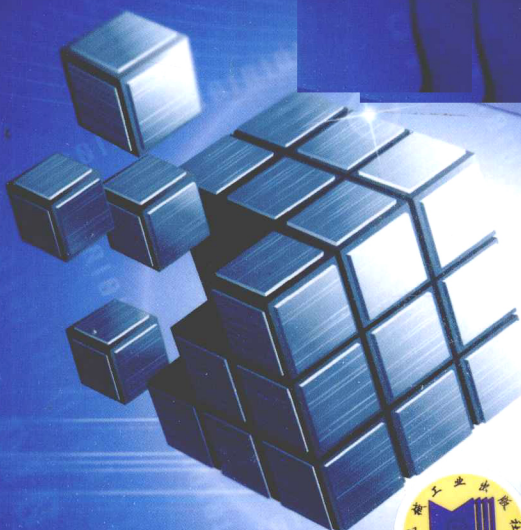




全国高等职业教育规划教材

Java程序设计及实训

黄能耿 编著



电子教案下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

Java 程序设计及实训

黄能耿 编著



机械工业出版社

本书介绍了 Java 语言基础、面向对象的程序设计、Java API 类库、异常处理与单元测试、I/O 处理与对象串行化、图形用户界面编程与 Applet、多线程、数据库编程和网络编程等基础知识。在内容的选取上,突出面向对象的程序设计技术和实用性技术,涵盖软件公司在 Java 软件开发中所需要的基本知识点。

本书的特点是将企业中 Java 软件开发的流程、编码规范、软件测试、进度控制等知识有机地融入到 Java 语言的基础教学中。为此,本书设计了 4 套实训项目案例,贯穿全书。实训项目的设计完全按照软件开发的要求,其中两套实训项目分为多个阶段,随着理论教学新授知识点而不断改进,最终成为较为完整的项目。

本书是 Java 语言的基础教材,既可作为高职高专院校的教学用书,也可作为自学者的参考资料。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ:81922385,电话:010-88379739)。

图书在版编目(CIP)数据

Java 程序设计及实训/黄能耿编著. —北京:机械工业出版社,2010.8

全国高等职业教育规划教材

ISBN 978-7-111-31627-5

I. ① J… II. ① 黄… III. ① Java 语言—程序设计—高等学校:技术学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2010)第 162219 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:鹿 征

责任印制:乔 宇

北京瑞德印刷有限公司印刷(三河市胜利装订厂装订)

2011 年 1 月第 1 版·第 1 次印刷

184mm×260mm·19.5 印张·484 千字

0001—3000 册

标准书号:ISBN 978-7-111-31627-5

定价:32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者服务部:(010)68993821

封面无防伪标均为盗版

全国高等职业教育规划教材计算机专业

编委会成员名单

主 任 周智文

副 主 任 周岳山 林 东 王协瑞 张福强
陶书中 龚小勇 王 泰 李宏达
赵佩华

委 员 (按姓氏笔画排序)

马 伟	马林艺	万雅静	万 钢
卫振林	王兴宝	王德年	尹敬齐
史宝会	宁 蒙	刘本军	刘新强
刘瑞新	余先锋	张洪斌	张 超
李 强	杨 莉	杨 云	罗幼平
贺 平	赵国玲	赵增敏	赵海兰
钮文良	胡国胜	秦学礼	贾永江
徐立新	唐乾林	陶 洪	顾正刚
康桂花	曹 毅	眭碧霞	梁 明
黄能耿	裴有柱		

秘 书 长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近60所高等职业院校的骨干教师对在2001年出版的“面向21世纪高职高专系列教材”进行了全面的修订和增补,并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

Java 语言自从诞生以来, 一直受到业界的追捧, 成为主流的程序设计语言之一, 特别是在企业级大型应用的开发方面占据了主导地位。本书是为高职高专院校计算机专业的学生或其他对 Java 语言有兴趣的读者编写的。本书作者对 ASP、PHP、JSP 有较多的开发经验, 本书内容经过精心设计, 具有如下特点。

- 着重细节: 重视 Java 语言技术的细节, 重点讲解面向对象的程序设计方法, 内容全面丰富, 讲解细致周到。
- 技术先进: 摒弃过时的编程技术, 采用主流的设计思想, 在不经意中融合了 UML 类图、MVC 设计模式等技术, 数据库编程直接采用面向对象的设计, 有利于学生进一步学习 Java EE 的 Struts2 + Spring + Hibernate 技术。
- 编程规范: 强调企业软件开发的流程和规范, 注重培养学生良好的编程习惯。
- 实用性强: 内容的选取以实用为主要依据, 例子代码有较多的注释。第 4 章及以后的内容都是根据开发经验有针对性的重点讨论编写而成。
- 学以致用: 设计了 11 个实训, 将 Java 软件开发的流程、编码规范、软件测试、进度控制等知识融入到 Java 语言的学习中, 使学生在潜移默化中接受代码编写规范的熏陶和初步的软件开发训练。

