

21世纪高等学校本科计算机专业系列实用教材

Java

程序设计实用教程

© 张永常 主编

(第2版)



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

21 世纪高等学校本科计算机专业系列实用教材

Java 程序设计实用教程

(第 2 版)

张永常 主编

杜明 杨兴运 副主编

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书主要讲解 Java 程序运行环境建设、Java 语言基础、数组和字符串、类和对象、包和接口、异常处理、Applet 小程序、图形用户界面应用、多线程、多媒体编程、网络编程、数据库编程、输入/输出流编程、Servlet 编程、JavaBean 编程等内容。另外，本书除了正文中的大量例题外，各章的最后附有综合应用举例、习题和实验，使读者的实践能力得到大幅度提高。

本书可作为高等院校学生学习 Java 的教材、Java 技术培训的教材和教学参考书，也可作为 Java 爱好者的学习参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Java 程序设计实用教程 / 张永常主编. —2 版. —北京: 电子工业出版社, 2010.1
(21 世纪高等学校本科计算机专业系列实用教材)
ISBN 978-7-121-09899-4

I. J… II. 张… 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 (2009) 第 210013 号

责任编辑: 刘海艳 (lhy@phei.com.cn)

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 22.5 字数: 605 千字

印 次: 2010 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 35.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

序 言

21 世纪是“信息”主导的世纪，是崇尚“创新与个性”发展的时代，体现“以人为本”、构建“和谐社会”是社会发展的主流。然而随着全球经济一体化进程的不断推进，市场与人才的竞争日趋激烈，对于国家倡导发展的 IT 产业，需要培养大量的、适应经济和科技发展的计算机人才。

众所周知，近年来，一些用人单位对部分大学毕业生到了工作岗位后，需要 1~2 年甚至多年的训练才能胜任工作的“半成品”现象反应强烈。从中反映出单位对人才的需求越来越讲求实用，社会要求学校培养学生的标准应该和社会实际需求的标准相统一。对于 IT 业界来讲，一方面需要一定的科研创新型人才，从事高端的技术研究，占领技术发展的高地；另一方面，更需要计算机工程应用、技术应用及各类服务实施人才，这些人才可统称“应用型”人才。

应用型本科教育，简单地讲就是培养高层次应用型人才的本科教育。其培养目标应是面向社会的高新技术产业，培养在工业、工程领域的生产、建设、管理、服务等第一线岗位，直接从事解决实际问题、维持工作正常运行的高等技术应用型人才。这种人才，一方面掌握某一技术学科的基本知识和基本技能，另一方面又具有较强的解决实际问题的基本能力，他们常常是复合性、综合性人才，受过较为完整的、系统的、有行业应用背景的“职业”项目训练，其最大的特色就是有较强的专业理论基础支撑，能快速地适应职业岗位并发挥作用。因此，可以说“应用型人才培养既有本科人才培养的一般要求，又有强化岗位能力的内涵，它是在本科基础之上的以‘工程师’层次培养为主的人才培养体系”，人才培养模式必须吸取一般本科教育和职业教育的长处，兼容并蓄。“计算机科学与技术”专业教学指导委员会已经在研究并指导实施计算机人才的“分类”培养，这需要我们转变传统的教育模式和教学方法，明确人才培养目标，构建课程体系，在保证“基础的前提”下，重视素质的养成，突出“工程性”、“技术应用性”、“适应性”概念，突出知识的应用能力、专业技术应用能力、工程实践能力、组织协调能力、创新能力和创业精神，较好地体现与实施人才培养过程的“传授知识，训练能力，培养素质”三者的有机统一。

在规划本套教材的编写时，我们遵循专业教学委员会的要求，针对“计算机工程”、“软件工程”、“信息技术”专业方向，以课群为单位选择部分主要课程，以计算机应用型人才培养为宗旨，确定编写体系，并提出以下的编写原则。

(1) 本科平台：必须遵循专业基本规范，按照“计算机科学与技术”专业教学指导委员会的要求构建课程体系，覆盖课程教学知识点。

(2) 工程理念：在教材体系编写时，要贯穿“系统”、“规范”、“项目”、“协作”等工程理念，内容取舍上以“工程背景”、“项目应用”为原则，尽量增加一些实例教学。

(3) 能力强化：教学内容的举例，结合实际应用，力争有针对性；每本教材要安排课程

实践教学指导，在课程实践环节的安排上，要统筹考虑，提供面向现场的设计性、综合性的实践教学指导内容。

(4) 国际视野：本套教材的编写要做到兼长并蓄，吸收国内、国外优秀教材的特点，人才培养要有国际背景和视野。

本套教材的编委会成员及每本教材的主编都有着丰富的教学经验，从事过相关工程项目（软件开发）的规划、组织与实施，希望本套教材的出版能为我国的计算机应用型人才的培养尽一点微薄之力。

