

全国电子信息类优秀教材
浙江省“十一五”重点建设教材
新工科建设之路·软件工程系列教材



Java 面向对象实用教程

第 4 版

杨晓燕 王仁芳 刘云鹏 邓 芳 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

教材
建设教材
工程系列教材

Java 面向对象实用教程

第4版

杨晓燕 王仁芳 刘云鹏 邓芳 编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

Java 程序设计语言具有完全面向对象、简单高效、与平台无关等特点,同时 Java 内建了对网络编程、数据库连接、多线程等高级程序设计任务的支持。自 1995 年发布以来,Java 在开发领域一直高歌猛进,其地位一直名列前茅。特别在“互联网+”迅速发展的今天,Java 有着非常广阔的应用领域。本书基于“案例分析——知识学习——案例实现”和“章节案例——综合案例”的结构,本着 Java 基础和架构应用并重的原则进行编写,教材内容主要包括:Java 的渊源和特点、Java 编程基础、Java 流程控制与算法、类的结构及应用、面向对象编程基本原则、多线程应用、Java 常用的 API 和 GUI 图形界面等。

本书中的程序在 JDK 7.0 中经过验证,并都给出运行结果,教材案例通过二维码辅以微视频进行在线解读。本书在编写过程中,重要内容注意提炼,重点内容重点提示,使平面教材具有立体感,使读者便于学习和理解。同时为了教师教学方便,主要的程序代码都增加了行号。课后习题分为学习内容“积木化”的练习和拓展研讨题,并提供参考答案,每章内容都安排实训内容,便于读者“用中学,学中用”。

本书既可作为大中专学校的 Java 程序设计的教材,又可作为 Java 自学者的入门用书。

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 面向对象实用教程/杨晓燕等编著. —4 版. —北京:电子工业出版社, 2019.1

ISBN 978-7-121-34715-3

I. ①J… II. ①杨… III. ①JAVA 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 150204 号

策划编辑:章海涛

责任编辑:章海涛

印 刷:北京七彩京通数码快印有限公司

装 订:北京七彩京通数码快印有限公司

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:20.75 字数:518 千字

版 次:2006 年 1 月第 1 版

2019 年 1 月第 4 版

印 次:2019 年 1 月第 1 次印刷

定 价:49.00 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, 88258888。

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。

本书咨询联系方式:192910558 (QQ 群)。

前 言

Java 自 1995 年诞生以来独树一帜,发展为计算机世界的“国际语言”。原 Sun 公司总裁兼首席运营官 Jonathan Schwartz 说:“Java 技术正在成为全球网络应用的事实标准,它将大大加快和简化提供移动、消费和企业市场的服务。”

计算机学科的应用型人才不仅应具有基本的科学理论基础,而且更重要的是能将理论和实践相结合,并具有解决实际问题的能力。

2005 年,在 Java 发布 10 周年之际,我们探索编写第 1 版 Java 程序设计教程,由人民邮电出版社出版,受到读者和出版社好评,连续加印 5 次。2010 年,本教材获得浙江省“十一五”重点教材建设项目立项;重新修订编写后于 2012 年在电子工业出版社出版了第 2 版,当年获全国电子信息类优秀教材;2015 年,在人民邮电出版社的支持下,编写了第 3 版,依然受到读者的诸多鼓励。

今年基于电子工业出版社的邀约和鼓励,本次修订是教材第 4 版,本次修訂立足点是重入门、重基础、重方法、重实用。同时,在 Java 内容体系和结构上做微调,增加章节实训,基于“互联网+”新时代特征,推荐优秀互联网资源,辅助和简化教材,线上、线下资源结合。本书基于编程语言动手实践的重要性,层次上分为教学与研学,研学内容主要为学生自主探究学习。将原来教材中的 JDBC 内容放在后续 Web 高级开发集成环境中,方便学生学习。同时本书内容体现了不断扩展的类库和相关新特性。通过案例的引领,突出“学中用”和“用中学”,语言方面保持了读者一致首肯的形象、生动的特点,不断推敲,使读者读起来更流畅,更易于理解。

