已渐渐成为当前人类社会中最重要的信息获取工具之一。同时由于国际互联网(Internet)与其上相关应用系统的快速普及，可以说我们当前正处于一个“信息爆炸”时代。不同国家、行业被 Internet 连接在一起，相互通信，共享全世界的计算机资源和信息。如何保证不同格式的数据或信息安全、高速、自由的交流、传输就成为一个需迫切解决的问题。TCP/IP 网络传输协议为我们提供了统一的传输协议，但该协议只是允许程序之间以无格式的二进制数据流来进行信息的传送，对语义的解释及维护还需程序双方共同进行，因而难以实现程序代码交换。1996 年 Sun 公司 Java 语言的正式发表可以说是 Internet 的一次技术革命，它实现了程序的运行与平台无关，在网络上不仅可以进行无格式的数据信息交换，还可以进行程序交换。自从 Java 推出以后，马上在全世界范围得以普及，现在，越来越多的程序设计人员开始用 Java 进行程序开发。在国外，80%以上的企业都在利用 Java 技术，各个数据中心大大小小的机器上更是离不开 Java。最近，我们常常能从广播上听到，全球各手机公司都在利用 Java 推出其第三代手机，Sony 公司所做的游戏机(以后能上网)里也嵌入了 Java。不远的将来，我们在下班的路上驾驶着汽车，甚至远在千里之外，将仍可以调节家中空调的温度，可以控制家里的电饭煲给亲人做饭……这一切，听起来是那么不可思议，但有了互联网，有了 Java 语言，有了其他先进的技术的结合，它们就会渐渐成为现实。难怪比尔·盖茨发出感叹:“Java 是最卓越的程序设计语言!”

为了使读者对 Java 有一个更好的理解，我们首先介绍 C、C++的一些基本特点，比较二者与 Java 的一些差异，为后续的学习作准备。

1.1 目前流行的编程语言简介

我们知道，当前的程序开发语言多种多样，比如说有微软公司(Microsoft)的 VC、VB 及 Borland 公司的 C++ Builder、Delphi，Sun 公司的 Java 等等。其中，最典型的当属 C、C++及 Java。C 语言曾是国际上广泛流行的计算机高级语言。

1.1.1 C/C++的一些概念

C 语言是贝尔实验室的 Dennis Ritchie 在 B 语言的基础上开发出来的，于 1972 年在一台 DEC PDP-11 计算机上实现了最初的 C 语言。C 是作为 UNIX 操作系统的开发语言而开始广为人们所认识的。当今许多新的、重要的操作系统都是用 C 或 C++编写的。在过去 20 年内，C 语言已经能够用在绝大多数计算机上了。由于 C 语言的一些显著特点，如与硬件

无关、设计严谨，使得用 C 语言编写的程序移植到大多数计算机上成为可能。到 20 世纪 70 年代末，C 已经演化为现在所说的“传统的 C 语言”。Kernighan 及 Ritchie 在 1978 年出版的《The C Programming Language》一书中全面地介绍了传统的 C 语言，当前，该书已经成为最权威的计算机学术著作之一。C 语言的不断发展导致出现了许多 C 语言版本，虽然大多版本是类似的，但通常都不兼容，这对希望开发出的代码能够在多种平台上运行的程序开发者来说是一个严重的问题。为了明确地定义与机器无关的 C 语言，1989 年美国国家标准协会制定了 C 语言的标准(ANSI C)。Kernighan 和 Ritchie 编著的《The C Programming Language》(第二版)介绍了 ANSI C 的全部内容。C 语言具有其独特的优点，包括：

