

高等学校计算机专业规划教材



# Java程序设计

## 第2版

黄岚 王岩 王康平 编著

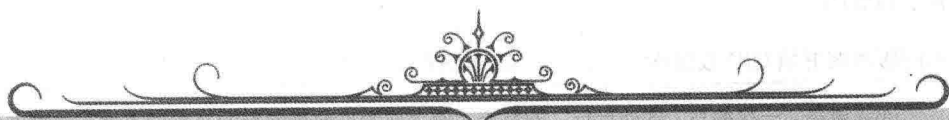


*I*ntroduction  
to Java Programming  
Second Edition



机械工业出版社  
China Machine Press

高等学校计算机专业规划教材



# Java程序设计

第2版

黄岚 王岩 王康平 编著

A large, stylized, cursive letter 'I' that serves as the first letter of the word 'Introduction'.

ntroduction

to Java Programming

Second Edition

## 图书在版编目 ( CIP ) 数据

---

Java 程序设计 / 黄岚, 王岩, 王康平编著. —2 版. —北京: 机械工业出版社, 2016.6  
(高等学校计算机专业规划教材)

ISBN 978-7-111-54190-5

I . J… II . ①黄… ②王… ③王… 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 (2016) 第 148724 号

---

Java 语言“一次编写, 到处运行”的特点, 使之成为目前最流行的开发语言之一。本书围绕“Java 程序设计”教学大纲, 深入浅出地介绍了 Java 语言的基础编程思想。

本书共分为 12 章, 主要包括 Java 语言基础、结构化程序设计、面向对象编程思想、图形化用户界面技术、异常处理技术、数组、I/O 流技术、并发控制技术、数据库和网络编程等基础内容, 以及 JDK 6 和后续版本中包含的容器、泛型和注解等高级技术。通过阅读本书, 读者可以全面掌握面向对象程序设计的基本方法, 领会 Java 语言的基本思想, 掌握 Java 语言技术的相关内容, 了解和理解蕴含在 Java 语言中的软件工程的基本设计思想。

本书可作为高等院校计算机及相关专业本科生 Java 程序设计的教材, 也可作为软件开发工程师与相关技术人员的自学参考书。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 王 彬 张梦玲

责任校对: 殷 虹

印 刷: 北京瑞德印刷有限公司

版 次: 2016 年 7 月第 2 版第 1 次印刷

开 本: 185mm × 260mm 1/16

印 张: 17.75

书 号: ISBN 978-7-111-54190-5

定 价: 39.00 元

---

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

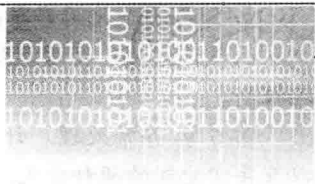
读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

# 前言



根据 TIOBE 开发语言排行榜 2016 年 4 月公布的流行开发语言的排名结果, Java 语言再次位居榜首。同时, 由于 Java 语言“一次编写, 到处运行”的特点, 使其在近年来国内外的软件开发企业中的使用比例排名第一。

Java 语言是一种广为使用的面向对象编程语言, 其设计思想代表了目前最为先进的高级语言程序设计思想。Java 语言是一种具有“简单、面向对象、分布式、解释型、健壮性(鲁棒性)、安全、平台无关、可移植、高性能、多线程和动态执行”等特性的语言, 是一种独立于平台的网络编程语言。在计算机技术发展到目前的互联网时代, Java 已经不仅仅是一种语言, 还可以说是一种技术, 这种技术涉及网络、编程、嵌入式等各个领域。

本书主要介绍 Java 语言基础、结构化编程思想、面向对象编程思想、图形化用户界面技术、异常处理技术、数组、I/O 流技术、并发控制技术、数据库和网络编程等基础内容, 以及 JDK 6 及后续版本中包含的容器、泛型和注解等高级技术。通过阅读本书, 读者可以全面掌握面向对象程序设计的基本方法, 领会 Java 语言的基本思想, 掌握 Java 语言技术的相关内容, 了解和理解蕴含在 Java 语言中的软件工程的基本设计思想。

本书可作为高等院校计算机及相关专业本科生 Java 程序设计的教材, 也可作为软件开发工程师与相关技术人员的自学参考书。Java 程序设计课程的教学以课堂教学为主, 通过对 Java 的基本语法与相关技术的讲解, 使学生充分掌握 Java 语言的基本概念、基本知识、基本内容, 了解和掌握 Java 语言的程序设计手段, 在完成学习之后具备较强的程序设计能力。坚持“既讲语言, 又讲技术, 语言为线, 技术为面, 相辅相成”的指导思想, 保证 Java 语言的语法和程序设计技术并重, 力求在清楚讲解 Java 语言编程技术的基础上, 传授 Java 编程思想和设计模式。在课堂教学的同时, 安排适当数量的实验教学, 使学生在学好基本理论和基础知识的同时, 能够得到充分的实践机会, 以验证课堂讲解的知识, 开拓编程思路, 切实锻炼和提高学生的程序设计能力。

