

21世纪软件工程专业规划教材

Java核心技术

马志强 张然 李雷孝 编著

10010101000101

111010010010

10010101000101

111010010010

1000101

10010101000101

111010010010



清华大学出版社

21世纪软件工程专业规划教材

Java核心技术

马志强 张然 李雷孝 编著

清华大学出版社
北京

TP312JA
1576

产品编号: 024458-01

1895750110

内 容 简 介

本书涵盖了Java平台标准版的全部基础知识和高级特性,包括Java编程基础知识、Java面向对象程序设计、数组、高级类特性、Java GUI程序设计、集合框架、泛型、Java异常处理、I/O流、JDBC访问数据库、线程编程、Socket网络编程等。

全书结构严谨,层次清晰,语言生动,理论论述精准深刻,程序实例丰富实用。不仅适合作为普通高校或职业培训教材,也可作为Java初学者和程序员的首选参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

Java核心技术/马志强,张然,李雷孝编著. —北京:清华大学出版社,2014

21世纪软件工程专业规划教材

ISBN 978-7-302-35205-1

I. ①J… II. ①马… ②张… ③李… III. ①JAVA语言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2014)第014310号

责任编辑:张 玥

封面设计:傅瑞学

责任校对:梁 毅

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 装 者:三河市李旗庄少明印装厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28 字 数:699千字

版 次:2014年4月第1版 印 次:2014年4月第1次印刷

印 数:1~2000

定 价:49.00元

前 言

P R E F A C E

近年来,在 TIOBE 公布的排名中,Java 始终保持位列三甲的成绩。排名在一定程度上说明 Java 在行业中得到广泛应用,使用 Java 技术进行软件开发的人员较多。要想成为 Java 技术类的开发人员,不仅需要扎实的 Java 语言功底,还要学习掌握软件设计与开发技术。

本书既叙述软件开发技术,又培养工程实践能力。以 IT 企业对开发人员的技术能力要求为基础,以工程能力培养为目标,阐述软件工程对计算机语言要求的知识点,并形成相应知识单元;按照工程需求顺序组织课程内容,便于学习和掌握;提供一定量案例,注重培养实践能力。

全书共分 13 章,内容安排以工程应用为主线。第 1 章为 Java 语言概述;第 2 章为 Java 语法基础;第 3 章为面向对象编程;第 4 章为数组;第 5 章为抽象类、接口、内部类以及反射;第 6 章为 GUI 基础、事件处理、适配器类以及常用 Swing 组件编程;第 7 章为 List、Set 和 Map 接口与其实现类的使用;第 8 章为泛型类、泛型方法的定义与使用;第 9 章为 Java 程序中异常的处理;第 10 章为 I/O 流的读/写操作;第 11 章为 JDBC 编程,事务处理以及 DAO 模式编程;第 12 章为线程编程;第 13 章为基于 TCP 和 UDP 协议的 Socket 编程。

本书具有以下特点。

(1) 遵照教指委计算机科学与技术 and 软件工程专业培养目标和方案,合理安排知识体系,结合 Java 技术方向的先行课程和后续课程组织知识点与内容。

(2) 注重理论和实践结合,融入面向对象软件开发过程和工程实践背景的项目案例,使学生在掌握理论知识的同时提高分析问题和解决问题的能力,加强创新意识,理论知识和实践技能得到全面发展。

(3) 每个知识点都包括基础案例,每章都有一个综合案例,知识内容层层推进,易于学习。每章综合案例以“大学生综合测评系统”为基础,以开发过程为主线,将知识点有机串联在一起,便于掌握理解。

(4) 章节习题提供一定数量的课外实践题目,采用课内外结合的方式提高学生的软件开发兴趣和工程实践能力,满足当前社会对软件开发人员的需求。

(5) 本书提供配套课件和案例的源码。

本书由马志强、张然和李雷孝编著。其中,马志强编写第 1、2、3、4、11 和 13 章并统稿,张然编写第 8 章、各章综合案例与习题,李雷孝编写第 5、6、7、9、10 和 12 章。编写过程中参阅了甲骨文(Oracle)公司、安博教育集团、青岛软件园、达内时代科技、杰普软件等