(1) 语言简洁、紧凑，使用方便、灵活。C 语言只有 32 个关键字，程序书写形式自由。

(2) 丰富的运算符和数据类型。

(3) C 语言可以直接访问内存地址，能进行位操作，能够胜任开发操作系统的工作。

(4) 生成的目标代码质量高，程序运行效率高。

(5) 可移植性好。

虽说 C 语言具有以上的一些优点，但在实际的程序开发过程中也暴露出了一些不足：

(1) C 类型检查机制比较薄弱，使得程序设计中的一些错误不能在编译时被发现。

(2) C 本身几乎没有支持代码重用的语言结构，因此一个程序员精心设计的程序，很难为其他程序共用。

(3) 当程序的规模达到一定程度的时候，程序员很难控制程序的复杂性。

为了满足管理程序的复杂性要求，贝尔实验室的 Bjarne Stroustrup 开始对 C 进行改进和扩充。最初的成果称为“带类的 C”，1983 年正式取名为 C++。在经历了 3 次修订后，于 1994 年制定了 ANSI C++标准的草案。以后又经过不断完善，成为目前的 C++。C++包含了整个 C，C 是建立 C++的基础。C++包括 C 的全部特征、属性和优点，同时添加了面向对象编程(OOP)的完全支持。

1.1.2 从 C 到 C++

经过上面的描述，大家对 C 语言的优、缺点有了一个具体的认识。C 语言从本质上说是属于过程性语言，其程序设计方法也是遵从于结构化程序设计。结构化程序设计的主要思想是功能分解并逐步求精，比如，当一些任务非常复杂以至无法描述时，可以将它拆分为一系列较小的功能部件，直到这些自完备的子任务小到易于理解的程度。但是，采用结构化程序设计方法的程序员很快发现，每一种相对于老问题的新方法都要带来额外的开销，通常称这为重复投入。基于可重用性的思想是指建立一些具有已知特性的部件，在需要时可以插入到程序之中。这可以说是一种模仿硬件组合方式的做法，当工程师需要一个新的晶体管时，当然不用自己去发明，只要去电子市场买一个就行了，对于软件工程师来说，在面向对象程序设计出现之前，一直缺乏具备这种能力的工具。

C++语言包括过程性语言部分和类部分，类部分是 C 中所没有的，它是面向对象程序设计的主体。可想而知，结构化程序设计随着 C++的出现也过渡到了面向对象程序设计，

所以从C到C++也是自然而然的事了。不过，C语言程序设计的经验非常有益，因为C程序设计开发锻炼了程序员进行抽象程序设计的能力，这正是C++更为抽象的概念和技术的基础。而且，C++是C语言的扩展，它分享了C的许多技术风格。程序员使用C的经验越丰富，编写C++程序也就越容易，所以，对C的学习，能够促进对C++的学习，换句话说，C++的发展并没有完全抛弃C的一些特点。

1.1.3 面向对象初步知识

面向对象程序设计的本质是把数据和处理数据的过程当成一个整体——对象。面向对象程序设计的实现需要封装和数据隐藏技术，需要继承和多态性技术。下面我们对这些概念作一描述。

1．封装和数据隐藏

当一个工程师要安装一台电脑时，他将各个设备组装起来，如果需要一个声卡时，不需要用原始的集成电路芯片和材料自己去做，而是去电脑公司买一个符合其需要的声卡就行了。工程师所关心的是声卡的功能，并不关心声卡内部的工作原理，因为声卡是由电子厂商所提供，是自成一体的。这就是所谓的封装性——无需知道封装单元内部是如何工作的。声卡的所有属性都封装在声卡中，不会扩展到声卡之外，用户无需知道其工作原理就可以有效地使用。

面向对象的程序设计通过建立用户定义类型或类支持封装性和数据隐藏。完好定义的类一旦建立，就可看成是完全封装的实体，可以作为一个整体单元使用。类的实际内部工作应当隐藏起来，使用完好定义的类的用户不需要知道类是如何工作的，只要知道如何使用它就行了。

2．继承和重用

要制造新的电视机，可以有两种选择：一种是从草图开始，另一种是对现有的型号加以改进。因为现有的型号可能已经令人满意，如果多加一个功能，就会更加完美，因而电视设计人员决不会推倒重来，一切从头开始，而是在原有的型号基础上增加一组电路或多加某些芯片来增强其已有的功能，完成之后，新型号的电视机就诞生了。这就是继承和重用的生活实例。