本书特色:

- 每章由“案例分析”开篇,“案例实现”收尾,案例大小适中,使读者能够快速入门,复杂案例附带视频讲解。
- 遵循“案例分析——知识学习——案例实现”以及“章节案例——综合案例”的结构,既便于读者理解,又把综合案例分解为可独立运行的子案例。
- 四化设计:核心知识案例化,抽象概念形象化,复杂问题通过分解尽量简单化,综合知识项目化。教材内容循序渐进,程序注重前后衔接和对比,环环相扣。为了便于教师讲解和理解,对教材中大部分程序的主要程序代码增加行号。
- 基于读者学习的实际需要,将输入/输出庞杂的内容进行了整合,基于应用和模型,删繁就简,突出重点,便于读者快速上手。
- 本书不仅注重一般概念和理论的解析,而且注重系统开发过程中结构和模式的研究。
- 课后练习的习题是“积木式”的内容重建,问题探究在知识广度和深度上拓展,SCJP/OCJP 题为读者打开一扇认证之窗。

本书编写的初衷是重在应用,每章的习题、问题探究及 SCJP 均配有参考答案,同时增加习题的二维码互动以及有关案例的视频演示和讲解。

在本书顺利完成之际,要感谢 2005 年我们一起编写教材的长辈和同事,他们是已经故去的尊敬的姜遇姬教授,还有我年轻的同事邓芳、刘臻等,以及我聪明的学生潘庆伟、王

贤挺等。感谢电子工业出版社章海涛和孟宇老师的积极推动、热心付出和敬业的指导。

PPT 等资料可以在华信教育资源网 <http://www.hxedu.com.cn> 下载。

由于编者水平所限，书中难免还存在一些缺点和错误，希望读者批评指正。联系方式 yangxy3225@163.com。

注：代码行号在程序编辑、编译、运行时是不需要的。*为自学内容，教材案例为研学内容。



课后习题及问题探究参考答案

编者

2018 年 12 月

目 录

第 1 章 Java 概述	1
1.1 Java 崛起	1
1.2 Java 与 C、C++	3
1.2.1 Java 和 C++	3
1.2.2 Java 与 C	3
1.3 Java 语言特点及更新	5
1.4 Java 程序的类型及其不同的编程模式	6
1.5 Java 开发工具入门	8
1.5.1 JDK 的下载、安装	8
1.5.2 配置环境变量	9
1.5.3 JDK 开发工具简介	11
1.6 Java 程序开发过程	11
1.7 实训	18
习题 1	18
问题探究 1	18
第 2 章 Java 编程基础	20
2.1 标识符、关键字和分隔符	20
2.1.1 标识符和关键字	20
2.1.2 分隔符	21
2.2 数据类型	22
2.2.1 基本数据类型	23
2.2.2 常量和变量	24
2.3 运算符与表达式	26
2.3.1 算术运算符	26
2.3.2 赋值运算符	27
2.3.3 关系运算符	28
2.3.4 逻辑运算符	29
2.3.5 条件运算符	29
2.3.6 其他运算符	30
2.3.7 运算符的优先级	30
2.4 Scanner 键盘输入	30
2.5 案例实现	33
习题 2	33
问题探究 2	34

