



教育部实用型信息技术人才培养系列教材

Java 程序设计

基础教程

龚永罡 陈昕 等编著

清华大学出版社





教育部实用型信息技术人才培养系列教材

Java 程序设计

基础教程

龚永罡 陈昕 等编著

清华大学出版社
北 京

内 容 简 介

本书是一本实用的 Java 程序设计教材,以面向高校进行 Java 语言程序设计教学实践活动和培训机构进行 Java 程序员开发培训为目标。全书共分为 14 章,包括基础知识篇、基础开发篇、中级开发篇。本书强调基本概念、技术和方法的阐述,注重理论联系实际的操作。内容包括 Java 程序设计概论,Java 基本的程序设计结构,类和对象,继承、接口与内部类,Java API,图形用户界面设计,多线程,文件和流,异常处理,Java 网络编程,Java Applet 基础,常见数据结构的 Java 实现,Java 数据库连接——JDBC,Java 多媒体编程——图像、动画和声音等。每章都列举许多经典实例,章后配有练习思考题,引导读者利用本章的知识点进行思考和程序设计,有利于提高读者分析解决问题的能力。

本书可以作为高等院校计算机专业及职业培训的教材,也可作为其他专业的选学和自学教材。随书配有电子课件、习题解答、源程序代码、实验指导书等相关辅助教学资料,凡使用本书作为教材的教师可登录网站下载。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java 程序设计基础教程 / 龚永罡等编著. —北京:清华大学出版社, 2009.7
ISBN 978-7-302-20040-6

I. J… II. 龚… III. JAVA 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2009)第 064680 号

责任编辑:冯志强

责任校对:徐俊伟

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:三河市李旗庄少明装订厂

经 销:全国新华书店

开 本:185×260 印 张:22.5 字 数:559 千字

版 次:2009 年 7 月第 1 版 印 次:2009 年 7 月第 1 次印刷

印 数:1~5000

定 价:36.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:029822-01

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被业界广泛认可和采用的工具，同时越来越多的高校也将其作为程序设计教学时主要的编程语言。随着 Java 语言的广泛应用，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 到定位于客户端程序应用的 J2SE，以及定位于企业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，本书的目的就在于帮助 Java 初学者，力求以最简单、最实用的文字和实例帮助初学者，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。

本书编写的初衷是设计一本真正能适合高校进行 Java 语言程序设计教学实践活动和培训机构进行 Java 程序员开发培训的基础教程。按照由浅入深、通俗易懂的原则介绍 Java 编程语言，让学生迅速了解、掌握 Java 技术的基本思想与应用开发技术，掌握基础知识和操作技能，编制面向对象和网络化的程序，并且能够根据实际需求编制出一些实用程序。

本书共分为 14 章，包括基础知识篇、基础开发篇、中级开发篇。基础知识篇介绍了 Java 程序设计概述，Java 基本的程序设计结构，类和对象，继承、接口与内部类；基础开发篇介绍了 Java API、图形用户界面设计、多线程、文件和流、异常处理、Java 网络编程；中级开发篇介绍了 Java Applet 基础，常见数据结构的 Java 实现，Java 数据库连接——JDBC，Java 多媒体编程——图像、动画和声音。

本书将章节安排为三大篇是考虑到目标不同的教学课程对学时要求不同。教师可根据学时长短选择授课内容。学时短的可以选择讲授基础知识篇和基础开发篇，中级开发篇可为学有余力的学生课后进一步学习提供参考；学时长的可将三大篇内容全部讲授。

此外，本书的后面还提供了 3 个附录，一是 Eclipse 3.3 开发集成环境的下载、安装和运行，包含下载网址、下载方法和安装运行说明，按照步骤操作可以很快搭建一个 Java 集成开发平台；二是 MyEclipse 7.0 的下载、安装和运行，MyEclipse 是基于 Eclipse 的一个插件，可以提供更全面的开发插件和功能更强大的开发环境；三是 MySQL 数据库的介绍，包括 MySQL 数据库和 JDBC 驱动程序的下载网址、下载方法和安装运行说明，指导读者快速掌握这款应用广泛的轻量级免费数据库服务器的安装和使用。