公司的教学科研成果,也吸取了国内外教材的精髓,在此表示由衷感谢。感谢柯展、闫瑞在案例代码的实现和测试中的贡献。

本书的出版得到刘利民教授、刘建兰老师的支持和帮助;还得到清华大学出版社的大力支持,在此表示诚挚的感谢。

本书受到全国高等学校计算机教育研究会 2013 年度高等学校计算机教材建设项目资助。

由于作者水平有限,书中难免有不妥和疏漏之处,恳请读者不吝赐教,作者邮箱为 mzq_bim@imut.edu.cn。

编者

2013 年 10 月

目 录

CONTENTS

第 1 章	Java 语言概述	1
1.1	Java 技术	1
1.1.1	Java 语言	1
1.1.2	Java 平台	2
1.1.3	Java API 使用	4
1.2	Java 程序的开发过程	5
1.2.1	简单的桌面应用程序结构	5
1.2.2	Java 程序的编译	7
1.2.3	装载运行	8
1.2.4	集成开发环境的使用	8
1.3	基本输入输出	11
1.3.1	基本输入方法	11
1.3.2	基本输出方法	14
1.4	案例	15
1.4.1	案例设计	15
1.4.2	案例演示	16
1.4.3	代码实现	16
1.5	习题	17
第 2 章	Java 语法基础	20
2.1	常量与变量	20
2.2	运算符与表达式	23
2.2.1	运算符	23
2.2.2	表达式	26
2.2.3	类型转换	27
2.3	控制结构	29
2.3.1	分支结构	29

2.3.2	循环结构	32
2.3.3	跳转语句	36
2.4	案例	38
2.4.1	案例设计	38
2.4.2	案例演示	39
2.4.3	代码实现	40
2.5	习题	44
第3章	面向对象编程	49
3.1	面向对象程序设计思想	49
3.2	类	51
3.2.1	类的定义	51
3.2.2	属性的定义	52
3.2.3	方法定义	53
3.2.4	方法参数的使用	54
3.2.5	构造方法(Constructor Method)	56
3.2.6	属性值的读取与设置方法	58
3.2.7	this 关键字	58
3.3	对象	60
3.3.1	对象声明	60
3.3.2	对象实例化	60
3.3.3	对象使用	61
3.3.4	对象销毁	63
3.4	继承	63
3.4.1	继承的定义	63
3.4.2	super 关键字	67
3.4.3	覆盖技术(Overriding)	68
3.4.4	Object 类	71
3.5	访问控制	72
3.5.1	包(package)	72
3.5.2	访问控制修饰符	74
3.6	非访问控制修饰符	75
3.6.1	static 修饰符	75
3.6.2	final 修饰符	78
3.7	String 类	78
3.7.1	String 的基本使用	78
3.7.2	正则表达式	82
3.8	案例	84

3.8.1	案例设计	84
3.8.2	案例演示	85
3.8.3	代码实现	86
3.9	习题	91
第 4 章 数组 96		
4.1	基本概念	96
4.2	一维数组	96
4.3	多维数组	99
4.4	数组 API 的使用	101
4.5	案例	104
4.5.1	案例设计	104
4.5.2	案例演示	104
4.5.3	代码实现	105
4.6	习题	110
第 5 章 高级类特性 114		
5.1	抽象类	114
5.2	接口	117
5.2.1	接口概念	117
5.2.2	接口定义	118
5.2.3	接口实现	118
5.2.4	接口的多重实现	120
5.2.5	接口的多继承	121
5.2.6	接口应用	122
5.3	内部类	125
5.3.1	内部类概念	125
5.3.2	实例内部类	127
5.3.3	静态内部类	128
5.3.4	局部内部类	129
5.3.5	匿名内部类	130
5.3.6	内部类应用	132
5.4	反射	133
5.4.1	反射概念	133
5.4.2	Class	133
5.4.3	其他反射相关 API	134
5.4.4	反射编程基本步骤	136
5.4.5	反射编程举例	137

