

►► 高等学校应用型本科“十三五”规划教材



- 以任务驱动教学，使学生在完成任务中学习
- 强调规范编码，使学生养成良好的编码习惯

Java软件开发基础

►► 主 编 何桂兰 陈素琼
副主编 姚骏屏 刘志强 郭新明

高等学校应用型本科“十二五”规划教材

Java 软件开发基础

主 编 何桂兰 陈素琼

副主编 姚骏屏 刘志强 郭新明

西安电子科技大学出版社

内 容 简 介

本书针对 Java 程序开发的初学者而编写,力求培养读者的基本编码能力、程序的调试与测试能力以及规范编码的习惯。本书内容分为两大部分,第一部分采取任务驱动的方式,读者在完成的过程中将学到开发环境的配置与开发工具的使用,数据类型的声明、使用和转换,方法与运算符,分支语句与命令行参数,控制语句之循环语句初步,循环结构之多重循环等知识;第二部分采取项目驱动的方式,选用读者熟悉的 ATM 机模拟程序,使读者在完成项目的过程中学到输入输出流、开关语句——switch 语句、字符串的处理、数组的使用、异常处理、文件的读写、软件测试基础等知识。

本书可作为应用型本科和高职院校学生的 Java 课教材,也可作为 Java 软件开发人员的自学参考书。

图书在版编目(CIP)数据

Java 软件开发基础/何桂兰,陈素琼主编. —西安:西安电子科技大学出版社,2015.8

高等学校应用型本科“十三五”规划教材

ISBN 978-7-5606-3832-4

I. ① J… II. ① 何… ② 陈… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2015)第 192926 号

策 划 李惠萍

责任编辑 李惠萍 孟秋黎

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2015 年 8 月第 1 版 2015 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 14.5

字 数 338 千字

印 数 1~3000 册

定 价 25.00 元

ISBN 978-7-5606-3832-4/TP

XDUP 4124001-1

如有印装问题可调换

西安电子科技大学出版社
高等学校应用型本科“十三五”规划教材

编审专家委员会名单

主任：鲍吉龙（宁波工程学院副院长、教授）

副主任：彭 军（重庆科技学院电气与信息工程学院院长、教授）

张国云（湖南理工学院信息与通信工程学院院长、教授）

刘黎明（南阳理工学院软件学院院长、教授）

庞兴华（南阳理工学院机械与汽车工程学院副院长、教授）

电子与通信组

组 长：彭 军（兼）

张国云（兼）

成 员：（成员按姓氏笔画排列）

王天宝（成都信息工程学院通信学院院长、教授）

安 鹏（宁波工程学院电子与信息工程学院副院长、副教授）

朱清慧（南阳理工学院电子与电气工程学院副院长、教授）

沈汉鑫（厦门理工学院光电与通信工程学院副院长、副教授）

苏世栋（运城学院物理与电子工程系副主任、副教授）

杨光松（集美大学信息工程学院副院长、教授）

钮王杰（运城学院机电工程系副主任、副教授）

唐德东（重庆科技学院电气与信息工程学院副院长、教授）

谢 东（重庆科技学院电气与信息工程学院自动化系主任、教授）

楼建明（宁波工程学院电子与信息工程学院副院长、副教授）

湛腾西（湖南理工学院信息与通信工程学院教授）

计算机大组

组 长：刘黎明（兼）

成 员：（成员按姓氏笔画排列）

刘克成（南阳理工学院计算机学院院长、教授）

毕如田（山西农业大学资源环境学院副院长、教授）

李富忠（山西农业大学软件学院院长、教授）

向 毅（重庆科技学院电气与信息工程学院院长助理、教授）

张晓民（南阳理工学院软件学院副院长、副教授）

何明星（西华大学数学与计算机学院院长、教授）

范剑波（宁波工程学院理学院副院长、教授）

赵润林（山西运城学院计算机科学与技术系副主任、副教授）

雷 亮（重庆科技学院电气与信息工程学院计算机系主任、副教授）

黑新宏（西安理工大学计算机学院副院长、教授）

前 言

Java 语言自从 1995 年诞生以来,在跨平台开发及互联网开发领域中扮演着越来越重要的角色,被公认为是功能最强大、最有前途的编程语言。

编者有多年软件开发的经验,并且在高职院校和应用型本科院校从事了多年的软件开发教学工作。软件工程师的核心职业能力基本划分为四个层次:基本编码与调试能力、面向对象分析及 C/S 软件开发能力、B/S 网站开发能力、主流框架应用开发能力。本书对应于第一层,适合于从零开始接触 Java 语言的读者。