面向对象的程序设计也采用继承和重用的思想：程序可以在扩展现有类型的基础上声明新类型；新的子类是从现有类型派生出来的，称为派生类，但已在原有类的基础上增加了新的功能。

3．多态性

类是通过继承的方法构造的，采用多态性可为每个类指定表现行为。举例来说，学生类应该有一个计算成绩的操作：对于中学生，计算成绩的操作可表示为对语文、数学、英语等课程成绩的计算；对于大学生，应可以继承中学生的基类，计算成绩的操作表现为对高等数学、计算机、大学物理等课程成绩的计算。

实际的应用中，继承和多态经常会用到，这是因为：有了继承性，使得多个对象可以共享许多相似的特征；有了多态性，一个对象可以有其独特的多种表现方式。

1.2 从 C/C++到 Java

前面我们已讨论了 C 及 C++语言的一些特点，相信大家对二者有了一定的认识。

C++语言虽在计算机行业广为使用，但它的复杂与繁琐也令一般用户为之伤透脑筋。所以，为使程序开发或设计人员从 C++“返朴归真”，就需开发一种全新的程序设计语言。

Java 是由 Sun 公司开发的新一代程序设计语言，正在逐步成为 Internet 应用的主力开发语言，成为 Internet 上的世界语。为了开拓消费类电子产品市场，Sun 公司于 1991 年成立了一个项目开发小组，其小组负责人是 James Gosling。在研究开发过程中，Gosling 感到消费类电子产品和工作站产品的开发存在较大的差异：消费类电子产品要求可靠性高、费用低、标准化、使用简单，而工作站用户需要较强的计算能力，不考虑价格以及操作的复杂性。消费类电子产品并不关心 CPU 的型号，只是要求整个系统与平台无关。Gosling 首先尝试从改写 C++编译器着手，但在改写过程中，感到 C++无法满足要求，这促使他打算开发一个新的语言——后来命名为 Java。该系统运行于一个巨大的、分布的、异质的网络环境中，以完成电子设备之间的通信与协同工作。为达到此目的，设计过程中采用了虚机器码技术(Virtual Machine Code)。编好的程序经过编译后产生的就是虚机器码，其不能单独运行。当操作系统安装了对应的解释器后，可通过该解释器解释执行虚机器码，至此，与平台无关的 Java 语言就产生了。

Java 的开发主要是以 C++作为蓝图，因此它的大部分语法与 C++相似，但为了达到真正的面向对象，C++中很多过程式程序设计语言必须舍弃掉。Java 开发的目标是期望其能成为一个简单，但面向对象的程序语言。为使读者有一个初步的印象，我们用 C，C++，Java 各举一个例子来输出字符串“Welcome to C/C++/Java World!”，读者可自行作一比较。

- C 语言实现：

```
#include <stdio.h>
    main()
    {
    printf("Welcome to C World! ");
    }
```

- C++语言实现：

```
#include <iostream.h>
void main()
{
cout<<"Welcome to C++ World! \n";
}
```

- Java 语言实现：

```
public class A {
public static void main(String args[])
```

```
{
    System.out.println("Welcome to Java World! ");
}
```

习 题

1. 试分析 Java 语言与 C/C++之间的优缺点，并给出你对 Java 语言的理解。
2. 举例说明 Java 语言的应用前景。
3. 自己安装、配置 JBuilder 运行环境，并编辑一个简单的例子来运行。