本书强调基本概念、技术和方法的阐述，注重理论联系实际的操作。每一章都列举许多经典实例和分析，每章后都附有练习思考题，习题内容由浅入深，侧重实际操作，引导读者利用本章的知识点进行思考和程序设计，以提高解决实际问题的能力。教师可根据情况安排课后习题作业及进行习题分析。另外，本书还配有电子版实验指导手册可供下载，通过程序设计实例的分析和上机实践，培养和提高学生对 Java 语言的应用能力

前言

和面向对象程序设计的能力。该实验指导手册也可以作为开放式实验、家庭作业以及自学参考用书。

随书配有教师电子课件、习题解答、源程序代码、实验指导手册等相关辅助教学资料，凡使用本书作为教材的教师可登录网站下载。

本书主要由龚永罡、陈昕编写，参与编写者还有王淼、陈涛、邓彬、李京云、王泽琳、夏天、刘林、肖雪芳、刘博、赵凯等。本书编写过程中还得到许多老师和同学的支持与帮助，他们参与了资料的搜集、实验及程序的编写和调试工作，在此一并表示衷心的感谢。由于作者水平有限，书中不足之处在所难免，希望读者给予批评、指正。多提宝贵意见。

作 者

2009年3月

基础知识篇

第 1 章 Java 程序设计概述..... 2

1.1 机器语言、汇编语言和高级语言 2

1.2 Java 语言概述 3

1.2.1 Java 语言的产生 3

1.2.2 Java 语言的发展 4

1.2.3 Java 虚拟机及 Java 的
跨平台原理 4

1.2.4 Java 语言的关键特点 5

1.3 Java 平台和主要应用方向 6

1.4 Java 的垃圾回收与代码
安全检查机制 7

1.5 典型的 Java 开发环境 8

1.5.1 下载、安装和了解 JDK 8

1.5.2 设置 Path 变量 10

1.5.3 设置 ClassPath 变量 11

1.5.4 测试环境变量 12

1.5.5 Eclipse 和 MyEclipse 13

1.6 Java 编程示例 14

1.6.1 第一个 Java Application 14

1.6.2 Java 程序的编辑 15

1.6.3 运行 Java Application
应用程序 15

1.6.4 第一个 Java Applet 17

1.6.5 运行 Applet 17

1.7 反编译工具的介绍 18

1.8 Java JDK 6 新功能介绍 18

1.9 练习思考题 19

第 2 章 Java 基本的程序设计结构..... 20

2.1 标识符和关键字 20

2.1.1 标识符 20

2.1.2 关键字 21

2.2 Java 的基本数据类型 21

2.2.1 布尔型 21

2.2.2 字符型 22

2.2.3 整数型 23

2.2.4 浮点型 24

2.2.5 基本数据类型的转换 24

2.3 运算符、表达式 26

2.3.1 算术运算符与算术表达式 27

2.3.2 比较和条件运算 28

2.3.3 逻辑和位运算 30

2.3.4 递增和递减运算 34

2.3.5 赋值运算符 36

2.4 语句 37

2.4.1 语句概述 37

2.4.2 控制语句 38

2.4.3 循环语句 42

2.4.4 break 和 continue 46

2.5 输入/输出 48

2.6 数组 50

2.6.1 数组的概念 50

2.6.2 一维数组 50

2.6.3 多维数组 51

2.7 练习思考题 53

第 3 章 类和对象 57

3.1 面向对象技术基础 57

3.1.1 面向对象的基本概念 57

3.1.2 面向对象的基本特征 58

3.1.3 面向对象程序设计方法 59

3.2 类的定义与使用 60

目录

3.2.1 类	60	3.8.2 import 语句	79
3.2.2 类声明	60	3.9 练习思考题	80
3.2.3 类体	61		
3.2.4 成员变量和局部变量	62	第 4 章 继承、接口和内部类	83
3.2.5 成员方法	63	4.1 继承、超类和子类	83
3.2.6 方法重载	65	4.1.1 子类	83
3.2.7 构造方法	66	4.1.2 子类的继承	83
3.2.8 值得注意的问题	67	4.1.3 super 关键字	85
3.3 对象	67	4.2 抽象类和接口	88
3.3.1 对象的生成	67	4.2.1 抽象类	88
3.3.2 对象的使用	68	4.2.2 接口	90
3.3.3 对象的引用和实体	70	4.2.3 接口的定义	90
3.3.4 参数传值	71	4.2.4 接口的实现	91
3.3.5 对象的清除	72	4.3 内部类	92
3.4 static 关键字	72	4.3.1 内部类的定义和使用	92
3.4.1 实例变量和类变量的区别	72	4.3.2 匿名类的定义和使用	94
3.4.2 实例方法和类方法的区别	73	4.4 Class 类	95
3.5 this 关键字	74	4.4.1 获取类的有关信息	95
3.6 final 关键字	76	4.4.2 使用 Class 实例化 一个对象	96
3.7 Java 类中的限定词	77	4.5 练习思考题	97
3.8 包	78		
3.8.1 包语句	78		