本书按照“教、学、做一体化”的模式编写,同样也适合“理论+实做”的教学模式。全书分为两大模块:任务驱动模块与项目驱动模块,改变了以往“理论+验证”式的教学方式。在课时设置上将每个任务设置为 2 个课时,项目的每个模块功能大体需 4 个学时。全书均采用“提出任务→重点知识讲解→任务概要流程分解→代码级详解”的流程进行教学,以学生能完成任务作为课堂知识掌握的评价标准。

本书强调了编码规范的重要性。编码规范对软件开发来说非常重要,而在以往的教材中却忽视了这一点。通过书中不断出现的编码规范提示,编者力图使读者在使用本书的过程中潜移默化地“养成”良好的编程习惯。

程序员都知道,软件开发离不开软件测试。而以往学生学习测试只停留在理论上,对学生而言,学习软件测试枯燥且难以理解。本书精心安排了项目测试单元,通过实际项目中所采用的真实测试用例模板进行实战演练,使学生很快理解什么是测试、如何测试。

本书由何桂兰、陈素琼担任主编,姚骏屏、刘志强、郭新明担任副主编。何桂兰负责全书总体框架的设计和全书的统稿工作,并编写了第 4、5、6、7、8、10、11 章的内容;陈素琼完成了全书的评审工作,并编写了第 1、2、3、9 章的内容;姚骏屏参与了全书的修改工作,并编写了第 4 章的部分内容;刘志强参与了全书的校正工作,并编写了第 12 章的内容;郭新明参与了第 7、8 章内容的编写。

由于作者水平有限,疏漏和错误之处在所难免,欢迎广大读者提出宝贵意见。

作 者

2015 年 7 月

目 录

第一部分 任务驱动课程模块

第 1 章 熟悉 Java 开发环境——我的第一个 Java 程序	3
1.1 搭建 Java 开发环境	3
1.1.1 Java 的由来与特点	3
1.1.2 Java 开发环境与开发工具包	7
1.2 使用文本编辑器开发 HelloWorld 程序	14
1.2.1 Java 开发工具简介	14
1.2.2 用记事本编写 Java 代码	16
1.3 使用 MyEclipse 开发 HelloWorld 程序	17
1.3.1 MyEclipse 的使用	17
1.3.2 MyEclipse 的一些常用技巧	22
1.4 巩固与提高	25
第 2 章 数据类型的声明、使用和转换	26
2.1 数据类型的声明	26
2.1.1 走近程序“语言”	26
2.1.2 Java 语言的标识符与保留字	27
2.1.3 数据类型概述	29
2.2 数据类型的使用	32
2.2.1 基本数据类型	32
2.2.2 引用数据类型	34
2.3 数据类型的转换	36
2.3.1 类型转换	36
2.3.2 ASCII 编码	37
2.4 使用 API 帮助文档	40
2.4.1 Java 的 API 帮助文档简介	40
2.4.2 如何使用 API 文档	40
2.5 巩固与提高	42
第 3 章 方法与运算符	44
3.1 编写求两数和的方法	44
3.1.1 方法	44
3.1.2 运算符与表达式	48
3.2 编写比较两数大小的方法	51

3.2.1 方法	51
3.2.2 运算符与表达式	52
3.3 求圆和圆柱相关参数的方法编写	57
3.3.1 运算符	57
3.3.2 运算符的优先级	58
3.4 巩固与提高	62
第 4 章 分支语句与命令行参数	65
4.1 实现 ATM 机登录密码验证	65
4.1.1 程序结构概述	65
4.1.2 分支语句	66
4.2 实现 ATM 机菜单选择功能	69
4.2.1 if-else 语句	70
4.2.2 嵌套 if 语句	71
4.3 验证取款金额的合法性	75
4.4 命令行参数的使用	79
4.5 巩固与提高	83
第 5 章 控制语句之循环语句初步	84
5.1 调换数字的顺序	84
5.1.1 循环语句概述	84
5.1.2 Java 语言提供的循环语句种类	85
5.2 从 1 加到 50	89
5.2.1 do-while 语句	89
5.2.2 for 语句	90
5.3 巩固与提高	94
第 6 章 循环结构之多重循环	96
6.1 打印由字符组成的三角形	96
6.1.1 什么是多重循环	96
6.1.2 双重循环的运行过程	97
6.1.3 双重循环的使用	98
6.2 求素数	101
6.3 巩固与提高	105