采用本书授课的课程, 前置课程是“C 语言程序设计”或“C++ 程序设计”, 建议授课课时 80, 实训课时 20, 见表 1。也可进行调整, 以适应不同的需要, 例如讲授第 1~8 章可安排为 64 课时, 安排一周的课程设计完成实训项目。

表 1 教学课时安排建议

章 节	授 课 课 时				实 训 课 时
	课堂教学课时	实验教学课时	小 计	累 计	
第 1 章 Java 语言概述	2	2	4	46	
第 2 章 Java 语言基础	4	4	8		
第 3 章 面向对象的程序设计	10	6	16		4
第 4 章 Java API 类库	6	6	12		4
第 5 章 异常处理与单元测试	2	4	6		2
第 6 章 I/O 处理与对象串行化	2	4	6	62	2
第 7 章 图形用户界面编程与 Applet	2	4	6		4
第 8 章 多线程	2	2	4		
第 9 章 数据库编程	4	6	10	78	2
第 10 章 网络编程	2	4	6		2
机动	2		2		
小计	38	42	80		20

课堂教学主要讲授 Java 语言编程实践中需要掌握的知识点, 教学的目的是学习 Java 语言编程技术。实验教学在计算机机房授课, 采用老师讲授和学生实验相结合的方式, 实验的

目的是在教师的指导下，通过学生的自主探索，进而理解和掌握知识点和相关技术。实训教学采用实训项目案例，实训的目的是学习如何将所学知识点和技术应用到项目的开发中，加深理解知识点以及其中的编程思想，并通过多次训练，达到熟练操作的程度，同时熟悉企业中项目开发的流程、规范和要求。

实训项目可以与课程教学同步进行，也可以安排在课程设计专用周中，建议采用前一种方式，学生每学习完一章即可通过实训项目对所学内容进行巩固和加深理解。实训项目与关键知识点的关系见表2。

表2 实训项目与关键知识点的关系

章 节	实 训 项 目	关键知识点	建议用时 ^[注]
第1章 Java 语言概述			
第2章 Java 语言基础	简易计算器项目	Java 基本语法	15
第3章 面向对象的程序设计	绘图项目（一）	类、接口、继承、多态、包	40
	学生管理项目（一）	类、接口、继承、多态、包	30
第4章 Java API 类库	绘图项目（二）	容器 API（List）	30/60
	学生管理项目（二）	容器 API（Map）、正则表达式	20/40
第5章 异常处理与单元测试	学生管理项目（三）	自定义异常、单元测试	40/80
第6章 I/O 处理与对象串行化	学生管理项目（四）	文件处理、对象串行化	20/90
第7章 图形用户界面编程与 Applet	绘图项目（三）	事件处理、按钮、绘图	60/100
	学生管理项目（五）	事件处理、按钮、对话框、文本框	40/120
第8章 多线程			
第9章 数据库编程	学生管理项目（六）	连接工厂、POJO 类、dao 类	60/180
第10章 网络编程	聊天室项目	UDP 编程、组播、多线程	30

注：建议用时是指在多次训练以后，完成项目所需要的时间（分钟），2 个数字表示阶段用时/全程用时。

本书提供 PPT 电子教案、实训项目视频录像、实训项目源代码、全书例子源代码和部分习题的参考答案，读者可从 www.ngweb.org 下载。

本书由黄能耿编著。在编写过程中，得到了无锡职业技术学院领导和老师们的大力支持和帮助，熊中祥工程师为实训项目的编写提供了宝贵的建议，无锡市服务外包人才培养（联盟）协会举办的 2009 “NIIT” 杯服务外包（软件）公开课竞赛以其独特的比赛方式和评奖标准为本书实训项目的创作提供了思路和启示，在此一并表示衷心的感谢。