第 2 章 Java 语言概述

1991 年，Sun 公司的 James Gosling 等人，为在电视机、烤面包箱等家用消费类电子产品上进行交互式操作而开发了一个名叫 Oak(一种橡树的名字)的语言。由于商业上的种种原因，这种语言始终没有投放到市场中，而且连 Oak 这个名字也成了问题，因为已经有上百家公司在使用这个名字，所以 Sun 公司根本无法将之注册为商标。最终，Sun 公司决定，将这种语言改名为 Java，并且在互联网上发布，免费提供下载。当时，由于 Oak 的失败，有一些传谣者鼓吹 Java 这些字母代表“只是又一个无意义的缩写词”(Just Another Valueless Acronym)。Sun 公司否认了这一说法，而且说 Java 是语言开发者在喝一种原产于印度尼西亚爪哇群岛的咖啡时，出于一时的灵感而碰撞出的火花。

几个月后，出乎所有人的意料，Java 成为赛博空间最热门的话题。Java 被越来越多的用户使用，受到越来越多的重视。上百个 Java 小应用程序在互联网上的多媒体应用中流行起来。一些著名的公司，如微软、IBM、苹果电脑、数字设备公司，纷纷购买了 Java 语言的使用权，随之大量出现了用 Java 编写的软件产品。Java 受到业界的重视与好评。微软总裁比尔·盖茨在悄悄地观察了一段时间后，也感慨地说：“Java 是长时间以来最卓越的程序设计语言”。

Java 为什么会在短时间内受到如此多的程序员欢迎？为什么在计算机行业竞争激烈的今天，一个计算机硬件公司开发出来的语言，会一下子得到几乎世界上各大计算机软、硬件公司的支持呢？

Java 最重要的特征在于它的操作平台无关性，这是以往任何一种语言都不具备的特征。也就是说，用 Java 语言编写的程序可以在任何一台计算机上运行，而不管该计算机使用何种操作系统，要知道，这可是广大程序员的一个梦想。

其次，Java 是一种面向对象的语言。长期以来，人们一直在设法争取问题空间同求解空间在结构上的一致，以使我们在分析、设计和实现程序时，同我们认识客观世界的过程尽可能一致，因此产生了面向对象的程序方法。Java 就是这样一种面向对象的语言，不仅如此，它还代表了面向对象程序设计方法在目前的最高应用水平。对一个程序员来说，这意味着可以将注意力集中在应用程序的数据和处理数据的方法上，而无需过多地考虑处理过程。

此外，Java 还是一种非常简单的语言。Java 的前身 Oak，是为家用电器产品设计的，只有简单易用才能推广开来，因此，这种语言被设计得简单而高效。程序员只需理解一些基本的概念，就可以用它编写适合各种情况的应用程序了。

最后，安全性也成为 Java 受青睐的一个方面。因为在网络环境中，安全是需要认真考虑的一个问题。没有安全的保障，用户绝对不会从 Internet 上随意一个站点上下载一个 Java 小应用程序，并在自己的计算机上运行。Java 语言提供了若干种安全机制来抵御病毒产生

或侵入文件系统。这一点也让用户们非常放心。

Java 的出现确实给计算机行业吹来了一股清风；它带来了很多新鲜而有趣的思想和观念；它甚至改变了人们使用计算机的方式。就连环球信息网 WWW 的创始人也说："计算机行业发展的下一个浪潮就是 Java，并且很快就会发生。"

如今，在美国硅谷，不懂得 Java 的人是无法找到工作的。在我国，许多计算机权威人士都断言，谁先掌握了 Java，谁就号准了世界的脉搏，就能在信息时代找到自己的立足之地。

