



Java 2

简明教程



学勤 主编
李诚 学勤 肖占彪 编著



清华大学出版社

Java 2 简明教程

学 勤 主编

李 诚 学 勤 肖占彪 编著

清华大学出版社

(京) 新登字 158 号

内 容 简 介

Java 语言是目前最为流行和通用的网络编程设计语言，在 Internet 上有着重要而广泛的应用。本书全面介绍了 Java 语言的特点和功能，主要包括：Java 编程基础知识、HTML 基础知识、Java 算术运算符、Java 编程语法、面向对象程序设计基本概念、系统类的使用和应用、图形用户界面设计、图形处理、多媒体处理、异常处理、多线程、网络编程等。

本书语言通俗易懂，实例丰富实用，对 Java 语言的编程技术进行了详细、生动的讲解，着重培养编程初学者解决实际问题的能力。

本书不但适合没有任何编程经验的初学者使用，也可作为高等学校、高等职业学校、成人高校 Java 程序设计课程的教材和教学参考书。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

书 名：Java 2 简明教程

作 者：李 诚 学 勤 肖占彪 编著

责任编辑：苗建强

出 版 者：清华大学出版社（北京清华大学学研大厦，邮编 100084）

<http://www.tup.tsinghua.edu.cn>

印 刷 者：北京通州区大中印刷厂

发 行 者：新华书店总店北京发行所

开 本：787×1092 1/16 印张：24.5 字数：580 千字

版 次：2002 年 7 月第 1 版 2002 年 7 月第 1 次印刷

书 号：ISBN 7-302-05523-8/TP·3250

印 数：0001~5000

定 价：37.00 元

前 言

Sun 公司研制的 Java 语言，目前正成为推广速度最快的程序设计语言。Java 作为一种面向对象的编程语言，它简单、面向对象、不依赖于机器的结构，具有跨平台、多线程、多媒体、安全性和稳定性等诸多优点，并且提供了开发的机制，具有很高的性能。特别是与 Internet 的成功结合，使 Java 成为风靡世界的网络编程语言。“一次编写，全球运行”的特点，更给计算机软件产业带来了一次新的革命。此外 Java 还提供了丰富的类库，使程序设计者可以很方便地建立起自己的系统。因此，学习 Java 语言可以说是顺应了时代潮流。

本书是一本全面介绍 Java 语言特点和应用技术的实用教材，主要是面向初学者和自学者的教材，由浅入深、循序渐进地讲解了 Java 的概念和编程技术。使广大用户既能够全面系统地了解 Java 技术的特点，又能掌握最新的实用技术。

全书主要内容包括：Java 的产生、现状和特点；Java 语言的基本语法；HTML 基本语法；面向对象编程技术的基本概念，对象、类以及相关概念；包、接口和异常；Java 系统类库；图形用户界面；多媒体技术；流；多线程和网络编程。本书从原理到方法，讲解了如何编写 Java 语言程序和使用 Java 语言获取网络资源等。

虽然在写这本书时，我们抱着比以往更严谨的态度，对书中的每个概念，每个说明都一再推敲，一再求证，但仍然无法满足每位读者的需求。因此，书中若有任何错误或用词不当，希望读者能不吝赐教；而当书中的说明让您感到疑惑时，欢迎您来信询问，我们一定给您满意的答复。请发电子邮件到：myran@163.net。

最后感谢所有帮助作者完成此书的朋友和同仁。本书由李诚、学勤、肖占彪、张艳伟、张志新、李盛、王宇霞、陈辉、杨新丽、冯怡等共同编写，程序员刘爱青、肖占国、张志民提供技术支持，在此表示由衷的感谢。正是他们的积极努力和全身心的投入，才使该书能早日与读者见面！同时向清华大学出版社金地公司编辑苗建强表示由衷的感谢。本书的成功出版，和苗建强先生的勤奋而高效的工作是分不开的。