本书分为 12 章, 主要内容如下:

第 1 章 Java 语言基础。计划 2 学时, 可扩展为 3 学时。这一章介绍 Java 语言的发展史、特征、运行机制、Java 程序的基本结构, 以及 JDK、Eclipse 等 Java 开发工具的使用。

第 2 章 结构化程序设计基础。计划 3 学时, 可扩展为 4 学时。这一章介绍标识符、关键字、数据类型、操作符、表达式、语句和控制流程等。

第 3 章 Java 类与对象。计划 3 学时, 可扩展为 4 学时。这一章介绍 Java 语言的基本程序设计方法和面向对象程序设计思想。

第 4 章 类的复用和抽象。计划 6 学时。这一章介绍类的复用、继承、多态、接口, 以及内部类。

第5章 图形化用户界面。计划4学时，可扩展为6学时。这一章介绍Java语言图形化用户界面设计方法、图形化组件类的使用、图形化用户界面的构建与布局管理器，以及事件处理机制。

第6章 异常处理。计划2学时，可扩展为4学时。这一章介绍异常处理的概念和机制，以及异常处理的两种方式。

第7章 对象的容纳。计划4学时，可扩展为6学时。这一章介绍数组、枚举与容器的基本概念，使得学生掌握如何根据实际需求选定不同对象容器。

第8章 I/O 流。计划2学时，可扩展为4学时。这一章介绍字节/字符流、I/O 流、节点流、处理流，以及其相关类的使用。

第9章 并发控制。计划2学时，可扩展为4学时。这一章介绍创建线程的两种方法、线程的生命周期，以及同步和协作的线程并发控制技术。

第10章 数据库应用。计划2学时，可扩展为3学时。这一章介绍DBMS相关概念、基本SQL语句、JDBC数据库编程技术，以及ORM模式。

第11章 网络应用。计划2学时，可扩展为3学时。这一章介绍网络编程相关类，以及TCP/UDP网络编程技术。

第12章 注解。为可选章节，可扩展1学时。

本课程建议讲授32学时，可扩展为48学时，扩展内容在目录中用\*号标注。如有条件可增加16学时的实验教学环节。

为配合课堂教学，编者精心制作了各章的课件，图文并茂地展示了编者的教学思想，读者可从hzbook.com上下载。书中还列举了丰富的实例程序，这些程序均在JDK和Eclipse环境下测试运行通过。

本书的编写组是由吉林大学多年讲授“Java程序设计”课程的教师，以及从事Java项目开发的教师组成，尤其感谢周袖和杜伟老师等为本教程的选材和编写所付出的辛勤劳动。本书第1版为吉林大学本科“十二五”规划教材，第2版在第1版的基础上扩展了Java教学体系参考、泛型扩展、GUI可视化编程等内容，并已列为吉林大学本科“十三五”规划教材。

