

普通高等院校应用型人才培养“十三五”规划教材

Java程序设计 (慕课版)

Java CHENGXU SHEJI (MUKEBAN)

焦 铭 王映龙 刘青云 主编



中国铁道出版社有限公司
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

普通高等院校应用型人才培养“十三五”规划教材

Java 程序设计

(慕课版)

焦 铭 王映龙 刘青云 主 编
蒋 劼 李 翔 范双南 副主编

常州大学图书馆
藏书章

内 容 简 介

本书以面向对象的设计思想为主线,结合 Java 的最新特性,详细讲解了 Java 的基础语法,逐步引入面向对象思想,重点解释面向对象的三大特征、接口及应用和程序设计方法等重要知识点,并深入讲解字符串处理、标准类库、异常处理、输入/输出处理、图形化界面、多线程、JDBC 和网络编程等方面的编程方法。

本书注重可读性和实用性,内容全面、讲解细致,所有例题都经过精心的设计,既能帮助学生理解知识,又具有启发性。

本书适合作为普通高等院校计算机等相关专业 Java 语言程序设计教材,也可以作为全国计算机等级考试(二级)Java 程序设计的辅导用书,以及 Java 编程爱好者的自学参考书。

图书在版编目(CIP)数据

Java 程序设计:慕课版/焦 铭,王映龙,刘青云主编. —
北京:中国铁道出版社有限公司,2019.9
普通高等院校应用型人才培养“十三五”规划教材
ISBN 978-7-113-25943-3

I. ①J… II. ①焦… ②王… ③刘… 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(2019)第 171755 号

书 名:Java 程序设计(慕课版)
作 者:焦 铭 王映龙 刘青云

策 划:曹莉群

编辑部电话:010-63589185 转 2019

责任编辑:周海燕 彭立辉

封面设计:刘 颖

责任校对:张玉华

责任印制:郭向伟

出版发行:中国铁道出版社有限公司(100054,北京市西城区右安门西街8号)

网 址:<http://www.tdpress.com/51eds/>

印 刷:北京铭成印刷有限公司

版 次:2019年9月第1版 2019年9月第1次印刷

开 本:787 mm×1 092 mm 1/16 印张:18.5 字数:446 千

书 号:ISBN 978-7-113-25943-3

定 价:49.00 元

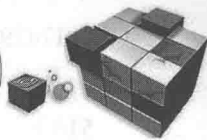
版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659



前 言



根据 2019 年 3 月 TOIBE 发布的编程语言排行榜,Java 语言仍是全球编程使用率最高的语言之一。市场调研机构 Gartner 公布了 2018 年第四季度智能手机市场报告,报告显示基于 Java 的 Android 操作系统占全球智能手机市场的 88%和平板计算机市场的 65%,Java 语言也广泛地应用于移动设备的编程。随着物联网和“互联网+”与传统行业的深度结合,社会需要大量精通 Java 的工程师,人才市场需求旺盛。

本书采用基础优先的方式,从编程基础开始,逐步引入面向对象思想,重点解释面向对象的三大特征和接口的编程方法,深入讲解 Java 输入/输出、异常处理、图形化界面、多线程、JDBC 和网络编程等知识点,注重教材的可读性和实用性,所有例题都经过精心的设计,既能帮助读者理解知识,又具有启发性。

本书的主要特点如下:

(1) 紧跟设计开发的步伐,很多章节都涉及 Java 的最新特性,软件都使用当前流行的较新版本。

(2) 每章都增加了一个综合案例设计,把本章及前面章节的知识点很好地串联起来,使读者能更好地理解和运用。

(3) 多线程是 Java 语言的一大特点,占有很重要的地位,通过有针对性的例子使读者掌握多线程的概念,并使用多线程来解决实际问题。

(4) 数据库的使用无处不在,本书中增加了深受中小企业欢迎的 MySQL 数据库的使用,重点讲解了 Java 使用 JDBC 操作 MySQL、SQL Server 数据库的方法。

(5) 增加了网络编程的知识,在读者学完 Java 语言的基础知识后,为学生深入学习网络编程打下基础。

(6) 开发了 MOOC 教学资源网站,方便线上线下学习。

本书由衡阳师范学院、江西农业大学、湖南交通工程学院的老师合作编写,由焦铭、王映龙、刘青云任主编,蒋劼、李翔、范双南任副主编。具体编写分工:第 1~3、12 章由焦铭和王映龙编写,第 6、7、9、11 章由蒋劼和范双南编写,第 4、5、8、13 章由刘青云编写,第 10 章由李翔编写。另外,林陆纲、雷天齐、赵军霞等老师参与了第 1、6、7 章的编写工作,全书由焦铭统稿。