第3章 程序流程控制结构和方法	36
3.1 语句和程序流程控制结构	36
3.2 选择结构	37
3.2.1 选择语句	37
3.2.2 多选择结构 switch 语句	41
3.3 循环结构	44
3.3.1 三种循环语句	44
3.3.2 循环程序结构小结	48
3.3.3 循环嵌套与 continue、break 语句	48
3.4 算法设计*	51
3.4.1 迭代算法	51
3.4.2 穷举算法	52
3.4.3 递归算法	54
3.5 案例实现	55
习题 3	55
问题探究 3	56
第4章 数组	57
4.1 数组的基本概念	57
4.2 一维数组	58
4.2.1 一维数组的声明	58
4.2.2 一维数组内存申请	58
4.2.3 一维数组的初始化	59
4.2.4 测定数组的长度	60
4.2.5 for each 语句与数组	61
4.3 二维数组	62
4.3.1 认识二维数组	62
4.3.2 二维数组的声明与创建	63
4.3.3 二维数组元素的初始化	64
4.3.4 二维数组的引用	65
4.4 案例实现* (课外研学)	68
习题 4	70
问题探究 4	71
第5章 Java 类和对象	73
5.1 面向对象编程	73
5.2 类的描述	76
5.2.1 类的定义	76
5.2.2 成员变量的访问控制符	78
5.2.3 成员方法	79

5.2.4	成员变量和局部变量	81
5.2.5	final 变量	82
5.3	对象的创建与使用	82
5.3.1	对象的创建	82
5.3.2	对象的比较	84
5.3.3	对象的使用	85
5.3.4	释放对象	89
5.3.5	Java 变量内存分配	89
5.3.6	匿名对象	90
5.4	类的构造方法	91
5.4.1	构造方法的作用和定义	91
5.4.2	this 引用	93
5.5	static 变量及 static 方法	99
5.5.1	static 变量	99
5.5.2	static 方法	100
5.6	对象初始化过程	103
5.7	成员方法	106
5.7.1	方法调用与参数传递方式	106
5.7.2	方法重载	111
5.7.3	final 最终方法和 abstract 抽象方法	113
5.8	复杂程序解决方案和方法	113
5.9	案例实现	117
	习题 5	120
	问题探究 5	123
第 6 章	类的继承和接口	126
6.1	类的继承	126
6.1.1	继承的概念	126
6.1.2	创建子类	127
6.1.3	关于父类的构造方法	128
6.2	成员变量的隐藏和成员方法的重构	131
6.3	抽象类	133
6.4	接口	135
6.4.1	接口概述	135
6.4.2	接口的定义	136
6.4.3	实现接口的类定义	136
6.4.4	接口的多态性	139
6.4.5	Java 8 接口扩展方法	141
6.5	泛型	142

6.5.1	泛型的概念和泛型类的声明	142
6.5.2	泛型应用	142
6.6	案例实现	144
习题 6		148
问题探究 6		149
第 7 章	Java API 初步	150
7.1	Java SE API 官网下载	150
7.2	Java 输入/输出	151
7.2.1	标准输出方法	151
7.2.2	命令行参数输入法的应用	153
7.2.3	流式交互输入/输出的应用	154
7.2.4	Java I/O 基本模型	155
7.2.5	文件数据的读/写	157
7.2.6	JOptionPane 对话框输入法	159
7.3	字符串类	160
7.3.1	创建 String 对象	161
7.3.2	创建 StringBuffer 对象	162
7.3.3	正则表达式与模式匹配实例	163
7.3.4	Java 中正则表达式常用的语法	164
7.3.5	模式匹配方法	164
7.4	颜色类与图形绘制类	166
7.4.1	图形的颜色控制	166
7.4.2	类 Graphics 的基本图形	167
7.5	集合 ArrayList	170
7.5.1	集合概述	170
7.5.2	类 ArrayList 的应用	171
7.5.3	ArrayList 的综合应用	174
7.5.4	类 Arrays	175
7.6	Java 8 新特性*	177
7.7	Java 9 入门体验	178
7.8	案例实现* (课外研学)	181
习题 7		182
问题探究 7		183
第 8 章	包和异常	185
8.1	包	185
8.1.1	创建包	186
8.1.2	类的包外引用	187
8.2	异常处理	189