由于作者水平有限，加之时间紧迫，本书内容虽经过多次修改，仍难免存在疏漏和错误之处，恳请广大师生和读者给予批评指正。

编者

2016年5月

# 目录



## 前 言

### 第 1 章 Java 语言基础.....1

#### 1.1 Java 概述.....1

##### 1.1.1 Java 起源.....1

##### 1.1.2 Java 的版本.....4

##### 1.1.3 Java 语言的特征.....5

##### 1.1.4 Java Application 与 Java Applet.....8

##### 1.1.5 Java 与 C/C++ 的比较.....9

##### 1.1.6 类库与 API 文档.....10

##### 1.1.7 Java 教学体系探索.....11

#### 1.2 Java 程序结构与运行机制.....11

##### 1.2.1 一个简单的 Java 程序.....12

##### 1.2.2 Java 运行机制.....13

#### 1.3 Java 开发环境.....15

##### 1.3.1 JDK.....15

##### 1.3.2 Eclipse.....16

##### 1.3.3 NetBeans.....17

##### 1.3.4 其他.....17

#### 1.4 Java 编程参考.....18

##### 1.4.1 Java 编程规范.....18

##### 1.4.2 Java 帮助文档.....18

##### 1.4.3 Java 相关讨论和社区.....18

#### 本章小结.....19

#### 习题.....19

#### 实验.....20

### 第 2 章 结构化程序设计基础.....21

#### 2.1 标识符、关键字和数据类型.....21

##### 2.1.1 标识符.....21

##### 2.1.2 关键字.....22

##### 2.1.3 数据类型.....23

##### 2.1.4 常量.....25

##### 2.1.5 变量的声明和初始化.....26

#### 2.2 操作符.....27

##### 2.2.1 赋值操作符.....27

##### 2.2.2 数学操作符.....28

##### 2.2.3 关系操作符.....29

##### 2.2.4 逻辑操作符.....30

##### 2.2.5 位操作符和位移操作符.....31

##### 2.2.6 字符串操作符.....32

##### 2.2.7 类型转换操作符.....32

##### 2.2.8 三元条件操作符.....33

##### 2.2.9 Java 语言操作符的优先级与 结合性.....34

#### 2.3 表达式与语句.....35

##### 2.3.1 表达式.....35

##### 2.3.2 语句与语句块.....35

#### 2.4 控制流程.....36

##### 2.4.1 顺序结构.....36

##### 2.4.2 选择结构.....36

##### 2.4.3 循环结构.....38

##### 2.4.4 break 语句和 continue 语句.....39

##### 2.4.5 return 语句.....41

#### 本章小结.....41

#### 习题.....41

#### 实验.....42

### 第 3 章 Java 类与对象.....43

#### 3.1 面向对象的基本概念.....43

#### 3.2 Java 类.....45

##### 3.2.1 类的定义.....45

##### 3.2.2 变量成员与方法成员.....46

##### 3.2.3 访问控制符.....50

##### 3.2.4 构造方法.....51

##### 3.2.5 终结处理方法.....52



6.2.4 异常处理原则	159	8.2 标准 I/O 流	215
6.3 自定义异常	159	8.3 文件 I/O 流	217
6.3.1 自定义异常的创建	159	8.3.1 File 类	217
6.3.2 抛出异常对象	159	8.3.2 基于字节的文件流	219
6.3.3 自定义异常抛出与捕获	161	8.3.3 基于字符的文件流	221
*6.3.4 异常转型	163	*8.4 随机访问文件	223
本章小结	164	*8.5 压缩	224
习题	164	本章小结	226
实验	165	习题	226
<b>第 7 章 对象的容纳</b>	166	实验	226
7.1 数组	166	<b>第 9 章 并发控制</b>	227
7.1.1 数组创建、初始化和使用	167	9.1 多任务系统介绍	227
7.1.2 多维数组	167	9.2 Java 多线程的概念	229
7.1.3 数组和数组引用	168	9.2.1 Java 语言中多线程的 作用	229
7.1.4 数组的工具类 Arrays	169	9.2.2 线程的创建	230
7.1.5 对象比较	170	9.2.3 线程的生命周期	232
7.2 枚举	171	9.2.4 线程的优先级	233
7.2.1 枚举的定义	172	9.3 线程之间的协作	234
7.2.2 枚举的使用	172	9.3.1 同步区域	235
7.3 容器	173	9.3.2 协作机制	236
7.3.1 List	173	9.3.3 死锁	238
7.3.2 泛型	178	*9.4 扩展模型	239
7.3.3 Set	181	9.4.1 ReentrantLock 锁	239
7.3.4 Map	188	9.4.2 await/signal 协作机制	240
7.3.5 容器的 Collections 工具类	193	本章小结	241
7.3.6 容器的选择	195	习题	241
*7.4 Java 泛型扩展	196	实验	242
7.4.1 无界通配符 “?”	196	<b>第 10 章 数据库应用</b>	243
7.4.2 通配符上界 “? extends T”	197	10.1 数据库基础	243
7.4.3 通配符下界 “? super T”	199	10.1.1 常用的关系数据库	243
本章小结	200	10.1.2 SQL 语言简介	245
习题	201	10.2 JDBC 概述	246
思考题	201	10.2.1 JDBC 驱动程序接口	246
实验	201	10.2.2 JDBC 应用程序接口	247
<b>第 8 章 I/O 流</b>	202	10.3 JDBC 编程步骤	248
8.1 流与相关类	202	10.4 一个简单的 JDBC 操作数据库 程序	248
8.1.1 流的概念	202	10.5 对象关系映射概述	251
8.1.2 字节流及其方法	203	10.5.1 ORM 的概念	252
8.1.3 字符流及其方法	207	10.5.2 ORM 技术特点	252
*8.1.4 节点流与处理流	211		



10.5.3 流行的 ORM 框架简介	252	习题	265
10.5.4 Hibernate 简介	253	实验	265
本章小结	253		
习题	254		
实验	254		
<b>第 11 章 网络应用</b>	<b>255</b>	<b>* 第 12 章 注解</b>	<b>266</b>
11.1 InetAddress 类	256	12.1 注解的语法	267
11.2 ServerSocket 类和 Socket 类	258	12.2 注解的使用	268
11.3 DatagramSocket 类	262	12.3 JUnit4 中注解的使用	270
本章小结	265	本章小结	271
		习题	272
		参考文献	273

## 【本章知识点】

- Java 语言历史
- Java 基本特性
- Java 简单程序实现
- Java 开发环境

本章着重讲述 Java 语言的产生历史和语言特点，以帮助读者了解 Java 语言作为一种纯粹的面向对象高级程序设计语言的实现意义和显著优势。本章领引读者初步接触 Java 程序，编写简单的 Java 代码，掌握 Java 程序结构和 Java 开发环境安装，并学习如何阅读 Java 帮助文档。