编 者
2002 年 6 月

目 录

第 1 章 Java 基础入门	1
1.1 概述	1
1.1.1 Java 的起源与现状	1
1.1.2 Java 是什么	2
1.1.3 Java 语言的特点	3
1.1.4 Java 语言的应用前景	7
1.2 Java 程序开发与运行	7
1.2.1 Java 虚拟机	8
1.2.2 J2SDK 实用程序	9
1.3 Java 程序举例	11
1.3.1 Java 应用程序	11
1.3.2 Java Applet	14
第 2 章 HTML 初步	17
2.1 基础介绍	17
2.1.1 WWW 的诞生	17
2.1.2 什么是 HTML	18
2.1.3 轻松踏进 HTML 之门	18
2.2 HTML 初步	19
2.2.1 HTML 文件基本架构	19
2.2.2 文字的语法与编排	21
2.2.3 HTML 文字的排列语句	28
2.2.4 HTML 显示表格语句	30
2.2.5 HTML 链接语句	32
2.2.6 HTML 图像语句	33
2.2.7 HTML 表单、输入框语句	35
2.2.8 HTML 窗口语句	38
2.2.9 HTML 多媒体语句	39
第 3 章 基本数据类型	42
3.1 数据类型综述	42
3.2 简单数据类型的实例化——变量与常量	42

3.2.1 标识符 (Identifiers)	43
3.2.2 常量 (Constantvariables)	43
3.2.3 变量 (Variables)	43
3.3 整数类型	44
3.3.1 整型常量	44
3.3.2 整型变量	45
3.4 浮点型 (实型) 数据	47
3.4.1 实型常量	47
3.4.2 实型变量	47
3.5 字符类型	49
3.5.1 字符型常量	49
3.5.2 字符型变量	50
3.6 布尔类型	51
3.7 各类型数据间的优先关系和相互转换	52
3.7.1 数据类型的默认值	52
3.7.2 自动类型转换	52
3.7.3 强制类型转换	53
第 4 章 运算符与表达式	55
4.1 算术运算符	55
4.2 关系运算符	57
4.3 布尔逻辑运算符	59
4.4 位运算符	61
4.5 赋值运算符	64
4.6 扩张赋值运算符	65
4.7 条件运算符	66
4.8 其他运算符	68
4.9 表达式	68
第 5 章 流程控制语句	70
5.1 结构化程序设计的三种基本流程	70
5.2 选择语句	71
5.2.1 if 语句	71
5.2.2 switch 语句	75
5.3 循环语句	78
5.3.1 while 语句	78
5.3.2 do-while 语句	80
5.3.3 for 语句	80
5.4 跳转语句	83

5.4.1	continue 语句	83
5.4.2	break 语句	84
5.4.3	return 语句	86
第 6 章	数组	89
6.1	一维数组定义	89
6.1.1	测定数组的长度	92
6.1.2	数组的越界	93
6.2	多维数组的定义	94
6.2.1	二维数组的定义	94
6.2.2	数组的应用	97
第 7 章	面向对象程序设计	109
7.1	面向对象程序设计基本概念	109
7.1.1	对象	109
7.1.2	面向对象的编程技术	110
7.1.3	消息	112
7.2	类	112
7.2.1	类的创建	112
7.2.2	类体	118
7.2.3	类的构造方法	119
7.3	成员变量	121
7.3.1	成员变量的说明	121
7.3.2	作用域修饰符	123
7.3.3	构造函数	124
7.3.4	对象的生成、使用和清除	127
第 8 章	类的继承性和多态性	131
8.1	类的继承	131
8.1.1	创建子类	131
8.1.2	类成员的覆盖	134
8.1.3	this 与 super	134
8.2	类的多态	137
8.2.1	成员方法的重载	137
8.2.2	构造方法的重载	140
8.2.3	类与类之间的数据类型转换	142
第 9 章	包、接口和异常	144
9.1	包	144
9.1.1	创建包	144

