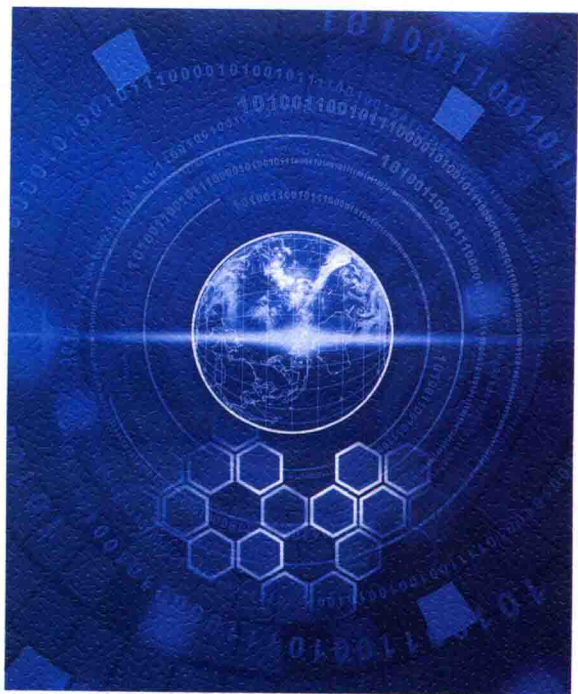


Java语言程序设计

- ◆ Java语言基础知识
- ◆ 结构化程序设计
- ◆ 类和对象、数组
- ◆ 常用类的使用
- ◆ 抽象类和接口
- ◆ 泛型和反射
- ◆ Java异常处理
- ◆ Java图形界面设计
- ◆ Swing组件编程
- ◆ Java的数据流
- ◆ 图形、图像绘制
- ◆ 进程和线程
- ◆ 网络编程
- ◆ 数据库编程



邹秀斌 向金海 刘警 主编



清华大学出版社

高等学校计算机应用规划教材

Java 语言程序设计

邹秀斌 向金海 刘警 主编



清华大学出版社

北 京

内 容 简 介

Java 语言是当前非常热门的计算机编程语言,它深受广大计算机专业人员和编程爱好者的喜爱。本书根据读者学习计算机语言的思维训练要求以及编者长期从事计算机编程的教学经验,全面、系统地介绍了 Java 语言的开发技术。全书共分 17 章,其中第 1~3 章讲解 Java 语言的基础知识、结构化程序设计,重在传统编程思想训练,同时,让读者掌握 Java 编程环境(Netbeans)。第 4~12 章重点讲解 Java 面向对象程序设计的基础知识,主要包括简单的类和对象、数组、复杂的类和对象、常见类的使用、抽象类和接口、泛型和反射、Java 异常处理、Java 的图形界面设计基础、Swing 组件编程。第 13~17 章主要讲解 Java 应用方面的技术,主要包括 Java 的数据流、绘制图形和图像、线程、网络编程、数据库编程。

本书着重传统编程思维训练,根据 Java 语言知识点,精心设计典型实例,让读者感觉耳目一新、受益匪浅;通过综合实例讲解软件开发全过程;课后习题丰富,读者可根据自己的情况选择部分或者全部习题进行练习。

本书可作为高等学校计算机相关专业教材,也可供广大 Java 程序员参考。

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 语言程序设计 / 邹秀斌, 向金海, 刘警 主编. —北京: 清华大学出版社, 2019

(高等学校计算机应用规划教材)

ISBN 978-7-302-51208-0

I. ①J… II. ①邹… ②向… 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(2018)第 207611 号

责任编辑: 刘金喜

装帧设计: 孔祥峰

责任校对: 牛艳敏

责任印制: 李红英

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

印 装 者: 北京密云胶印厂

经 销: 全国新华书店

开 本: 185mm×260mm 印 张: 24.25 字 数: 683 千字

版 次: 2019 年 1 月第 1 版 印 次: 2019 年 1 月第 1 次印刷

定 价: 59.00 元

产品编号: 078924-01

前 言

Java 程序设计语言是当前全球第一大编程语言，是一种跨平台、面向对象、网络化的高级编程语言，它是国内外大学必选的一门高级程序设计课程。为了配合计算机程序设计课程教学变化以及满足大学培养创新型人才的需要，编者结合长期教学、开发实践以及学生在学习编程语言过程中出现的各种问题，精心编写了本书。