## 1.1 Java 概述

### 1.1.1 Java 起源

#### 1. 背景

Java 是印尼的一个盛产咖啡的小岛，而程序员往往喜欢喝咖啡，因此设计者将该语言取名为 Java。目前，Java 这杯咖啡已经飘香在世界各地。作为一种通用的支持面向对象的高级编程语言，其语法类似于 C++，是一种针对网络应用程序开发的面向对象的程序设计语言，奉行“一次编写，到处运行”（“Write once, run anywhere”）的理念。

1991 年，Sun MicroSystem 公司的 Jame Gosling、Bill Joe 等人，为在电视、烤面包箱等家用消费类电子产品上进行交互式操作，而开发了一个名为 Oak 的软件（Oak 是一种橡树的名字），但当时并没有引起人们的注意。直到 1994 年下半年，Internet 的迅猛发展，环球信息网（World Wide Web, WWW）的快速增长，促进了 Java 语言研制的进展，使它逐渐成为 Internet 上备受欢迎的开发与编程语言。俗话说：“有心栽花花不开，无心插柳柳成荫”，Sun 公司绝没想到原本想用于消费电子产品开发的编程语言，却率先在网络中得到了广泛应用。但也可以说是“东方不亮西方亮”，正是因为 Java 语言在设

计目标上的正确性,使得它备受青睐。

1995年,美国 Sun MicroSystem 公司正式向 IT 业界推出了 Java 语言,该语言具有安全、跨平台、面向对象、简单、适用于网络等显著特点。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运行。一些著名的计算机公司,包括 IBM、Oracle、微软、Netscape、Apple、SGI 等,纷纷与 Sun MicroSystem 公司签订合同,以获得授权使用 Java 平台技术。Java 语言因此被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品(计算机类就此一项入选)。随之出现了大量用 Java 编写的软件产品,受到工业界的重视与好评。可以说“Java 是 20 世纪 80 年代以来计算机界的一件大事”。微软公司总裁比尔·盖茨先生在经过研究后认为“Java 语言是长时间以来最卓越的程序设计语言”。目前,Java 语言已经成为最流行的网络编程语言。根据最新 TIOBE 开发语言排行榜宣布的流行开发语言的排名结果,Java 语言仍然高居榜首。在国内的软件开发企业中,Java 语言的使用比例也排名第一,这也使得 Java 软件开发人才的需求持续增长。许多大学纷纷开设 Java 课程,Java 正逐步成为程序员使用最多的编程语言。因此,Java 的出现正在对整个计算机产业产生深远的影响,并对传统计算模型提出新的挑战。

在 1997 年,3 周内超过 22 万的开发者下载了 JDK 1.1 软件,超过 8000 名开发者参加了 JavaOne 大会,同年 Java Card 2.0 平台发布。随后的 1998 年, JDK 1.1 正式发布,下载次数超过 200 万, VISA 公司应用 Java Card 技术推出了世界上首张智能卡, Java 社区 (JCP) 成立。1999 年, Java 2 平台源代码公开, JavaOne 大会参会人数增长到 20000 人, J2EE veta 软件发布。2002 年, J2EE SDK 下载次数达 200 万, 78% 的决策者认为, J2EE 技术是具有影响力的网络构建开发平台。2004 年, Java 2 Platform Standard Edition 5 发布,成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE 1.5 更名为 Java SE 5.0。2005 年, JavaOne 大会召开, Sun 公司公开 Java SE 6。此时, Java 的各种版本已经更名,以取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。直至今今最新的 Java 7 版本仍延续这种版本名的命名方法。2010 年, Oracle 宣布正式接管 Sun 的一切,在业界引起了轩然大波。一年后, Oracle 公司发布 Java SE 7。

接下来,让我们回顾一下程序设计语言的历史。人们要与计算机交流信息需要解决“语言”问题,计算机的程序设计语言正是用来解决这一问题的。计算机程序设计语言是人们为了向计算机准确描述实际情况而设计的,它经历了机器语言、汇编语言、高级程序设计语言的发展历程。1946 年第一台数字电子计算机 ENIAC、1954 年高级语言 FORTRAN、1970 年结构化语言 Pascal、1973 年的 C 语言、80 年代提出面向对象的程序设计软件设计思想,这些事件都是计算机程序设计发展的里程碑。目前计算机程序设计语言种类不下百种,如图 1.1 所示列举了部分流行的计算机语言,较为接近人们思维逻辑的计算机程序语言主要是基于面向对象的程序设计语言,而 Java 则是一种纯粹的面向对象的程序设计语言。

