



Java程序设计

杨文艳 田春尧 主编

 北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

Java 程序设计

主 编：杨文艳 田春尧

副主编：王 皓 陈 磊 赵 妍 张旭辉 徐晓东

参 编：邹 飞

主 审：洪运国



内 容 简 介

本书通过 12 个模块 21 个项目案例，系统、简明地介绍了 Java 语言程序设计的核心技术。它简明扼要地介绍了面向对象程序设计的基本概念、Java 语言的特点以及 Java 程序的开发过程；快速回顾了编程基础语法，系统介绍了 Java 面向对象编程基础和高级语法，并通过典型应用案例对异常处理、常用 API、集合类、流式 I/O、JDBC 编程、多线程、GUI 编程、网络编程等专项进行学习和训练。

本书附带配套微课视频、源代码、知识题库、编程题库、教学 PPT、课程教案等资源，方便读者学习和使用。

本书既可用于各院校计算机相关专业 Java 程序设计课程的教材，也可作为 Java 自学者的入门用书。阅读本书只需要对计算机和编程有一般性了解。

版权专有 侵权必究

图书在版编目 (CIP) 数据

Java 程序设计 / 杨文艳, 田春尧主编. —北京: 北京理工大学出版社, 2018. 3

ISBN 978-7-5682-5467-0

I. ①J… II. ①杨… ②田… III. ①JAVA 语言-程序设计-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 059882 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

(010) 82562903 (教材售后服务热线)

(010) 68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 山东临沂新华印刷物流集团有限责任公司

开 本 / 787 毫米×1092 毫米 1/16

印 张 / 20

字 数 / 470 千字

版 次 / 2018 年 3 月第 1 版 2018 年 3 月第 1 次印刷

定 价 / 75.00 元

责任编辑 / 高 芳

文案编辑 / 高 芳

责任校对 / 周瑞红

责任印制 / 施胜娟

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换

前 言

Java 语言是一种优秀的程序设计语言，是计算机世界的“国际语言”，因其具有平台无关性、性能优异、安全性等特点，自 1995 年问世以来便受到了广大编程人员的喜爱。Java 拥有最广泛的应用领域和软件开发市场，从大型复杂的企业级开发到移动物联网设备的开发，随处都可以看到 Java 的身影。

作为软件开发类课程的入门和基础课程，“Java 程序设计”秉承“做中学，理论为指导、实践出真知”的理念，在把握编程知识的基础上，通过阅读和编写大量的程序实例让读者轻松理解并快速掌握编程技能。全书通过 12 个模块、大量的例题程序和 21 个阶段案例，由浅入深、从易到难地介绍了 Java SE 的核心技术。

立体化和阶段案例的巧妙穿插是本教材的特色和亮点。针对 12 个模块的重点内容，全面设置了以单个知识点或技能点为单位的微课视频，使学习者随时可以利用碎片时间或在需要的时候来完成学习，有效补充了课堂学习时间的不足，弥补了个体学习的差异性；每个模块精心设计的 1~2 个阶段案例将知识和技能融入其中，由案例描述、设计目标、实现思路、参考代码 4 部分组成。其中，“案例描述”通过对案例任务的解读和运行结果的展示，使学习者可以对所完成的案例任务一目了然；“设计目标”是对知识点的掌握要求和对案例项目的掌握要求；“实现思路”是阶段案例的精髓，通过分析实现思路，让学习者对如何应用所学知识和技能完成任务有清晰的逻辑思路；有了思路后，每个人就可以按照自己的理解来通过代码实现案例功能，而不是照搬照抄参考代码，真正达到提高编程技能的目的。

模块 1 主要介绍了 Java 语言的特点和面向对象程序设计的基本概念、Java 开发环境的搭建和程序的开发过程。

模块 2 主要学习 Java 编程的基础知识，包括 Java 程序基本结构、标识符、关键字、数据类型、变量和方法、运算符与表达式，以及选择结构、循环结构的使用、数组的使用等。

模块 3 和模块 4 系统学习 Java 面向对象编程基础和高级语法，使学习者掌握类和对象、构造方法、static 关键字及内部类的使用；掌握类的继承、抽象类、接口、多态及类和类成员的访问控制方法。