依据 Java 语言的知识点以及计算机编程语言的教学要求，本书覆盖了 Java 语言绝大部分知识点。本书具有如下特点。

(1) 以训练编程思想为指导思路，且辅之于大量实例。

(2) 将 Java 语言的各知识点融合到项目开发过程中，使学生或自学者具备分析问题、解决问题的能力，同时提高其自身的编程技术能力。

(3) 课后习题丰富，且都是专为各章知识点量身定制，难度适中，学生或自学者可以自由选择部分或全部题目来练习。

全书共分 17 章，其中第 1~3 章讲解 Java 语言的基础知识、结构化程序设计，重在训练传统编程思想，同时，让读者掌握 Java 编程环境(Netbeans)；第 4~12 章重点讲解 Java 面向对象程序设计的基础知识，主要包括简单的类和对象、数组、复杂的类和对象、常见类使用、抽象类和接口、泛型和反射、Java 异常处理、Java 的图形界面设计基础、Swing 组件编程；第 13~17 章主要讲解 Java 应用方面的技术，主要包括 Java 的数据流、绘制图形和图像、线程、网络编程、数据库编程。

本书每章提供了大量的课后习题，而且针对性非常强。读者通过完成这些习题，可以深刻掌握 Java 知识，同时，还可以巩固所学知识，从而达到学以致用的目的。

编者还为本书编写了配套的实验教材《Java 程序设计实验教程》。读者学习时，配合该实验教材，学习效果会更好。

参加该书编写工作的教师有邹秀斌、向金海、刘警。具体分工情况是：第 1 和第 2 章由刘警编写；第 3~6 章由向金海编写；邹秀斌负责第 7~17 章的编写；全书最后由邹秀斌统编定稿。

在此，感谢江汉大学数计学院各级领导及本人所在教研室的同仁们，他们为本书编写提供了许多宝贵建议以及部分教学素材。感谢我的妻子高群玲女士，由于她一直默默操持家务，才使我有充足的时间编写此书。

在本书编写过程中，作者精益求精，但难免存在不足和纰漏，欢迎广大读者指正并提出宝贵意见和建议，可通过 xbz1234@163.com 邮箱直接与作者联系。

本书 PPT 课件和案例源文件可通过 <http://www.tupwk.com.cn/downpage> 下载。服务邮箱：wkservice@vip.163.com。

编 者
2018 年 11 月

目 录

第 1 章 了解 Java 语言	1
1.1 Java 的发展史	1
1.2 Java 技术	2
1.3 Java 语言的特点	3
1.4 Java 的应用领域	4
1.5 Java 开发环境	5
1.6 NetBeans IDE 8.1 环境介绍	6
1.7 一个简单的 Java 应用程序	8
习题 1	9
第 2 章 Java 语言基础知识	10
2.1 Java 的标识符	10
2.1.1 Java 的关键词	10
2.1.2 分隔符	11
2.2 注释	11
2.3 Java 的基本数据类型	13
2.4 常量	14
2.4.1 整型常量	14
2.4.2 浮点常量	14
2.4.3 字符常量	14
2.4.4 字符串常量	15
2.5 变量	15
2.6 运算符和表达式	16
2.6.1 算术运算符	17
2.6.2 关系运算符	18
2.6.3 逻辑运算符	19
2.6.4 位运算符	19
2.6.5 赋值运算符	21
2.6.6 条件运算符	21
2.6.7 其他运算符	22
2.6.8 运算符优先级	22
习题 2	23
第 3 章 结构化程序设计	24
3.1 结构化程序设计的基本结构	24
3.2 简单语句	25