在进一步认识 Java 语言前,我们不得不提起 C 和 C++ 语言。C 语言是面向过程的语言,也是使用率非常高的语言。而面向对象的思想引入到编程语言之后,在 C 语言基础上产生了面向对象的 C++ 语言,它得到了广泛的应用。但是 C++ 语言必须兼容 C 语言,因此 C++ 语言是面向过程和面向对象混合的语言。Java 语言产生于 C++ 语言之后,是完全面向对象的编程语言。它充分吸取了 C++ 语言的优点,采用了程序员所熟悉的 C 和 C++ 语言的许多语法,同时又去掉了 C 语言中指针、内存申请和释放等影响程序健壮性的部分。可以说 Java 语言是站在 C++ 语言这个“巨人的肩膀上”前进的。



图 1.1 多种流行计算机语言

在经历了以大型机为代表的集中计算模式和以 PC 为代表的分散计算模式之后，互联网的出现使得计算模式进入了网络计算时代。网络计算模式的一个特点是计算机是异构的，即计算机间的硬件和操作系统的类型不一定相同，例如 Sun 工作站的硬件是 SPARC 体系，软件是 UNIX 中的 Solaris 操作系统，而 PC 的硬件是 Intel 体系，操作系统是 Windows 或 Linux。因此相应的编程语言（如 COBOL、FORTRAN、C/C++ 等）基本上只适用于单一的系统。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移，这就迫切需要一种跨平台的编程语言，使得用它编写的程序能够在网络中的各种计算机上正常运行。Java 就是在这种需求下应运而生的。

Java 语言的一个目标是跨平台，因此采用了解释执行而不是编译执行的运行环境。在执行过程中，它根据所在的不同的硬件平台把字节码程序解释为当前的机器码，实现跨平台运行。而动态装载程序代码的机制完全是为了适应网络计算的特点，程序可以根据需要实时地从服务器中下载代码并执行，在此之前还没有任何一种语言能够支持这一点。正是因为 Java 语言符合了互联网时代的发展要求，才使它获得了巨大的成功。

Sun MicroSystem 公司的总裁 ScottMcNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。WWW 的创始人 Berners-Lee 曾说“计算机事业发展的下一个浪潮就是 Java，并且将很快发生”，而今天使用 Java 已成大势所趋。

Apple、HP、IBM、Microsoft、Novell、SGI、SCO、Tandem 等公司均将 Java 并入各自开发的操作系统，负责开发并推广 Java 技术的 SunSoft 公司 (Sun 下属的一个子公司)，通过颁发许可证的办法允许各家公司把 Java 虚拟机和 Java 的 Applets 类库嵌入他们开发的操作系统。这样各类开发人员就能更容易地选择多种平台来使用 Java 语言编程，不同的用户也就能脱离 Web 浏览器来运行 Java 应用程序。这无疑是很受广大用户欢迎的，也为 Java 语言的应用开拓了极为广阔的前景。在 Java 推广阶段，甚至有人预言：Java 将是网络上的“世界语”，今后所有的用其他语言编写的软件统统都要用 Java 语言来改写。

## 2. Java 语言对软件开发技术的影响

工业界不少人预言：“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作一番改动——耗时费力。而 Java 语言能在执行码（二进制码）上兼容，这样以前所开发的软件就能运行在不同的机器上，只要所用的机器能提供 Java 语言解释器即可。

Java 语言将对软件的开发产生重要影响，可从如下几个方面考虑：

1) 软件的需求分析：可将用户的需求进行动态、可视化的描述，为设计者提供更加直观的要求。而用户的需求是各式各样的，不受地区、行业、部门、爱好的影响，都可以用 Java 语言描述清楚。

2) 软件的开发方法：由于 Java 语言的面向目标的特性，所以完全可以用面向对象的技术与方法来开发，这符合最新的软件开发规范要求。

3) Java 语言的动画效果远比 GUI 技术更加逼真，尤其是利用 WWW 提供的巨大动画资源空间，可以共享全世界的动态画面的资源。

4) 软件最终产品：用 Java 语言开发的软件具有可视化、可听化、可操作化的效果，这要比电视、电影的效果更为理想，因为它可以做到“即时、交互、动画与动作”，要它停就停，要它继续就继续，而这是在传统程序语言中难以做到的。

5) 其他：使用 Java 语言对开发效益优化、开发价值提高都有比较明显的影响。

### 3. 工业界对 Java 语言的评价

1996年5月29日至31日，在美国旧金山召开了全世界 Java 语言开发者大会，出席会议的多达 6500 多人，来自工业界的就超过一半。有人评价说：“这是近年来计算机界最光辉的一次盛会。”一些工业界的老总们相当看好 Java 语言，认为它的使用将会引起一场软件革命。从软件的设计风格、设计方法、设计目标到设计过程，都会产生彻底的变革，甚至会改变人们的生活方式。