在本书的编写过程中,编者根据多年的教学经验,结合 Java 的最新特性,整理出适于读者学习,并提供相应编程思路的源程序;开发了 MOOC 教学资源网站,方便线上线



下学习, Java 语言程序设计优质课程建设网站网址 <http://mooc1.chaoxing.com/course/201741979.html>。

本书得到湖南省普通高等学校教学改革研究项目(湘教通〔2018〕436号, No: 516)、教育部2018年第一批产学合作协同育人项目(201801193033)和衡阳师范学院优质课程“Java 语言程序设计”项目的支持,涉及的案例具有很强的实用性和代表性。

本书提供教学课件、源程序和习题答案,有教学需要的老师可以在中国铁道出版社有限公司的网站上下载,也可以发邮件向编者索取。编者的联系方式: jiaoge@126.com。

由于时间仓促,编者水平有限,书中疏漏和不妥之处在所难免,恳请广大读者批评指正。

编者
2019年5月

目 录



第 1 章 Java 开发入门	1
1.1 Java 语言的发展	1
1.2 Java 语言的特点	3
1.2.1 Java 语言的优点	3
1.2.2 Java 与其他程序设计语言的异同	4
1.3 Java 的运行与开发环境	5
1.3.1 Java JDK 下载与安装	5
1.3.2 配置环境变量	6
1.3.3 Java API 文档	7
1.4 Java 程序举例	8
1.4.1 Java 程序开发步骤	8
1.4.2 编写 Java 源程序	9
1.4.3 编译和运行	9
1.5 Eclipse 开发工具	10
1.5.1 Eclipse 的安装	10
1.5.2 Eclipse 的汉化	12
1.5.3 Eclipse 进行程序开发	13
1.5.4 Eclipse 程序调试	17
1.6 综合案例	19
小结	20
习题	21
第 2 章 Java 程序设计基础	22
2.1 标识符和关键字	22
2.1.1 标识符	22
2.1.2 关键字	23
2.1.3 注释	23
2.2 常量和变量	24
2.2.1 常量	24
2.2.2 变量与赋值	25
2.2.3 变量的作用域	25
2.3 基本数据类型	28
2.3.1 数据类型的分类	28
2.3.2 整数类型	29
2.3.3 浮点类型	30
2.3.4 字符类型	30
2.3.5 布尔类型	31
2.3.6 字符串类型	31



2.3.7	数据类型转换.....	32
2.4	运算符和表达式.....	33
2.4.1	算术运算符与算术表达式.....	33
2.4.2	赋值运算符与赋值表达式.....	34
2.4.3	关系运算符与关系表达式.....	35
2.4.4	逻辑运算符与逻辑表达式.....	36
2.4.5	条件运算符.....	36
2.4.6	位运算符.....	37
2.4.7	运算符的优先级.....	39
2.5	程序控制语句.....	40
2.5.1	语句概述.....	40
2.5.2	选择语句.....	40
2.5.3	循环语句.....	44
2.5.4	跳转语句.....	47
2.6	综合案例.....	48
小结	50
习题	51
第3章	数组.....	53
3.1	一维数组.....	53
3.1.1	一维数组的声明.....	53
3.1.2	一维数组分配空间.....	54
3.1.3	一维数组的初始化.....	55
3.1.4	一维数组的访问.....	55
3.1.5	一维数组的应用举例.....	56
3.2	二维数组.....	57
3.2.1	二维数组的声明.....	57
3.2.2	创建二维数组.....	58
3.2.3	二维数组的初始化.....	58
3.2.4	二维数组的访问.....	59
3.2.5	二维数组的应用举例.....	60
3.3	Arrays类.....	60
3.3.1	数组的排序.....	61
3.3.2	填充数据元素.....	61
3.3.3	数组的比较.....	62
3.3.4	数组的复制.....	62
3.3.5	数组的查找.....	64
3.4	综合案例.....	65
小结	67
习题	68
第4章	面向对象程序设计.....	70
4.1	Java程序结构.....	70
4.1.1	Java程序层次结构.....	70
4.1.2	Java工程结构及源码示例.....	71