基础开发篇

第 5 章 Java API	102	5.5.1 Hashtable 类	114
5.1 Java API 的概念	102	5.5.2 Properties 类	116
5.2 Object 类	103	5.5.3 利用 Properties 类读取配置 文件信息	117
5.3 String 类和 StringBuffer 类	105	5.6 System 类与 Runtime 类	120
5.3.1 String 类	106	5.6.1 System 类	120
5.3.2 StringBuffer 类	110	5.6.2 Runtime 类	121
5.3.3 String 类和 StringBuffer 类下字符串的区别	112	5.7 Date 与 Calendar 类	122
5.4 基本数据类型的对象类	112	5.7.1 Date 类	122
5.4.1 Double 类和 Float 类	112	5.7.2 Calendar 类	123
5.4.2 Byte、Integer、Short、 Long 类	113	5.8 Math 与 Random 类	125
5.4.3 Character 类	113	5.8.1 Math 类	125
5.5 Hashtable 与 Properties 类	114	5.8.2 Random 类	127
		5.9 练习思考题	128

第 6 章 图形用户界面设计	130
6.1 Java 的 GUI	130
6.2 用 AWT 生成图形化用户界面	131
6.2.1 java.awt 包	131
6.2.2 组件	131
6.3 常用容器	132
6.3.1 Frame	132
6.3.2 Panel	133
6.3.3 ScrollPane	134
6.4 布局管理器	135
6.4.1 FlowLayout 布局	136
6.4.2 BorderLayout 布局	137
6.4.3 GridLayout 布局	138
6.4.4 CardLayout 布局	139
6.4.5 容器的嵌套	140
6.4.6 总结	141
6.5 AWT 事件处理模型	141
6.5.1 事件处理机制	141
6.5.2 事件类	143
6.5.3 事件监听器	143
6.5.4 AWT 事件及其相应的 监听器接口	144
6.5.5 事件适配器	146
6.6 AWT 组件库	147
6.6.1 按钮	147
6.6.2 单行文本输入框	149
6.6.3 文本输入区	150
6.6.4 选择框和选择框组	151
6.6.5 下拉式列表	154
6.6.6 滚动列表	155
6.6.7 画布	157
6.6.8 对话框	159
6.6.9 菜单栏、菜单和菜单项	162
6.7 练习思考题	163
第 7 章 多线程	166
7.1 线程简介	166
7.1.1 进程与线程	166
7.1.2 线程的状态	167

7.2 Thread 类的子类创建线程	169
7.3 实现 Runnable 接口	171
7.4 基本的线程控制	172
7.5 线程的调度	176
7.6 多线程的互斥与同步	177
7.6.1 临界资源问题	177
7.6.2 互斥锁	178
7.6.3 多线程的同步	179
7.7 Daemon 线程	181
7.8 练习思考题	182
第 8 章 文件和流	184
8.1 I/O 流概述	184
8.1.1 I/O 流的层次	184
8.1.2 InputStream 和 Output- Stream	185
8.1.3 I/O 中的例外	186
8.2 文件处理	187
8.2.1 文件描述	187
8.2.2 文件的顺序处理	189
8.2.3 随机访问文件	190
8.3 过滤流	191
8.3.1 过滤流类层次	191
8.3.2 几种常见的过滤流	192
8.4 字符流的处理	192
8.4.1 Reader 和 Writer	192
8.4.2 InputStreamReader 和 Output- StreamWriter	193
8.4.3 BufferedReader 和 Buffered- Writer	194
8.5 对象的序列化 (Serialization)	195
8.5.1 序列化的定义	195
8.5.2 序列化方法	196
8.5.3 序列化的注意事项	197
8.6 其他常用的流	198
8.6.1 管道流	198
8.6.2 内存的读/写	200
8.6.3 顺序输入流	202
8.7 练习思考题	203