研究机构 Evans Data 公司最近公布的调查结果显示，Java 将在 2003 年超过 C/C++成为全球软件开发人员的首选语言。 参加本次调查的编程人员来自 60 多个国家，他们中 60%的人在 2003 年将更多地使用 Java，所用时间超过使用 C/C++或 VB。公布本次调查结果的 Evans Data 公司副总裁 Garvin 说，自该公司 1998 年首次开始跟踪 Java 的使用情况起，Java 用户总数不断增加。她说："Java 在北美之外的发展更强劲。至少一半被调查的来自各国的开发人员目前使用 Java。事实上，他们使用 Java 的平均时间由 1999 年的 9.1%上升到目前的 17.7%。"尽管最初存在由于 Sun 公司的专有立场和该公司与操作系统社区的矛盾而造成的有关 Java 的争议，但是 Java 不断作为技术"热点"蚕食 C/C++市场。其主要原因是 Java 具有许多 C++所没有的优点，如简单性、更好的内存管理和跨平台功能。相反，在过去三年里，C++在各国开发人员中的占有率减少。Evans Data 说，目前被调查的开发人员有 25%的时间使用 C++，而这一数字将在 2002 年进一步减少。但是这家市场研究机构仍认为 C/C++由于其已经生成的巨大代码量，不可能很快消逝。至于 C#，一种与 C++相似但与 Microsoft.Net 倡议密切相连，具备类似 Java 语言特性的新语言，这份调查显示，3/4 的开发人员表示还没有采用它的计划。Garvin 说，有可能采用 C#的开发人员可能是那些已经使用某种 Microsoft 编程语言的用户。

2.1 Java 语言的优势与特点

Sun 公司这样形容自己的 Java 语言：它是一种简单、面向对象、分布式、解释型、稳定、安全、结构中立、易移植、高性能、多线程的动态语言。这段长长的定语准确地描述了 Java 语言的基本特征，也道出了 Java 为何流行的秘密。

可以说，程序设计语言的优点，Java 几乎全都拥有。从实际的应用开发来看，也确实如此。但有一点需注意的是，由于 Java 是经过编译器生成字节码后再通过解释器对其解释执行，其运行速度会比较慢。下面就上述的特点逐一进行阐述。

1. 简单性

Java 是一种简单的语言。这点主要表现为简单易学，在形式上它和 C/C++极为相似，而且其运行系统小(Java 的基本解释器只有 40 KB，加上标准库和线程支持也不过 215 KB)。

我们知道，C++中的结构、联合和类的概念重合之处很多，而 Java 只保留了类的概念，减少了复杂性。运算符重载是 C++的一大特点，一度被认为简化了程序设计，但实际上用得非常少，作用也不大，反而加重了程序员的负担，Java 把它也给去掉了。多重继承一直是一个有争议的问题，赞成者认为它为面向对象语言带来了方便性和通用性，增加了语言

的表达能力，反对者声称它是一个不必要的特征，容易混淆，难以使用，浪费机器资源，实际价值有限，所以 Java 把它也取消了，而带之以接口“interface”。Gosling 同时认为，C++的预处理程序、标题文件、goto 语句和隐式类型转换，都增加了程序的不可读性，容易出问题，所以全部给取消了。

2．面向对象

在程序的开发设计过程中，程序员一直在设法争取问题空间同求解空间在结构上尽可能一致，以使我们在分析、设计和实现程序时，同我们认识客观世界的过程尽可能地一致，因此产生了面向对象程序设计方法。所谓面向对象的设计方法，是基于信息隐藏和抽象数据类型概念，把系统中所有资源，如数据、模块以及系统都看成是对象。每个对象封装数据和方法，而方法实施对数据的处理，并且通过继承机制实现代码复用。传统的面向对象语言各有不足，如 C++语言，由于继承了 C 的大量特性，如独立的函数概念，因而变得复杂而难用，而纯粹的面向对象语言如 Smalltalk、Eiffel，却是动态性有余，效率不高。Java 完全具备面向对象的四大特点：封装、继承、多态和动态。其封装性比 C++好，它没有全程变量，没有主函数 main。在 Java 中绝大部分成员是对象，只有简单数字类型、字符类型、布尔类型除外，以便保持高性能。Java 提供给用户一系列的类(class)，一个或多个 class 可以组成一个包(package)。Java 的 class 和 C++的一样有层次结构，子类可以继承父类的属性和方法。Java 类中方法均缺省为虚函数。