第二部分 项目驱动课程模块

第 7 章 输入输出流	109
7.1 实现 ATM 机登录功能	109
7.1.1 输入/输出基础	109

7.1.2 读取控制台输入	113
7.2 显示 ATM 机操作菜单	119
7.3 巩固与提高	123
第 8 章 开关语句——switch 语句	124
8.1 ATM 机功能菜单选择的实现	124
8.1.1 switch 语句概述	124
8.1.2 switch 语句语法	124
8.2 ATM 机显示欢迎信息(扩展任务)	129
8.2.1 switch 语句执行过程	129
8.2.2 获取当前时间	132
8.3 巩固与提高	138
第 9 章 字符串的处理	141
9.1 实现 ATM 机密码修改的功能	141
9.1.1 字符串的基本概念	141
9.1.2 字符串变量	143
9.1.3 字符串的“+”连接运算	144
9.1.4 几种字符串内置方法的使用	145
9.2 巩固与提高	158
第 10 章 数组的使用	160
10.1 ATM 机取款功能的实现	160
10.1.1 一维数组的定义及使用	160
10.1.2 二维数组的定义及使用	163
10.1.3 数组基本方法的使用	165
10.2 巩固与提高	174
第 11 章 异常处理	175
11.1 ATM 机转账功能的实现	175
11.1.1 异常概述	175
11.1.2 异常情况分类	176
11.1.3 处理异常的方式	177
11.2 改造 ATM 机的取款代码	191
11.3 巩固与提高	196
第 12 章 文件的读写	199
12.1 文件创建	199
12.1.1 文件概述	199

12.1.2	绝对路径和相对路径	199
12.1.3	文件名称	200
12.1.4	File 类	200
12.2	实现 ATM 机余额查询与退出功能.....	202
12.2.1	流类 Stream	202
12.2.2	文件的读写	203
12.3	巩固与提高	214
第 13 章	软件测试基础	215
13.1	ATM 机程序功能测试.....	215
13.1.1	测试的基本概念	215
13.1.2	如何进行测试	217
13.2	巩固与提高	222

第一部分

任务驱动课程模块

通过“任务提出→重点知识讲解→任务过程分解→任务代码分析→巩固与提高→软件测试”的形式进行 Java 开发环境搭建，数据类型声明、使用和转换，运算符、表达式和方法的定义以及分支语句、循环语句的分析和应用。

第1章 熟悉 Java 开发环境

——我的第一个 Java 程序

1.1 搭建 Java 开发环境



工作目标

本节你将学习：

- (1) Java 的历史、特色和运行机制。
- (2) JDK 的安装与环境变量的配置。



工作描述

要搭建 Java 开发环境，首先要了解 Java 的发展历史和 Java 的三个版本，以及 JDK 的下载、安装和环境变量的配置。



工作分解——理论阐述

1.1.1 Java 的由来与特点

Java 是由美国 Sun 公司开发的新一代面向对象编程语言，其底层由 C++ 语言开发。在与网络的融合中，它显现出强大的生命力和广阔的前景。Java 的语法简练，学习和掌握起来比较容易，可以在各式各样的不同种机器、不同种操作平台的网络环境中使用。