第 9 章 异常处理	205		
9.1 异常处理基础	205		
9.1.1 异常示例	205		
9.1.2 异常类的层次	206		
9.1.3 异常类的分类	206		
9.2 异常的处理	207		
9.2.1 捕获异常	208		
9.2.2 声明抛弃异常	209		
9.3 自定义异常类	209		
9.3.1 继承 Exception 类	210		
9.3.2 继承 Throwable 类	210		
9.3.3 自定义异常类的使用	211		
9.4 练习思考题	212		
第 10 章 Java 网络编程	213		
10.1 网络编程基础知识	213		
10.1.1 网络基础知识	213		
10.1.2 网络基本概念	214		
10.1.3 传输协议: TCP、UDP	214		
10.2 URL 及应用示例	215		
10.2.1 统一资源定位器 URL	215		
10.2.2 URL 的组成	216		
10.2.3 创建一个 URL	217		
10.2.4 解析一个 URL	217		
10.2.5 从 URL 读取 WWW 网络资源	218		
		10.2.6 通过 URLConnetction 连接 WWW	219
		10.3 InetAddress 类	221
		10.4 基于套接字的低层次 Java 网络编程	221
		10.4.1 Socket 通信	221
		10.4.2 Socket 通信的一般过程	222
		10.4.3 创建 Socket	222
		10.4.4 客户端的 Socket	223
		10.4.5 服务器端的 Server- Socket	223
		10.4.6 打开输入/输出流	224
		10.4.7 关闭 Socket	225
		10.4.8 简单的 Client/Server 程序设计	225
		10.4.9 支持多客户的 Client/Server 程序设计	228
		10.5 UDP 数据报	230
		10.5.1 什么是 Datagram	230
		10.5.2 DatagramSocket、Datagram- Packet	230
		10.5.3 基于 UDP 的简单的 Client/ Server 程序设计	232
		10.5.4 用数据报进行广播通信	235
		10.6 练习思考题	237

中级开发篇

第 11 章 Java Applet 基础	240		
11.1 Java Applet 的运行原理	240		
11.1.1 Applet 的运行环境	240		
11.1.2 什么是 appletviewer	241		
11.1.3 Java Applet 程序开发 步骤	242		
11.1.4 Applet 的安全机制	244		
11.2 Applet 类及其方法	244		
11.2.1 Applet 类的基本方法	244		
11.2.2 Applet 运行状态 控制方法	245		
		11.2.3 Applet 应用的有关参数	247
		11.3 一个显示动画的 Applet 程序	248
		11.3.1 重载 update()	248
		11.3.2 使用双缓冲技术	248
		11.3.3 例子	249
		11.4 Applet 中使用 URL	253
		11.5 Applet 中建立新线程	255
		11.6 Applet 中播放声音	256
		11.7 练习思考题	259

第 12 章 常见数据结构的 Java

实现 261

- 12.1 链表 261
 - 12.1.1 链表的创建 262
 - 12.1.2 LinkedList 类中的常用方法 263
 - 12.1.3 使用 Iterator 类遍历链表 264
- 12.2 栈 269
 - 12.2.1 栈的常用方法 270
 - 12.2.2 用链表实现栈 271
- 12.3 树集 272
 - 12.3.1 用构造方法 TreeSet() 创建一个树集 273
 - 12.3.2 用构造方法 TreeSet (Comp-aratorc) 创建一个树集 273
 - 12.3.3 TreeSet 类的一些常用方法 275
- 12.4 树映射 279
- 12.5 散列表 Hashtable 281
 - 12.5.1 Hashtable 类的常用方法 281
 - 12.5.2 Hashtable 类的应用 282
- 12.6 散列集 286
- 12.7 向量 287
 - 12.7.1 Vector 类的常用方法 288
 - 12.7.2 Vector 类的应用 288
- 12.8 练习思考题 291

第 13 章 Java 数据库连接——

JDBC 293

- 13.1 JDBC 概述 293
- 13.2 JDBC 的类型 294
 - 13.2.1 类型 1: JDBC-ODBC 桥 294
 - 13.2.2 类型 2: Native-API Bridge 295
 - 13.2.3 类型 3: JDBC-middlewarc 295
 - 13.2.4 类型 4: Pure Java Driver 295
- 13.3 JDBC 主要的接口和核心类 296