由于作者水平有限，书中的错误和不妥之处，敬请读者批评指正。

编 者

目 录

出版说明

前言

第1章 Java 语言概述 1

1.1 Java 语言的历史和特点 1

1.1.1 Java 语言的历史 1

1.1.2 Java 语言的特点 2

1.1.3 Java 语言与 C++ 的比较 2

1.1.4 Java 语言的核心技术 3

1.1.5 Java 语言的局限性 4

1.2 Java 开发环境及其安装 5

1.2.1 Java 运行环境和开发环境 5

1.2.2 Java 运行环境的安装 5

1.2.3 Java 开发环境的安装和配置 5

1.3 Java 语言的应用 8

1.3.1 Java 程序的分类 8

1.3.2 Java 应用程序 8

1.3.3 Java Applet 程序 10

1.4 Eclipse 的使用 10

1.4.1 Eclipse 介绍 11

1.4.2 使用 Eclipse 进行开发 12

1.4.3 使用 Eclipse 中经常遇到的问题 13

1.5 本章小结 15

1.6 习题 15

第2章 Java 语言基础 16

2.1 Java 语言的基本要素 16

2.1.1 关键字 16

2.1.2 Java 标识符 16

2.1.3 分隔符 17

2.1.4 代码注释 18

2.2 变量、数据类型、运算符和

表达式 20

2.2.1 常量和变量 21

2.2.2 基本数据类型 23

2.2.3 空类型 27

2.2.4 引用数据类型 27

2.2.5 运算符 28

2.2.6 表达式、语句和语句块 32

2.3 控制结构 33

2.3.1 顺序结构 33

2.3.2 分支结构 33

2.3.3 循环结构 37

2.3.4 跳转控制语句 40

2.4 数组 42

2.4.1 一维数组 42

2.4.2 二维数组 46

2.4.3 数组的操作 50

2.5 简单的输入/输出 52

2.5.1 输出 52

2.5.2 输入 53

2.6 实训1 简易计算器项目 55

2.6.1 项目需求分析、设计与管理 55

2.6.2 项目实施 56

2.6.3 实施指导 56

2.7 本章小结 56

2.8 习题 56

第3章 面向对象的程序设计 58

3.1 面向对象的基本概念 58

3.1.1 面向对象的术语 58

3.1.2 面向对象的基本特征 59

3.2 类的封装 60

3.2.1 类的声明 60

3.2.2 类的实例——对象 63

3.2.3 构造方法 66

3.2.4 修饰符 67

3.2.5 变量的引用 73

3.2.6 方法调用和参数传递 74

3.2.7 初始化 77

3.2.8 内部类 78

3.2.9 对象的使用	81	3.10 本章小结	121
3.3 类的继承	83	3.11 习题	121
3.3.1 继承的实现	83	第4章 Java API 类库	125
3.3.2 this 和 super	86	4.1 Java 类库介绍	125
3.3.3 抽象类和抽象方法	89	4.1.1 Java API 类库	125
3.3.4 类对象之间的类型转换	92	4.1.2 第三方类库	126
3.3.5 类型检测	95	4.2 java. lang 包	126
3.4 接口	95	4.2.1 Object 类	126
3.4.1 接口的声明	95	4.2.2 Runtime 类	127
3.4.2 接口的实现	96	4.2.3 ClassLoader 抽象类	128
3.4.3 接口的继承	97	4.2.4 Class 类	129
3.4.4 接口的使用	99	4.2.5 基本数据类型的包装类	131
3.4.5 UML 类图	99	4.2.6 Math 类	132
3.4.6 接口的特点	101	4.2.7 System 类	133
3.4.7 再论继承	102	4.2.8 String 类	135
3.5 聚集	102	4.2.9 StringBuffer 类	135
3.5.1 聚集与代码复用	102	4.2.10 异常处理类	135
3.5.2 继承与聚集的选择	103	4.2.11 线程控制类	135
3.6 类的多态	103	4.3 字符串处理	135
3.6.1 方法的重载	104	4.3.1 字符串	135
3.6.2 方法的覆盖	105	4.3.2 String 类字符串	135
3.6.3 多态的优点	107	4.3.3 StringBuffer 类字符串	141
3.6.4 再论接口	107	4.3.4 String 类和 StringBuffer 类的 比较	144
3.7 包	108	4.4 java. util 包	146
3.7.1 包的声明	108	4.4.1 日期类库	147
3.7.2 包的导入	109	4.4.2 容器类类库	151
3.7.3 CLASSPATH	111	4.4.3 正则表达式	162
3.7.4 再论修饰符	111	4.5 其他常用类库介绍	168
3.7.5 包的使用	112	4.5.1 其他 Java API 类库	168
3.7.6 项目打包和部署	115	4.5.2 第三方类库	168
3.7.7 项目开发过程总结	116	4.6 实训4 绘图项目(二)	169
3.8 实训2 绘图项目(一)	116	4.6.1 项目需求分析、设计与管理	169
3.8.1 项目需求分析、设计与管理	116	4.6.2 项目实施	170
3.8.2 项目实施	118	4.6.3 实施指导	170
3.8.3 实施指导	118	4.7 实训5 学生管理项目(二)	171
3.9 实训3 学生管理项目(一)	119	4.7.1 项目需求分析、设计与管理	171
3.9.1 项目需求分析、设计与管理	119	4.7.2 项目实施	172
3.9.2 项目实施	121	4.7.3 实施指导	172
3.9.3 实施指导	121		