1. Java 的由来

Sun 公司在 1991 年准备开发一个机顶盒的项目，能够将用户的请求发送到电视台，电视台再将节目发送给用户，实现点播的功能。

这个机顶盒能够转发用户的请求，那么这里面就应该有一个芯片，工作小组考虑到产

品(机顶盒)开发出来需要 2~3 年的时间,要是到时候没有芯片了怎么办?这个问题曾经出现在 IBM 的 8096 芯片上,因此令开发商高度重视。IBM 在 8051 芯片上取得了很大的成功,后来又开发了 16 位的 8096 芯片,其性能更优越,于是许多公司开始开发基于 8096 的产品,但当各大研发公司万事俱备就等 IBM 的 8096 芯片时,IBM 不再生产这种芯片了,这使开发商损失很大。以致于以后研发时都把这个问题列为必须考虑的风险之一。James Gosling 提出了一个建议:开发一个解释器,设计一种语言,解释器负责执行这种语言,而解释器可以根据不同的芯片进行设计。这样就使得开发工作一开始就不用与硬件打交道,这种解释器能够在 Sun 公司的 Solaris 平台上运行,等开发完成后,再针对要使用的芯片设计一种解释器就可以了。三年之后的 1994 年,这种机顶盒设计出来了,而这种语言取名为 OAK(橡树, James Gosling 的别墅旁边有一棵橡树, James Gosling 每天早上起来都能看到它,所以命名为 OAK)。但等到 Sun 公司向电视台推销这种产品的时候,电视台却不认可这种装置, Sun 公司的这个项目只能以失败告终。后来, James Gosling 想到既然 HTML 语言可以在各种浏览器里运行,那么可不可以把他们的这种语言嵌入到浏览器里,开发一种新的浏览器,使它不仅仅是显示静态的 HTML 页面,还能够执行程序。

1995 年, Sun 公司和好莱坞开展了一个技术交流会,借以宣传公司的产品。James Gosling 带着他们的 OAK 系统来到了技术交流会,但他不能再用 OAK 这个名字了,因为 OAK 已经被一家半导体公司注册。当 James Gosling 四处寻找灵感时,他看到了一个泡咖啡的杯子,杯子上印着 Java(印度尼西亚的爪哇岛,以生产咖啡著名),他就把这种语言改名为 Java。Java 语言在交流会上大获成功。网景公司(一家著名的开发互联网浏览器的公司)找到了 Sun 公司,想把 Java 语言嵌入他们的 Netscape 浏览器,而当时 Netscape 浏览器在市场上占据了 95% 的份额。1995 年 5 月 23 日, Sun 公司宣布 Java 语言正式诞生,5 月 23 日也就成了 Java 的生日,而项目组领导 James Gosling 就成了 Java 之父(见图 1-1)。

Java 语言一推出,便以其网络上编程的独特优势赢得了许多著名公司的青睐,得到 Netscape、IBM、Microsoft、Oracle 等公司的支持,迅速得到推广。1996 年, Sun 公司成立了维护 Java 的 Javasoft 分公司, Java 图标如图 1-2 所示。

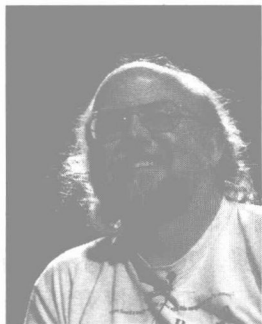


图 1-1 Java 之父 James Gosling



图 1-2 Java 图标

随着 Java 技术的不断发展,它根据市场进一步细分为针对企业应用的 J2EE(Java 2 Enterprise Edition)企业版、针对普通桌面应用的 J2SE(Java 2 Standard Edition)标准版,以及针对嵌入式设备及消费类电器的 J2ME(Java 2 Micro Edition)微缩版,共三个版本。

2. Java 语言的特点

Java 语言是一种高级的(High Level)、通用的(General Purpose)、面向对象的(Object Oriented)程序设计语言。其语法与 C 或 C++ 类似,但在组织结构上却截然不同。它是一种完全面向对象的程序设计语言,其程序的基本处理单位是类。不仅如此,在设计 Java 的时候也倾向于把它设计成一种具有生产力的语言,而不只是研究性的语言,所以在学习运用 Java 语言的时候,能很快地感受到它的便利与强大的功能。

Java 是一种简单的、面向对象的、分布式的、健壮的、结构中立的、安全的、可移植的、可解释的、性能优异的、支持多线程的、动态的语言。

1) 简单性

Java 最初是为对家用电器进行集成控制而设计的一种语言,因此它必须简单明了。Java 语言的简单性主要体现在以下方面:

系统精简,但功能齐备。对 Java 而言,其基本解释器只有 40 KB 左右,加上标准类库和线程的支持也只有 210 KB 左右,可谓短小精悍,但功能毫不逊色,对面向对象、多线程和多媒体都提供了全面的支持。