- 13.4 JDBC 进行数据库操作的步骤 300
- 13.5 使用 JDBC 进行数据操作 301
 - 13.5.1 建立一个数据源类 302
 - 13.5.2 Statement、ResultSet 304
 - 13.5.3 PreparedStatement 307
 - 13.5.4 LOB 读写 309
 - 13.5.5 事务 (Transaction) 312
 - 13.5.6 批处理 313
 - 13.5.7 ResultSet 光标控制 313
 - 13.5.8 ResultSet 新增、更新、删除数据 315
 - 13.5.9 ResultSetMetaData 316
- 13.6 练习思考题 317

第 14 章 Java 多媒体编程: 图形、图像和声音 320

- 14.1 AWT 绘图 320
 - 14.1.1 Graphics 类 320
 - 14.1.2 绘制文本、建立字体 321
 - 14.1.3 绘制基本图形 323
- 14.2 Java 2D 图形 325
 - 14.2.1 绘制基本图形 326
 - 14.2.2 控制图形线条的粗细 327
 - 14.2.3 颜色渐变 329
 - 14.2.4 旋转图形 330
- 14.3 Java 的图像处理 332
 - 14.3.1 在小应用程序中绘制图像 332
 - 14.3.2 在应用程序中绘制图像 334
- 14.4 打印图形、图像 335
- 14.5 Java 媒体框架——JMF 336
 - 14.5.1 JMF 的体系结构 337
 - 14.5.2 一个播放多媒体的例子 338
- 14.6 练习思考题 342

附录 A Eclipse 3.3 的下载、安装和运行 343

附录 B MyEclipse 7.0 的下载、安装和运行 346

附录 C MySQL 5 数据库下载、安装和运行 350

基础知识篇

第1章 Java 程序设计概述

本章旨在使读者对 Java 语言程序设计有一个初步的认识，为以后的学习打下基础。其内容主要包括 Java 语言的产生、背景、发展和主要应用方向，重点主要包括编程语言的发展、Java 语言的特点和应用方向、JDK 常用工具的说明、Java 开发环境的搭建、Java 程序的运行机制，同时举例演示了应用程序和小应用程序的编写步骤和运行方法。学习本章后，读者应能够熟练搭建基于 JDK 的 Java 开发环境，了解 JDK 中常用工具的用途，理解 Java 程序的开发流程和运行机制。

本章目标

- 学习 Java 语言的发展历史
- 了解 Java 语言的特点
- 熟悉典型的 Java 开发环境
- 亲自编写运行一个 Java 程序
- 了解 Java 开发中的基本知识：垃圾回收与析构、反编译
- 了解 JDK 1.6 的新功能

1.1 机器语言、汇编语言和高级语言

自 20 世纪 40 年代第一台计算机问世以来，计算机技术的发展已经走过了半个多世纪的历程。语言作为一种形式化的符号，可以使人们更加方便、简单地使用计算机解决一个问题。计算机程序设计语言的种类非常多，总的来说可以分成机器语言、汇编语言、高级语言 3 大类。

机器语言是计算机唯一能接受和执行的语言，与硬件紧密相连。机器语言由二进制码组成，每一串二进制码叫做一条指令。一条指令规定了计算机执行的一个动作。一台计算机所能懂得的指令的全体叫做这个计算机的指令系统。不同型号计算机的指令系统不同。指令种类包括数据传送指令、算术运算指令、位运算指令、程序流程控制指令、串操作指令、处理器控制指令等。

一条指令通常由两个部分组成：操作码 + 操作数。操作码规定了计算机要执行的基本操作；操作数规定了操作对象或操作对象的地址。

汇编语言（Assembly Language）是面向机器的程序设计语言。汇编语言是一种功能很强的程序设计语言，是利用计算机所有硬件特性直接控制硬件的语言。汇编语言作为一门语言，对应于高级语言的编译器需要一个“汇编器”来把汇编语言源文件汇编成机器可执行的代码。用汇编语言编写的程序叫汇编语言源程序，计算机无法执行，必须用汇编程序把它翻译成机器语言目标程序，计算机才能执行。这个翻译过程称为汇编过程。汇编程序的功能就是把由汇编语言编写的程序（称为汇编语言源程序）翻译成机器语言程序，计算机才能执行该程序，这个翻译过程称为汇编。高级的汇编器如 MASM、TASM