8.2.1	异常的基本概念	190
8.2.2	异常处理机制	193
8.2.3	自定义异常类	197
8.2.4	GUI 应用程序的异常处理	198
8.3	案例实现	200
习题 8		203
问题探究 8		203
第 9 章	面向对象程序设计的基本原则及初步设计模式*	204
9.1	UML 类图	204
9.1.1	类的 UML 图	205
9.1.2	UML 接口表示	205
9.1.3	UML 依赖关系	205
9.1.4	UML 关联关系	206
9.1.5	UML 聚合关系	206
9.1.6	UML 组合关系	207
9.1.7	泛化关系	207
9.1.8	实现关系	208
9.2	面向对象程序设计的基本原则	209
9.2.1	发现变化, 封装变化	209
9.2.2	单一职责原则和最少知识原则	212
9.2.3	开放—封闭原则	212
9.2.4	子类型能够替换基类型原则	213
9.2.5	合成/聚合复用原则	215
9.3	案例实现	215
习题 9		219
问题探究 9		219
第 10 章	图形用户界面	221
10.1	图形用户界面概述	221
10.1.1	图形用户界面组件	221
10.1.2	组件分类	222
10.1.3	常用容器类的应用	223
10.2	事件处理	227
10.2.1	基本概念	227
10.2.2	事件处理机制	229
10.2.3	事件处理的实现方式	230
10.2.4	适配器类	234
10.3	一般组件	237
10.3.1	标签	237

10.3.2	按钮	237
10.3.3	文本框	238
10.3.4	文本区	238
10.3.5	列表框	240
10.3.6	滚动窗格	242
10.3.7	复选框和单选按钮	243
10.3.8	滑动条	245
10.4	菜单与对话框	250
10.4.1	创建菜单	250
10.4.2	弹出式菜单	254
10.4.3	对话框	255
10.5	布局管理器*	260
10.5.1	顺序布局	261
10.5.2	边界布局	261
10.5.3	网格布局	262
10.5.4	卡片布局	263
10.5.5	手工布局	264
10.6	JApplet 类的使用	265
10.7	Java 事件类方法列表	266
10.8	案例实现	267
	习题 10	271
	问题探究 10	274
第 11 章	多线程	275
11.1	多线程概述	275
11.1.1	基本概念	276
11.1.2	线程的状态与生命周期	277
11.1.3	线程的调度与优先级	279
11.2	创建和运行线程	279
11.2.1	利用 Thread 类创建线程	280
11.2.2	用 Runnable 接口创建线程	282
11.3	线程间的数据共享	284
11.4	多线程的同步控制*	287
11.4.1	线程同步相关概念	287
11.4.2	synchronized 应用	289
11.4.3	synchronized 的进一步说明	292
11.5	案例实现	293
	习题 11	295
	问题探究 11	296

第 12 章 综合案例——聊天通信298

12.1 界面及源代码298

12.2 应用程序框架分解302

12.2.1 Socket 连接的建立303

12.2.2 基于 TCP 的 Socket 数据通信架构305

12.2.3 图形用户界面与事件处理界面的设计308

12.3 网络通信基础知识310

12.3.1 网络通信的层次310

12.3.2 通信端口311

12.3.3 Java 网络编程中主要使用的类和可能产生的异常311

12.3.4 Socket 通信模式312

12.3.5 Socket 类和 ServerSocket 类的构造方法及常用方法312

12.3.6 API 系统中 DataInputStream 和 DataOutputStream 的应用313

12.3.7 多线程处理机制315

习题 12316

参考文献317

第 1 章 Java 概述

多年来，为什么 Java 语言一直在程序设计语言排行榜中独占鳌头，发展成为程序设计语言的常青树，而精通 Java 语言也为职业规划提供很多优势？让我们循序渐进，走进 Java，学习 Java，掌握 Java。