9.1.2	程序包的引用	145
9.1.3	Java 的系统程序包	147
9.2	接口	148
9.2.1	接口概述	149
9.2.2	接口的声明	149
9.2.3	接口的实现	150
9.2.4	接口的多态	153
9.3	异常	155
9.3.1	编程中的错误	155
9.3.2	异常与异常类	156
9.3.3	异常处理	161
9.3.4	创建用户自己的异常	175
9.3.5	总结	182
第 10 章	Java 系统类库	183
10.1	Java 的类库结构	183
10.1.1	Java 类库	184
10.1.2	Java 1.1 类库	185
10.1.3	Java 1.2 简介	186
10.2	语言基础类库	186
10.2.1	Object 类	187
10.2.2	数据类型类	192
10.2.3	Math 类	194
10.2.4	字符串类	199
10.2.5	System 类	226
10.3	工具类库	238
10.3.1	向量 (Vector) 类	238
10.3.2	日期类	245
10.3.3	随机数类 Random	250
第 11 章	图形用户界面	254
11.1	常用组件	255
11.1.1	标签	255
11.1.2	按钮	257
11.1.3	选项框	258
11.1.4	复选框和选项按钮	259
11.1.5	列表框	262
11.1.6	文本框	263
11.1.7	文本区	264

11.1.8 滚动条	265
11.2 组件布局管理	267
11.2.1 顺序布局	267
11.2.2 边界布局	268
11.2.3 网格布局	269
11.2.4 网格包布局	270
11.2.5 卡片布局	272
11.2.6 面板的使用	273
11.2.7 手工布局	275
11.3 事件处理	276
11.4 事件处理编程范例	277
第 12 章 窗口、菜单和对话框	284
12.1 窗口	284
12.2 菜单	287
12.2.1 为窗口加入菜单	288
12.2.2 使用菜单子项快捷键	290
12.2.3 定义二级菜单	290
12.2.4 弹出式菜单	293
12.3 对话框	295
12.3.1 自定义对话框	295
12.3.2 文件对话框	301
12.4 Swing 简介	304
第 13 章 图形处理	308
13.1 基本图形	308
13.1.1 直线	308
13.1.2 矩形	310
13.1.3 椭圆	312
13.1.4 圆弧	314
13.1.5 多边形	315
13.2 画布	316
13.3 文字输出	318
13.3.1 字体、字型和字号	319
13.3.2 颜色	321
13.4 绘图模式控制	324
第 14 章 多媒体编程	327
14.1 图像	327
14.1.1 图像的种类与显示	327

14.1.2	幻灯机效果	331
14.1.3	生成图像	332
14.1.4	图像的旋转与透明处理	333
14.2	动画制作	335
14.2.1	动画原理	335
14.2.2	动态显示文字	339
14.2.3	图像高级处理	343
14.3	数字音频	346
第 15 章	多线程和网络编程	353
15.1	多线程	353
15.1.1	多线程的主要概念	353
15.1.2	多线程的实现	357
15.1.3	多线程的同步	362
15.2	网络编程	366
15.2.1	URL 编程	366
15.2.2	底层网络通信编程	369

第 1 章 Java 基础入门

1.1 概 述

最近几年 Internet 的迅速发展及万维网 WWW (World Wide Web) 的日益普及和快速增长,使得整个计算机世界正在经历着深刻的变革。1989 年超文本标记语言 HTML (Hypertext Markup Language, 超文本标记语言) 和万维网 WWW 的产生是 Internet 数据描述语言的一次飞跃,万维网把全球的信息资源用 HTML 格式统一起来。1996 年, Sun 公司的 Java 语言正式发表,则是一次 Internet 的技术革命。在 Internet 上, Java 语言和用 Java 编写的浏览器 HotJava 的出现在当时是特别吸引人的重大事件。Java 的诞生从根本上解决了 Internet 的异质、代码交换以及网络程序的安全性等诸多问题。其中 Java 语言是与平台无关的语言。Java 语言程序编译后,生成字节代码,运行在 Java 虚拟机上。一个操作系统平台只要提供 Java 虚拟机, Java 程序就可以在上面直接运行了。从理论上讲, Java 语言程序可运行在所有的操作系统平台上,从而在根本上解决了 Internet 的异质问题。

Java 语言不仅仅是一个程序设计语言,更是一个网络操作系统。因为:

1. Java 语言可不依赖于任何现有操作系统,可直接运行在某种机器或 CPU 上;
2. Java 语言有自己的内存管理机制;
3. Java 语言支持多线程程序,可对处理器资源进行分配和管理;
4. Java 语言可对设备进行管理,例如:屏幕、键盘、鼠标等;
5. Java 语言支持网络文件管理,Java 语言可打开远程机器上的文件,Java 语言文件的概念已经扩充到整个 Internet 网络。

Java 是具有很强吸引力的面向对象的编程语言,又是目前最为流行的网络编程语言。Java 的出现引起了软件开发的重大变革,成为推动 IT 业蓬勃发展的最新动力。它的出现对整个计算机软件业的发展产生了重大而深远的影响,对传统的计算模式提出了新的挑战。

1.1.1 Java 的起源与现状

Java 语言是一种新型的面向对象的程序设计语言。自从 1996 年初 Java 语言 1.0 版的正式发表,就引起了整个计算机工业界的极大兴趣,在各类杂志、报纸上充满了对 Java 语言的各种评论。Java 语言已经成为最卓越的程序设计语言之一。

Java 语言诞生于 1991 年,它是由 Sun Microsystems 的一个开发小组在开发 Green 项目时完成的。该项目最初是开发一种用于消费类电子产品的与平台无关的软件技术,主要用

来解决如电视机、电话、闹钟等家用电器的控制和通信问题的。开始时，该项目组成员准备采用 C++ 语言，但是他们感到 C++ 语言太复杂，安全性也差，无法满足项目设计的需要，最后决定基于 C++ 语言开发一种新的编程语言。该小组最初将其称为 Oak（一种橡树的名字），后来正式改名为 Java 语言。在 1991 年到 1993 年期间，这种语言一直被认为是用来开发消费类电子产品和交互式电视控制器的利器，但由于市场的因素，Sun 在最后放弃了继续开发和研制。但随后 Internet 的异常火爆，使 Sun 看到了 Oak 在计算机网络上的广阔前景，于是通过改造，就产生了 Java。

Java 语言是 Sun 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序的开发，它的硬件和软件平台无关性直接威胁到 Windows 和 Intel 的垄断地位。一时间“连 Internet，用 Java 编程”成为编程技术人员的一种时尚。

Java 的出现迅速引起了 IT 和工业界的高度重视，并成为 Internet 的重要编程语言。由于 Java 提供了强大的图形、图像、动画、音频、视频、多线程和网络交互能力，使它在设计交互式、多媒体网页和网络应用方面大显身手，成为当今推广最快的计算机编程语言。

1.1.2 Java 是什么

1. Java 是面向对象的网络编程语言

Java 是由 Sun 公司开发的新一代面向对象的网络编程语言。Java 的目标是建立一种在任意一种机器、任意一种操作系统的网络环境中运行的软件。Java 的原则是“程序只写一次，但可在任何时候、任何地方永远工作下去。”正是因为如此，Java 成为当今 Internet 上最流行、最受欢迎的一种程序开发语言。

Java 是第一个能编写可嵌入 Web 网页中的所谓小应用程序 (applets) 的程序设计语言。Java 使您能够编写可以在支持该语言的计算机上正常运行的应用程序。甚至还可以编写既作为普通的应用程序，又作为小应用程序运行的程序。由于近来 Java 的日趋成熟，所以它正在成为编写需要运行在不同种类的计算机系统中的应用程序的候选语言。

Java 是 Internet 上的世界语，它正日益成为客户机、数据库和其他服务器资源之间进行通信的“中间桥梁”，而且目前正流行的嵌入系统，也是 Java 的天地，成为了手持设备、因特网最好的编程语言之一。

Java 也是最好的网络编程语言，它具有强大的移植性、多线程处理和连网能力，成为最适合网络编程人员、最出色的编程语言之一。

2. Java 是产业

“Java 语言的出现，将会引起一场软件革命”。这是因为传统的软件往往都是与具体的实现环境有关，换了一个环境就需要重新做一番变动，造成人力和财力的浪费，而 Java 语言能在执行码（二进制码）上兼容，这样以前开发的软件就能直接运行在不同的机器上了。Java 语言作为软件开发的一种革命性的技术，其作用和突出影响主要表现在：