模块 5~模块 8 介绍了 Java 异常处理、常用 API、集合类、流式 I/O，这些知识是今后 Java 开发中常用的知识，应在学习时认真理解和把握，完成每个案例程序。

模块 9~模块 12 介绍了 JDBC、多线程、GUI 编程及网络编程专项知识和技术，可以根据后续课程需要和专业方向重点学习部分或全部内容，各院校和学习者可根据课时和需要灵活把握。

本书不仅可以作为传统课堂教学的教材，也适用于翻转课堂教学模式和自学者。方便灵活的二维码扫描可以让学习者随时随地利用碎片时间完成知识学习，并在课堂等大块时间完成程序和案例的编写调试，解决学习中遇到的问题。配合学习网站，可以让老师和学生更方便地完成教和学以及考核、反馈等。

本书由杨文艳、田春尧主编，多所院校一线任课教师共同完成教材的编写工作。在编写过程中，各位老师付出了很多辛勤的汗水，在此一并表示衷心的感谢。

编 者

目 录

模块 1 欢迎走进 Java 世界	1
1.1 Java 与面向对象程序设计	1
1.1.1 什么是面向对象程序设计	1
1.1.2 OOP 的三大特性	3
1.2 开始 Java 程序开发	4
1.2.1 Java 的起源及特点	4
1.2.2 JDK 的使用及环境变量	5
1.2.3 HelloWorld 程序开发	10
1.3 集成开发环境 Eclipse	11
1.3.1 Eclipse 安装与启动	11
1.3.2 使用 Eclipse 开发程序	13
【案例 1】ATM 存取款系统界面设计	15
习题 1	16
模块 2 Java 编程基础语法	18
2.1 Java 基本语法	18
2.1.1 Java 程序基本结构	18
2.1.2 标识符和关键字	19
2.1.3 Java 数据类型	20
2.2 Java 变量与方法	22
2.2.1 变量的定义及类型转换	22
2.2.2 方法的定义及方法重载	23
2.2.3 变量的作用域	25
2.3 运算符和表达式	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 条件运算符	29
2.3.5 运算符的优先级与结合性	30
【案例 2-1】数字分割	31
2.4 结构化程序设计	31
2.4.1 选择结构	32
2.4.2 循环结构	36
2.4.3 跳转语句与多重循环	39
【案例 2-2】猜数字游戏	41



2.5	数组	42
2.5.1	一维数组的定义及使用	42
2.5.2	多维数组的定义及使用	44
	【案例 2-3】商品查询器	46
习题 2		48
模块 3	面向对象基础	49
3.1	类与对象	49
3.1.1	类的定义	49
3.1.2	对象的创建与使用	51
3.1.3	类的封装	53
3.2	构造方法及 this 关键字	55
3.2.1	构造方法的定义	55
3.2.2	构造方法的重载	55
3.2.3	this 关键字	56
	【案例 3-1】简单几何图形类的封装	58
3.3	static 关键字	61
3.3.1	静态变量	61
3.3.2	静态方法	62
3.3.3	静态代码块	63
3.3.4	单例模式	64
3.4	内部类	65
	【案例 3-2】银行卡开户程序设计	68
习题 3		70
模块 4	面向对象进阶	72
4.1	类的继承及 super 关键字	72
4.1.1	继承的实现	72
4.1.2	方法的重写	74
4.1.3	super 关键字	75
4.2	final 关键字	77
4.2.1	final 类	78
4.2.2	final 方法	78
4.2.3	final 变量	79
4.3	抽象类和接口	80
4.3.1	抽象类	80
4.3.2	接口	82
	【案例 4-1】图形计算程序设计	85
4.4	多态	88
4.4.1	对象的类型转换	88
4.4.2	多态性的实现	90