138	5.5 案例	143
138	5.5.1 案例设计	143
138	5.5.2 案例演示	144
138	5.5.3 代码实现	145
	5.6 习题	150
第 6 章 GUI 编程		
148	6.1 GUI 基础	154
148	6.1.1 GUI 编程概述	154
148	6.1.2 组件	155
148	6.1.3 容器	155
148	6.1.4 布局管理器	156
148	6.1.5 容器嵌套	160
148	6.2 GUI 事件处理	161
148	6.2.1 GUI 事件处理机制	161
148	6.2.2 GUI 事件类型	165
148	6.2.3 多重监听器	166
148	6.2.4 适配器类	168
148	6.3 Swing 编程	170
148	6.3.1 Swing 简介	170
148	6.3.2 常用 Swing 组件	171
148	6.4 案例	185
148	6.4.1 案例设计	186
148	6.4.2 案例演示	186
148	6.4.3 代码实现	188
148	6.5 习题	198
第 7 章 集合框架		
158	7.1 集合概念	200
158	7.2 集合框架结构	200
158	7.2.1 集合接口	200
158	7.2.2 集合实现类	204
158	7.3 List	204
158	7.4 Set	207
158	7.5 Map	214
158	7.6 案例	217
158	7.6.1 案例设计	217
158	7.6.2 案例演示	217

7.6.3	代码实现	219
7.7	习题	229
第 8 章	泛型	232
8.1	泛型简介	232
8.2	使用泛型	233
8.2.1	集合框架中的泛型使用	233
8.2.2	泛型的向后兼容性	235
8.3	泛型进阶	236
8.3.1	类型参数	236
8.3.2	类型通配符	238
8.3.3	泛型方法	241
8.3.4	受限制的类型参数	243
8.4	案例	245
8.4.1	案例设计	245
8.4.2	案例演示	246
8.4.3	代码实现	247
8.5	习题	256
第 9 章	异常	257
9.1	异常概念和分类	257
9.1.1	异常概念	257
9.1.2	异常分类	260
9.1.3	常见异常	261
9.2	异常处理	262
9.2.1	异常处理机制	262
9.2.2	捕获-处理异常	263
9.2.3	声明抛出异常	267
9.2.4	人工抛出异常	269
9.3	自定义异常	270
9.4	案例	271
9.4.1	案例设计	271
9.4.2	案例演示	272
9.4.3	代码实现	272
9.5	习题	274
第 10 章	流	277
10.1	流的基本概念	277

10.2	流的分类	278
10.3	流的体系结构	278
10.3.1	InputStream	279
10.3.2	OutputStream	280
10.3.3	Reader	280
10.3.4	Writer	281
10.4	常用流的使用	282
10.4.1	流的操作步骤	282
10.4.2	字节流	283
10.4.3	字符流	285
10.4.4	字节字符转换流	287
10.4.5	随机读取文件流	288
10.4.6	PrintStream/PrintWriter	290
10.4.7	标准 I/O	291
10.5	对象序列化	292
10.6	文件操作	294
10.7	案例	299
10.7.1	案例设计	300
10.7.2	案例演示	300
10.7.3	代码实现	301
10.8	习题	304
第 11 章 JDBC 访问数据库		308
11.1	JDBC 体系结构	308
11.2	JDBC 编程的基本概念	309
11.2.1	注册驱动	309
11.2.2	建立连接	310
11.2.3	新建状态	314
11.2.4	执行 SQL 语句	315
11.2.5	处理结果集	322
11.2.6	关闭资源	327
11.2.7	事务	327
11.3	DAO 模式的应用	328
11.3.1	构建实体类	328
11.3.2	对象操作接口	333
11.3.3	对象操作代理类	335
11.3.4	对象操作实现类	337
11.3.5	工厂模式访问 DAO	341