在这次大会上，Java 的创始人之一 James Gosling 说：“Java 不仅仅是 Applets，它能做任何事情。”Dta 咨询公司的高级软件工程师 RichKadel 说：“Java 不仅仅是一种程序设计语言，更是现代化软件再实现的基础；Java 还是未来新型 OS 的核心；将会出现 Java 芯片；将构成各种应用软件的开发平台与实现环境，是人们必不可少的开发工具。”

由于各界都看好它，因此，各大公司都纷纷表示支持 Java。Intel、Xerox 公司声称将把 Java 嵌入他们的产品中去，就连华尔街金融界也在投入资金和人力用 Java 开发电子贸易、金融软件。

所以有人说：“现在第三方的开发商都团结在 Java 大旗周围了！”纷纷推出用 Java 开发的各种软件产品，以期尽快地占领市场。

### 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、可操作化的软件的开发。

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(选择交互、定向交互、控制流程等)。

5) Internet 的系统管理功能模块的设计，包括 Web 页面的动态设计、管理和交互操作设计等。

6) Intranet(企业内部网)上的软件开发(直接面向企业内部用户的软件)。

7) 各类数据库连接查询的 SQL 语句实现。

8) 其他应用类型的程序。

## 1.1.2 Java 的版本

Java 至今已发展到版本号 8，虽然有众多的 Java 开发版本号，但其每个版本号都由 3 个

版本名组成，即 Java SE、Java EE 和 Java ME。

### 1. Java SE (Java Platform Standard Edition)

标准版允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。Java SE 是基础包，但是也包含了支持 Java Web 服务开发的类，并为 Java EE 提供基础。

### 2. Java EE (Java Platform Enterprise Edition)

企业版本帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的，它提供 Web 服务、组件模型、管理和通信 API，可以用来实现企业级的面向服务体系结构 (Service-Oriented Architecture, SOA) 和 Web 2.0 应用程序。出于在安全和服务完备性等方面的考虑，Java EE 在数据维护上比较烦琐，属于重量级架构。为适应快速开发需求，越来越多的企业级开发趋向于轻量级架构，如广为流行的 SSH (Struts+Spring+Hibernate) 架构等。

### 3. Java ME (Java Platform Micro Edition)

微缩版为在移动设备和嵌入式设备 (如手机、PDA、电视机顶盒、打印机等) 上运行的应用程序提供一个健壮且灵活的环境。Java ME 包括灵活的用户界面、健壮的安全模型、许多内置的网络协议，以及对可以动态下载的联网和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次，就可以用于许多设备，而且可以利用每个设备的本机功能。

我们可以总结出：Java Standard Edition (Java SE) 适用于客户端桌面单机系统应用和 Applet 小程序；Java Enterprise Edition (Java EE) 适用于企业服务端应用环境开发，如 servlet 和 Server Pages 等；Java Micro Edition (Java ME) 适用于小型设备和智能卡，如移动设备、手机等。本书将以 Java SE 为主介绍 Java 编程。

## 1.1.3 Java 语言的特征

Java 是广泛使用的网络编程语言，它是一种新的计算概念，Sun 在 Java “白皮书”中指出：“Java 语言是一种简单、面向对象、分布式、解释执行、鲁棒、安全、体系结构中立、可移植、高性能、多线程以及动态性的编程语言。”

### 1. 简单性

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作。另外，Java 也适合于在小型机上运行，它的基本解释器及类的支持只有 40KB 左右，加上标准类库和线程的支持也只有 215KB 左右。

在程序的执行过程中，部分内存在使用后将处于废弃状态，如果不及时进行无用内存的回收，就会导致内存泄漏，进而导致系统崩溃。在 C++ 语言中是由程序员进行内存回收的，程序员需要在编写程序的时候把不再使用的对象内存释放掉。但是这种人为的管理内存释放的方法却往往由于程序员的疏忽而致使内存无法回收，同时也增加了程序员的工作量。而在 Java 运行环境中，始终存在着一个系统级的线程，专门跟踪内存的使用情况，定期检测出不再使用的内存，并进行自动回收，避免了内存的泄露，也减轻了程序员的工作量。

### 2. 面向对象

Java 是纯粹的面向对象语言，从最开始就是遵照面向对象方法学设计的，而面向对象是一种越来越流行的编程实现技术，并逐渐替代传统的基于过程的编程技术。

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对

象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏。而类则提供了一类对象的原型，并且通过继承机制，子类使用父类提供的方法，实现了代码的复用。

### 3. 分布性

分布式计算涉及多台计算机通过网络协同工作，Java 语言使分布式计算易于实施。Java 是面向网络的语言，通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL 地址在网络上很方便地访问其他对象。