Java 的风格类似于 C++,因而对于 C++ 程序员而言非常容易掌握 Java 技术。

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等。

Java 提供了丰富的类库。

2) 面向对象的特性

面向对象可以说是 Java 最重要的特性。Java 语言完全是面向对象的,它不支持类似 C 语言那样的面向过程的程序设计技术。

3) 分布式处理的特性

Java 包含广泛的类库,可处理像 HTTP 和 FTP 这样的 TCP/IP 协议。Java 应用程序可通过一个特定的 URL 来打开并访问网络资源对象,就像访问本地文件系统那样简单。

4) 健壮性

Java 在编译和运行程序时都要对可能出现的问题进行检查。类型检查可以帮助人们检查出许多开发早期出现的错误。Java 提供自动垃圾收集机制来进行内存管理,减少了内存出错的可能性。Java 提供面向对象的异常处理机制,在编译时能进行空指针检测、数组边界检测、异常出口字节代码校验等。

5) 结构中立的特性

为了使 Java 能独立于不同的网络平台,Java 将它的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。Java 源程序被编译成一种高层次的与机器无关的字节代码,只要安装了 Java 运行系统,Java 程序就可以在任意的处理器上运行。

6) 安全性

Java 的安全性可从四个方面得到保证:

(1) Java 语言自身提供的安全性。在 Java 语言里,如指针和释放内存等 C++ 的功能被封装了起来,避免了非法内存操作。

(2) 编译器提供的安全性。当 Java 用来创建浏览器时,语言功能和一些浏览器本身提供的功能结合起来,使 Java 更安全。Java 代码在机器上执行前,要经过很多次的检测。它需要通过代码校验、检查代码段的格式、检测指针操作、检测对象操作是否过分以及是否正试图改变一个对象的类型等测试。

(3) 字节码校验。由字节码校验器对程序中的代码进行四趟扫描,如果字节代码通过代码校验没有返回错误,代码没有堆栈上溢出和下溢出,所有操作代码参数类型都是正确的,并且没有发生非法数据转换(如将整数转换成指针),访问对象操作是合法的等,才允许代码运行。

(4) 类装载。由类装载器把一个类装入 Java 虚拟机中,类装载将本机中的类与网络资源类的名称分开,以此来保证安全性。因为调入类时总要经过检查并校验,这样也就避免了特洛伊木马现象的出现。

7) 可移植性

同体系结构无关的特性使 Java 应用程序可以在配备了 Java 解释器和运行环境的任何计算机系统上运行,这成为 Java 应用软件便于移植的良好基础。通过定义独立于平台的基本数据类型及其运算,Java 数据得以在任何硬件平台上保持一致。另外,Java 的编译器由 Java 语言实现,解释器由标准 C 实现,因此 Java 本身也具有可移植性。

8) 可解释性

Java 编译器将 Java 源文件编译生成类文件,扩展名为.class,类文件可通过 Java 命令加载、解释和执行,将 Java 字节码转换为机器可执行代码。Java 解释器(运行系统)能直接运行目标代码指令。

9) 高性能

如果解释器速度不慢,Java 可以在运行时直接将目标代码翻译成机器指令。Sun 公司用直接解释器一秒钟内可调用 300 000 个进程,翻译目标代码的速度与 C/C++ 的性能没有什么区别。使用“实时(Just In Time)”编译器(也称为 JIT 编译器),可以提高 Java 代码的执行速度。Sun 公司在 Java2 发行版中提供了 JIT 编译器,JIT 编译器是 Java 虚拟机的一部分。JIT 的主要用途是将字节码指令集转换为针对某一特定微处理器的机器代码指令。

10) 多线程

Java 语言内置多线程的功能,使得在一个程序里可同时执行多个小任务。多线程带来的最大好处是更好的交互性和实时控制性。

11) 动态特性

Java 的动态特性是其面向对象设计方法的发展。它允许程序动态地装入运行时所需要的类,这是 C++ 语言所无法实现的。

Java 的这一特点使它适合于一个不断发展的环境。在 Java 类库中可以自由加入新的方法和实例变量而不会影响用户程序的执行。

3. Java 的运行机制

1) Java 虚拟机

Java 虚拟机(JVM)是软件模拟的计算机,可以在任何处理器上(无论是在计算机中还是

在其他电子设备中)安全并且兼容地执行保存在 .class 文件中的字节码。 .class 文件有时也称为字节码文件。

Java 程序的跨平台性主要是指字节码文件可以在任何具有 Java 虚拟机的计算机或者电子设备上运行，Java 虚拟机中的 Java 解释器负责将字节码文件解释成特定的机器码并投入运行。Java 源程序需要通过编译器编译成 .class 文件(字节码文件)。Java 程序的编译和执行过程如图 1-3 所示。