4.8 本章小结	172	6.3.2 对象串行化的实现	205
4.9 习题	172	6.3.3 串行化的注意事项	208
第5章 异常处理与单元测试	176	6.4 实训7 学生管理项目(四)	208
5.1 异常概述	176	6.4.1 项目需求分析、设计与管理	208
5.1.1 异常的概念	176	6.4.2 项目实施	209
5.1.2 几种常见的异常	176	6.4.3 实施指导	209
5.1.3 异常的分类	177	6.5 本章小结	209
5.2 异常处理	178	6.6 习题	209
5.2.1 捕获异常	179	第7章 图形用户界面编程	211
5.2.2 声明抛出异常	181	与 Applet	211
5.3 自定义异常	183	7.1 GUI 介绍	211
5.3.1 自定义异常类	183	7.2 组件、容器、布局和观感	212
5.3.2 抛出异常	183	7.2.1 组件	212
5.3.3 自定义异常类的使用	183	7.2.2 容器	214
5.4 单元测试	185	7.2.3 观感	215
5.4.1 JUnit 介绍	185	7.2.4 布局管理器	215
5.4.2 Test Case	185	7.3 事件处理	219
5.4.3 Test Suite	189	7.3.1 事件处理介绍	219
5.4.4 单元测试的设计和编写	189	7.3.2 使用内部类和匿名类作为 监听器	220
5.4.5 单元测试的执行和记录	189	7.3.3 事件处理模型	221
5.5 实训6 学生管理项目(三)	190	7.3.4 事件、监听器和适配器	222
5.5.1 项目需求分析、设计与管理	190	7.4 图形的绘制	226
5.5.2 项目实施	191	7.4.1 Graphics 类	227
5.5.3 单元测试	191	7.4.2 绘图实例	228
5.5.4 实施指导	191	7.5 Applet 编程	231
5.6 本章小结	191	7.5.1 Applet 概述	231
5.7 习题	192	7.5.2 Applet 和 HTML	233
第6章 I/O 处理与对象串行化	193	7.5.3 Applet 的界面设计	234
6.1 文件处理	193	7.5.4 Applet 的应用	234
6.1.1 文件处理概述	193	7.6 实训8 绘图项目(三)	235
6.1.2 文件类的使用	193	7.6.1 项目需求分析、设计与管理	235
6.2 I/O 处理	197	7.6.2 项目实施	236
6.2.1 I/O 处理概述	197	7.6.3 实施指导	236
6.2.2 字节流	198	7.7 实训9 学生管理项目(五)	237
6.2.3 字符流	202	7.7.1 项目需求分析、设计与管理	237
6.2.4 随机读/写文件	205	7.7.2 项目实施	238
6.2.5 I/O 处理的注意事项	205	7.7.3 项目发布	238
6.3 对象的串行化	205	7.7.4 实施指导	238
6.3.1 串行化概述	205		

7.7.5 最终项目成果	239	9.3.4 dao 类	267
7.8 本章小结	239	9.3.5 单元测试	271
7.9 习题	239	9.4 实训 10 学生管理项目(六) ...	273
第 8 章 多线程	241	9.4.1 项目需求分析、设计与管理	273
8.1 多线程的概念	241	9.4.2 项目实施	275
8.2 多线程的实现方法	244	9.4.3 项目发布	275
8.2.1 继承 Thread 类	244	9.4.4 实施指导	276
8.2.2 实现 Runnable 接口	244	9.4.5 最终项目成果	276
8.2.3 线程的优先级	246	9.5 本章小结	276
8.2.4 Thread 类的常用方法	246	9.6 习题	276
8.2.5 两种实现方法的比较	246	第 10 章 网络编程	278
8.2.6 线程的状态和生命周期	247	10.1 网络基础知识	278
8.3 多线程的同步、通信与死锁	248	10.1.1 TCP/IP 协议	278
8.3.1 多线程的同步	248	10.1.2 IP 协议	278
8.3.2 线程间的通信	251	10.1.3 TCP 和 UDP 协议	279
8.3.3 死锁	254	10.1.4 应用层协议	281
8.3.4 线程安全	255	10.1.5 域名	281
8.4 本章小结	255	10.1.6 C/S 模式与 B/S 模式	281
8.5 习题	255	10.2 URL 编程	282
第 9 章 数据库编程	256	10.2.1 URL 访问远程资源	282
9.1 数据库编程概述	256	10.2.2 URLConnection 访问远程资源 ...	283
9.1.1 JDBC	256	10.3 TCP 编程	286
9.1.2 JDBC 的功能	256	10.3.1 Socket 介绍	286
9.1.3 MySQL 介绍	256	10.3.2 ServerSocket 服务器端的编程 ...	287
9.2 数据库访问流程	259	10.3.3 Socket 客户端的编程	290
9.2.1 数据准备	259	10.3.4 支持多客户的服务器端程序 ...	292
9.2.2 添加 JDBC 类库	259	10.4 UDP 编程	295
9.2.3 加载 JDBC 驱动程序	260	10.4.1 Datagram 编程	295
9.2.4 创建数据库连接	260	10.4.2 组播通信	297
9.2.5 执行 SQL 语句	261	10.5 实训项目 11 聊天室项目	299
9.2.6 接收并处理 SQL 的返回结果	261	10.5.1 项目需求分析、设计与管理	299
9.2.7 关闭创建的对象	262	10.5.2 项目实施	300
9.3 数据库编程方法	264	10.5.3 实施指导	300
9.3.1 程序设计	264	10.6 本章小结	300
9.3.2 连接工厂	265	10.7 习题	301
9.3.3 POJO 类	267	参考文献	302