3.3 数据的输入和输出	26
3.4 分支语句	28
3.4.1 if 语句	28
3.4.2 switch 语句	30
3.5 循环语句	31
3.5.1 while 语句和 do while 语句	31
3.5.2 for 语句	32
3.5.3 continue 语句	34
习题 3	35
第 4 章 简单的类和对象	38
4.1 面向对象技术	38
4.2 属性和方法	39
4.3 类的定义和创建对象	41
4.4 构造方法	42
4.4.1 构造方法定义	43
4.4.2 关键字 this	45
4.5 对象成员属性的初始化	47
4.5.1 static 修饰符	48
4.5.2 变量的作用域	50
4.6 访问权限修饰符	51
4.7 方法的参数传值方式	54
习题 4	55
第 5 章 数组	57
5.1 一维数组	57
5.1.1 一维数组定义	57
5.1.2 一维数组初始化	59
5.2 多维数组	60
5.2.1 二维数组定义	60
5.2.2 二维数组初始化	61
5.3 数组综合举例	62
5.4 Arrays 类	67
习题 5	68
第 6 章 复杂的类和对象	70
6.1 子类的定义	70

6.2	派生类的构造方法	71	9.2.2	常用的泛型类	127
6.3	方法继承、覆盖、重载	75	9.3	反射	141
6.3.1	方法继承	75	9.3.1	反射概念	141
6.3.2	方法覆盖	76	9.3.2	与反射相关的类	142
6.3.3	方法重载	77	习题 9		148
6.3.4	多态	78	第 10 章	Java 异常处理	151
6.3.5	异类集合	80	10.1	Java 异常层次结构	151
6.3.6	final 关键字	83	10.2	Java 异常处理语法	152
习题 6		85	10.3	抛出异常	153
第 7 章	常用类的使用	87	习题 10		155
7.1	String 类	87	第 11 章	Java 的图形界面设计基础	156
7.1.1	String 对象的初始化	87	11.1	AWT	156
7.1.2	String 类的主要方法	88	11.1.1	组件	156
7.2	StringBuffer 类	91	11.1.2	GraphicsEnvironment 类	159
7.3	正则表达式	93	11.1.3	颜色类 Color	160
7.3.1	正则表达式的相关知识	93	11.1.4	Font 类	160
7.3.2	Java 语言处理正则表达式	95	11.2	容器概念	162
7.4	基本数据类型的包装类	97	11.3	窗格概念	163
7.5	Object 类	98	11.4	布局管理器概念	165
7.6	Math 类	99	11.4.1	流式布局管理器	165
习题 7		101	11.4.2	边界布局管理器	167
第 8 章	抽象类和接口	103	11.4.3	网格布局管理器	167
8.1	抽象类的概念	103	11.4.4	卡式布局管理器	168
8.2	接口概念	107	11.5	Java 事件处理	170
8.2.1	接口定义	107	11.5.1	事件的种类	174
8.2.2	接口的实现	108	11.5.2	事件适配器	177
8.3	枚举类 Enum	111	习题 11		180
8.3.1	为什么需要枚举类型	111	第 12 章	Swing 组件编程	182
8.3.2	定义枚举类型	112	12.1	Swing 包的介绍	182
8.3.3	自定义枚举类型	113	12.2	分析 NetBeans 环境下的 Swing 应用程序	184
习题 8		115	12.3	常用对话框	188
第 9 章	泛型和反射	117	12.4	Swing 中常用控件类的使用	191
9.1	泛型	117	12.4.1	字体和颜色	191
9.1.1	泛型概念	117	12.4.2	JComponent	191
9.1.2	泛型定义	118	12.4.3	标签和图像位图	192
9.1.3	<? extends T>、<? super T> 和<?>	120	12.4.4	按钮	193
9.2	常用的泛型接口和类	124	12.4.5	文本框	197
9.2.1	常用的泛型接口	124	12.4.6	列表框和组合框	202