目前计算机产业的许多大公司购买了 Java 语言的使用许可证，Java 语言已经得到了工业界的认可。其中包括 IBM、Apple、Adobe、Microsoft 等公司都拥有了 Java 的使用许可证。

众多的软件开发商开始支持 Java 语言的软件产品。今天的世界是以网络为中心的计算机时代，不支持 HTML 和 Java 语言，应用程序的应用范围只能局限于同质的环境中。

Intranet（企业内部网）正在成为企业信息系统最佳的解决方案，而其中 Java 语言将发挥不可替代的作用。Intranet 的目的是把 Internet 用于企业的内部信息系统，它的优点主要是便宜、易于管理和使用。用户不管使用何种类型的机器和操作系统，界面是统一的 Internet 浏览器，而数据库、网页、应用程序则存在于 WWW 服务器上，无论是开发人员，还是管理人员或是用户都可以受益于该解决方案。

Java 已经逐步从一种单纯的计算机高级编程语言发展成为一种重要的 Internet 平台，并进而引发、带动了 Java 产业的发展 and 壮大，成为当今计算机业界不可忽视的力量和重要的发展潮流与方向！

1.1.3 Java 语言的特点

Java 语言是由 C++ 语言发展而来的，是一种彻底的纯面向对象的程序设计语言。作为一种纯面向对象的程序设计语言，它非常适合于大型软件的开发。同时它去掉了 C++ 语言的一些容易引起错误的特性。例如：Java 语言没有指针，避免了使用指针直接访问物理寄存器带来的风险，以及 Java 语言用接口代替了 C++ 语言中容易引起混乱的多重继承机制等。

Java 语言是一个简单的、面向对象的、分布的、解释的、健壮的、安全的、独立于平台的、可移植的、可扩展的、高性能的、多线程的、动态的程序设计语言。

1. Simple（简单）

Java 语言最初是为家用电器进行集成控制而设计的一种语言，因此它必须简单明了，易于学习。Java 语言与 C++ 语言的风格极为相似，但却比 C++ 语言简单多了，从某种意义上讲，Java 语言是 C/C++ 语言的一个变种。

Java 语言遗弃了 C++ 语言中容易引起程序错误的地方，如指针和内存管理。它没有一些高级编程语言的特性及 C++ 中一些不是绝对必要的功能。它没有头文件、指针算法、结构、单元、运算符重载、虚拟基础类等。Java 的简单实在让人着迷，它的功能如此强大，但应用起来却如此的简单。在 Java 中不存在许多模棱两可的概念。

这些为程序开发者提供了丰富的类库，使程序编写变得容易、简单。

Java 简单的另一个原因就是“小”，它采用特殊的软件构造方法，可在小机器上独立运行。其基本的解释器以及类支持模块大概仅 40KB，即使再加入基本的标准库，以及对线程的支持，也只需多加 175KB。

2. Object-Oriented（面向对象语言）

面向对象其实是现实世界模型的自然延伸。现实世界中任何实体都可以看作是对象。对象之间通过消息相互作用。另外，现实世界中任何实体都可归属于某类事物，任何对象都是某一类事物的实例。如果说传统的过程式编程语言是以过程为中心、以算法为驱动的话，面向对象的编程语言则是以对象为中心、以消息为驱动。用公式表示，过程式编程语言为：程序=算法+数据；面向对象编程语言为：程序=对象+消息。

所有面向对象编程语言都支持三个概念：封装、多态性和继承，Java 也不例外。现实世界中的对象均有属性和行为，映射到计算机程序上，属性则表示对象的数据，行为表示对象的方法（其作用是处理数据或同外界交互）。所谓封装，就是用一个自主式框架把对象的数据和方法连在一起形成一个整体。可以说，对象是支持封装的手段，是封装的基本单位。Java 语言的封装性较强，因为 Java 无全程变量，无主函数，在 Java 中绝大部分成员是对象，只有简单的数字类型、字符类型和布尔类型除外。而对于这些类型，Java 也提供了相应的对象类型以便与其他对象交互操作。