第 1 章 Java 语言概述

本章要点

本章介绍 Java 语言的发展历史、特点和工作机制, Java 与 C++ 的比较, Java 语言开发环境的安装和使用, Java 应用程序和 Java Applet 程序的编写, Java 集成开发工具的使用等入门知识。

1.1 Java 语言的历史和特点

1.1.1 Java 语言的历史

Java 语言源自 Sun Microsystem 公司于 1991 年开发的一个面向家用电子产品编程的项目。该项目开发一个在电冰箱、电视机等产品上使用的软件通用运行平台。这个项目先后被称为 Green 和 Oak 项目。它的设计思路是开发一种可移植的语言, 采用这种语言开发的应用软件可以在不同的硬件平台上运行。例如, 对于不同品牌的电视机使用不同的 CPU 控制芯片, 当采用传统方式开发时, 在不同芯片上开发应用软件需要不同的语言和技术, 而上述项目的目标则是为不同的硬件产品设计一种通用的运行平台, 采用这种语言开发出来的软件可以在不同的硬件平台上运行, 从而大幅度地降低开发成本。

到了 1994 年, Oak 项目虽然在技术上取得了成功, 但并没有在商业上取得成功, 无法进行推广, 其主要原因是当时家用电器的性能有限, 主要是内存小和 CPU 速度慢这两个致命因素限制了它的发展。

此时的网络技术正在蓬勃发展, 需要一种跨平台的计算机语言, Oak 项目正好可以满足这种需求。计算机的性能比家用电器的性能高很多, 内存和 CPU 都不能成为制约发展的瓶颈。因此, Sun Microsystem 公司决定将该项目应用到网络编程领域, 并在当时取得了极大的成功。由于 Oak 这个商标已被注册, 因此 Oak 项目的开发人员就以他们最爱喝的 Java 咖啡(产于印尼爪哇岛)命名了这个产品。

1995 年 Java 被正式发布, 并立即引起了极大的轰动, 成为网络编程的主流语言之一。表 1-1 列出了 Java 语言发展史上的重要版本 (Version), 其平台类型如表 1-2 所示。

表 1-1 Java 语言发展史上的重要版本 (Version)

发布日期	版本	特 点
1995 年 6 月	1.0	正式发布, 立即引起极大轰动, 提出 “Write Once, Run Anywhere” (一次编写, 到处运行) 的口号
1998 年 12 月	1.2	一个划时代的版本, 即 Java 2, 演变出 J2SE、J2EE 和 J2ME
2002 年 2 月	1.4	一个稳定的版本, 其中的 1.42 版目前仍被广泛使用
2004 年 9 月	1.5 (5.0)	重大改进, 并且 J2SE、J2EE 和 J2ME 改名为 Java SE、Java EE 和 Java ME
2006 年 12 月	6.0	当前使用的版本

表 1-2 Java 语言的平台类型

平 台	全 称	用 途
Java SE(J2SE)	Java Standard Edition	Java 标准版,分为 Java JRE 和 Java SDK
Java EE(J2EE)	Java Enterprise Edition	Java 企业版,是一系列标准的集合,用于企业应用的开发。它包括 Java SE
Java ME(J2ME)	Java Micro Editon	Java 微型版,用于手机等小电器的编程。它是 Java SE 的子集以及专用的功能