4.4.3 匿名内部类	91
4.5 包与访问权限	93
4.5.1 package 关键字	93
4.5.2 import 关键字	94
4.5.3 访问权限控制	95
【案例 4-2】银行存款程序设计	98
习题 4	102
模块 5 Java 异常处理	104
5.1 异常及其分类	104
5.1.1 什么是异常	104
5.1.2 异常分类	105
5.2 异常的处理	107
5.2.1 捕获异常	107
5.2.2 抛出异常	109
5.3 自定义异常	111
【案例 5-1】学生信息的录入	112
习题 5	116
模块 6 Java 常用 API	117
6.1 Java 类库	117
6.2 数据类型包装类	118
6.3 字符串	119
6.3.1 String 类	119
6.3.2 StringBuffer 类	121
6.3.3 StringTokenizer 类	123
【案例 6-1】统计单词个数	123
6.4 日期类	124
6.4.1 Date 类	125
6.4.2 Calendar 类	126
6.4.3 GregorianCalendar 类	128
6.5 数据操作类 Math 与 Random	129
6.5.1 Math 类	129
6.5.2 Random 类	131
【案例 6-2】随机安排座位号	132
习题 6	134
模块 7 集合类	136
7.1 集合概述	136
7.1.1 集合的概念和分类	136
7.1.2 Collection 接口	137
7.2 List 接口	137



7.2.1 ArrayList 集合	138
7.2.2 LinkedList 集合	139
7.2.3 Iterator 迭代器	141
7.2.4 foreach 循环	143
7.2.5 泛型	144
【案例 7-1】图书查询程序设计	147
7.3 Set 接口	149
7.3.1 HashSet 集合	149
7.3.2 TreeSet 集合	152
7.4 Map 接口	156
7.4.1 HashMap 集合	156
7.4.2 TreeMap 集合	160
7.5 集合及数组工具类	160
7.5.1 Collections 工具类	161
7.5.2 Arrays 工具类	163
【案例 7-2】学生成绩排序程序设计	165
习题 7	167
模块 8 Java 流式 I/O 技术	169
8.1 流式 I/O 概述	169
8.1.1 Java I/O 简介	169
8.1.2 I/O 流的分类	169
8.2 文件操作类	170
8.2.1 File 类	170
8.2.2 RandomAccessFile 类	174
【案例 8-1】文件检索系统	176
8.3 字节流	181
8.3.1 字节输入流 InputStream	181
8.3.2 字节输出流 OutputStream	182
8.3.3 文件字节流	183
8.4 字符流	185
8.4.1 字符输入流 Reader	185
8.4.2 字符输出流 Writer	186
8.4.3 文件字符流	187
8.4.4 缓冲流	189
8.4.5 转换流	191
【案例 8-2】简易文本文件编辑器	193
8.5 其他 I/O 流	197
8.5.1 对象输入/输出流	197
8.5.2 PrintStream	199



8.5.3 管道输入/输出流	201
8.5.4 字节数组输入/输出流	202
习题 8	202
模块 9 Java 数据库连接技术	204
9.1 MySQL 数据库管理系统	204
9.1.1 下载、安装 MySQL	204
9.1.2 建立数据库	207
9.2 JDBC 技术	211
9.2.1 JDBC 概述	211
9.2.2 JDBC 常用 API	212
9.2.3 数据库常见操作	217
9.2.4 使用 PreparedStatement	220
9.2.5 使用 CallableStatement	222
【案例 9-1】使用 JDBC 实现学生成绩管理系统	225
习题 9	228
模块 10 多线程编程	230
10.1 多线程概述	230
10.1.1 进程与线程	230
10.1.2 线程的生命周期及状态转换	231
10.1.3 线程的优先级	234
10.2 线程的创建	234
10.2.1 继承 Thread 类创建多线程	235
10.2.2 实现 Runnable 接口创建多线程	236
10.2.3 两种实现多线程方式的对比	237
10.3 线程控制问题	238
10.3.1 线程休眠	238
10.3.2 线程让步与插队	240
10.3.3 线程同步与死锁	242
【案例 10-1】模拟铁路售票系统程序设计	249
习题 10	251
模块 11 Java GUI 编程	252
11.1 GUI 编程概述	252
11.2 GUI 界面设计	254
11.2.1 界面组件类	254
11.2.2 界面布局管理	263
11.2.3 菜单及菜单组件	271
【案例 11-1】学生成绩管理系统界面设计	273
11.3 GUI 事件处理	276
11.3.1 事件处理机制	276