12.4.7 滑块	207	15.5 线程调度	303
12.4.8 微调器	208	15.6 线程通信	305
12.4.9 进度条	210	15.6.1 循环查询方式	305
12.5 菜单组件	212	15.6.2 线程同步	307
12.6 工具栏	217	15.6.3 等待/通知机制	313
12.7 表格	221	15.7 定时器	316
习题 12	227	15.8 Java 进程	319
第 13 章 Java 的数据流	229	15.8.1 ProcessBuilder	320
13.1 数据流的概念	229	15.8.2 Runtime 类	320
13.2 基本字节数据流	231	15.8.3 Process 类	321
13.2.1 文件数据流	231	习题 15	322
13.2.2 过滤数据流	232	第 16 章 网络编程	324
13.2.3 数据输入/输出流	235	16.1 网络基础知识	324
13.2.4 对象流	238	16.2 Java 地址类和接口	326
13.3 基本字符流	240	16.2.1 InetAddress 类	326
13.3.1 字符集	240	16.2.2 URL 类	327
13.3.2 与字符集相关的类	241	16.2.3 URLConnection 类	328
13.3.3 基本字符流	244	16.2.4 NetworkInterface 接口	329
13.3.4 字节流与字符流转换	245	16.3 Socket 编程	329
13.3.5 字符文件流	247	16.4 UDP 编程	333
13.4 文件与目录操作	249	16.5 多播编程	337
13.5 随机存取文件流	256	16.6 广播编程	341
习题 13	259	习题 16	343
第 14 章 图形、图像	260	第 17 章 数据库编程	344
14.1 图形	260	17.1 数据库基础知识	344
14.1.1 绘图类	260	17.1.1 数据库基本概念	344
14.1.2 绘图设置	262	17.1.2 常用的 SQL 语句	345
14.1.3 绘制基本图形和文字	265	17.2 JDBC 基础知识	347
14.1.4 图形操作	275	17.2.1 与数据连接相关的接口	
14.1.5 图形运算	279	和类	348
14.2 绘制图像	283	17.2.2 创建数据库连接方法	349
14.3 输入/输出图像	286	17.2.3 与执行 SQL 语句相关的	
14.4 绘制组件	288	接口	351
习题 14	291	17.3 访问常用数据库	356
第 15 章 线程	293	17.3.1 访问 Access 数据库	356
15.1 进程和线程的概念	293	17.3.2 访问 mysql 数据库	357
15.2 线程定义	294	17.4 数据操作	360
15.3 线程状态	296	17.4.1 查询操作	360
15.4 守护线程	301	17.4.2 更新记录集操作	363
		17.4.3 插入记录操作	365

17.4.4 删除记录操作·····	368	17.6.3 在 DBCon 类中新创建的 方法·····	375
17.4.5 JDBC 事务·····	370	17.6.4 登录类 teacherLogin 的设计···	376
17.5 SQL 数据类型与 Java 数据类型 相互转化·····	372	17.6.5 teacherSelectCourse 类·····	377
17.6 应用举例·····	374	习题 17·····	378
17.6.1 数据表及其表结构·····	374	参考文献·····	380
17.6.2 程序界面设计·····	375		

第1章 了解Java语言

本章知识目标:

- 了解 Java 语言的发展史、Java 技术、Java 语言的特点、Java 语言的应用领域。
- 掌握 Java 语言的开发环境、JDK 的安装以及 Netbeans IDE 8.1 开发平台。
- 掌握如何创建一个简单的 Hello 程序, 使读者更加熟悉 Netbeans IDE 8.1 开发平台。

1.1 Java 的发展史

1991年, 美国 Sun 公司开始研究家用消费类电子设备。在 Sun 公司内部, James Gosling 领导的 Green 小组专注于软件方面的研究, 该小组在开始阶段选择已经非常成熟的 C/C++语言进行开发和设计, 可是却发现执行 C++程序需要消耗大量的内存, 而且还不能兼容不同的设备。因此, 在 C/C++语言的基础上, Green 小组开发出一种名为 Oak 的新语言(Java 语言的前身)。

当时, 由于 Green 小组并未将 Oak 语言产品化, 因而未给 Sun 公司产生经济效益, 导致 Oak 语言面临夭折的危险。直到 1995 年, 随着互联网的出现并迅速蓬勃发展, 急需一种面向网络编程, 能够在不同终端设备、不同的操作系统上运行的语言。与此同时, Marc Andreessen 开发的 Mosaic(后来称为 Netscape)启用 Oak 项目组成员, 开发出基于 Oak 语言的 HotJava 浏览器, 并得到 Sun 公司首席执行官 Scott McNealy 的全力支持, 从而开启了 Oak 语言进军 Internet 的契机。由于 Oak 名字的版权问题, Sun 公司将 Oak 语言重新命名为 Java, 同时图标亦设计成冒着热气的咖啡(以爪哇岛咖啡命名)。