1.1.2 Java 语言的特点

Sun Microsystem 公司在 Java 语言的产品白皮书中总结了 Java 语言的 11 个特点（见表 1-3）。

表 1-3 Java 语言的特点

特 点	说 明
简单性	系统精简，但功能齐备 风格类似于 C ++，对 C ++ 程序员而言非常容易掌握 Java 摒弃了 C ++ 中容易引发程序错误的地方，如指针操作和内存管理 提供丰富的 API 类库
面向对象	完全面向对象。而 C ++ 既支持面向对象的程序设计技术，同时也支持面向过程的程序设计技术
分布式处理	支持 TCP/IP。Java 应用程序访问远程对象，就像访问本地文件系统那样简单
健壮性	提供面向对象的异常处理机制，有较强的编译时代码检查能力 提供自动垃圾收集功能，自动进行内存管理
结构中立	将源程序编译成一种结构中立的中间文件格式。只要有 Java 运行系统的机器都能执行这种中间代码
安全性	Java 语言提供的安全 编译器提供的安全 字节码校验时进行安全检查 类装载时进行安全检查
可移植性	Java 应用程序可以在配备了 Java 运行系统的任何机器上运行，而与体系结构无关
解释性	Java 编译器将 . java 源文件编译成类文件 . class，类文件通过 Java 命令解释执行，将字节码转换为机器码
高性能	采用 JIT 编译技术，提高 Java 代码的执行速度
多线程	内置多线程功能，在一个程序里同时执行多个小任务。其优点是具有更好的交互性和实时控制性
动态性	允许程序动态地装入运行过程中所需要的类，这是 C ++ 进行面向对象程序设计时所无法实现的

1.1.3 Java 语言与 C ++ 的比较

C 语言诞生于 1971 年，后来在 20 世纪 80 年代发展为 C ++，40 年来一直受到开发人员的青睐。C/C ++ 语言具有语法结构清晰、功能强大、性能极高等众多的优点，但是也有一些缺点，例如指针的滥用、内存管理不严、数据类型复杂、可移植性差等。

Java 语言的设计者借鉴了 C/C ++ 语言的优点，摒弃了其缺点，完全采用面向对象技术，设计了这样一种优秀的程序设计语言。表 1-4 是 Java 语言与 C ++ 语言的比较。

表 1-4 Java 语言与 C++ 语言的比较

项 目	Java 语言	C++ 语言
全局变量	不存在全局变量	存在
数据类型	无结构、联合，整数类型无符号的区别	数据类型丰富，使用复杂
数据类型转换	转换时进行相容性检查，安全性好	通过指针可进行任意的类型转换
范围运算符	无	有作用域运算符“::”
数组	通过对象操作，有 length 属性	通过指针操作，无 length 属性
内存管理	内存使用后不必回收，有自动垃圾回收机制	内存使用后必须回收，否则造成内存崩溃
提前声明	不必提前声明	需先声明后使用
预处理	无	有预处理机制
头文件	import 包	include 头文件
指针	无	有，功能强，但带来安全问题
goto 关键字	作为保留的关键字，无作用	有，可以使用
virtual 关键字	无	有
注释文档	提供对注释文档的支持	无

1.1.4 Java 语言的核心技术

Java 语言的核心设计理念是跨平台 and 安全性。为此，Java 发展了 Java 虚拟机、Java 字节码和垃圾回收机制 3 大核心技术。

1. Java 虚拟机

实现跨平台的基础是 Java 虚拟机 (Java Virtual Machine, JVM)，它是执行以 Java 字节码 (见下述) 为指令集的软 CPU，负责对字节码文件 (扩展名为 .class，也称为类文件) 的解释执行。

Java 虚拟机包含类加载器、字节码确认器以及 JIT 实时编译器 (见图 1-1)。

- 类加载器：从网络或本地取得类文件并加载到内存。
- 字节码校验器：校验类文件是否符合正确的格式。
- JIT 实时编译器：使虚拟机能提高到编译式的运行效率。
- Java 解释器：将字节码解释翻译成机器码。

2. 字节码

C++ 程序经编译、连接后的产品是机器码，即可以被本地机器识别并直接执行的可执行文件 (见图 1-2)。因此，针对不同的操作系统，需要编译、连接生成可被不同操作系统识别的可执行文件。C++ 的跨平台性是源代码级的跨平台，可执行文件是不能跨平台的。Java 程序编译后的产品是字节码文件，与机器码不同，它不是真实 CPU 可执行的机器代码，故又称伪代码。字节码文件需要由 JVM 在执行期间翻译成能被本地机器识别的机器码，然后再交给操作系统执行 (见图 1-1)。因此，Java 的跨平台性不仅是源代码级的跨平台，而且可执行文件也是跨平台的，只要目标操作系统安装了 JVM，就可以执行任意的字节码文件，从而实现 “Write Once, Run Anywhere” (一次编写，到处运行)。