本章主要内容

- Java 与 C、C++
- Java 语言的特点
- Java 开发工具
- Java 程序的类型及其不同编程模式

【案例分析】

使用面向对象方法，描述现实世界中的一个实体——售报亭，如图 1.1 所示。

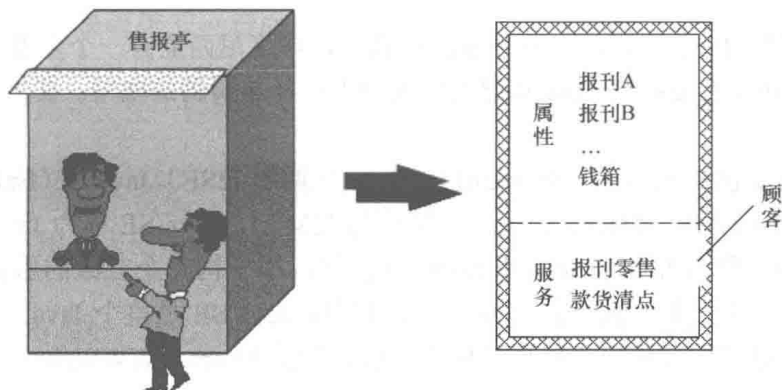


图 1.1 售报亭的对象封装

面向对象方法解决问题的思路是：从现实世界中的客观对象（如人和事物）入手，尽量运用人类的自然思维方式来构造软件系统。

在面向对象方法中，把一切都看成对象。把对象的属性和服务操作结合成一个独立的系统单位，其属性与操作刻画了事物的性质和行为，并尽可能隐蔽对象的内部细节，向外部只提供接口。软件对象是数据和方法的封装体。如图 1.1 所示，属性对应软件对象的数据，服务对应软件对象的方法。

在面向对象系统中，无论系统的构成成分，还是通过这些成分之间的关系而体现的系统结构，都可直接地映射问题域。这使得运用面向对象方法有利于我们正确理解问题域及系统责任。

1.1 Java 崛起

1991 年，美国 Sun Microsystems 公司启动了名为“Green Project”的研究项目，研究

解决家用电器的智能通信和控制问题。开发小组最初构想以当时颇为流行的 C++ 语言开发此项智能软件。后来,由于 C++ 语言本身的复杂性、安全性及平台移植方面的障碍等问题,因此项目组最后决定另辟蹊径,最终他们基于 C++ 重新开发了一套新的语言系统——Java 语言。

Java 语言的始创者是 Sun 公司的 James Gosling。起先,他根据办公室窗外的一棵橡树(Oak)将其命名为 Oak 语言。在申请注册时,因为命名冲突问题,后来将其改名为 Java 语言。

Java 语言可以称得上是一种精巧而安全的语言。然而,当时 Sun 公司一开始就遭遇了“智能化家用电器”市场的萧条。同时, Sun 公司以它投标一个自认为乐观的交互式电视项目时也折戟而归,未能成功。在这种情况下, Java 语言似乎生不逢时, Green 项目几乎走入了绝境。

可谓绝处逢生,峰回路转,1993 年万维网空前流行起来。Java 的发展转向了网络应用领域。

Java 语言具有平台无关性,使得 Java 程序适应了 Internet 上多样化的服务器站点环境,Java 程序既可以在 Windows 平台上运行,又可以在 Unix、Linux 等平台上运行。这造就了 Sun 公司宣传的“Write Once, Run Anywhere”(一次编写,随处运行)的优势。

1995 年 5 月, Sun 公司正式对外发布了 Java 语言,并随着互联网的飞速发展,逐渐确定了自己网络编程语言的地位。当年 Java 就被美国著名杂志《PC Magazine》评为十大优秀科技产品之一。1996 年 1 月, JDK 1.0 发布。2009 年 4 月,甲骨文公司收购 Sun 公司,取得了 Java 版权。

之所以命名为 Java 语言,有两种说法:其一,印度尼西亚有一个重要的岛屿——爪哇岛,盛产咖啡,开发人员起名 Java 寓意为世人端上一杯热腾腾的咖啡;其二,美洲俚语 Java 有咖啡之意。

Java 平台主要包括 Java SE(Standard Edition,早期的 J2SE)、Java EE(Enterprise Edition,早期的 J2EE)、Java ME(Micro Edition,早期的 J2ME)。Java SE 称为 Java 标准版或 Java 标准平台;Java EE 称为 Java 企业版或 Java 企业平台,用于构建企业级的服务应用;Java ME 为 Java 微型版,用于移动/嵌入式开发平台。其中,Java SE 是整个 Java 开发的基础。