Java 的面向对象与许多语言都有相通之处，与 C++自不必说，因为 Java 几乎是 C++的一个子集，熟悉 C++编程的人很快会习惯 Java 编程。学过 Pascal 的读者对 Java 可能也有似曾相识的感觉，因为 Java 的 package 和 Pascal 中的 Unit 非常相似，都是表示某些程序块的输入。运算符重载在 C++中一向被认为是一种优雅的多态机制，但在实践中，人们发现，运算符重载会使程序变得难以理解，Java 的设计者最后取消了运算符重载，只在字符串连接运算中留了一些运算符重载的痕迹，例如用“+”实现字符串的连接。

Java 语言面向对象结构的动态性很高，C++中，如果修改了某一个类，整个程序都得重新编译。在 Java 中，可以在类库中自由地加入新的方法和实例变量，而不影响用户的程序执行。虽然 C++也可以实现这种动态性，但使用起来非常复杂，代价也很高。

3．分布式

Java 是一种面向对象的程序设计语言，它也支持网络上的应用程序，是一种分布式(distributed)程序设计语言。使用 Java 提供的类库，比如 java.net，可以方便地支持 TCP/IP 协议，完成各种层次上的网络连接。请看下面一个简单的 ftp 登录的例子：

```
...
FtpClient f_client;
f_client=new FtpClient("hostname");
f.login("anonymous","");
f.binary();
...
```

另外，Java 提供一个 Socket 类，这个类可以提供可靠的流式网络连接。这样，我们可以非常方便地创建分布式的 Client 和 Server 应用程序。传统的网络编程是一件复杂的事情，

但是通过Java提供的网络类库，可以轻易地构造出网络应用，如客户机/服务器应用，浏览器/服务器应用，大大简化了工作难度。正如Java的设计者Gosling所说，Java的设计就是为了“最大限度地利用网络”。

4．解释型

我们知道，Java的编译器产生的是字节代码，可把它理解为一种中间代码，而不是特定的机器码。该字节码必须运行在一个解释器上，所以说，Java是一种解释型语言。由于产生的是中间代码——字节代码，因而可达到与平台无关的目的，从而可高效地在不同平台之间传输。同时，该程序可在任何平台上运行，只要这个平台上安装了Java解释器和运行系统即可。

在解释环境中，程序开发过程中标准的“链接”过程没有了，Java的“链接”实际上是把一个新类加载到当前的环境中，这和传统的程序开发过程中的编译、链接、测试有较大的区别。

5．健壮性

分布式计算环境要求软件具有高度的健壮性。C++程序员都知道其在稳定性方面的最大问题在于指针的使用和缺乏自动的内存管理。这使得程序员可能编写出在语法和语义上均正确，但却可能对系统产生巨大破坏作用的软件。Java是一种比C++还强的强壮型语言。它要求显式的方法声明，保证编译器可以发现方法调用错误。Gosling认为指针的主要作用在于数组和结构的访问及使用。Java的数组可以解决前者的问题，但加强了对数组下标的检查。结构的访问和使用可通过类及类的方法访问变量解决，因而可以取消指针的概念。

Java语言稳定性的另一个方面是自动的内存管理。用过C/C++的读者知道，比如我们可以调用函数malloc()来进行内存的分配，但在使用完后，必须要对占用的资源进行释放，即再调用free()函数释放掉分配的内存，如果稍不小心，就可能造成系统故障或空间的浪费。Java中专门有一个后台垃圾自动收集程序——Garbage Collector。它以较低的线程优先级对存储器进行扫描，自动释放掉不再使用的存储碎片，从而使程序员不用再担心内存的使用，只是专注于程序的设计即可。

异常处理也是Java健壮性的一个方面。一般认为，异常处理是成熟语言的标志。在Java中，通过使用try/catch/finally语句，程序员可以把一组错误处理代码放在一个集中的地方统一处理，这可简化错误的处理及错误的恢复。