随着互联网的进一步发展, 以及 Java 语言与浏览器的结合, 产生了 Applet 技术。尽管此技术现在已被 Flash 替代, 但是此项技术使 Sun 公司的 Java 研发小组获得了重生, 让 Java 语言成为迄今为止最伟大的计算机语言。

JDK 发布历程如下。

1995年3月, Sun 公司正式向外界发布 Java 语言, Java 语言正式诞生。

1996年1月, Sun 公司公开发布 JDK 1.0。

1997年2月, Sun 公司公开发布 JDK 1.1。

1998年12月, Sun 公司公开发布 JDK 1.2, 这是 Java 语言的里程碑, Java 也被首次划分为 J2SE/J2EE/J2ME 3 种开发技术。

从此以后, 国内开发者开始学习和使用 Java 语言。

2000年5月, Sun 公司公开发布了 JDK 1.3。

2002年2月, Sun 公司公开发布了 JDK 1.4。

2004年10月, Sun 公司公开发布了 JDK 1.5, 同时 Sun 公司还将 JDK 1.5 改名为 J2SE 5.0。

2006年6月, Sun 公司公开发布了 JDK 1.6, 也称 Java SE 6.0, 同时 Java 的各版本去掉 2 的称号, J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2006年12月, Sun 公司发布 JRE 6.0。

2009年4月, Oracle 公司(甲骨文公司)收购 Sun 公司, 从而取得了 Java 的版权。

2011年7月, Oracle 公司发布了 Java 7.0 的正式版。

2014年3月，Oracle公司发布了Java 8.0的正式版，该版本增加了lambda、Default、Method等特性。

2017年9月，Oracle公司发布了Java 9.0的正式版。

2018年4月，Oracle公司发布了Java 10.0的正式版。

当前Java语言编程已非常流行。TIOBE编程语言社区排行榜是编程语言流行趋势的一个指标，每月更新，这份排行榜排名基于互联网上有经验的程序员、课程和第三方厂商的综合统计结果。2018年5月，TIOBE(<https://www.tiobe.com>)编程语言排行榜如图1-1所示。

May 2018	May 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.380%	+1.74%
2	2		C	14.000%	+7.00%
3	3		C++	7.668%	+2.92%
4	4		Python	5.192%	+1.64%
5	5		C#	4.402%	+0.95%

图 1-1 2018年5月的TIOBE编程语言排行榜

1.2 Java 技术

Java几乎是任何网络应用的基础，也是开发和提供嵌入式应用、游戏、Web内容和企业软件的全局标准。Java在全球拥有超过900万名开发人员，可高效地开发和部署功能强大的各种应用和服务。当前Java技术主要包括Java嵌入技术、Java SE、Java EE以及Java云。

1. Java 嵌入技术

Java Embedded产品经专门设计和优化，可出色满足微控制器、传感器和网关等各种嵌入式设备的独特需求，实现智能的M2M通信以及物联网系统，从而为各种设备赋予更加丰富的功能。M2M是Machine-to-Machine/Man的简称，是一种以机器终端智能交互为核心的、网络化的应用与服务。M2M协议规定了人机和机器之间交互需要遵从的通信协议。

2. Java SE

Java SE平台(Java Platform Standard Edition)旨在为各种计算平台开发安全、可移植、高性能的应用。通过开发适用于异构环境的应用，企业能够大大提高用户的生产力，改善沟通与协作，并显著降低企业应用与消费类应用的拥有成本。

3. Java EE

Java EE平台(Java Platform Enterprise Edition)是企业Java计算的行业标准。凭借一系列可增强HTML5支持、提高开发人员工作效率以及更好地满足企业需求的新特性，Java EE有效降低了样板代码编写工作量，具有更好的新Web应用和框架支持，同时还为开发人员提供了更强大的可扩展性和更丰富的功能。

4. Java 云

Java云服务为在云中开发和部署应用提供了一个企业级平台，不仅能够即时访问支持所有标准Java EE应用的云环境，还具有集成的安全性和数据库访问功能，以实现更高的工作效率——这一切均由Oracle Web Logic Server驱动。