编委会

前 言

天基人才网 2008 年 9 月 23 日消息：“由天基人才网的统计数据显示，9 月中原才市中三类岗位需求加大，增量急剧上升，……，Java 开发人员在技术岗位中排名第一，……。目前招聘市场中一些软件开发和信息服务公司对 Java 人才需求很大。其中设置的岗位有测试工程师、Java 开发工程师、高级程序员、系统架构工程师等。从薪酬情况来看，Java 人才薪资 2000~8000 元不等。根据企业的不同规模及不同性质，支付能力差异明显，但 Java 人才前景一片看好。”请注意，这是在发生了金融危机的 2008 年下半年的统计数据。

由于 Java 的跨平台特性比较突出，企业对 Java 及 JavaEE 开发有着巨大的需求，随着高等教育大众化，就业竞争的形势日趋激烈，计算机专业学生学习 Java 语言的热情也随之不断高涨。有些学校原来把 Java 程序设计这门课程放在专业选修课中让学生选修，但几乎所有的学生都选修，成为事实上的必修课，所以学校也就适应市场需求，把这门课程改为必修课程。

大家都注意到，教育部考试中心已经把“Java 语言程序设计”列入了全国计算机等级考试项目。全国计算机等级考试（National Computer Rank Examination, NCRE），是经原国家教育委员会（现教育部）批准，由教育部考试中心主办，面向社会，用于考查应试人员计算机应用知识与技能的全国性计算机水平考试体系，把“Java 语言程序设计”列入全国计算机等级考试，从另一方面说明了 Java 课程在学生就业方面的重要作用。

由于高等教育大众化和市场经济对人才需求的不断深化，企业往往对求职者——高校的毕业生提出了直接上岗的能力要求，但是，刚刚从高校毕业的学生往往缺乏工作经历，而企业招聘人才时对 Java 程序员的需求却是巨大的，这就催生了一些培训企业，这些企业专门做高校毕业生毕业后到企业上岗前的这一段空间的培训工作。这些培训的企业开发了一些成功的项目案例，让刚刚毕业或尚未毕业的学生真刀真枪地进行项目训练，在这些企业培训过后，受训者的项目能力得到了较大幅度地提高，用人单位就愿意接收他们。这些培训企业往往对毕业生提出应该有 Java 基础知识和基本实践能力的要求。普通高校培养出来的大学生带有的青涩，经过培训企业的催熟，使得学生就业变得顺利起来。

本教程的第 1 版于 2006 年出版，最近三年 Java 技术又有了新的发展，但是，作为基础课程的 Java 程序设计教程，老师们认为仍然要突出其基础性和实用性。因此，我们在编写第 2 版时，注重了在保持该课程的基础地位，对后续课程有良好的支撑作用，强化该课程培养学生实践能力的特色，适当兼顾 Java 新版本中新特性的应用。

在编写本教程之前，我们调研并总结了第 1 版教材的使用情况，修订后的第 2 版主要有以下特色：仍然秉承第 1 版的体系结构，但是，对部分章节适当扩充；更正了第 1 版中存在的不足；在各章后面增加了综合应用举例、课后实践的习题，实训的习题、实验题增加到了 190 个，较大幅度地丰富了学生的实践项目内容；添加了部分 Java 新的特性应用；增加了习题的参考答案。

本教程中用到的全部程序代码已在 JDK1.6.13 环境中调试运行通过。虽然我们提供了本教程的所有例题、习题的源代码电子稿，但是我们仍然建议初学 Java 语言的读者逐个输入各程序的源代码。这样做的最大优点是，可以培养严谨的学风，熟悉 Java 的程序结构、便于记忆和熟练地运用 Java 中的类、方法等，同时输入和调试的过程也是学习的过程。

读者若使用本教程的源程序、PPT 时，可与电子工业出版社刘海艳编辑 (E-mail: lhy@phei.com.cn) 联系索取或登录到出版社的网站下载。

本教程由张永常任主编，并完成全部书稿的统稿工作。其中第 1~4 章由张永常编写，第 5~8 章由费贤举编写，第 9~12 章由杜明编写，第 13~15 章由杨兴运编写，张旭隆、曹言敬完成了编写过程中程序的调试和校对工作。