4.2	类和对象	72
4.2.1	类的定义	72
4.2.2	对象的创建和使用	74
4.2.3	指针、引用与对象	75
4.2.4	对象的 this 引用	76
4.2.5	类成员和实例成员	77
4.3	成员变量与局部变量	78
4.3.1	成员变量的定义和使用	78
4.3.2	局部变量的定义和使用	79
4.3.3	变量同名问题	80
4.4	构造器	82
4.4.1	构造器的定义	82
4.4.2	构造器的调用	83
4.5	初始化块	83
4.5.1	代码块	84
4.5.2	普通初始化块	86
4.5.3	静态初始化块	88
4.5.4	初始化块的执行	89
4.6	成员方法	91
4.6.1	方法的定义和调用	92
4.6.2	方法的重载	93
4.6.3	方法的参数传递机制	93
4.6.4	实参长度可变的方法	95
4.7	封装性	96
4.7.1	包 package	96
4.7.2	Java 常用包	97
4.7.3	类成员的封装和访问控制	97
4.7.4	类的访问控制和 import 语句	99
4.8	继承性	100
4.8.1	类的继承	101
4.8.2	方法的重写	102
4.8.3	super 关键字	103
4.9	多态性	105
4.9.1	实现多态	105
4.9.2	引用变量的强制类型转换	107
4.9.3	instanceof 关键字	108
4.10	综合案例	109
	小结	111
	习题	111
第 5 章	面向对象高级程序设计	114
5.1	类和对象的生存周期	114
5.1.1	类的加载	114
5.1.2	对象的销毁	115

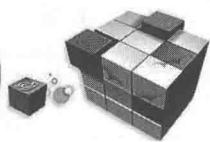


5.1.3	堆栈和常量池	116
5.2	final 修饰符	118
5.2.1	final 修饰成员变量	118
5.2.2	final 修饰局部变量	119
5.2.3	final 修饰变量的本质	120
5.2.4	final 修饰方法	121
5.2.5	final 修饰类	121
5.3	抽象类	122
5.3.1	抽象类和抽象方法	122
5.3.2	抽象类的作用	123
5.4	接口	124
5.4.1	接口的定义	125
5.4.2	接口继承和使用	126
5.5	内部类	128
5.5.1	成员内部类的定义	128
5.5.2	成员内部类的使用	130
5.5.3	匿名内部类	132
5.6	枚举类	133
5.6.1	枚举类的简单使用	133
5.6.2	枚举类的定义	135
5.7	其他特殊类	136
5.7.1	java.lang.Object	136
5.7.2	单例类	137
5.7.3	不可变类	138
5.8	综合案例	139
	小结	141
	习题	142
第 6 章	字符串处理	144
6.1	String 类	144
6.1.1	构造字符串	144
6.1.2	操作字符串的主要方法	145
6.2	StringBuffer 和 StringBuilder 类	146
6.3	综合案例	147
	小结	152
	习题	152
第 7 章	Java 标准类库	153
7.1	数据类型包装器	153
7.1.1	基本数据类型对应的包装类	153
7.1.2	包装类的使用	154
7.2	System 和 Runtime 类	154
7.3	Math 和 Random 类	155
7.4	日期时间实用工具类	159
7.5	集合类	165

7.5.1	集合框架和泛型	165
7.5.2	Collection 接口	167
7.5.3	Set 接口及其实现	167
7.5.4	List 接口及其实现	171
7.5.5	Map 接口及其实现	172
7.6	综合案例	174
	小结	182
	习题	183
第 8 章	异常处理机制	185
8.1	异常概述	185
8.2	异常的体系结构	186
8.3	异常处理机制	187
8.3.1	捕获异常	187
8.3.2	声明抛弃异常	190
8.3.3	访问异常信息	191
8.4	手动抛出异常	191
8.5	自定义异常类	192
8.6	异常处理规则	194
8.7	综合案例	195
	小结	200
	习题	200
第 9 章	输入/输出处理	203
9.1	I/O 流的概念和划分	203
9.2	字节流和字符流处理	204
9.3	串行化	206
9.4	综合案例	208
	小结	210
	习题	211
第 10 章	多线程	213
10.1	多线程的概念和创建	213
10.2	线程的调度及生命周期	214
10.2.1	线程调度	215
10.2.2	线程生命周期	218
10.3	多线程同步	219
10.3.1	多线程同步	219
10.3.2	Lock 锁的使用	222
10.4	线程间死锁与通信	223
10.4.1	线程死锁问题	223
10.4.2	线程间通信	225
10.5	综合案例	227
	小结	229
	习题	230