11.3.2 GUI 事件处理	278
【案例 11-2】Java 简易计算器设计	283
习题 11	287
模块 12 网络编程	288
12.1 网络编程基础	288
12.1.1 TCP/IP 协议	288
12.1.2 IP 地址和端口号	288
12.1.3 InetAddress	289
12.1.4 UDP 与 TCP 协议	290
12.2 Socket 编程	291
12.2.1 Socket 概述	291
12.2.2 Socket 类和 ServerSocket 类	292
【案例 12-1】Server 和多客户的通信程序	296
12.3 数据报编程	298
12.3.1 数据报通信概述	298
12.3.2 UDP 通信程序	300
【案例 12-2】聊天程序设计	303
习题 12	307

模块 1

欢迎走进 Java 世界

学习目标：

- 了解面向对象编程 OOP 的基本概念和主要特性
- 了解 Java 语言及其相关特性，掌握 Java 开发环境的搭建
- 熟悉 Eclipse 开发环境，能进行 HelloWorld 程序的开发

1.1 Java 与面向对象程序设计



OOP 简介

首先，欢迎大家走进 Java 世界。Java 是一种优秀的面向对象程序设计语言，是一种非常流行的软件开发技术框架，拥有最广泛的应用领域和软件开发市场。为了更好地理解和掌握 Java 编程技术，在正式开始学习 Java 之前，先来了解一下什么是面向对象程序设计（OOP）以及它的特性。

1.1.1 什么是面向对象程序设计

面向对象程序设计（Object Oriented Programming）简称 OOP，是一种符合人类思维习惯的编程思想，也是一种主流的程序开发方法。现实生活中存在各种不同的事物，这些事物之间存在着各种联系，在程序中使用对象来描述事物，使用对象之间的关系来描述事物之间的联系，这种思想就是面向对象。

面向对象和面向过程是两种不同的程序开发方法。面向过程就是分析解决问题的步骤，然后用函数把这些步骤一一实现，使用时依次调用，函数是基本的编程单位；面向对象则是找出构成问题的各个对象，通过对象之间的消息传递（方法调用）来解决问题，类是基本的编程单位。

OOP 的基本概念包括对象、类、抽象和消息等。

1. 对象

对象（Object）是用来描述客观事物的一个实体，它可以指具体的事物，也可以指抽象的事物，一个人、一本书、一个玩家（如五子棋的白方）、一个棋盘、一个规则等都可以看做对象。每个对象都有自己的静态特征和动态行为，如图 1-1 中的对象“张三”“李四”。在 OOP 中，把静态特征称为属性，而动态行为称为方法。

2. 类

类（Class）是对一组具有相同静态特征和动态行为的对象的抽象，是对象的模板。对象和类的关系，是具体和抽象的关系：类是在对象之上的抽象，对象则是类的具体化，是类的实例。例如，对“张三”“李四”等对象进行抽象，就得到了人类 Person，如图 1-2 所示。

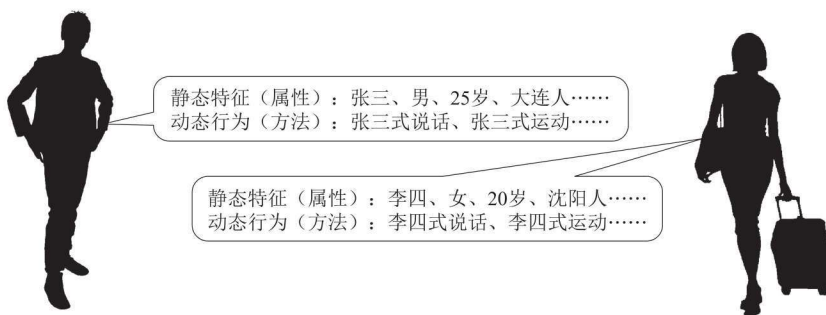


图 1-1 对象“张三”“李四”

对象和类的关系，也是变量和类型的关系：类 **Person** 相当于一个新定义的数据类型，而对象“张三”则相当于该类型的一个变量，也叫做实例变量，如图 1-3 所示。