本教程的编写工作是在徐州师范大学、常州工学院、徐州工程学院的领导和同事们的大力支持、鼓励与帮助下完成的，向他们表示衷心的感谢！

本教程第 2 版的编写还要感谢使用第 1 版教程的老师和读者们给我们提出的宝贵意见和建议！

再次感谢全体作者的家庭成员给予的支持。

由于作者教学任务繁重且水平有限，加之时间紧迫，对于书中存在的错误和不妥之处，诚挚欢迎批评指正，作者联系邮箱：yczhang@xznu.edu.cn。

张永常

目 录

第 1 章 绪论	1	2.5.3 Java 缩进排版规范	41
1.1 Java 发展简史	1	2.5.4 Java 文件名规范	41
1.2 为什么要学习 Java	2	2.5.5 Java 声明规范	42
1.2.1 高层评价	2	2.5.6 Java 语句规范	42
1.2.2 主要特点	3	2.5.7 Java 编程规范	42
1.2.3 Java 运行机制	6	2.6 如何获得 Java 帮助	42
1.2.4 应用前景	7	2.6.1 下载帮助	42
1.3 面向对象程序设计的主要概念与举例 ..	8	2.6.2 使用帮助	44
1.3.1 对象、抽象和类	8	2.6.3 Java 资源	44
1.3.2 封装	9	本章小结	44
1.3.3 继承	9	习题 2	44
1.3.4 多态性	9	实验 1 Java 环境搭建与简单编程	45
1.4 Java 与 C/C++ 的简单比较	10	第 3 章 Java 语言基础	46
1.4.1 C 语言	10	3.1 关键字	46
1.4.2 C++ 语言	10	3.2 基本数据类型	47
1.4.3 Java 语言	12	3.2.1 整数类型 (int、short、long、 byte)	47
本章小结	14	3.2.2 浮点类型 (float、double) ..	47
习题 1	14	3.2.3 布尔类型 (boolean)	48
第 2 章 Java 程序运行环境建设	15	3.2.4 字符类型 (char)	48
2.1 Java 运行环境的安装	15	3.3 常量和变量	50
2.1.1 JDK 的下载	15	3.3.1 常量	50
2.1.2 JDK 的安装	16	3.3.2 变量	51
2.1.3 设置 Java 运行环境	19	3.4 运算符与表达式	52
2.2 如何编写并运行 Java 程序	21	3.4.1 赋值运算符和语句	52
2.2.1 Java Application 程序编辑、 编译与运行举例	21	3.4.2 算术运算符和算术表达式	53
2.2.2 Java Applet 程序编辑、编译 与运行举例	23	3.4.3 关系运算符和关系表达式	54
3.4.4 逻辑运算符和逻辑表达式	55		
2.3 Java 集成开发环境简介	26	3.4.5 位运算符	56
2.3.1 NetBeans	27	3.4.6 条件赋值运算符	57
2.3.2 Eclipse	30	3.4.7 广义赋值运算符	58
2.4 Java 程序的基本输入与输出	34	3.4.8 对象运算符	58
2.4.1 输入流和输出流概述	34	3.4.9 括号与方括号运算符	58
2.4.2 简单输入与输出举例	35	3.4.10 运算符的优先级	58
2.5 Java 编码规范简介	40	3.5 类型转换	59
2.5.1 Java 命名规范	40	3.5.1 自动转换	59
2.5.2 Java 注释规范	41	3.5.2 强制转换	60

3.6 流程控制	61	5.4.1 static	123
3.6.1 选择语句	61	5.4.2 final	124
3.6.2 循环语句	64	5.4.3 volatile、native、synchronized	125
3.6.3 转移语句	67	5.5 Java 的名字空间和包	125
3.7 综合应用举例	68	5.5.1 Java 的名字空间	125
本章小结	72	5.5.2 Java 提供的包	126
习题 3	72	5.5.3 自定义包	126
实验 2 Java 语言基础	73	5.5.4 包的引用	127
第 4 章 数组和字符串	75	5.6 接口	129
4.1 数组	75	5.6.1 接口的声明	129
4.1.1 一维数组	75	5.6.2 接口的实现	130
4.1.2 多维数组	79	5.6.3 接口的使用	131
4.1.3 复制数组的方法	83	5.6.4 扩展接口与接口的多重继承	132
4.1.4 数组元素的排序方法	84	5.7 综合应用举例	134
4.1.5 数组元素的查找方法	86	5.7.1 案例 1: 通用加密类 Clipher	
4.1.6 数组元素的填充方法	87	的设计	134
4.2 字符串	88	5.7.2 案例 2: 多态与继承综合应用	
4.2.1 字符串的声明与创建	88	举例	136
4.2.2 与字符串有关的方法	89	本章小结	140
4.2.3 StringBuffer 中有关的方法	93	习题 5	141
4.3 向量*	98	实验 4 Java 面向对象程序设计	142
4.3.1 向量的声明	98	第 6 章 Java 的异常处理	143
4.3.2 Vector 类的方法	98	6.1 异常的概念	143
4.4 综合应用举例	100	6.2 异常处理	144
本章小结	103	6.2.1 Java 中的 Throwable 类	145
习题 4	104	6.2.2 try-catch-finally 语句	147
实验 3 数组和字符串编程	105	6.2.3 throw 和 throws 语句	149
第 5 章 Java 面向对象程序设计	106	6.3 自定义异常处理类	151
5.1 类	106	6.4 综合应用举例	154
5.1.1 类的定义	106	本章小结	156
5.1.2 类的使用	110	习题 6	157
5.1.3 方法重载	114	实验 5 Java 异常处理	158
5.1.4 内部类	116	第 7 章 Applet 小程序	160
5.2 访问控制符和封装	116	7.1 Applet 概述	160
5.3 继承	117	7.1.1 Applet 小程序简介	160
5.3.1 子类 and 简单继承举例	117	7.1.2 Applet 安全性及与 Application	
5.3.2 变量隐藏和方法覆盖	118	的区别	161
5.3.3 super 和 this	119	7.1.3 Applet 类及 Applet 的生命	
5.3.4 运行时多态与多重继承	121	周期	162
5.3.5 对象类型转换	122	7.2 在 HTML 中嵌入 Applet 程序	164
5.3.6 抽象类	123	7.2.1 HTML 代码的基本结构	164
5.4 非访问控制符	123	7.2.2 Applet 标记	165