11.4	案例分析	342
11.4.1	案例设计	342
11.4.2	案例演示	343
11.4.3	代码实现	343
11.5	习题	356
第 12 章 线程		359
12.1	线程基础	359
12.1.1	线程的基本概念	359
12.1.2	线程的概念模型	360
12.1.3	线程的创建	360
12.2	线程的状态	368
12.2.1	线程状态转换	368
12.2.2	常用的线程状态转换方法	369
12.3	线程同步	376
12.3.1	临界资源问题	376
12.3.2	线程同步	379
12.4	线程死锁	381
12.5	线程通信	383
12.6	案例	386
12.6.1	案例设计	386
12.6.2	案例演示	387
12.6.3	代码实现	387
12.7	习题	390
第 13 章 网络编程		395
13.1	网络基本概念	395
13.2	java.net 包	396
13.3	基于 TCP 的 Socket 编程	397
13.3.1	InetAddress 类	397
13.3.2	Socket 编程模型	399
13.3.3	服务器程序	401
13.3.4	客户端程序	402
13.3.5	多客户端的服务器程序	404
13.4	基于 UDP 的 Socket 编程	406
13.4.1	UDP 编程模型	406
13.4.2	接收端程序	407
13.4.3	发送端程序编程	409

13.5	URL(Uniform Resource Locator)	411
13.6	案例	423
13.6.1	案例设计	423
13.6.2	案例演示	423
13.6.3	代码实现	424
13.7	习题	429
附录 A 综合案例说明		
A.1	数据库安装和导入	432
A.2	开发工具的安装和工程导入	432
A.3	程序运行	433
参考文献		
		434

Java 语言概述

作为一门纯面向对象程序设计语言,Java 在软件开发领域得到了广泛应用。本章主要介绍了 Java 技术的构成、Java 桌面应用程序的开发过程以及基本的输入输出方法。

1.1 Java 技术

面向对象编程语言起源于 20 世纪 60 年代开发的 Simula 67,其首次在程序设计中提出了对象的概念;20 世纪 70 年代出现的 Ada 语言是一种重要的支持数据抽象、基于对象的语言,但没有全面地支持继承。接着出现的 Smalltalk 语言也是一门真正的面向对象编程语言,它丰富了面向对象的概念,实现了面向对象技术的机制。后来所有面向对象语言都受到它的启发。

20 世纪 90 年代初,随着互联网应用的兴起,一门可以跨平台应用的面向对象程序设计语言——Java 出现了。特别是在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

1.1.1 Java 语言

Java 技术包括 Java 语言和 Java 平台两部分。Java 语言是一门高级的面向对象语言,具有简单性、面向对象性、分布式、多线程、动态性、体系结构的中立性、可移植性、高性能、健壮性和安全性的特点。

(1) 简单性(Simple): Java 语言的语法与 C 和 C++ 语言很接近,同时丢弃了 C++ 中很少使用和很难理解的特性,如操作符重载、多继承、自动的强制类型转换,因此大多数程序员很容易掌握。特别地,Java 语言不使用指针,并提供了自动垃圾收集功能,因此程序员不必为内存管理担忧。

(2) 面向对象性(Object Oriented): Java 提供了类、接口和继承等原语。为了简单起见,只支持类之间的单继承,但支持接口之间的多继承,并支持类与接口之间的实现机制。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。总之,Java 语言是一个纯面向对象程序设计的语言。

(3) 分布式(Distributed): Java 支持 Internet 应用的开发,在基本 Java 应用编程接口中有一个网络应用编程接口(Java Net),它提供了用于网络应用编程的类库,包括 URL、URLConnection、Socket、ServerSocket 等。Java 的远程方法调用(RMI)机制也是开发分布式应用的重要手段。

(4) 多线程(Multithreaded): 线程是一种特殊的对象,它的活动由一组方法来控制。Java 语言支持多个线程同时执行,并提供多线程之间的同步机制(最简单的方法是使用关键字 Synchronized)。

(5) 动态性(Dynamic): Java 类能够动态地载入到运行环境,也可以通过网络载入需要的类,有利于软件的升级。

(6) 体系结构的中立性(Architecture Neutral): 程序在 Java 平台上被编译为体系结构中立的字节码格式,然后可以在装有 Java 平台的任何系统中运行。这种途径适于异构的网络环境和软件的分发。

(7) 可移植性(Portable): Java 系统本身是中立的,具有很强的可移植性。Java 还严格规定了各个基本数据类型的长度,Java 编译器是用 Java 实现的,Java 的运行环境是用 ANSIC 实现的。