3. 垃圾回收机制

C++ 在需要使用内存时，必须申请分配内存空间，一旦使用完毕，必须由程序员通过代码释放该空间；否则将造成内存的大量占用，最终导致应用程序崩溃。

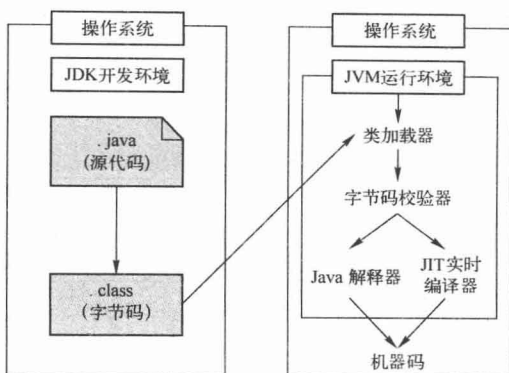


图 1-1 Java 虚拟机与字节码的关系

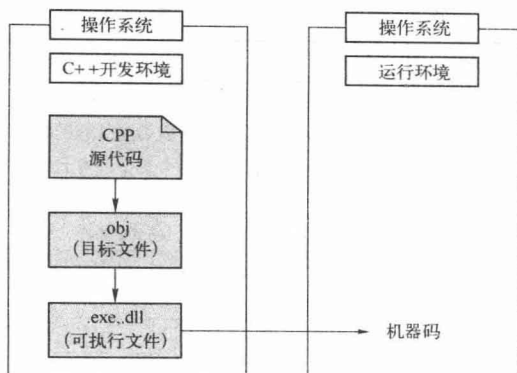


图 1-2 C++ 的运行机制

Java 语言在这方面作了改进，引入自动垃圾回收机制，即程序员不必也不需要关心内存空间的释放，这一切工作由 Java 的自动垃圾回收机制完成。

Java 语言的核心技术对软件技术发展作出了很大的贡献，主要表现在以下两个方面。

- 在 Java 技术的基础上，出现了多种语言。如 JRuby、Jython、Scala 和 Groovy 语言。前两种是原有语言（Ruby 和 Python）在 JVM 平台上的实现，而后两种则是全新的语言，它们虽然与 Java 语言在语言结构上不同，但都是编译成 Java 字节码文件，同样需要 JVM 的支持才能运行。可以说，Java 不仅仅是一种语言，更是一个平台，多种语言可以共享的一个平台。
- Java 虚拟机技术也渗透到其他软件语言的设计中。例如，微软公司的 .NET 技术也采用了虚拟机技术——通用语言运行时（Common Language Runtime, CLR）、字节码技术——微软中间语言（Microsoft Intermediate Language, MSIL）以及自动垃圾回收机制。因此 .NET 平台上的语言，例如 C# 与 Java 非常相像，同样有自动垃圾回收机制，也同样是编译成字节码，也具有跨平台运行的潜力。

1.1.5 Java 语言的局限性

Java 语言在早期的发展中，曾受到性能因素的制约，一般来说，使用 Java 语言开发的应用软件要比用 C++ 开发的应用软件在性能上相差 2 ~ 10 倍。这在当时是最严重的缺陷之一。目前，由于 Java 技术的发展，特别是 JIT 技术的引入，性能问题已经得到了很好的解决。

Java 语言的另一个主要制约因素是缺少一种好的图形界面开发环境。近年来，IBM 公司推出了 Eclipse 以及 Eclipse 插件的开发，Sun Microsystem 公司对 NetBean 作了大幅改进，使这种状况得到了极大的改善。

Java 语言的另一个缺点是对图形界面编程的支持不够完善，限制了 Java 语言在桌面应用软件方面的应用。为此，IBM 公司在开发 Eclipse 的过程中，开发了一套独立的图形类库（SWT），这是对 Java 语言的一个重大改进。

Java 语言被设计为不适合于底层软件的开发，如操作系统的开发、驱动程序的开发等，而这些领域却是 C/C++ 语言最擅长的领域。因此，虽然 Java 语言在网络编程以及其他广泛的领域内取得了巨大的成功，但 Java 语言和 C/C++ 语言仍是各有所长，都将在比较长的时期内占据主流编程语言的地位。

1.2 Java 开发环境及其安装

1.2.1 Java 运行环境和开发环境

1. Java 运行环境

采用 Java 语言开发的应用软件需要 JVM 的支持才能运行。Java 的运行环境称为 JRE (Java Runtime Environment)，它包括 JVM 以及相应的 API 类库（如图形 API 类库）。因此，所有需要运行 Java 应用软件的计算机都必须安装 JRE。

JRE 本身是与操作系统相关的，不同的操作系统有不同的 JRE 版本，安装时需要选择与操作系统匹配的 JRE 版本。一旦安装了 JRE，就能运行各种 Java 应用软件。

2. Java 开发环境

对于开发人员来说，除了需要上述的运行环境以外，还需要开发环境的支持，Java 的开发环境主要由以下两部分组成。

- Java 开发工具包 (Java Development Kit, JDK)：主要由 Java 编译器、调试工具等组成，是 Java 开发必备的工具。JDK 内含 JRE，不再单独需要 JRE。
- 集成开发环境 (Integrated Development Environment, IDE)：除了可以直接使用记事本等文本编辑器来开发 Java 程序以外，绝大多数开发人员都会选择一种集成开发环境。