7.2.3	在 HTML 中传递 Applet 程序	166	9.1.2	多线程简介	212
	使用的参数	166	9.1.3	Java 中的多线程机制	213
7.3	Applet 应用	168	9.2	线程类及其线程创建	214
7.3.1	Applet 的字体和颜色	168	9.2.1	线程类	214
7.3.2	Applet 的应用	169	9.2.2	线程的创建	215
	本章小结	174	9.2.3	线程的状态与控制	218
	习题 7	174	9.3	线程的同步	221
	实验 6 Applet 小程序	174	9.3.1	线程同步的概念	221
第 8 章	图形用户界面应用	176	9.3.2	方法同步	221
8.1	AWT 和 Swing 简介	176	9.3.3	对象同步	223
8.1.1	组件和容器	176	9.3.4	同步方法的缺点	224
8.1.2	Swing 组件及其之间的		9.4	线程组	224
	层次关系	178	9.4.1	线程组简介	224
8.1.3	组件类	179	9.4.2	ThreadGroup 类	225
8.2	布局管理	180	9.4.3	线程之间的通信	227
8.2.1	顺序布局	180	9.5	综合应用举例	229
8.2.2	边界布局	181		本章小结	234
8.2.3	卡片布局	182		习题 9	234
8.2.4	网格布局	183		实验 8 多线程	236
8.2.5	面板的使用	184	第 10 章	多媒体编程	238
8.3	事件处理	185	10.1	检索与使用声音文件	238
8.3.1	事件与事件处理方法	185	10.1.1	简单声音文件播放	238
8.3.2	事件处理机制	186	10.1.2	AudioClip 接口	239
8.3.3	事件处理举例	189	10.1.3	把声音加载到 AudioClip	
8.3.4	事件适配器类	192		对象中	239
8.4	常用组件的应用	193	10.2	图形与动画	241
8.4.1	标签	193	10.2.1	图形与文本	241
8.4.2	按钮	194	10.2.2	图像	243
8.4.3	列表框	195	10.2.3	动画	246
8.4.4	选项框	196	10.2.4	双缓冲	250
8.4.5	复选框	196	10.3	综合应用举例	252
8.4.6	文本框	198		本章小结	258
8.4.7	文本区	198		习题 10	258
8.4.8	菜单	199		实验 9 多媒体编程	259
8.4.9	窗口与对话框	201	第 11 章	网络编程	260
8.4.10	综合实例	207	11.1	网络编程的基本概念	260
	本章小结	210	11.1.1	TCP/IP 协议概述	260
	习题 8	210	11.1.2	Socket 套接字	261
	实验 7 图形用户界面	210	11.1.3	端口	262
第 9 章	多线程	212	11.1.4	URL	262
9.1	多线程的概念	212	11.2	使用 URL 获取网络资源	263
9.1.1	进程和线程	212	11.2.1	URL 类	263

11.2.2	获取图像	264	13.1.2	文件的处理	311
11.2.3	获取声音	267	13.2	Reader 与 Writer 类应用举例	312
11.2.4	获取文档	268	13.2.1	Reader 和 Writer	313
11.3	使用 TCP 协议的 Socket 编程	270	13.2.2	InputStreamReader 和 OutputStreamWriter	313
11.3.1	Socket 套接字	270	13.3	InputStream 与 OutputStream 类应用 举例	315
11.3.2	一对一的 Socket C/S 通信	271	13.3.1	InputStream 类	315
11.3.3	TCP 协议通信的实现	272	13.3.2	OutputStream 类	316
11.3.4	一对多的 Socket C/S 通信	274	13.4	数据流应用举例	317
11.3.5	一对多通信的实现	274	13.5	对象流应用举例	319
11.4	综合应用举例	276	本章小结	320	
	本章小结	282	习题 13	320	
	习题 11	282	实验 12 输入与输出	321	
	实验 10 网络编程	283	第 14 章 Servlet 编程	322	
第 12 章 数据库编程		284	14.1 Servlet 简介	322	
12.1	JDBC 技术简介	284	14.1.1 Servlet 概述	322	
12.1.1	JDBC 概述	284	14.1.2 Servlet 开发环境搭建	322	
12.1.2	JDBC 的基本功能	285	14.1.3 Servlet 的工作原理与生命 周期	325	
12.2	JDBC 接口与驱动程序	285	14.2 Servlet 编程示例	327	
12.2.1	JDBC API	286	本章小结	331	
12.2.2	JDBC Drive API	287	习题 14	331	
12.3	JDBC 在数据库访问中的应用	288	实验 13 Servlet 编程	331	
12.3.1	建立与数据库的连接	288	第 15 章 JavaBeans 编程	333	
12.3.2	执行查询语句	288	15.1 JavaBeans 简介	333	
12.3.3	对数据库进行更新操作	290	15.1.1 JavaBeans 概述	333	
12.3.4	事务	291	15.1.2 JavaBeans 的组件模型	333	
12.4	JDBC 综合应用示例	291	15.1.3 JavaBeans API 应用简介	334	
12.4.1	建立数据库	292	15.2 JavaBeans 的组件构成	336	
12.4.2	建立数据源	292	15.2.1 JavaBeans 组件的创建	336	
12.4.3	数据库操作程序	294	15.2.2 开发工具中的 JavaBeans	338	
12.5	综合应用举例	298	15.3 JavaBeans 组件的应用示例	338	
	本章小结	306	本章小结	344	
	习题 12	306	习题 15	345	
	实验 11 数据库编程	307	实验 14 JavaBeans 编程	345	
第 13 章 输入/输出流编程		308	参考文献	347	
13.1	File 类应用举例	308			
13.1.1	File 类用来获取文件或目录 的信息	308			