### 4. 鲁棒性

Java 的鲁棒性可以体现为：1) Java 在编译和运行程序时，都要对可能出现的问题进行检查，以消除错误的产生。作为一种强制类型语言，Java 在编译和连接时都进行大量的类型检查，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。如果引用一个非法类型，或执行一个非法类型操作，Java 将在解释时指出该错误。2) 在 Java 程序中不能采用地址计算的方法通过指针访问内存单元，大大减少了错误发生的可能性，而且 Java 的数组并非用指针实现，这样就可以在检查中避免数组越界的发生。3) Java 提供自动垃圾收集来进行内存管理，防止程序员在管理内存时产生的错误。4) Java 通过集成的面向对象的异常处理机制，在编译时会提示出可能出现但未被处理的异常，帮助程序员正确地进行选择以防止系统的崩溃。

### 5. 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵。Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止黑客使用“特洛伊”木马等欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

字节码的执行需要经过三个步骤，首先由类装载器 (classloader) 负责把类文件 (.class 文件) 加载到 Java 虚拟机中，在此过程需要检验该类文件是否符合类文件规范。然后字节码校验器 (bytecodeverifier) 检查该类文件的代码中是否存在着某些非法操作，例如 Applet 程序中写本机文件系统的操作。如果字节码校验器检验通过，由 Java 解释器负责把该类文件解释成为机器码进行执行。Java 虚拟机采用的是“沙箱”运行模式，即把 Java 程序的代码和数据都限制在一定内存空间里执行，不允许程序访问该内存空间外的内存。如果是 Applet 程序，还不允许访问客户端机器的文件系统。

安全和方便总是相对矛盾的。Java 编程语言的出现使得客户端机器可以方便地从网络上下载 Java 程序到本机上运行，但是如何保证该 Java 程序不携带病毒或者不怀有其他险恶目的呢？如果 Java 语言不能保证执行的安全性，那么它就不可能使用至今。虽然有时候少数程序员会抱怨说 Applet 连文件系统也不能访问，但是正是各种安全措施的实行才确保了 Java 语言的生存。

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袭。Java 在运行应用程序时，严格检查其访问数据的权限，比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一个核实工具，一旦字节代码被核实，便由 Java 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 Java 极高的鲁棒性也增强了 Java 的安全性。虽然 Java Applet 充分考虑了安全性，但其运行效率和 Sun 公司对其滞后的推广策略，导致 Flash 等其他网络应用迅速替代了 Applet 的位置。

### 6. 体系结构中立

Java 解释器生成与体系结构无关的字节码指令，只要安装了 Java 运行时系统，Java 程

序就可在任意的处理器上运行。这些字节码指令对应于 Java 虚拟机中的表示, Java 解释器得到字节码后, 对它进行转换, 使之能够在不同的平台运行。

网络上充满了各种不同类型的机器和操作系统, 如 SPARC 结构、x86 结构、MIPS 和 PPC 等嵌入式处理芯片, UNIX、Linux、Windows 和部分实时操作系统等。为使 Java 程序能在网络的任何地方运行, Java 编译器编译生成了与体系结构无关的字节码结构文件格式。任何种类的计算机, 只有在其处理器和操作系统上有 Java 运行时环境, 字节码文件才可以在该计算机上运行。即使是在单一系统的计算机上, 结构中立也有非常大的作用。随着处理器结构的不断发展变化, 其他语言的程序员不得不编写各种版本的程序以在不同的处理器上运行, 而使用 Java 将使同一版本的应用程序可以运行在所有的平台上(虽然微软的 .NET 架构正在效仿 Java 的这一跨平台理念)。

### 7. 可移植性

与平台无关的特性使 Java 程序可以方便地被移植到网络上的不同机器。同时, Java 的类库中也实现了与不同平台的接口, 使这些类库可以移植。体系结构的中立也使得 Java 系统具有可移植性, Java 运行时系统可以移植到不同的处理器和操作系统上。Java 的编译器是由 Java 语言实现的, 解释器是由 Java 语言和标准 C 语言实现的, 因此可以较为方便地进行移植工作。

### 8. 解释执行

编译型语言是一次性编译成机器码, 脱离开发环境独立运行, 所以运行效率较高, 但是由于编译成的是特定平台上的机器码, 所以可移植性差。编译型语言的典型代表有 C、C++、FORTRAN、Pascal 等。解释型语言是专门的解释器对源程序逐行解释成特定平台上的机器码并执行的程序。解释型语言通常不会进行整体性的编译和链接处理, 解释型语言相当于把编译型语言的编译和解释过程混合到了一起同时完成。于是, 每次执行解释型语言的程序都要进行一次编译, 因此解释型语言的程序运行效率通常较低, 而且不能脱离解释器独立运行。但解释型语言跨平台容易, 只需要提供特定的平台解释器即可。解释型语言可以方便地进行程序的移植, 但是以牺牲程序的执行效率为代价的。

Java 语言的特殊性在于用 Java 编写的程序先要经过编译, 但不会生成特定平台的机器码, 而是生成一种平台无关的字节码, 即 .class 文件。这种字节码并不是可执行的, 必须通过 Java 解释器来解释执行。因此可以认为 Java 是半编译半解释型语言。