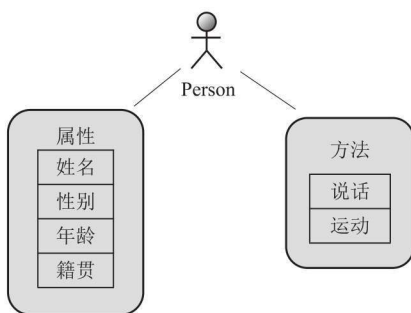


图 1-2 Person 类

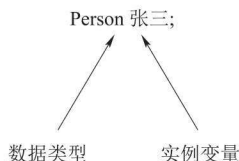


图 1-3 类和对象的关系

3. 抽象

抽象是不断提炼事物关键元素的过程，这些元素往往是该事物区别于其他事物的关键，这些元素构成了事物的本质。如对人“张三”的大量静态特征和动态特征进行提炼（抽象），只用关键属性“张三、男、25岁、大连人”和关键方法“说话、运动”来描述此人，便抽象出了对象张三。

在 OOP 中，软件设计的过程就是一个不断抽象的过程，抽象使人们更接近于事物的本质，可以使我们暂时忽略问题域中具体的、细节的东西。通过抽象，呈现在人们面前的是一个相对简单的问题域，可以使人们较容易地解决复杂问题。

4. 消息

消息是对象之间进行交互作用和通信的工具。在 OOP 中，对象功能的实现通常由另一对象对其传递消息开始，传递消息一般由三部分组成：接收消息的对象、消息名、实际变元，这三部分其实就是对象、方法名、方法参数，也就是方法的调用。如对对象“张三”发出“学习一小时”的消息，“张三”就是接收消息的对象，而“学习”就是消息名（方法名），“一小时”是实际变元（方法参数）。

软件设计除了不断抽象，另一个重要的过程就是设计对象之间的消息传递。



1.1.2 OOP 的三大特性

1. 封装性

封装是将数据和对数据进行操作的方法集中定义在一个类中，并对外部环境隐藏内部细节。封装的目的是把类的设计者和使用者分开：使用者只能见到类的外部接口（能接收哪些消息？具有哪些处理能力？），而内部实现细节是隐藏的。如数学类 `Math`，封装了数学常量 `PI`、`E`，以及三角函数 `sin(x)` 等数学运算方法，使用时，只需要知道对外接口（如何调用）就可以。

在 OOP 中，类是封装的最基本单位。良好的封装可以对外提供一致的公共接口而不影响内部实现，提高代码的安全性和可维护性。

2. 继承性

继承是指在已有类的基础上，通过增加新的属性和方法创建新类的过程。通过继承创建的新类称为“子类”或“派生类”，被继承的类称为“基类”“父类”或“超类”，例如在父类 `Person` 的基础上定义子类 `Student`，只需增加 `number` 属性和 `study` 方法；定义子类 `Teacher` 只需增加 `degree` 属性和 `teach` 方法，如图 1-4 所示。子类和父类是特殊和一般的关系，子类是特殊的父类，父类却不一定是子类。

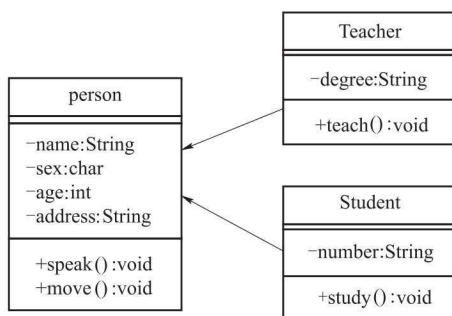


图 1-4 类的继承

3. 多态性

对象根据所接收的消息产生行为，同一消息被不同的对象接收时可能产生完全不同的行为，这种现象称为多态性。多态有两种实现途径：方法重载和方法覆盖。方法重载是指在一个类中定义多个同名方法，这些方法的参数是不同的，如 `Student` 类的三个重载的 `study` 方法；方法覆盖发生在子类 and 父类之间，在子类中重新定义父类中的方法，名字相同，但方法体不同。如图 1-5 所示。