第1章

绪论

本章要点

- ◆ 了解 Java 的发展简史;
- ◆ 理解 Java 对软件开发技术的影响;
- ◆ 了解 Java 的应用前景及特点;
- ◆ 通过 Java 与 C/C++ 的比较, 加深理解面向对象程序设计的基本概念;
- ◆ 为本课程后续的学习打下思想和认识的基础。

1.1 Java 发展简史

1991 年, Sun MicroSystem 公司决定由 James Gosling、Bill Joe 等人组成一个开发小组, 开发在 PDA (Personal Digital Assistant, 个人数字助理)、机顶盒、手机、烤面包机等家用消费类电子产品上运行的交互式系统。这就需要开发一种小型的计算机语言, 用于上述各种消费类设备, 因为这些设备是由不同的厂商生产的, 完全可能选用不同的 CPU, 它们的功能和内存均不足以运行大程序。此外, 还要求这种语言不能受限于任何一种独立的体系结构, 因为用这种语言开发的系统需要运行在种类非常繁杂的各个厂家的设备上, 完成各种电子设备之间的通信与协同工作, 即要求这个系统必须具有跨平台的性能。

刚开始时, 他们试图在 C++ 语言的基础上, 将其做功能上的修改, 但是, 在开发的过程中他们发现, 对于消费类电子产品来说, C++ 语言过于庞大和复杂, 一直难以克服出现在编译器上的难题, 加上安全性问题也不能令人满意, 无奈之下, 只好另起炉灶开发了一个全新的程序设计语言。这种语言吸收了 C/C++ 语言的优点, 抛弃了 C/C++ 语言的不足, 并且是完全面向对象的程序设计语言。James Gosling 在制定这个全新的程序设计语言的树状结构和为该语言起名字时, 看到办公室外高大的橡树枝繁叶茂, 就将该语言命名为 Oak (是一种橡树的名称), 预示着这种语言将会有广泛的用途和良好的发展势头。

由于 Sun MicroSystem 公司在一次争取电视机上的计算机控制器的项目中失利, 造成开发小组被解散, 只剩下 James Gosling 等几个人继续研究如何将这种计算机技术应用到多媒体上, 以至于在 Oak 早期的发展过程中, 并没有引起人们对它的注意。当 Sun MicroSystem 公司把这种程序设计语言开发成功要为其注册时, 才发现 Oak 是一种已经被别人注册过的计算机语言, 他们只好再为其起别的名字。当他们看到办公桌上的咖啡后, 灵感使他们把该语言命名为太平洋上一个盛产味道非常美妙的咖啡的岛屿名字——Java, 向学习或应用该语言的人预示: 应用 Java 开发语言环境时就像喝咖啡那样惬意。

1994年下半年, Internet 得到了迅猛的发展, 看到 WWW 的快速增长, James Gosling 等人马上意识到如果把 Java 语言应用到 Web 上, 将会大大促进 Java 语言研制的进展。这个设想付诸实施后, 使得 Java 语言成为 Internet 上越来越受欢迎的开发与编程语言, 一些著名的计算机公司纷纷购买了 Java 语言的使用权, 如 Microsoft、IBM、Netscape、Novell、Apple、DEC、SGI 等。1995年5月23日召开的 SunWorld'95 大会上, 一种全新的浏览器(今天的 HotJava 的前身)亮相, 标志着 Java 的诞生, 即 Java 1.0 版正式发布了。