等为人们写汇编程序提供了很多类似于高级语言特征的功能,比如结构化、抽象化功能等,在这样的环境中编写的汇编程序,有很大一部分是面向汇编器的伪指令,已经类似于高级语言。汇编语言的长处在于编写高效且需要对机器硬件精确控制的程序。

在汇编语言中,用助记符(Mnemonic)代替操作码,用地址符(Symbol)或标号(Label)代替地址码。这样用符号代替机器语言的二进制码,就把机器语言变成了汇编语言。因此汇编语言也称为符号语言。

机器语言和汇编语言都是面向机器的,高级语言是面向用户的。

由于汇编语言依赖于硬件体系,且助记符量大、难记,于是人们又发明了更加易用的所谓高级语言。高级语言的语法和结构更类似于普通英文,而且由于远离对硬件的直接操作,使得一般人经过学习之后都可以编程。用高级语言编写的程序叫做高级语言源程序,必须翻译成机器语言目标程序才能被计算机执行。高级语言的翻译有两种方式:编译方式和解释方式。

编译方式:先由编译程序把高级语言源程序翻译成目标程序,执行时运行目标程序。现在大多数的编程语言都是编译型的,例如 Visual C++、Visual FoxPro、Delphi 等。

解释方式:在运行高级语言源程序时,由解释程序对源程序边翻译边执行。经典的 Basic 语言就是以解释方式运行的。

1.2 Java 语言概述

1.2.1 Java 语言的产生

任何事物的产生既有必然的原因也有偶然的因素,Java 语言的出现也验证了这一点。1991 年,美国 Sun Microsystems 公司的某个研究小组为了能够在消费电子产品上开发应用程序,开始积极寻找合适的编程语言。由于消费电子产品种类繁多,包括 PDA、机顶盒、手机等,所以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不相同,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, Sun 公司的研究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 Bill Joy 先生领导的研究小组就着手设计和开发出一种语言,称之为 Oak。该语言采用了许多 C 语言的语法,提高了安全性,并且是面向对象的语言,但是 Oak 语言在商业上并未获得成功。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、安全等特点非常符合互联网的需要,于是改进了该语言的设计,要达到以下几个目标。

- 创建一种面向对象的程序设计语言,而不是面向过程的语言。
- 提供一个解释执行的程序运行环境,使程序代码独立于平台。
- 综合 C 和 C++ 的优点,使程序员容易掌握。
- 去掉 C 和 C++ 中影响程序健壮性的部分,使程序更安全。例如指针、内存申请和释放。

- ❑ 实现多线程, 使得程序能够同时执行多个任务。
- ❑ 提供动态下载程序代码的机制。
- ❑ 提供代码校验机制以保证安全性。

最终, Sun 公司给该语言取名为 Java 语言, 造就了一代成功的编程语言。

1.2.2 Java 语言的发展

1995 年, 美国 Sun Microsystems 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进行运行。IBM、Oracle、微软、Netscape、Apple、SGI 等大公司纷纷与 Sun Microsystems 公司签订合同, 被授权使用 Java 平台技术。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。在国内, 根据权威部门统计, 截止到 2005 年, 我国对 Java 技术人员的需求量已高达 20 余万人, 许多大学纷纷开设 Java 课程, Java 已逐步成为世界上程序员使用最多的编程语言。

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, 即计算机的类型和操作系统是不一样的, 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 或者是 Linux, 因此相应的编程语言基本上只是适用于单机系统, 例如 COBOL、FORTRAN、C、C++等; 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移植, 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。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。

1.2.3 Java 虚拟机及 Java 的跨平台原理

Java 虚拟机 (Java Virtual Machine, JVM) 是一个想象中的机器, 在实际的计算机上通过软件模拟来实现。Java 虚拟机有自己想象中的硬件, 如处理器、堆栈、寄存器等, 还具有相应的指令系统。