第 11 章 GUI 程序设计	232
11.1 GUI 类的发展	232
11.2 JavaFX 程序的基本结构	232
11.3 事件驱动的 GUI 程序	233
11.4 综合案例	235
小结	238
习题	238
第 12 章 JDBC 数据库编程	239
12.1 JDBC 体系结构	239
12.1.1 JDBC 的结构	239
12.1.2 JDBC API	240
12.2 MySQL 数据库	240
12.2.1 在 Windows 系统上安装 MySQL	240
12.2.2 使用 MySQL 命令行工具	242
12.2.3 使用 Navicat 操作 MySQL 数据库	245
12.3 通过 JDBC 访问数据库	246
12.3.1 数据库的访问步骤	246
12.3.2 访问 MySQL 数据库	248
12.3.3 访问 SQLServer 数据库	252
12.4 综合案例	255
小结	263
习题	264
第 13 章 网络编程	266
13.1 网络编程基础	266
13.1.1 网络基础知识	266
13.1.2 InetAddress 类	267
13.2 URL 通信	269
13.2.1 URL 类	269
13.2.2 URLConnection 类	271
13.3 TCP 通信	272
13.3.1 使用 ServerSocket 创建服务器端	272
13.3.2 使用 Socket 进行通信	274
13.4 UDP 通信	275
13.4.1 DatagramSocket 类和 DatagramPacket 类	275
13.4.2 使用 DatagramSocket 通信	277
13.5 综合案例	279
小结	283
习题	284
参考文献	286



学习目标:

- 了解 Java 语言的发展及特点。
- 掌握 JDK 的安装与环境变量的配置。
- 掌握 Java 程序的编写、编译和运行,学会使用 javac 命令编译程序,使用 java 命令执行程序。
- 了解字节码与 Java 虚拟机,理解 Java 程序的运行机制。
- 掌握 Eclipse 创建 Java 程序的方法。
- 掌握 Eclipse 调试程序的方法。

Java 语言是目前十分流行的面向对象程序设计语言。它具有简单性、跨平台性、安全性和分布性等优点。Java 语言不但在网络编程和面向对象编程中占据主导地位,而且在移动设备和企业开发中也有广泛应用。本章将对 Java 语言的特点、开发运行环境、运行机制以及如何使用 Eclipse 创建、调试和运行 Java 程序等内容进行介绍。



1.1 Java 语言的发展

新技术的应用离不开编程语言,根据 2019 年 3 月 TOIBE 发布的编程语言排行榜。(见图 1-1),Java 程序设计语言以 16.904% 的占有率继续排名榜首,从 2002—2018 年编程语言 TOIBE 指数走势图(见图 1-2)可以看出,Java 语言仍是全球编程使用率最高的语言之一。市场调研机构 Gartner 公布了 2018 年第四季度智能手机市场报告,报告显示基于 Java 的 Android 操作系统占全球智能手机市场的 88% 和平板计算机市场的 65%,Java 语言也广泛应用于移动设备编程。

Java 语言是由 James Gosling 等人开发的一种面向对象的编程语言。Java 最早是在 1991 年的 GREEN 项目诞生的,但是其原本的名字不叫 Java,而是称为 OAK(橡树)。Sun 公司于 1995 年 5 月 23 日正式发布 Java 语言。2009 年 4 月 20 日,Oracle 公司收购 Sun 公司,之后由 Oracle 公司负责 Java 的维护和版本升级。Java 语言的发展历程如图 1-3 所示。

Java 不但可以用来开发网站后台、PC 客户端和 Android APP,还在大数据分析、



网络爬虫、云计算等领域大显身手。Java 平台按应用领域可分为 Java SE、Java EE 和 Java ME 三大版本。

Jan 2019	Jan 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.904%	+2.69%
2	2		C	13.337%	+2.30%
3	4	^	Python	8.294%	+3.62%
4	3	v	C++	8.158%	+2.55%
5	7	^	Visual Basic .NET	6.459%	+3.20%
6	6		JavaScript	3.302%	-0.16%
7	5	v	C#	3.284%	-0.47%
8	9	^	PHP	2.680%	+0.15%
9	-	^	SQL	2.277%	+2.28%
10	16	^	Objective-C	1.781%	-0.08%