Sun 公司于 1996 年初发布了 Java 1.02, 遗憾的是, Java 1.02 还未摆脱其小型语言的影子, 只适合用来做诸如网页上一个随机移动的文字之类的工作, 并不适合于正规的程序开发。这个时候, 可以说, Java 1.02 作为一种正规的编程语言, 准备得很不充分。

1998 年 12 月, Java 1.2 问世了。它是一个功能全面的、具有高度扩展能力的新版本。Java 1.2 很快被改进成 Java 2, 向“编写一次, 到处运行 (Write once, run anywhere)”的目标前进了一大步。

在美国, 1999 年对 Java 程序员的需求量首次超过对 C++ 程序员的需求量。现在包括我国国内的许多企业对使用 Java 作为开发平台有了明确要求, 这已经确凿地表明了 Java 语言在全世界范围内正如火如荼地流行, 并被广泛地应用在 Internet 的数据库、多媒体、CGI 及动态网页的制作方面。

1.2 为什么要学习 Java

现在, 许多企业明确提出开发时必须使用 Java 技术, 因此, 相关软件企业招聘员工时也有了明确要求。例如, 某公司招聘 Java 工程师负责金融管理信息系统、CRM 系统的开发。要求: 计算机或相关专业毕业; 掌握 Java 技术, 有 2~3 年的 Java 编程经验; 有 DB2、SQL Server、Oracle 等一种或多种数据库的开发经验。《2008—2009 年中国 Java/Brew 业务发展研究年度报告》指出, 随着手机应用软件与智能手机的普及, 由电信运营商提供的 Java/Brew 增值业务发展迅速; 市场的迅速拓展又刺激了 SP、CP 等产业链上游企业在 Java/Brew 平台上加大投入。2008 年中低端手机 Java/Brew 平台的普及更刺激了这一市场的迅速发展, 目前 Java/Brew 平台上的软件应有已不仅仅局限于手机游戏, 手机地图、手机证券、手机银行等应用层出不穷。作为与 3G 业务联系最为紧密的 Java/Brew 业务在市场的迅速发展下, 已呈现出勃勃生机。

应该认识到, Java 不仅是一种程序设计语言, 更重要的它是一种技术、一种开发环境, 所以, 学习 Java 就是在学习一门技术。Java 也不只是在大学教学计划中要求学习或自己没有事情时随便学习的一门课程, 而更多地应认识到学习 Java 与自己的就业需要密切相关。

1.2.1 高层评价

Java 语言被美国的著名杂志 PC Magazine 评为 1995 年十大优秀科技产品, 要知道, 在所有的计算机类产品中就此一项入选。

微软公司 (Microsoft Co.) 总裁比尔·盖茨 (Bill Gates) 在悄悄地观察了一段时间后, 不无感慨地说: “Java 是长时间以来最卓越的程序设计语言”, “Java 是用来推翻我们以前所创建的东西的一种语言”。并确定微软整个软件开发的战略从 PC 单机时代向着以网络为中心的计算机时代转移, 而购买 Java 则是他重大战略决策实施的主要步骤。因此, Java 的诞生必将对整个计算机产业产生深远的影响, 对传统的计算模型提出了新的挑战。

Sun MicroSystem 公司的总裁 Scott McNealy 认为 Java 为 Internet 和 WWW 开辟了一个崭新的时代。

万维网 WWW 的创始人 Berners-Lee 说：“计算机事业发展的下一个浪潮就是 Java，并且将很快会发生的。”

在麻省理工学院，计算机科学系早在 1997 年就用 Java 取代了 C++ 作为学生必须掌握的主要软件开发语言。2002 年春季，加州大学伯克利分校就开设了 25 门 Java 课程，而针对微软编程语言的课程只有 7 门。当时麻省理工学院的副教授 Daniel Jackson 说：“照我看，Java 将很快取代其他所有的编程语言，尤其是 C++。”这位学者提醒大家注意，大学生甚至在上 Java 课程之前就在逐渐地自学 Java 了。在两年的时间里，学院将把针对高中生开设的大学级学分课程的高级测验部分由 C++ 转向 Java。当然，Daniel Jackson 这个预言到现在并没有完全实现，但是这说明了人们对 Java 的期待。

根据 2000 年度 IDC 研究报告，当时有 300 万人在使用 C 和 C++ 作为自己的主要编程语言，Visual Basic 开发人员有 230 万，Java 开发人员则只有 120 万。而市场研究公司 Evans Data 则报告说，使用 Java 的人越来越多，相比之下 C 和 C++ 开发人员的数量则在显著减少。

1.2.2 主要特点

1. 语言特点

(1) 简单性

Java 自身小巧玲珑，基本解释程序、类库和多线程支持，对硬件的要求很低，这也体现了其简单性。JDK 6 Update 10 以来的版本 Java 内核大大缩小了，由原来的大约十几 MB 缩小到 4MB，同时也提高了 Java 的启动速度，而其他的 Java 库在需要的时候可以后台下载，这样也缩短了等待和安装的时间。

Java 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。在 Java 中略去了运算符重载、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。