总之,在计算机领域中很少出现过像现在所发生的 Internet/WWW/Java 这样的“火爆”现象。

延伸阅读:万维网和因特网

WWW 是环球信息网(World Wide Web)的缩写,简称为 Web,中文名称为“万维网”。万维网包括 WWW 服务器和 WWW 浏览器。万维网是一个资源空间,由“统一资源标识符”(URL)标识。这些资源通过超文本传输协议(Hypertext Transfer Protocol)传送给使用者,而后者通过点击链接来获得资源。

因特网(Internet)是当前全球最大的、开放的、由众多网络相互连接而成的计算机网络。万维网常常被视为因特网的同义词,其实万维网是依赖因特网运行的一项服务。万维网基于因特网,万维网被广泛应用于因特网之上。

延伸阅读:精神贵族——蒂姆·伯纳斯·李

蒂姆于 1955 年出生于英国伦敦,他是万维网的发明者。1989 年仲夏之夜,蒂姆成功开发出世界上第一个 Web 服务器和第一个 Web 客户机。并随后在 1990 年 12 月 25 日他成

功通过 Internet 实现了 HTTP 代理与服务器的第一次通信。2017 年，他因发明万维网、第一个浏览器和使万维网得以扩展的基本协议和算法而获得 2016 年度的图灵奖。他虽然放弃了万维网 http 的专利申请，但他成为了精神上最富有的人，是互联网的精神贵族，http 是他献给世界上每个人的互联网礼物。视频资料参阅央视纪录片《互联网时代》第一集 (<http://tv.cctv.com/2014/10/15/VIDA1413360557873609.shtml>)。

1.2 Java 与 C、C++

随着程序规模的不断扩大，在 19 世纪 60 年代末期出现了软件危机，当时的程序设计范型都无法克服错误随着代码的增多而级数般地扩大的问题，这个时候就出现了一种新的程序设计范型——面向对象程序设计。

1.2.1 Java 和 C++

Sun 微系统公司的 Java 开发小组汲取了 C++ 的精华，并将其融入到 Java 中，同时舍弃了 C++ 的低效率和不利于程序设计人员使用的缺点。Java 小组也创造了一些新的特性，给予 Java 开发基于 Internet 的应用程序时所必需的动态性。

Java 的目的并不是改进 C++ 进而最终取代 C++，C++ 和 Java 这两种语言是用来解决不同问题的。Java 用来设计必须共存于不同机器的应用程序，即常常是基于 Internet 的基础之上。相反，C++ 用来开发在一台特定机器上运行的程序，尽管 C++ 程序被重新编译后能够在其他机器上运行。

Java 语言的许多基本结构与 C++ 是相似的，有时甚至是相同的。例如，Java 是一种面向对象编程语言，它用类来创建对象的实例，类具有数据成员和方法成员，这和 C++ 中的类是相似的。

但是 Java 没有指针，而在 C/C++ 编程语言中指针是基石。在 C++ 中正确使用指针能使程序富有效率，但是指针难以掌握，若使用不当则会导致运行错误。

Java 带有自动的垃圾自动回收机制，这是在 C/C++ 中没有的功能。垃圾自动回收机制是一个常规程序，它收集程序中不再使用的内存。这样，程序设计人员就不必编写代码来释放之前使用的内存。

在不同的平台上使用 C/C++ 程序使系统会对每种数据类型根据平台的不同进而分配不同的字节数。而在 Java 中，Java 会为各种数据类型分配合理的固定位数，且位数在每种平台上都不改变，这样便保证了 Java 的平台无关性。

C++ 中支持多重继承，一个类可以有多个父类，这种方式使 C++ 中的类可以使用多个父类的属性和方法，但其结构复杂，容易引起混乱。而在 Java 中，一个类只能有一个父类，但是可以实现多个接口，这样既达到了多重继承的目的，又保证了结构比多重继承更加清晰。