6．安全性

Java的安全性和健壮性是紧密联系的，由于其主要应用于网络程序的开发，因而如果没有较高的安全性作为保障的话，用户从网络上下载程序将是非常危险的。正如我们前面所描述的，Java取消了指针，杜绝了不怀好意的程序对内存恶意篡改。

Java语言的安全机制基于“不存在可信任的代码”的概念。其运行环境提供了以下四级安全保障机制：

① 字节码校验器
② 类装载器
③ 运行时内存布局
④ 文件访问机制

当 Java 字节码进入解释器时，首先必须经过字节码校验器的检查。校验器就象一个忠诚的卫士，不能通过检查的一律不准入内。即使是 Java 编译器生成的完全正确的字节码，校验器也必须再次对其检查，因为 Java 程序的编译和解释执行期间，字节码可能会无意或恶意地被篡改过。然后，Java 解释器将决定程序中类的内存布局，这就意味着不怀好意的程序无法预先知道一个类的内存布局结构，因而也就无法利用该信息来修改或破坏系统。随后，类装载器负责把来自网络的类加载到单独的内存区域，类和类之间相互不会干扰。最后，客户端用户还可以限制从网络上加载的类只能访问某些被允许的系统，如文件和硬盘。所以，当这几种机制加在一起，再加上其他的增强办法，如传输过程中使用加密解密算法，程序中做上特殊的标记等，使得 Java 成为了最安全的系统之一。

7. 可移植性

Java 语言的可移植性具有深远的意义，它不仅给软件开发者带来了“一次性开发”的方便，而且迎合了网络计算的思想。在 Java 以前，人们只是将 Internet 当作一个巨大的硬盘，里面有无数的静止信息。当 Java 出现后，Internet 则变成了一个巨大的操作系统，Java 就是这个系统的语言。用这个语言写的任何程序，如电子表格软件，存放在某台机器上，Internet 网上的任何用户只要得到许可，就可以下载并运行这个程序，从而节省了大量的硬盘空间，也省去了管理的麻烦。

Java 的设计者采用了多种机制来保证可移植性，最主要的有：

(1) Java 从本质上讲是解释型的，这意味着任何一台机器，只要配备了 Java 解释器，就可以运行 Java 程序。

(2) Java 的数据类型在任何机器上都是一致的，它不支持特定硬件环境的数据类型。

总之，Java 的可移植性决定了它将成为未来网络环境的“世界语”。

8. 高性能

由前所述，我们知道 Java 是一个解释型语言。按照系统结构的观点，解释型语言除不可能达到编译型语言的速度外，其他一些性能并不比编译型语言差。有人作过测试，Java 程序的平均运算速度是 C 语言的 1/20 倍。为了解决高性能问题，Java 的设计者正在努力开发出一种更具效率的编译器，这种编译器可以在运行时把 Java 的字节码翻译成特定 CPU 的机器码，从而使转化成机器码的字节码在性能上接近于 C 或 C++。

9. 多线程

Java 高性能的另一个方面是它的多线程能力，他可以同时运行多个线程，处理多个任务。多线程可称为“轻量级进程”，它有点像 Unix 下的进程概念，多线程即多个模块并行运行；通过消息协调操作；通过信号灯和锁保证关键模块的执行不被中断。线程之间可以共享内存和全局变量，比进程的开销要小得多，因此也可以将线程看作可以同时运行的函数。多个线程的并行执行，仿佛有多个 CPU 在运行。比如说，一个线程在执行复杂的计算，而另一个线程完成与用户的交互。用户不必停下来等待 Java 程序完成冗长的操作，所以，多线程能够增强用户的实时交互能力，提高程序的运行效率。

传统的语言程序中，多线程编程是相当繁琐的，因为同一时刻要发生许多事情，程序员必须考虑它们的执行顺序、同步管理、资源争用等情况。例如当使用 C 或 C++语言开发多线程应用程序，首要的困难是要保证多个例程可被若干并发线程运行。如果一个例程改