多态性就是多种表现形式，具体来说，可以用“一个对外接口，多个内在实现方法”表示。举一个例子，计算机中的堆栈可以存储各种格式的数据，包括整型、浮点或字符。不管存储的是何种数据，堆栈的算法实现是一样的。针对不同的数据类型，编程人员不必手工选择，只需使用统一接口名，系统可自动选择。运算符重载（operator overload）一直被认为是一种优秀的多态机制体现，但由于考虑到它会使程序变得难以理解，所以 Java 最后还是把它取消了。

继承是指一个对象直接使用另一对象的属性和方法。事实上，我们遇到的很多实体都有继承的含义。例如，若把汽车看成一个实体，它可以分成多个子实体，如：卡车、公共汽车等。这些子实体都具有汽车的特性，因此，汽车是它们的“父亲”，而这些子实体则是汽车的“孩子”。Java 提供给用户一系列类（class），Java 的类有层次结构，子类可以继承父类的属性和方法。与另外一些面向对象编程语言不同，Java 只支持单一继承。

纯面向对象语言，程序代码以类的形式组织，由类来定义对象的各种状态和行为。面向对象的编程技术是当今世界软件开发中最常见的技术之一，Java 就是一种新型的面向对象的程序设计语言，它使用类的软件对象，代码可重复利用和可扩展。这样，可以将这些由变量和方法组成的类作为一个模板，增加其他功能来创建其他的类，无须重写父类或超类的编码，使应用程序的开发变得更加简单和容易。

3. Distributed（分布式）

Java 提供了包容广泛的例程库，可处理像 HTTP 和 FTP 这样的 TCP/IP 协议。Java 应用程序可通过一个特定的 URL，来打开及访问对象，就像访问本地文件系统那样简单。

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上；操作分布是指把一个计算分散在不同主机上处理。

Java 支持 WWW 客户机/服务器计算模式，因此，它支持这两种分布性。对于前者，Java 提供了一个称作 URL 的对象，利用这个对象，可以打开并访问具有相同 URL 地址的对象，访问方式与访问本地文件系统相同。对于后者，Java 的 Applet 小程序可以从服务器下载到客户端，即部分计算在客户端进行，提高系统执行效率。

Java 提供了一整套网络类库，开发人员可以利用类库进行网络程序设计，方便地实现 Java 的分布式特性。

4. Interpreted（解释型）

Java 是解释型的，它编译后并不生成特定的 CPU 机器代码，而是 Java 字节代码。而 Java 语言运行时要借助于 Java 语言运行系统（Java 语言解释器，即 Java 虚拟机），首先该系统对 Java 语言编译后的字节代码进行解释，然后再执行它。Java 虚拟机是与硬件、软件

平台有关的，它使 Java 语言程序在某一特定硬、软件平台环境中直接运行目标代码指令。这种链接程序通常比编译程序所需资源少，所以程序员可以花上更多的时间在创建源程序上，而不必考虑运行环境。

Java 语言解释器直接对 Java 字节码进行解释执行。字节代码本身携带了许多编译信息，使得 Java 语言执行时的链接过程更加简单。

Java 通过预先将源代码编译为接近机器指令的字节码，有效地克服了传统解释型语言的性能瓶颈，同时又保持了解释型语言的可移植性。Java 解释器能直接在任何机器上执行 Java 字节码。

5. Robust (健壮)

Java 将大量重点放在早期的静、动态检查，排除出现错误的条件，异常处理，取消指针，内存保护等能力上。Java 有一个指针模型，可以避免改写内存和毁损数据的情况发生。

Java 语言要求显式方法声明，从而保证编译器可以发现方法调用错误，保证了程序更加可靠。

Java 语言程序不可能造成计算机系统的崩溃，因为 Java 语言系统仔细检测对内存的每次访问，确认它是安全合法的，而且不至于引起任何问题。如果出现某种出乎意料的事情，系统不会崩溃，而把该例外抛弃。系统会发现该类例外，并加以处理。