图 1-1 2019 年 1 月编程语言排行榜

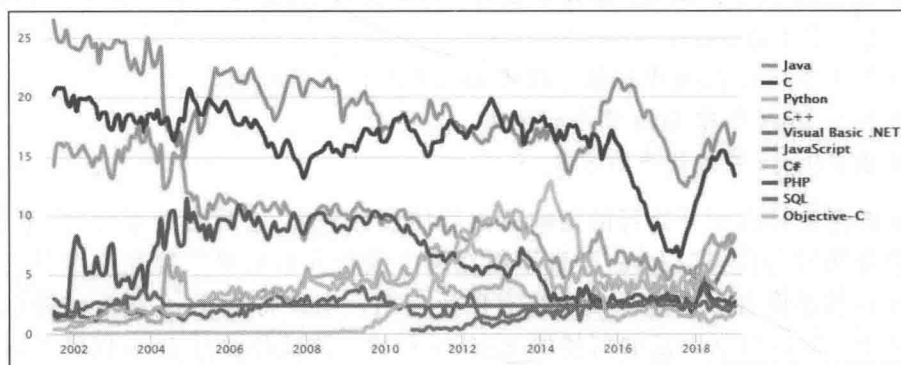


图 1-2 编程语言指数走势（2002—2018）

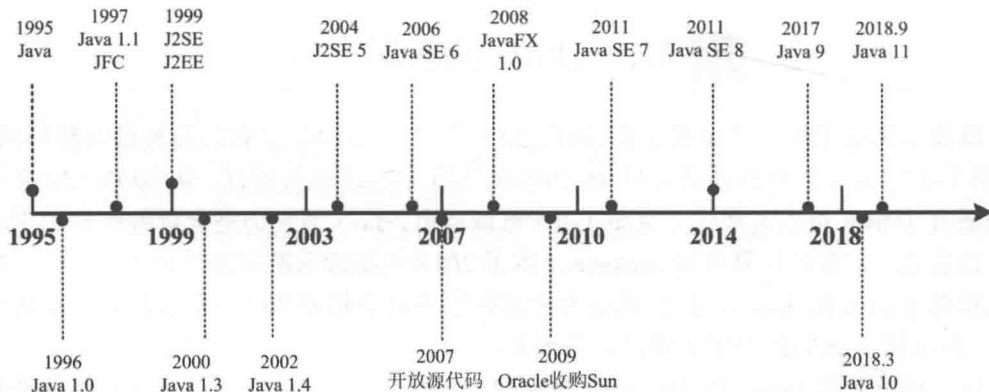


图 1-3 Java 语言发展历程

Java SE (Java Platform Standard Edition, Java 平台标准版) 以前称为 J2SE, 它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 包含了支持 Java Web 服务开发的类, 并为 Java EE 提供基础, 如 Java 语言基础、JDBC 操作、I/O 操作、网络通信以及多线程等技术。

Java EE (Java Platform Enterprise Edition, Java 平台企业版) 以前称为 J2EE。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 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。

Java ME (Java Platform Micro Edition, Java 平台微型版) 以前称为 J2ME, 也称 K-Java。Java ME 为在移动设备和嵌入式设备 (手机、PDA、电视机顶盒和打印机) 上运行的应用程序提供一个健壮且灵活的环境。

1.2 Java 语言的特点

1.2.1 Java 语言的优点

Java 是目前使用最广泛的网络编程语言之一, 它具有语法简单、面向对象、平台无关性、多线程、分布式、健壮性和安全性等优点。

1. 简单性

Java 语言的语法与 C 语言和 C++ 语言很相近, 使得很多程序员学起来很容易。Java 舍弃了很多 C++ 中难以理解的特性, 如操作符的重载和多继承等, 而且不使用指针, 加入了垃圾回收机制, 解决了程序员需要管理内存的问题, 使编程变得更加简单。

2. 面向对象

Java 是一种面向对象的语言, 它对面向对象中的类、对象、继承、封装、多态、接口、包等均有很好的支持。为了简便, Java 只支持类之间的单继承, 但是可以使用接口来实现多继承。使用 Java 语言开发程序, 需要采用面向对象的思想设计程序和编写代码。