(8) 高性能(High Performance): Java 是解释型语言。但是,与普通的解释型高级脚本语言相比,Java 的确是高性能的。事实上,随着 JIT(Just-In-Time)编译技术的发展,Java 的运行速度越来越接近 C++。

(9) 健壮性(Robust): Java 采用强类型检查、异常处理、垃圾回收等机制实现健壮性。对指针的丢弃是 Java 的明智选择,Java 的安全检查机制使得 Java 更具健壮性。

(10) 安全性(Secure): Java 常被用在网络环境中,Java 提供了一种安全管理机制(类 SecurityManager),防止恶意代码攻击。

1.1.2 Java 平台

Java 平台由两部分组成: Java 虚拟机(Java Virtual Machine,JVM)和 Java 应用程序接口(Java Application Programming Interface,API)。Java 平台将 Java 程序和硬件平台隔离开,实现了“一次编写,到处执行”的特点,Java 平台体系结构如图 1-1 所示。

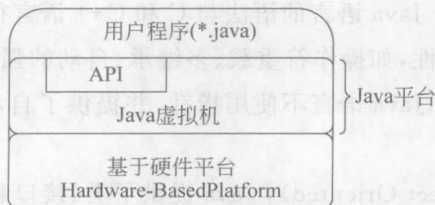


图 1-1 Java 平台体系结构

Java 虚拟机(以下简称 JVM)是运行 Java 代码的虚拟计算机,是一台通过软件模拟实现的真实计算机系统。JVM 是一种用于计算机设备的规范,可以采用不同方式(软件

或硬件)加以实现。所以,JVM 有自己的硬件,如处理器、堆栈、寄存器等;还有相应的指令系统,为 Java 字节码程序提供了独立的运行环境。

执行字节码时,JVM 需要把字节码解释成具体平台上的机器指令执行。因此,只要根据 JVM 规格描述编写对应计算机系统上的 JVM,就能保证经过编译的任何 Java 代码能够在该系统上运行,实现“一次编写,到处执行”。

JVM 的体系结构包括 4 部分,分别是类装载子系统、执行引擎子系统、本地方法接口和运行时数据区。JVM 体系结构如图 1-2 所示。

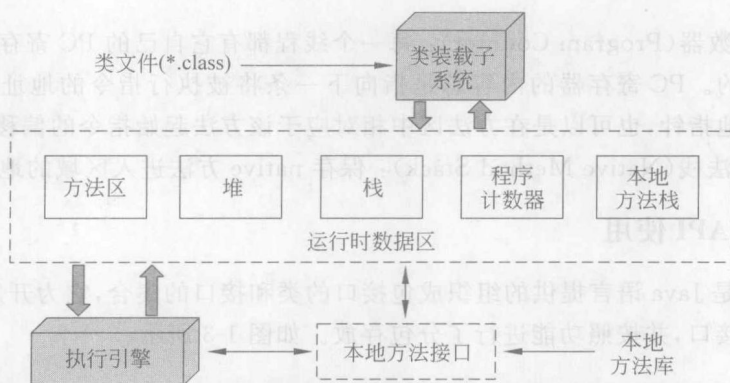


图 1-2 JVM 体系结构

1. 类装载子系统(Class Loader Subsystem)

根据给定的类全名,如 `java.lang.Object` 来装载类(class)文件的内容到运行时数据空间(Runtime Data Area)中的方法区域(Method Area)。程序员可以继承 `java.lang.ClassLoader` 类来写自己的类装载器。

2. 执行引擎子系统(Execution Engine)

执行引擎子系统由一套字节码指令集构成,通过执行引擎完成字节码的执行,执行类(class)中的指令。第一代 JVM 执行引擎采用解释技术;第二代 JVM 执行引擎采用 JIT 编译技术;目前的 JVM 执行引擎采用自适应优化技术。

3. 本地方法接口(Native Method Interface)

与本地方法库(Native Libraries)交互,是其他编程语言交互的接口。调用本地方法时,就进入了不再受虚拟机限制的空间,因此很容易出现 JVM 无法控制的本地堆溢出(Native Heap Out of Memory)的情况。