传统的计算机语言程序可以任意访问计算机的全部内存。程序可能修改内存中的任意值，这就会造成问题。Java 语言不支持指针，也就杜绝了内存的非法访问。Java 语言程序只能访问内存中允许它们访问的那些部分，所以 Java 语言程序不可能修改不允许修改的内存值。Java 语言自己操纵内存减少了内存出错的可能性，它利用所提供的自动垃圾收集器来进行内存管理，防止程序员在管理内存时容易产生的错误。通过集成的面向对象的例外处理机制，在编译时 Java 语言提示可能出现但未被处理的例外，帮助程序员正确地进行选择，以防止系统的崩溃。另外，Java 语言在编译时可捕获类型声明中的许多常见错误，防止动态运行时不匹配问题的出现。

6. Secure (安全性)

Java 将重点用于网络/分布式运算环境，确保建立无病毒且不会被侵入的系统。内存分配及布局由 Java 运行系统决定，字节码验证，可以轻松构建出防病毒、防黑客的系统。

Java 最初设计目的是应用于电子类消费产品，因此要求较高的可靠性。Java 虽然源于 C++，但它消除了很多 C++ 不可靠因素，可以防止许多编程错误。首先，Java 是强类型的语言，要求显式的方法声明，这保证了编译器可以发现方法调用错误，保证程序更加可靠；其次，Java 不支持指针，杜绝了内存的非法访问；第三，Java 的自动单元收集防止了内存丢失等动态内存分配导致的问题；第四，Java 解释器运行时实施检查，可以发现数组和字符串访问的越界；最后，Java 提供了异常处理机制，程序员可以把一组错误代码放在一个地方，这样可以简化错误处理任务，便于恢复。

由于 Java 主要用于网络应用程序开发，因此对安全性有着较高的要求。如果没有安全保证，用户从网络下载程序执行就非常危险。Java 通过自己的安全机制防止了病毒程序的产生和下载程序对本地系统的威胁破坏。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，然后，Java 解释器将决定程序中类的内存布局。随后，类装载器负责把

来自网络的类装载到单独的内存区域，避免应用程序之间相互干扰破坏。最后，客户端用户还可以限制从网络上装载的类只能访问某些文件系统。上述几种机制结合起来，使得 Java 成为安全的编程语言。

7. Architecture-Neutral (平台无关性)

Java 是平台无关的语言，是指用 Java 写的应用程序不用修改就可在不同的软硬件平台上运行。平台无关有两种：源代码级和目标代码级。C 和 C++具有一定程度的源代码级平台无关，表明用 C 或 C++写的应用程序不用修改只需重新编译就可以在不同平台上运行。

Java 主要靠 Java 虚拟机 (JVM) 在目标代码级实现平台无关性。JVM 是一种抽象机器，它附着在具体操作系统之上，本身具有一套虚拟机器指令，并有自己的栈、寄存器组等。但 JVM 通常是在软件上而不是在硬件上实现 (目前，SUN 系统公司已经设计实现了 Java 芯片，主要使用在网络计算机 NC 上。另外，Java 芯片的出现也会使 Java 更容易嵌入到家用电器中)。JVM 是 Java 平台无关的基础，在 JVM 上，有一个 Java 解释器用来解释 Java 编译器编译后的程序。Java 编程人员在编写完软件后，通过 Java 编译器将 Java 源程序编译为 JVM 的字节代码。任何一台机器只要配备了 Java 解释器，就可以运行这个程序，而不管这种字节码是在何种平台上生成的。另外，Java 采用的是基于 IEEE 标准的数据类型。通过 JVM 保证数据类型的一致性，也就确保了 Java 的平台无关性。

Java 应用程序能够在网络上任何地方执行，实现了字节代码平台无关性、语言版本完全统一无关性的特性，同时访问底层操作系统功能的扩展类库也不依赖于具体系统。Java 编译器生成的是一种与平台无关性的文件格式，只要用户安装了 Java 运行时间库，编译好的代码就可以在很多处理器上执行。它能生成特殊字节码指令，同任何一种特定的计算机系统都无关。它不仅在任何机器上都易于解释，也易于动态翻译成本机代码。