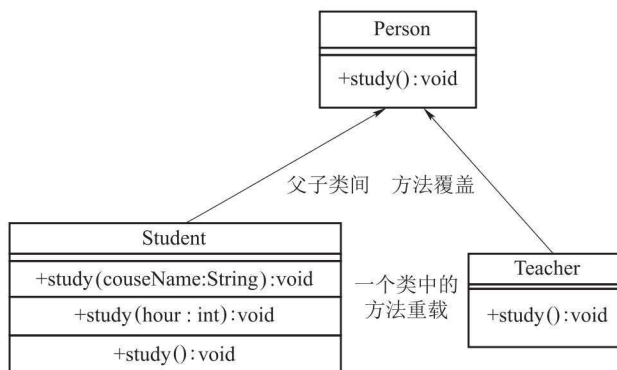


图 1-5 多态性

多态允许对任意指定的对象自动地使用正确的方法，并通过在程序运行过程中将对象与恰当的方法进行动态绑定来实现。



面向对象的特征如图 1-6 所示。简单地讲：继承、封装、多态是面向对象的三大特性，抽象是面向对象的基础。

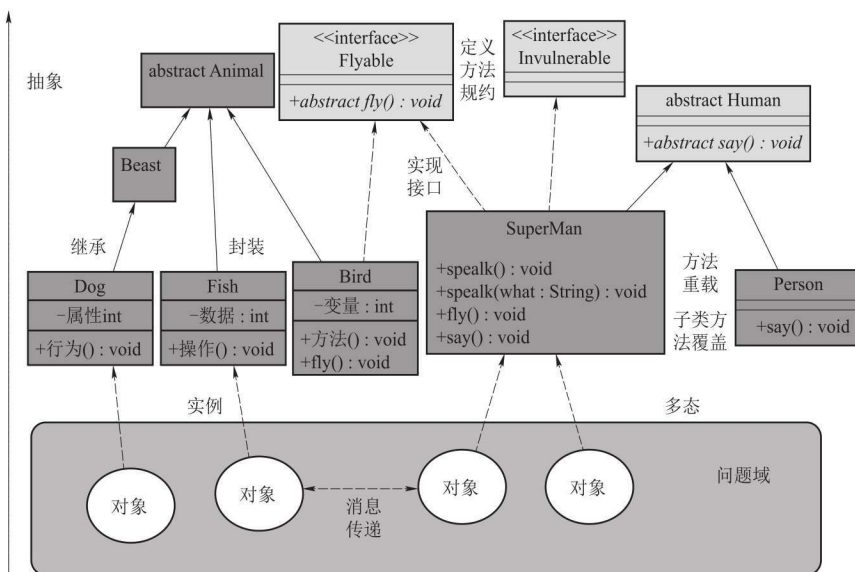


图 1-6 OOP 的特征总结

1.2 开始 Java 程序开发

1.2.1 Java 的起源及特点

Java 起源于 Oak，在印度一个盛产咖啡的岛——爪哇岛被命名，是由 Sun 公司（已被 Oracle 公司收购）的 JamesGosling 等几位工程师于 1995 年 5 月推出的一种可以编写跨平台应用软件、完全面向对象的程序设计语言。



Java 的起源及特点

Java 语言自问世以来发展得非常快，是目前最热门的编程语言之一。Java 之所以应用广泛、受到大家的欢迎，是因为它有众多突出的特点，其中最主要的特点有以下几个。

1. 面向对象

Java 语言是纯面向对象的。它提供了类、接口和继承等原语，支持类、接口之间的单继承以及接口和类之间的多实现机制；Java 语言全面支持动态绑定，而 C++ 语言只对虚函数使用动态绑定。

2. 语法简单

Java 语言的语法结构类似于 C 和 C++，但 Java 丢弃了 C++ 中很难理解的运算符重载、多重继承等模糊概念，特别是 Java 语言不使用指针，而是使用引用，并提供了自动垃圾回收机制，使程序员不必为内存管理而担忧。

3. 安全性高

Java 特别强调安全性。Java 程序运行之前会进行代码的安全性检查，确保程序不会存在



非法访问本地资源、文件系统的可能，保证了程序在网络间传送运行的安全。

4. 平台无关性

Java 引入虚拟机概念，Java 虚拟机（JVM）建立在硬件和操作系统之上，用于实现对 Java 字节码文件的解释和执行，为不同平台提供统一的接口。这使得 Java 应用程序可以运行于不同的系统平台，实现平台无关性，非常适合网络应用。

5. 支持多线程

Java 语言是支持多线程的。所谓多线程可以理解为程序中有多个任务并发执行，Java 语言提供的同步机制可保证各线程对共享数据的正确操作。多线程可以在很大程度上提高程序的执行效率。