除此之外，与 C++ 不同，Java 中不支持结构和联合，不支持宏定义，不支持头文件，不支持友元，大大保证了 Java 程序的安全性。

1.2.2 Java 与 C

C 语言为面向过程的程序设计语言。面向过程程序设计语言在程序设计过程中都倾向于

面向行为。在 C 语言中，程序设计的单元是函数，C 编程人员着重于编写函数。执行同一个任务的一系列动作构成函数，一系列函数再构成程序。这种语言的主要问题是程序中的数据和操作分离，不能够有效地组成与自然界中的具体事物紧密对应的程序成分。

Java 是纯面向对象的程序设计语言，Java 语言中程序设计的单元是类，从类中创建实例对象。Java 编程人员着重创建用户自定义的类，每个类均可包含数据属性和若干操作数据的函数，一个类的函数部分称为方法。C 和 Java 编程与执行过程的区别如下。

Windows 下 C 语言开发过程如图 1.2 所示。C 语言程序在执行之前需要把程序编译成机器语言文件，程序执行效率高，依赖专门编译器，跨平台性稍差。

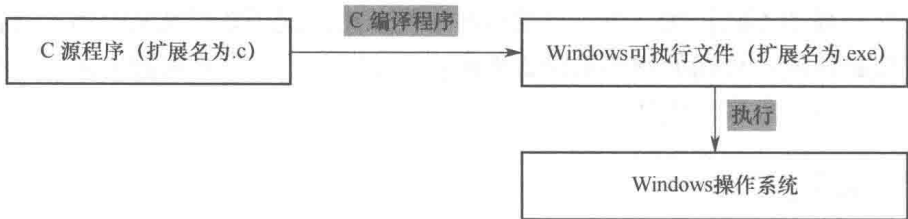


图 1.2 Windows 下 C 语言开发过程

Java 语言开发过程如图 1.3 所示。

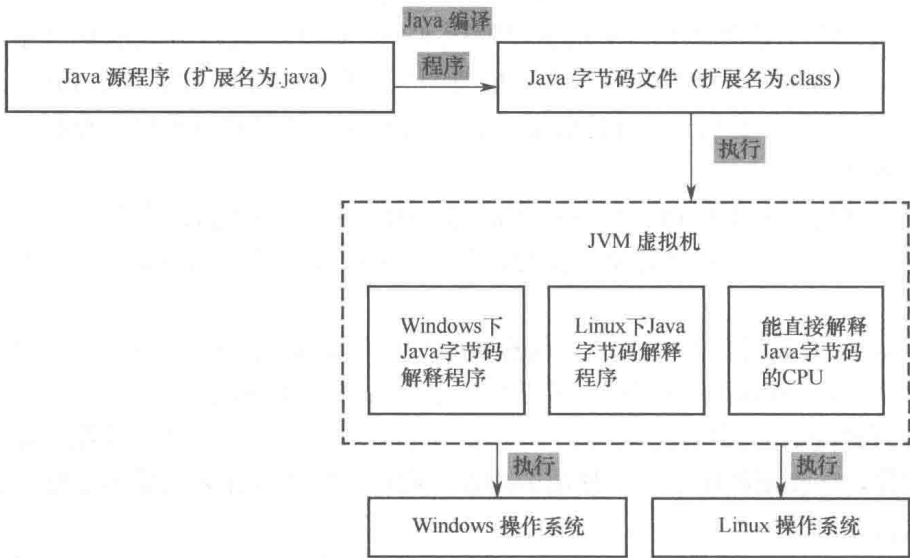


图 1.3 Java 语言开发过程

从图 1.2 和图 1.3 的比较中可以看出，Java 源程序编译后生成的字节码文件就相当于 C 源程序编译后 Windows 上的 exe 可执行文件，JVM（Java Virtual Machine）虚拟机的作用类似 Windows 操作系统。在 Windows 上运行的是 exe 文件，在 JVM 上运行的是 Java 字节码文件，即编译后生成的后缀为.class 的文件。