8. Portable (可移植)

Java 没有与具体环境有关的概念，其原始数据类型的大小是固定的，而且 Java 本身环境也具有很强的可移植性。

9. Multi-Threaded (多线程)

线程是操作系统的一种新概念，它又被称作轻量进程，是比传统进程更小的、可并发执行的单位。C 和 C++采用单线程体系结构，而 Java 却提供了多线程支持。

Java 语言嵌入了对并发控制的功能——多线程控制，大大简化了多线程应用程序的开发。多线程的优点是具有更好的交互性以及实时行为。多线程和进程两者是既有区别，又有联系的。进程是指一个正在运行当中的程序，它有自己的管理的一组系统环境和一个独立的存储空间。进程的特点是它所涉及的数据、内存是独立的。所以，多进程系统一定带有进程之间的通信机制，而为了实现进程通信，就要花费很多时间，也会让系统做出许多开销。线程也是指一个正在运行中的程序。但两者的区别是，可以多个线程共用同一个内存区域，也共享同一组系统资源。对每个线程来说，只有堆栈和寄存器数据是独立的。所以，线程之间进行通信和切换时，系统开销要比进程机制小得多。

Java 在两方面支持多线程。一方面，Java 环境本身就是多线程的，若干个系统线程运行负责必要的无用单元回收、系统维护等系统级操作；另一方面，Java 语言内置多线程控制，可以大大简化多线程应用程序开发。Java 提供了一个类 Thread，由它负责启动运行，

终止线程，并可检查线程状态。Java 的线程还包括一组同步原语。这些原语负责对线程实行并发控制。利用 Java 的多线程编程接口，开发人员可以方便地写出支持多线程的应用程序，提高程序执行效率。必须注意的是，Java 的多线程支持在一定程度上受运行时支持平台的限制。例如，如果操作系统本身不支持多线程，Java 的多线程特性可能就表现不出来。

10. Dynamic (动态)

Java 语言的设计使它适合于一个不断发展的环境。在类库中可以自由地加入新的方法和实例变量而不会影响用户程序的执行，并且 Java 语言通过接口来支持多重继承，使之比严格的类继承具有更灵活的方式和扩展性。这就是 Java 语言的动态特性，它可以动态地从本地或网上加载类。Java 语言的类有运行时的表述，这样 Java 程序在运行时可以分辨类之间的关系和类型信息以及把一个类动态地链接到运行系统中去。

Java 语言的动态特性也是其面向对象设计方法的扩展。它允许程序动态地装入运行过程中所需要的类，这是 C++ 语言进行面向对象程序设计所无法实现的。在 C++ 程序设计过程中，每当在类中增加一个实例变量或一种成员函数后，引用该类的所有子类都必须重新编译，否则将导致程序崩溃。

11. 可收集无用存储单元

用 C/C++ 编写软件的编程人员必须仔细跟踪所用的内存单元，当某一单元不再使用时，他们务必让程序释放它，方可再用。在大的项目中，这可能很困难，并往往成为出错和内存不足的根源。

在 Java 语言的情况下，编程人员不必为内存管理操心，Java 语言系统有一个称作“无用单元收集器”的内置程序，它扫描内存，并自动释放那些不再使用的内存单元。

Java 语言的这种自动垃圾收集机制，对不再被程序引用的对象自动取消其所占资源，这就彻底消除了出现存储器泄漏之类的错误，并免去了程序员管理存储器的繁琐工作。

1.1.4 Java 语言的应用前景

Java 语言有着广泛的应用前景，大体上可以从以下几个方面来考虑其应用：

1. 所有面向对象的应用开发，包括面向对象的事件描述、处理、综合等。
2. 计算过程的可视化、可操作化的软件开发。
3. 动态画面的设计，包括图形图像的调用。
4. 交互操作的设计。
5. Internet 的系统管理功能模块的设计、Web 页的动态设计、管理交互操作设计等。
6. 与各种数据库连接查询的 SQL 语言实现。

1.2 Java 程序开发与运行

用 Java 可以写两种类型的程序：小程序（又叫 Java Applet）和应用程序（Java