针对不同的开发市场，Java 分为 3 个技术平台：JavaSE、JavaEE 和 JavaME。

1. Java SE 标准版（Java Standard Edition）

主要用于普通 PC 机、工作站的 Java 控制台或桌面程序的基础开发。Java SE 是 3 个平台中最核心的部分，JavaEE 和 JavaME 都是从 JavaSE 的基础上发展而来的，JavaSE 平台中包括了 Java 最核心的类库，是本书的主要学习内容。

2. Java ME 小型版（Java Micro Edition）

用于移动设备、嵌入式设备上的 Java 应用程序开发和部署。例如，为手机开发新的游戏和通讯录管理功能、为家用电器开发智能化控制和联网功能等。

3. Java EE 企业版（Java Enterprise Edition）

JavaEE 是为开发企业级应用程序提供的解决方案。它可以被看作技术平台，该平台用于开发、部署和管理企业级应用程序，包括 Servlet、JSP、JavaBean、EJB、Web Service 等技术。

1.2.2 JDK 的使用及环境变量

1. 认识 JDK

JDK（Java Development Kit）是 Oracle 公司提供的免费的 Java 开发工具包。从 1995 年到目前为止，Sun 和 Oracle 公司先后发布了多个 JDK 版本，目前最新的版本是 JDK8.0，如图 1-7 所示。



JDK 的使用及
环境变量

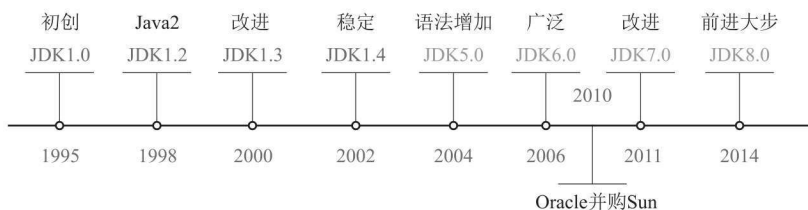


图 1-7 JDK 版本更替

JDK 是编译 Java 源文件、运行 Java 程序必需的环境，主要包括 Java 虚拟机（JVM）、Java 运行环境（JRE）以及 Java 的编译工具等。如果要开发 Java 程序，JDK 是必需的。

JRE（Java Runtime Enviroment，Java 运行环境）：运行 Java 程序所必需的环境集合，主要包含 JVM 及 Java 核心类库 API。如果只运行 Java 程序，无须 JDK，只要安装 JRE 即可。

JVM（Java Virtual Machine，Java 虚拟机）：由软件仿真出来的负责解释、执行 Java 程序的虚拟计算机。JVM 屏蔽了与具体操作系统平台相关的信息，负责解释执行 Java 字节码



文件，实现跨平台。

简单地说，JDK 是开发环境，JRE 是运行环境，JVM 是确保跨平台运行 Java 字节码的关键。JDK 包含 JRE，JRE 包含 JVM。三者的关系如图 1-8 所示。