Java 语言的一个非常重要的特点就是与平台的无关性, 而使用 Java 虚拟机是实现这一特点的关键。一般的高级语言如果要在不同的平台上运行, 至少需要编译成不同的目标代码; 而引入 Java 语言虚拟机后, Java 语言在不同平台上运行时不需要重新编译。Java 语言使用 Java 虚拟机屏蔽了与具体平台相关的信息, 使得 Java 语言编译程序只需生成能够在 Java 虚拟机上运行的目标代码 (字节码), 就可以在多种平台上不加修改地运行。Java 虚拟机在执行字节码时, 把字节码解释成具体平台上的机器指令执行。

Java 虚拟机是 Java 语言底层实现的基础, 对 Java 语言感兴趣的人都应对 Java 虚拟

机有个大概的了解，这有助于理解 Java 语言的一些性质，也有助于使用 Java 语言。

Java 虚拟机支持 Java 语言的基本数据类型如下。

- **byte** 1 字节有符号整数的补码。
- **short** 2 字节有符号整数的补码。
- **int** 4 字节有符号整数的补码。
- **long** 8 字节有符号整数的补码。
- **float** 4 字节 IEEE754 单精度浮点数。
- **double** 8 字节 IEEE754 双精度浮点数。
- **char** 2 字节无符号 Unicode 字符。

几乎所有的 Java 类型检查都是在编译时完成的。上面列出的原始数据类型数据在 Java 执行时不需要用硬件标记。操作这些原始数据类型数据的字节码（指令）本身就已经指出了操作数的数据类型，例如 `iadd`、`ladd`、`fadd` 和 `dadd` 指令都是把两个数相加，其操作数类型分别是 `int`、`long`、`float` 和 `double`。虚拟机没有给 `boolean`（布尔）类型设置单独的指令，`boolean` 型的数据是由 `Integer` 指令，包括 `Integer` 返回来处理；`boolean` 型的数组则是用 `byte` 数组来处理的。

虚拟机支持的其他数据类型包括 `Object`，对一个 `JavaObject`（对象）的 4 字节引用，Java 数组被当作 `Object` 处理。

从 Java 虚拟机支持的数据类型可以看出，Java 对数据类型的内部格式进行了严格规定，这样使得各种 Java 虚拟机的实现对数据的解释是相同的，从而保证了 Java 的平台无关性。

1.2.4 Java 语言的关键特点

Java 语言的特点很多，归纳起来可以体现在以下几个主要方面。

1. 简单性

Java 语言借鉴了 C++ 的成功，其大部分语法结构都与 C++ 相似，因此一般熟悉 C++ 语言的编程人员学习起来非常容易，同时 Java 所具有的自动内存管理机制也大大简化了 Java 程序设计开发。与 C++ 相比，Java 去掉了很容易混淆和较少使用的特性，增加了内存空间自动垃圾回收功能，使开发工作变得简单而可靠。

2. 平台无关性

Java 运行于虚拟机（JVM）上，Java 虚拟机建立在硬件和操作系统之上，用于实现 Java 字节码（Byte Code）的解释和执行等功能，为不同系统平台提供统一的 Java 接口。因此 Java 开发的程序可以运行于不同的系统平台。Java 字节码是一种近似于机器码的中间码，不受计算机硬件设备和操作平台种类的限制，只要计算机中有 Java 运行的环境，Java 字节码就可以在其上运行，这也正是 Java 最为突出的特征：一次编译，随处运行。配合严格规定的数据类型和类结构标准，使 Java 编写的程序具有非常好的可移植性。

3. 面向对象性

Java 是一种纯面向对象的程序语言，吸收了 C++ 面向对象的概念，将数据封装于类中，利用类的优点实现了程序的简洁和便于维护性。用 Java 语言设计程序时，通常以面向对象的方式来思考，程序员只需要把主要精力用在类和接口的设计及应用上即可，而不是专注于程序流程上。

Java 提供了众多的一般对象的类，通过继承即可使用父类的方法。与 C++ 不同的是，在 Java 中类的继承关系是非多重的，一个子类只有一个父类。

4. 安全性

Java 的设计目的是提供一个用于网络分布式的计算环境。因此 Java 语言特别强调安全性。

从内存管理的角度看，一方面 Java 不允许用户利用程序强制对内存进行存取操作，即对程序员来说内存分配是透明的，这种机制使内存出错的几率大大减小；另一方面 Java 编译器也没有对内存分配的決定权，而是通过系统所在的软硬件平台来决定的，Java 虚拟机在解释 Java 字节码的同时会对每段代码进行安全检测，对所有不合法的数据和无权限的存取操作加以阻隔，所以对于不合法的字节码是无法被解释和执行的。