Java 语言里解释执行字节码文件的是 JVM (Java 虚拟机)。JVM 是可执行 Java 字节码文件的虚拟计算机。所有平台上的 JVM 向编译器提供相同的编程接口, 而编译器只需要面向虚拟机, 生成虚拟机能理解的代码, 由虚拟机来解释执行。在一些虚拟机的实现中, 还会将虚拟机代码转换成特定系统的机器码执行, 从而提高执行效率。

### 9. 高性能

虽然 Java 是解释执行的, 但它仍然具有非常高的性能, 在一些特定的 CPU 上, Java 字节码可以快速地转换成机器码进行执行。而且 Java 字节码格式的设计就是针对机器码的转换, 实际转换时相当简便, 自动的寄存器分配与编译器对字节码的一些优化可使之生成高质量的代码。Java 虚拟机的改进和“即时编译”(Just In Time, JIT)技术的出现使得 Java 的执行速度更快。

另外, 与其他解释执行的语言如 Basic、TCL 不同, Java 字节码的设计使之能很容易地直接转换成对应于特定 CPU 的机器码, 从而得到较高的性能。



## 10. 多线程

当运行一个应用程序时，就启动了一个进程，当然有些会启动多个进程。启动进程的时候，操作系统会为进程分配资源，其中最主要的资源是内存空间，因为程序是在内存中运行的。在进程中，有些程序流程块是可以乱序执行的，并且这个代码块可以同时被多次执行。实际上，这样的代码块就是线程体。线程是进程中乱序执行的代码流程。当多个线程同时运行的时候，这样的执行模式称为并发执行。多线程的目的是最大限度地利用 CPU 资源。

多线程机制使应用程序能够并行执行，而且同步机制保证了对共享数据的正确操作。通过使用多线程，程序设计者可以分别用不同的线程完成特定的行为，而不需要采用全局的事件循环机制。Java 编写程序都运行在 JVM 中，在 JVM 的内部，程序的多任务是通过线程来实现的。每用 Java 命令启动一个 Java 应用程序，就会启动一个 JVM 进程。在同一个 JVM 进程中，有且只有一个进程，就是它自己。在这个 JVM 环境中，所有程序代码的运行都是以线程来运行的。一般常见的 Java 应用程序都是单线程的。比如，用 Java 命令运行一个最简单的 HelloWorldDemo 的 Java 应用程序时，就启动了一个 JVM 进程，JVM 找到程序的入口 main 方法，然后运行 main 方法，这样就产生了一个线程，这个线程叫作主线程。当 main 方法结束后，主线程运行完成，JVM 进程也随即退出。多线程编程集成于 Java 环境中，而其他语言多数是需要调用操作系统级别的特定过程才能支持多线程。

## 11. 动态性

Java 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。Java 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。Java 在执行过程中，可以动态地加载各种类库，这一特点使之非常适合于网络运行，同时也非常有利于软件的开发，即使是更新类库也不必重新编译使用这一类库的应用程序。

### 1.1.4 Java Application 与 Java Applet

Java 语言的特性使它可以最大限度地利用网络，Java 的小应用程序 (Applet) 可在网络上传输而不受 CPU 和环境的限制。Applet 是 Java 的小应用程序，它是动态、安全、跨平台的网络应用程序。Java Applet 嵌入 HTML 语言，通过主页发布到 Internet。网络用户访问服务器的 Applet 时，这些 Applet 从网络上进行传输，然后在支持 Java 的浏览器中运行。由于 Java 语言的安全机制，用户一旦载入 Applet，就可以放心地来生成多媒体的用户界面或完成复杂的计算而不必担心病毒的入侵。虽然 Applet 可以和图像、声音、动画等一样从网络上下载，但它不同于这些多媒体的文件格式，它可以接收用户的输入，动态地进行改变，而不仅仅是动画的显示和声音的播放。由于 Java Applet 已被 Flash 和 HTML5 等新技术取代，本书仅对其基本特性做简单介绍，而不做进一步详细介绍，有兴趣的读者可以参考 Oracle 公司的官方网站 <http://www.oracle.com/technetwork/java/javase/overview/index.html>。

Java Applet 和 Java Application 在结构方面的主要区别表现在：

- 1) 运行方式不同。Java Applet 程序不能单独运行，它必须依附于一个用 HTML 语言编写的网页并嵌入其中，通过与 Java 兼容的浏览器来控制执行。Java Application 是完整的程序，可以独立运行，只要有支持 Java 的虚拟机，它就可以独立运行而不需要其他文件的支持。

- 2) 运行工具不同。运行 Java Applet 程序的解释器不是独立的软件，而是嵌在浏览器中作为浏览器软件的一部分。Java Application 程序被编译后，用普通的 Java 解释器就可以使其