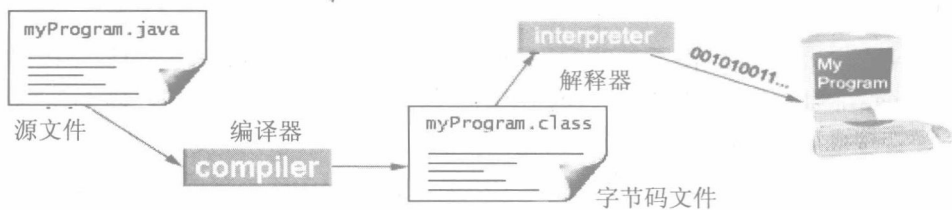


图 1-3 Java 程序的编译和执行过程

注意：Java 虚拟机的建立需要针对不同的软硬件平台做专门的实现，既要考虑处理器的型号，也要考虑操作系统的种类。如图 1-4 所示，目前在 SPARC 结构、X86 结构、MIPS 和 PPC 等嵌入式处理芯片上，以及在 UNIX、Linux、Windows 和部分实时操作系统(RTOS)上都有不同的 Java 虚拟机。



图 1-4 Java 的跨平台性

2) 垃圾内存自动回收机制

在程序的执行过程中，部分内存在使用过后就处于废弃状态，如果不及时进行无用内存的回收，就会导致内存泄漏，进而导致系统崩溃。

C++ 语言中是由程序员进行内存回收的，程序员需要把不再使用的对象内存释放掉，但是这种人为的管理方法却往往由于程序员的疏忽而致使内存无法回收，同时也增加了程序员的工作量。在 Java 运行环境中，始终存在着一个系统级的线程，专门跟踪内存的使用情况，定期检测出不再使用的内存，并进行自动回收，避免了内存的泄露，也减轻了程序员的工作量。

当然，Java 的垃圾内存自动回收机制也存在缺点，那就是由于这个系统级的线程始终在运行，从而会影响到 Java 应用的运行性能。

1.1.2 Java 开发环境与开发工具包

1. JDK——Java 开发工具包

JDK(Java Development Kit)是整个 Java 的核心,包括了 Java 运行环境 JRE(Java Runtime

Envirnment)以及 Java 工具和 Java 基础类库(rt.jar)。JDK 开发工具包是 Sun Microsystems 针对 Java 开发员的产品。自从 Java 推出以来, JDK 已经成为使用最广泛的 Java SDK(Software Development Kit)软件开发工具包。不论什么样的 Java 应用服务器, 实质都是内置了某个版本的 JDK。因此掌握 JDK 是学好 Java 的第一步。主流的 JDK 是 Sun 公司发布的 JDK, 除了 Sun 之外, 还有很多公司和组织都开发了自己的 JDK, 例如 IBM 公司开发的 JDK, BEA 公司的 Jrocket, 还有 GNU 组织开发的 JDK 等。

2. Java SDK 目前版本

J2SE(Java2 标准版), 提供了基础 Java 开发工具、执行环境与 API(应用程序接口)。

J2EE(Java2 企业版), 是由 Sun 公司提出的一组技术规格, 对企业用户以 Java2 技术开发、分发、管理多层式应用结构进行了规划, 一般用于 Web 程序开发。

J2ME(Java2 微型版), 适用于消费性电子产品, 提供嵌入式系统所使用的 Java 开发工具、执行环境与 API。

3. jdk 目录结构

安装好后的 jdk 目录如图 1-5 所示。



图 1-5 jdk 目录

bin 目录: 存放一些开发工具, 这些开发工具能够帮助开发、执行、调试 Java 程序。

jre 目录: 是 Java2 SDK 的运行环境, 包括虚拟机、类库和其他支持 Java 程序运行的文件。

lib 目录: 包含开发工具所需要的附加类库和支持文件。