由于 Java 在语法上与 C++ 类似，对于有过 C/C++ 编程经验的程序员可以很容易地学会 Java 语法。而对于没有编程基础的读者来说，应该正确理解 Java 的简单性，只要树立信心，也会很快学会 Java 编程。当然，本人建议读者要把“不经学习就可掌握”（例如××三日通、××21日通等）的美丽说法理解成容易学会就行了。

(2) 面向对象

许多读者学习或使用过 C++，甚至把 C++ 作为面向对象程序设计标准，若用 Java 与 C++ 相比，Java 的面向对象技术更加彻底，这是因为 Java 中所有内容都封装在类中，即以类作为程序的基本单位，所以 Java 是一种纯面向对象的程序设计语言。

C++ 为了向下兼容 C，不得不保留了很多 C 语言的特性，众所周知 C 语言是面向过程的语言，实际上 C++ 成了一个“混血儿”语言。

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，子类可以使用父类所提供的方法，实现了代码的复用。

(3) 分布性

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。

分布式包括数据分布和操作分布。数据分布是指数据库可以分散存放于网络上的不同主机上。操作分布则是把计算工作分散到不同主机,由不同的主机进行计算处理。Java 支持 WWW 客户机/服务器计算模式,因此,它可以支持这两种分布性。

(4) 鲁棒性(也称健壮性)

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。它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出现容易产生的错误。通过集成的面向对象的异常(例外)处理机制,在编译时,Java 提示出可能出现但未被处理的异常(例外)。另外,Java 在编译时还可捕获类型声明中的许多常见错误,防止动态运行时出现不匹配问题。

Java 与 C/C++ 的最大区别是,采取了一个安全的指针模型,能降低重写内容和崩溃数据的可能性。

Java 在语法中取消了 C++ 中说起来很强大、实际上很不安全的特性,如取消了头文件、指针算法、结构、单元等。C/C++ 最关键的一个特性是强大的指针算法,例如可以用指针访问字符串、数组甚至文件,但这样很容易产生指针溢出,意外地改写内存和损毁数据。Java 取消了对指针的支持,使得对内容的非法访问被杜绝了。Java 中有一个指针模型,对于普通的任务,如访问字符串、数组,根本不用指针,而一旦需要,如访问对象、文件时,也可以获得指针的强大能力;但 Java 同时又有高度的指针安全保障,用户永远不可能访问一个坏掉的指针、不可能造成内存分配错误,也用不着专门提防可能出现的内存漏洞。

自动垃圾回收机制是 Java 的又一个特色,这种机制防止了内容丢失等动态内存分配导致的问题。

Java 提供了较完善的异常(例外)处理机制,程序员可以把一组可能产生运行异常的代码放在异常处理结构中,这样大大简化了异常处理过程,也使程序更加健壮。

在 Java 中,取消了 C++ 的一些复杂而低效的特性,例如,用接口技术代替了 C++ 的多重继承。C++ 中,一个类允许有多个超类,这个特性叫做“多重继承”,多重继承使得编译器非常复杂且效率不高;Java 的类只允许有一个超类,而用接口(Interface)技术实现与 C++ 的多重继承相类似的功能。其他被取消的特性包括虚拟基础类、运算符重载等。例如,在 C/C++ 语言中,if(x=3) 是合法的判断语句结构;但在 Java 语言中,将这里可能隐含的错误进行了改进,只允许 if(x==3) 这样的判断语句存在。

(5) 安全性

用于网络、分布环境下的程序必须要防止非法的入侵。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,这样就防止程序员使用“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。Java 通过自己的安全机制预防了病毒下载程序对本地系统的破坏和威胁。

(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 程序就可以在任意的处理器上运行。Java 虚拟机能够识别这些字节码指令,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上运行。

(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。另外,Java 编译器是由 Java 语言实现的,Java 运行时系统由标准 C 实现,这就是说,Java 系统本身也具有可移植性。

例如, Java 编译后并不生成可执行文件 (exe 文件), 而是生成一种中间字节码文件 (class 文件)。任何操作系统, 只要装有 Java 虚拟机 (JVM), 就可以解释并执行这个中间字节码文件。这正是 Java 实现可移植的机制。

在 Java 中的原始数据类型存储方法是固定的, 避开了移植时可能产生的问题。例如, 在任何计算机上, 整型都是 32 位, 而 C++ 中整型的存储依赖于目标计算机, 对 16 位处理器 (例如 8086), 整数用两个字节表示; 在像 Sun SPARC 这样的 32 位处理器中, 整数用 4 个字节表示。在 Intel Pentium 处理器上, 整数类型由具体的操作系统决定: 对于 DOS 和 Win32 来说, 整数用两个字节表示; 对于 Windows 9x、Windows NT 和 Windows 2000, 整数用 4 个字节表示。当然, 使整数类型平台无关之后, 性能必然有所下降, 但就 Java 来说, 这个代价是值得的。Java 的字符串则采用标准的 Unicode 格式保存。可以说, 没有这个特性, Java 的可移植性也不可能实现。