3. 平台无关性

Java 是“一次编写, 到处运行”的语言, 它并不生成可执行文件 (.exe), 而是生成一种中间字节码文件 (.class), 任何操作系统, 只要安装了 Java 虚拟机 (Java Virtual Machine, JVM) 就可以解释并执行中间字节码文件。Java 语言使用 Java 虚拟机机制屏蔽了具体平台的相关信息, 使得 Java 语言编译的程序只需生成虚拟机上的目标代码, 就可以在多种平台上不加修改地运行。因此, 采用 Java 语言编写的程序具有很好的可移植性。

4. 多线程

Java 语言是多线程的, 这也是 Java 语言的一大特性, 它必须由 Thread 类和它的子类来创建。Java 支持多个线程同时执行, 并提供多线程之间的同步机制。任何一个线程都有自己的 run() 方法, 要执行的方法就写在 run() 方法体内。

5. 分布式

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。因此, Java 应用程序可凭借 URL 打开并访问网络上的对象, 其访问方式与访问本地文件系统几乎完全相同。Java 的远程方法调用 (Remote Method Invocation, RMI) 机制也是开发分布式应用的重要手段。



6. 健壮性

Java 的强类型机制、异常处理、垃圾回收机制等都是 Java 健壮性的重要保证。对指针的丢弃是 Java 的一大进步。另外, Java 的异常机制也是健壮性的一大体现。

7. 安全性

Java 通常被用在网络环境中, 为此, Java 提供了一个安全机制以防止恶意代码攻击。Java 语言除了具有许多安全特性以外, 还对通过网络下载的类增加一个安全防范机制, 分配不同的名字空间以防替代本地的同名类, 并包含安全管理机制。

Java 语言的众多特性使其在众多的编程语言中占有较大的市场份额, Java 语言对对象的支持和强大的 API 使得编程工作变得更加容易和快捷, 大大降低了程序的开发成本。

1.2.2 Java 与其他程序设计语言的异同

Java 语言是一种纯粹的面向对象语言, 它继承了 C++ 语言面向对象的核心技术, 使用了类似于 C/C++ 的语法, 而去除了 C/C++ 中许多不合理的内容, 以实现其简单、健壮、安全等特性。下面列出几点主要的区别:

1. 全局变量

Java 语言中不存在全局变量, 只能通过类中的公用的静态变量实现全局变量, 全局变量被封装在类中, 这样更好地保证了安全性。而 C++ 兼具面向对象和面向过程变量的特点, 可以定义全局变量。

2. 指针

与 C/C++ 语言相比, Java 语言中没有指针的概念, 有效防止了 C/C++ 语言中操作指针可能引起的系统问题, 从而使程序变得更加安全。同时, 数组在 Java 中用类来实现, 很好地解决了数组越界的问题。

3. 内存管理

Java 和 C++ 一样通过 new 创建一个对象分配内存空间, 但与 C++ 不同的是 Java 并没有与 new 相对应的 delete 操作符, 而是提供了垃圾回收器来实现垃圾的自动回收, 不需要程序显式地管理内存的分配。在 C++ 语言中, 通常都会把释放资源的代码放到析构函数中; Java 语言中虽然没有析构函数, 但却引入了一个 finalize() 方法, 当垃圾回收器将要释放无用对象的内存时, 会首先调用该对象的 finalize() 方法, 因此, 开发人员不需要关心也不需要知道对象所占的内存空间何时会被释放, 只需要停止对一个对象的引用, 一段时间后垃圾回收器会自动收集这个对象所占的内存。

4. 类型转换

C++ 支持自动强制类型转换, 这会导致程序不安全; Java 不支持自动强制类型转换, 必须由开发人员进行显式的强制类型转换。C++ 同一个数据类型在不同的平台上会分配不同的字节数, Java 对每种数据类型都分配固定长度, 例如, int 类型总是占据 32 位。

5. 结构和联合

C++ 中结构和联合的所有成员均为公有, 这会导致安全性问题的发生, 而 Java 根本就不包含结构和联合, 所有的内容都封装在类中。



6. 预处理

C++语言支持预处理,而Java语言没有预处理器。Java虽然不支持预处理功能(头文件、宏定义等),但它提供的import机制和C++中的预处理器功能类似。

7. 编译方式

Java为解释性语言,其运行过程为:程序源代码经过Java编译器编译成字节码,然后由JVM解释执行。而C/C++为编译型语言,源代码经过编译和连接后生成可执行的二进制代码。因此,Java的执行速度比C/C++慢,但是Java能够跨平台执行,而C/C++不能。