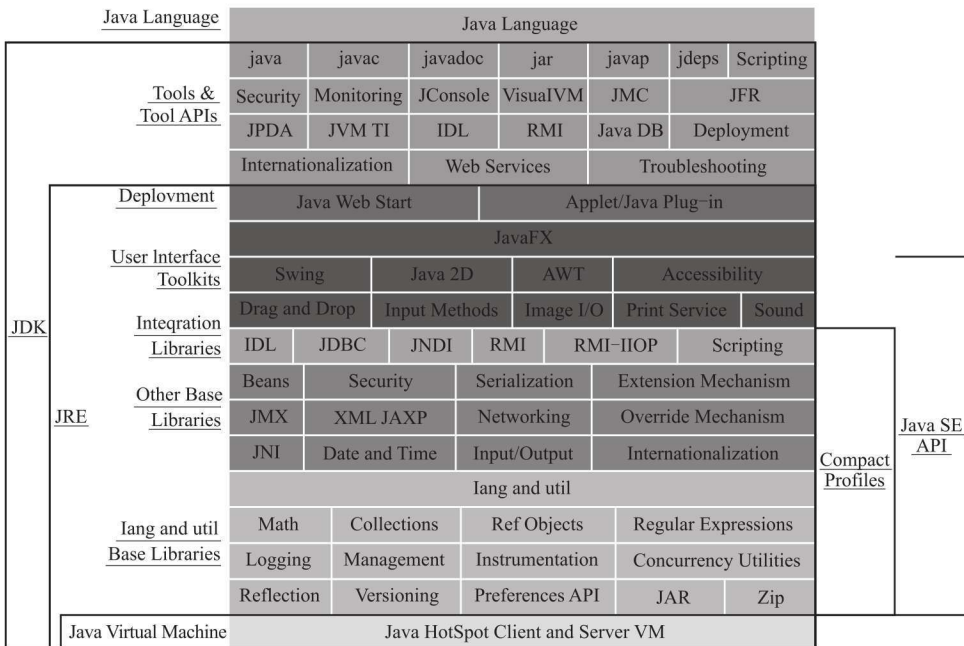


图 1-8 JDK、JRE、JVM 的关系

2. 下载并安装 JDK

(1) 下载 JDK。

JDK 的官方下载地址是“<http://www.oracle.com/technetwork/java/index.html>”，也可以输入“<http://java.sun.com>”，自动跳转到 Oracle 官网下载地址。在下载页选择 Java SE，进入如图 1-9 所示的 Java SE 下载页面后，根据自己机器的软硬件环境，选择下载合适的 JDK。

Java SE Development Kit 8u121		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.86 MB	jdk-8u121-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.83 MB	jdk-8u121-linux-arm64-vfp-hflt.tar.gz
Linux x86	162.41 MB	jdk-8u121-linux-i586.rpm
Linux x86	177.13 MB	jdk-8u121-linux-i586.tar.gz
Linux x64	159.96 MB	jdk-8u121-linux-x64.rpm
Linux x64	174.76 MB	jdk-8u121-linux-x64.tar.gz
Mac OS X	223.21 MB	jdk-8u121-macosx-x64.dmg
Solaris SPARC 64-bit	139.64 MB	jdk-8u121-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.07 MB	jdk-8u121-solaris-sparcv9.tar.gz
Solaris x64	140.42 MB	jdk-8u121-solaris-x64.tar.Z
Solaris x64	96.9 MB	jdk-8u121-solaris-x64.tar.gz
Windows x86	189.36 MB	jdk-8u121-windows-i586.exe
Windows x64	195.51 MB	jdk-8u121-windows-x64.exe

图 1-9 JDK 下载



(2) 安装 JDK。

本例以 64 位的 Windows 10 操作系统为例来安装 JDK。双击运行相应的 JDK 安装文件进行安装，单击“下一步”按钮，如图 1-10 所示。

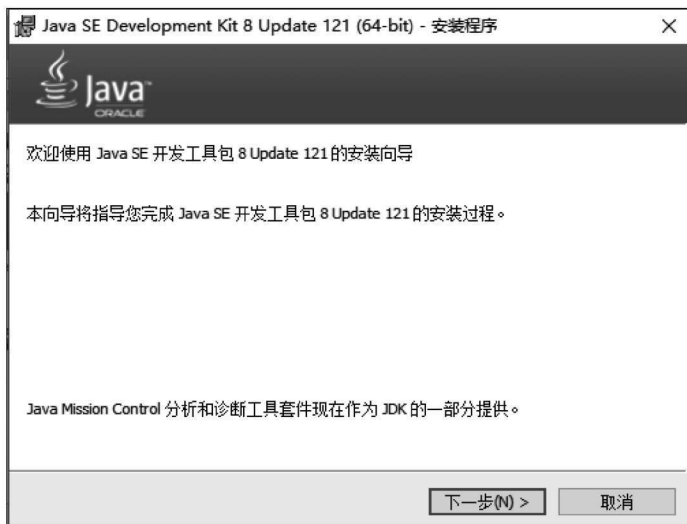


图 1-10 安装 JDK

选择安装位置，默认安装即可，单击“下一步”按钮，如图 1-11 所示。



图 1-11 选择 JDK 安装位置

选择公共 JRE 的安装路径，不做更改，仍安装于默认目录下，如图 1-12 所示。

进入安装完成界面，如图 1-13 所示。单击“关闭”按钮，完成 JDK 的安装。

3. 配置环境变量

安装完 JDK 后，会在硬盘上生成一个 JDK 安装目录 C:\Program Files\Java\jdk1.8.0_121，如图 1-14 所示。此目录也称为 JDK 主目录。