4. 运行时数据区(Runtime Data Area)

运行时数据区就是 JVM 的内存,包括堆、栈、方法区、程序计数器和本地方法栈 5 部分。

① 堆(Heap): 每个 JVM 实例中只存在一个堆空间,实现虚拟机中的数据共享。它是一个可动态申请的内存空间,负责存储对象的数据。所有使用 new 类名()构造出的对象都在堆中存储。

② 栈(Stack): 是一个先进后出的数据结构,通常用于保存方法中的参数、局部变量、所有基本类型和引用类型。它的存储空间小,但速度比较快。

③ 方法区(Method Area): 被装载的类信息存储在方法区的内存中。当虚拟机装载某个类型时,它使用类装载机定位相应的类文件,然后读入这个类文件内容,并把它传输到虚拟机中。

④ 程序计数器(Program Counter): 每一个线程都有它自己的 PC 寄存器,也是该线程启动时创建的。PC 寄存器的内容总是指向下一条将被执行指令的地址,这里的地址可以是一个本地指针,也可以是在方法区中相对应于该方法起始指令的偏移量。

⑤ 本地方法栈(Native Method Stack): 保存 native 方法进入区域的地址。

1.1.3 Java API 使用

Java API 是 Java 语言提供的组织成包接口的类和接口的集合,它为开发者提供了各种有效的类和接口,并按照功能进行了分包存放。如图 1-3 所示。

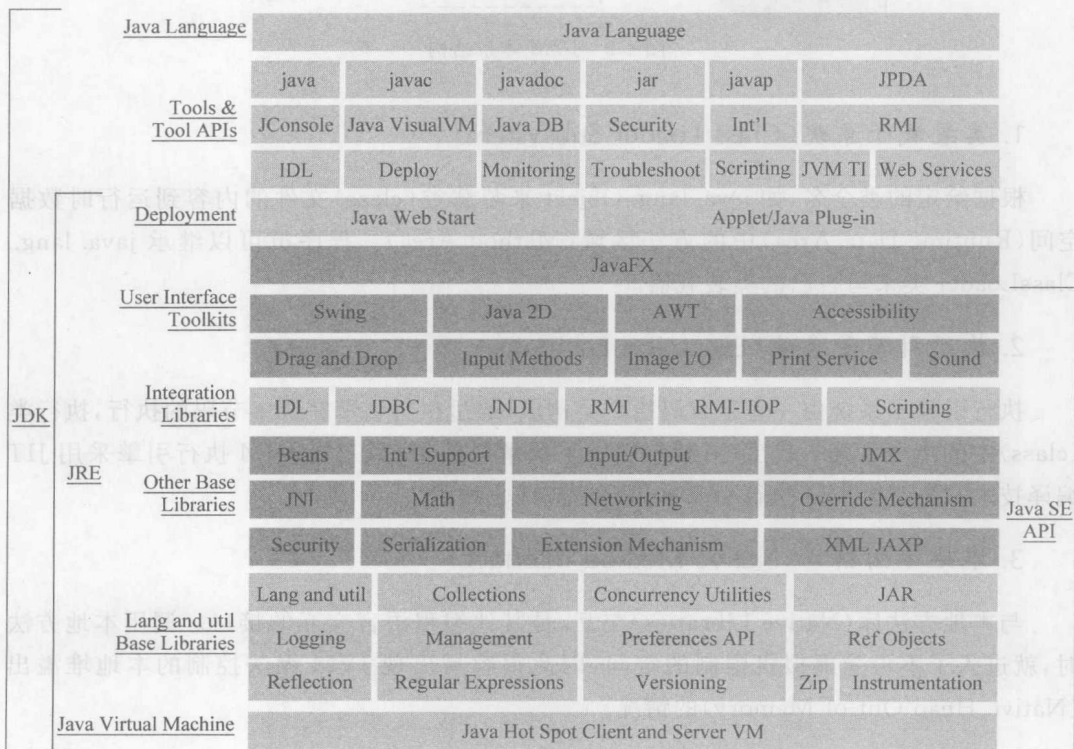


图 1-3 Java API^①

① 引自 Oracle Java 7.0 用户手册。