Windows 执行 exe 可执行文件的过程，就是从 exe 文件中取出一条条计算机指令交给 CPU 去解释、执行。字节码并不是机器指令，它与特定的平台无关，不能被任何平台直接识别、执行。字节码是可以被 Java 虚拟机识别、执行的代码，即 Java 虚拟机负责解释、运行字节码，将字节码翻译成所在平台的机器码，并让当前平台运行该机器码。可见，只要能实现特定平台下的解释器程序，Java 字节码就能通过解释器程序在该平台上运行，这是 Java

跨平台的根本特点。

Java 兼顾解释性与编译性语言的特点, Java 源文件转换成 class 字节码文件过程是编译型的, class 字节码文件在操作系统上运行的过程则是解释型的。Java 虚拟机充当了解释器的作用, C/C++ 都是编译型的语言, 运行速度较快。

延伸阅读: 程序设计语言发展脉络

计算机程序设计语言的发展是一个不断演化的过程, 从最开始的机器语言到汇编语言再到各种结构化高级语言, 最后发展到面向对象程序设计语言。

机器语言是第一代计算机语言, 是最原始的编程语言, 用二进制代码(0 或 1)书写, 能被机器直接识别, 二进制是计算机语言的基础。在计算机发展初期, 软件工程师们只能用晦涩的机器语言来编写程序。汇编语言将一个特定指令的二进制串机器指令映射为简洁的英文助记符。例如, 用“ADD”代表加法, 用“MOV”代表数据传递等, 它是比机器语言更“高级”的符号语言。高级语言是采用命令或语句的语言, 屏蔽了机器的细节问题, 提高了语言的抽象层次, 如我们正在学习的 Java。

1.3 Java 语言特点及更新

Java 语言是一种彻底的面向对象的程序设计语言。作为一种纯粹的面向对象的程序设计语言, 它非常适合大型软件的开发, 同时它又简单易学。2005 年 6 月, Sun 公司发布 Java SE 6, 此时, Java 的各种版本更名, J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。2014 年 3 月, Oracle 公司发表 Java SE 8。Java 9 于 2017 年 9 月正式发布。Java 9 带来了很大的变化, 其中最重要的改动是 Java 平台模块系统的引入。为了更快地迭代, 以及跟进社区反馈, Java 的版本更新周期加快, Java 10 于 2018 年 3 月发布, 最新特性大家可以参阅网上的最新资料。这里主要介绍 Java 语言核心特点。

1. 面向对象

对象是程序的基本单元和构件。在面向对象的程序语言中, 对象是类的实例, 而类则是描述对象的模板。类是具有相同属性和服务的一组对象的抽象、一般描述。抽象是事物的泛化, 抽象的目的是提取重要的特征而忽略不重要的细节。对象是现实世界中某个实际存在的事物, 软件对象是数据和方法的封装体。类与对象的关系如同一个模具与用这个模具铸造出来的铸件之间的关系, 如同自行车图纸与自行车的关系。

封装是面向对象的一个重要原则。它有两个含义, 第一个含义是, 把对象的全部属性和全部服务结合在一起, 形成一个不可分割的独立单位(对象); 第二个含义也称为“信息隐蔽”, 即尽可能隐蔽对象的内部细节, 对外形成一个边界(或者形成一道屏障), 只保留有限的对外接口使其与外部发生联系。这主要是指对象的外部不能直接地存取对象的属性, 只能通过几个允许外部使用的服务(或称方法)与对象发生联系。

2. 跨平台

这里所指的平台是由操作系统(OS)和处理器(CPU)所共同构成的平台。跨平台或与平台无关是指应用程序不因操作系统、处理器的变化而导致程序无法运行或出现运行错误。

用 Java 语言编写的程序, 经过 Java 编译器编译后生成 Java 语言特有的字节码(Bytecode),