(8) 解释型

Java 解释器直接对 Java 字节码进行解释执行。字节码本身携带了许多编译时的信息, 使得连接过程更加简单。Java 语言的程序可以在提供 Java 语言解释器和实时运行系统的任意环境中运行。

由于 Java 是采用解释型语言来执行的, 所以在单机上运行时速度较慢。据某个统计, 用 Java 语言编写的程序的运行速度要比其他语言 (例如 C 语言等) 低 20 倍, 但是在 Internet 上运行, Java 就有了用武之地, 其速度并不慢。

(9) 高性能

和其他解释执行的语言 (例如 BASIC) 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。用 Java 编写的程序在网络上运行时, 其速度要比 C/C++ 编写的程序的运行速度快得多。

(10) 多线程

多线程技术允许同一个程序中有两个以上的执行线路, 即同时做两件事情, 这样可以满足一些复杂软件的需要。多线程是当今软件技术领域里的又一种成果, 该技术已经成功地运用到操作系统、应用开发等多种领域。

在 Java 中内置了对多线程的支持, 在多线程模型中, 多个线程共存于同一块内存中, 且共享资源。这样, 使用多线程机制提高了程序性能, 每个线程分配有限的时间片来处理任务, 由于 CPU 在各个线程之间的切换速度非常快, 用户感觉不到, 从而认为并行运行。

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, 这样就很容易地实现网络上的实时交互行为。

(11) 动态性

Java 自身的设计使它比 C/C++ 更具有动态性, 从而更加适合于一个不断发展的环境。在 Java 类库中可以自由地加入新的方法和实例变量, 而不会影响用户应用程序的执行。另外, 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。

2. Java Applet

Java 语言的特性使它可以最大限度地利用网络。

在 Java 中, Applet 被称为小应用程序, 它是动态、安全、跨平台的网络应用程序。Java Applet 是一类特殊的 Java 程序, 它被嵌入在 HTML 语言中, 通过主页发布到 Internet 上。利

用 Applet, 用户可以非常简单地实现 Internet 上的程序编写, 实现多媒体的用户界面或进行复杂的计算。Java 程序的简单性使其与其他语言相比大大缩短了应用程序的下载时间。

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, 但它并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。

3. 丰富的类库

Java 程序是由类构成的, 学习 Java 就要学会使用类来解决问题, 但是如果所有的类都要学习者自己来完成, 问题就变得太复杂、难度太大, 因此, Sun 公司为 Java 提供了丰富的支持类库。也就是说, 并不需要学习者编写所有的类, 因为 Sun 公司提供了许多常用的类。这些类库存放在 Java 系统的各种包 (Package) 中, 其中有接口、类、异常 (例外) 处理等内容。

在 Java 语言包 (java.lang) 中, 提供利用 Java 编程语言进行程序设计的包括字符串处理、多线程处理、数学函数处理等基础类, 可以用它简单地实现 Java 程序的运行平台。

在 Java 的实用程序包 (java.util) 中, 提供的支持包括哈希表、堆栈、可变数组、时间和日期、集合框架、事件模型、国际化及各种实用工具类 (字符串标记生成器、随机数生成器和位数组)。

在 Java 的输入/输出包 (java.io) 中, 提供了用统一的“流”模型来实现所有格式的 I/O, 包括文件系统、网络输入。通过数据流、序列化和文件系统提供系统的输入和输出。

在 Java 的网络包 (java.net) 中, 提供了用于实现网络编程的 URL、Socket 编程。网络包支持 Internet 的 TCP/IP 协议, 提供了与 Internet 的接口。它支持 URL 连接、WWW 的即时访问, 并且简化了客户机/服务器模型的程序设计。

在 Java 的抽象图形用户接口包 (java.awt 和 javax.swing) 中, 提供了实现不同平台的计算机图形用户接口部件, 包括窗口、菜单、滚动条、对话框等, 使得 Java 可以移植到不同平台的计算机上。

在 Java 的数据库包 (java.sql) 中, 提供了用户进行数据库操作的驱动程序和应用程序接口等。

1.2.3 Java 运行机制

1. Java 程序运行流程

Java 是一种解释型的程序设计语言, 即 Java 源程序需要通过 Java 编译器先进行编译生成一种称为字节码的二进制文件, 之后用 Java 解释器来解释执行这种字节码文件。其流程如图 1.1 所示。

字节码的最大优点是可以跨平台运行, 实际上, 正是字节码实现了“编写一次, 到处运行”的目标。

2. Java 虚拟机及运行机制

Java 虚拟机 (Java Virtual Machine, JVM) 是一种用软件模拟硬件功能实现的“虚拟”计算机。

Java 语言采用虚拟机技术实现 Java 程序的跨平台运行。虚拟机通常在形式上表现为一个软件, 任何可以运行 Java 字节码的程序都可以看成是 JVM, 例如浏览器和 Java 的开发工具