5. 分布式

Java 建立在扩展 TCP/IP 网络平台上，库函数提供了用 HTTP 和 FTP 协议传送和接收信息的方法，这使得程序员使用网络上的文件和使用本机文件一样容易。

6. 多线程

Java 的多线程 (Multi Threading) 机制保证程序可以并行运行。多线程让程序同时拥有数个线程来运行不同的工作，比如当一个线程处理计算工作时，另一个线程可同时处理输出操作，而不需要采用全局的事件循环机制。使用 Java 开发出来的多线程程序比传统的单一线程程序所占用的 CPU 资源少。

7. 动态性

Java 的设计使其适应于不断发展的环境，在类库中可以自由地加入各种新的类和方法以适应新的环境要求，且不会影响用户程序的执行。

1.3 Java 平台和主要应用方向

从某种意义上来说，Java 不仅是编程语言，还是一个开发平台。Java 技术给程序员提供了许多工具：编译器、解释器、文档生成器和文件打包工具等，同时 Java 还是一个程序发布平台，有两种主要的发布环境，首先是 Java 运行时环境 (Java Runtime Environment, JRE) 包含了完整的类文件包；其次是许多主要的浏览器都提供了 Java 解释器和运行时环境。目前 Sun 公司把 Java 平台划分成 J2EE、J2SE、J2ME 共 3 个平台，

针对不同的市场目标和设备进行定位。

当前 Java 技术的平台架构包括以下 3 个方面。

- **J2SE (Java 2 Standard Edition)** 标准版, 主要目的是为台式机和 workstation 提供一个开发和运行的平台。包括标准的 Java 2 SDK、工具、运行时环境和 API, 用于那些使用 Java 程序设计语言编写、部署和运行 Applet 及应用程序的开发人员, 是桌面开发和低端商务应用的解决方案。
- **J2EE (Java 2 Enterprise Edition)** 企业版, 主要目的是为企业计算提供一个应用服务器的运行和开发平台。J2EE 本身是一个开放的标准, 任何软件厂商都可以推出自己符合 J2EE 标准的产品, 使用户可以有多种选择。J2EE 通过基于标准化模块组件和一整套的相关服务能够自动处理应用程序行为的许多细节, 从而大大简化了多层企业应用程序的构造和部署, 是以企业为环境而开发应用程序的解决方案。IBM、Oracle、BEA、HP 等 29 家企业已经推出了自己的产品, 其中以 BEA 公司的 WebLogic 产品和 IBM 公司的 WebSphere 最为著名。
- **J2ME (Java 2 Micro Edition)** 小型版, 主要面向消费电子产品, 为消费电子产品提供一个 Java 的运行平台, 使得 Java 程序能够在手机、机顶盒、PDA 等产品上运行包含高度优化的 Java 运行时环境, 致力于电子消费产品和嵌入式设备的解决方案。

Java 语言目前在服务器端确立了强大的战略优势, 同时由于其独有的特性, 在嵌入式系统方面的应用前景非常被看好, 未来的发展方向更是与互联网的发展需求紧密地联系在一起。目前 Java 已作为一门综合性技术在众多领域得到快速的发展和应用。使用 Java 开发的主要领域有以下几方面。

- Web 页面动态设计、网站管理和交互操作等基于互联网的应用。
- 嵌入式系统的开发与应用。
- 交互式、可视化图形软件的开发。
- 分布式计算系统的开发与应用。
- 电子商务系统的开发与应用。
- 多媒体系统的设计与实现。

1.4 Java 的垃圾回收与代码安全检查机制

在 Java 程序的执行过程中, 部分内存存在使用过后就处于废弃状态, 如果不及时进行无用内存的回收, 内存会越占越多, 导致内存泄漏, 进而使系统崩溃。在 C++ 语言中内存是由程序员人为进行回收的, 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, 但是这种人为管理内存释放的方法却往往由于程序员的疏忽而导致内存无法回收, 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, 避免了内存的泄漏, 也减轻了程序员的工作量。

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