8. 面向对象

(1) Java为纯面向对象语言,所有代码(包括函数、变量等)必须在类中实现,除基本数据类型(包括int、float等)外,所有类型都是类。

(2)与C++语言相比,Java语言不支持多重继承,但是Java引入了接口的概念,可以同时实现多个接口。由于接口也具有多态特性,因此在Java语言中可以通过实现多个接口来实现与C++语言中多重继承类似的功能。

(3)C++语言支持运算符重载,而Java语言不支持运算符重载。



1.3 Java 的运行与开发环境

1.3.1 Java JDK 下载与安装

Java提供了一个免费的Java开发工具集(Java Development Kits, JDK)。JDK是一种用于构建在Java平台上发布的应用程序、Applet和组件的开发环境,即编写Java程序必须使用JDK,它提供了编译和运行Java程序的环境。本书中使用的是JDK 8,读者在编译和运行本书的程序时,请使用JDK 8或更高版本。

可从Oracle官方网站<http://www.oracle.com/>免费下载JDK,根据计算机的操作系统不同下载相应的文件。JDK安装包被集成在Java SE中,因此下载Java SE即可。假设下载的64位JDK 8,文件名为jdk-8u201-windows-x64.exe,要安装在64位的Windows 7上,双击该文件开始安装,默认路径是C:\Program Files\Java\jdk1.8.0_201目录,可以通过单击“更改”按钮指定新的位置,如图1-4所示。



图 1-4 选择安装组件及路径



安装完 JDK 后系统自动安装 JRE，安装过程可以采用默认的配置方式，所以可以连续单击“下一步”按钮直至安装完成。安装完成后，在安装位置打开 JDK 的文件夹，内容和目录结构如图 1-5 所示。

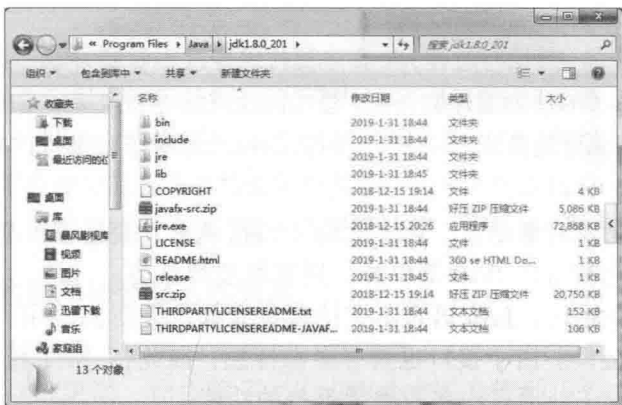


图 1-5 JDK 内容和目录结构

从图 1-5 可以看出，JDK 安装目录下具有多个子目录和一些网页文件，其中重要目录和文件的说明如下：

(1) bin：提供 JDK 工具程序，包括 javac、java、javadoc、appletviewer 等可执行程序。

(2) include：存放用于本地访问的文件。

(3) jre：存放 Java 运行环境文件。

(4) lib：存放 Java 的类库文件，工具程序实际上使用的是 Java 类库。JDK 中的工具程序，大多也由 Java 编写而成。

(5) src.zip：Java 提供的 API 类的源代码压缩文件。如果需要查看 API 的某些功能是如何实现的，可以查看这个文件中的源代码内容。

1.3.2 配置环境变量

JDK 环境变量的配置过程如下：

(1) 右击桌面上的“计算机”图标，在弹出的快捷菜单中选择“属性”命令，单击“高级系统设置”，打开如图 1-6 所示“系统属性”对话框。

(2) 在“系统属性”对话框中单击“环境变量”按钮，在打开的“环境变量”对话框（见图 1-7）中单击“系统变量”列表框下方的“新建”按钮。

(3) 在打开的“新建系统变量”对话框的变量名中输入 JAVA_HOME，在变量值中输入 JDK 的根目录 C:\Program Files\Java\jdk1.8.0_201，然后单击“确定”按钮，返回“环境变量”对话框。再次单击“新建”按钮，分别输入 CLASSPATH 和“.;%JAVA_HOME%\lib;”，如图 1-8 所示，然后单击“确定”按钮。

(4) 在如图 1-7 所示的“系统变量”列表框中双击 Path 变量，打开“编辑系统变量”对话框。在“变量值”文本框的最后添加“;%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin”，最后单击“确定”按钮，如图 1-9 所示。