Java IDE 有多种，表 1-5 列出了比较常见的 4 种，目前大多数软件企业都是使用 Eclipse 进行开发。

表 1-5 常用的 Java IDE

IDE 工具	最新版本	特 点
Eclipse	3.5.1 (2009.9 发布)	由 IBM 公司提供，最为流行，开源免费，功能强大，并且有各种插件可用
NetBeans	6.8 (2009.12 发布)	由 Sun Microsystem 公司提供，开源免费，曾不被看好，但有后来居上的趋势
JBuilder	2008 (2008.4 发布)	Borland 公司的产品，现被 Embarcadero 收购。商业版本，曾经是最好的 IDE
IntelliJ IDEA	9.0 (2009.12 发布)	JetBrains 公司的产品。商业版本(有开源的简化版)，功能强大，但使用者不太多

1.2.2 Java 运行环境的安装

在计算机上必须安装 JRE，才能运行由 Java 语言开发出来的应用软件。JRE 软件可以从 <http://java.sun.com> 下载，下载时需要注意选择合适的操作系统版本，下载后按普通的软件安装方法安装，并参照下述 JDK 的安装步骤，配置 3 个环境变量。

1.2.3 Java 开发环境的安装和配置

要学习 Java 语言，首先必须安装 Java 开发环境，包括 JDK 和 IDE。

1. JDK 的安装

目前 Sun Microsystem 公司提供的 JDK 是开源软件，可以从 <http://java.sun.com> 网站自由下载。该网站提供 JRE 和 JDK 的下载，而 JDK 中已经包含 JRE，因此下载 JDK 时就不必再下载 JRE，在安装 JDK 的同时也将安装 JRE。下载时需要注意选择合适的操作系统版本，下载的文件名中含有版本号。例如，jdk-6u18-windows-i586.exe 是 Windows 操作系统上 32 位版本的

JDK, 版本号是 6.0_18; 64 位版本的则是 jdk-6u18-windows-x64.exe。安装过程分为以下两步。

- 1) 安装 JDK 开发工具: 直接运行下载的安装文件, 安装过程中不需要进行任何配置。
- 2) 设置环境变量: 安装结束后, 需要设置如下的环境变量。

- JAVA_HOME = Java SDK 所在的目录。
- path = javac 和 java 命令所在的目录。
- CLASSPATH = 存放字节码文件的目录。

在 Windows 平台上, 如果 jdk 的安装目录是 C:\Program Files\Java\jdk6.0_18, 则上述环境变量的设置是:

```
JAVA_HOME = C:\Program Files\Java\jdk6.0_18
path = ;%JAVA_HOME%\bin
CLASSPATH = .;%JAVA_HOME%\lib\tools.jar;%JAVA_HOME%\lib\dt.jar
```

设置的方法是: 从“我的电脑”右键菜单中选择“属性”选项, 打开“系统属性”对话框, 然后在“高级”选项卡中, 单击“环境变量”按钮, 打开“环境变量”对话框。在“系统变量”区域按上述要求设置环境变量 (见图 1-3)。

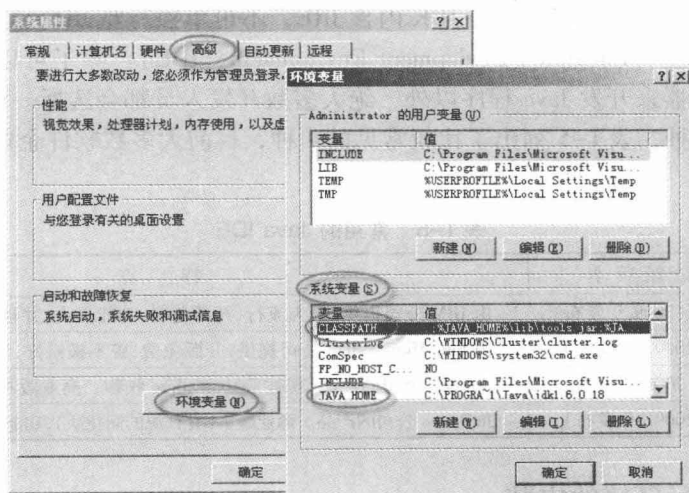


图 1-3 设置环境变量

设置时需要注意以下几点。

- 环境变量的各个值之间用分号分隔。
- 系统中已经存在 path 变量, 千万不能删除或更改原有的值, 只能在后面添加。不要忘记用分号“;”与原有的值分隔开来。
- CLASSPATH 值的第一个字符是句点, 它表示当前目录。

安装后要测试安装是否成功, 可以在命令行下运行:

```
java -version
```

如果安装正确, 则应该显示 Java 的版本号等有关信息。

JDK 提供了完整的 Java 开发工具, 其中常用的有下列几种。