1.3 Java 语言的特点

Java 语言是一种支持网络计算的面向对象程序设计语言。Java 语言吸收了 C++语言和 Smalltalk 语言的优点,并增加了其他特性,如支持网络通信、并发程序设计和多媒体数据控制等。Java 语言具有如下主要特点。

(1) 同 C++语言相比,Java 语言相对简单。其语法与 C++语言非常相似,这让大多数程序员学习和使用 Java 时都感到非常容易。另外,Java 抛弃了 C++语言中很少使用、难以理解、令人迷惑的一些特性,如多继承、操作符重载、自动强制类型转换等。特别是 Java 语言不再使用指针,并提供了自动废料收集机制,使编程者不必考虑内存管理的问题。

(2) Java 语言是一种面向对象的程序设计语言。Java 语言提供类、继承、接口等规则。为了简单和安全,Java 仅仅支持类之间的单向继承,抛弃了 C++语言类之间的多继承功能。而且 Java 还支持接口之间的继承,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 语言仅对虚函数使用动态绑定。总之,Java 语言是一种纯正的面向对象程序设计语言。

(3) Java 语言是面向计算机网络的高级编程语言,该语言可实现分布式编程。Java 语言支持 Internet 应用的开发。在基本的 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各种类库,包括 URL、URLConnection、Socket、ServerSocket 等。Java 的 RMI(远程方法激活)机制还是开发分布式应用的一种重要手段。

(4) Java 语言是健壮的。Java 提供了异常处理、强类型机制、废料的自动收集等功能,这些都是 Java 程序健壮性的重要技术,对指针的放弃更是 Java 的明智之举。另外,Java 的安全检查机制使其更具健壮性。

(5) Java 语言是安全的。Java 通常用于网络环境中,为此 Java 提供了一个安全机制以防恶意代码的攻击。除了 Java 语言具有的许多安全特性以外,Java 对通过网络下载类具有的一种安全防范机制(类 ClassLoader),例如,分配不同的名字空间以防替代本地的同名类。通过字节代码检查,提供了安全管理机制(类 SecurityManager)。

(6) Java 语言是跨平台的。Java 源程序(后缀名为 java 的文件)在 Java 平台上被编译为字节码文件(文件扩展名是 class 的文件),该字节码文件可以在实现该 Java 平台的任何系统中运行。

(7) Java 语言是可移植的。Java 严格规定了各种基本数据类型的长度。Java 系统本身也具有很强的可移植性,Java 编译器是用 Java 实现的,Java 的运行环境是用 ANSI C 实现的。

(8) Java 语言支持即时编译器。Java 程序在 Java 平台上被编译为字节码格式,然后可以在任何系统中的 Java 平台上运行。在运行时,Java 平台中的 Java 解释器对这些字节码进行解释执行,执行过程中需要的类就会被载入运行环境中。

(9) Java 运行效率非常高。同那些解释型的高级脚本语言相比,Java 运行效率非常高。随着 JIT(Just-In-Time)编译器技术的发展,Java 的运行速度越来越接近于 C 语言。

(10) Java 语言支持多线程。在 Java 语言中,线程是一种特殊的对象,必须由 Thread 类或其子孙类来创建。Java 语言可用两种方法创建线程:其一,使用实现了 Runnable 接口的类来包装成一个线程;其二,从 Thread 类派生子类并且重新写 run 方法,使用该子类创建的对象即为线程。值得注意的是,Thread 类已经实现了 Runnable 接口,因此,任何一个线程均有自身的 run 方法,而 run 方法中包含了线程所要执行的代码。线程的活动由一组方法控制。Java 语言支持多个线程同时执行,并提供了多线程之间的同步机制。

(11) Java 语言支持动态加载。Java 语言的设计目标之一是适应于动态变化的环境。Java 将需要的

各种类动态地加载到运行环境中，还可以通过网络来加载需要的类，这也利于软件以后的升级。另外，Java 语言能够进行运行时刻的类型检查。

Java 语言的优良特性使 Java 应用具有无比的健壮性和可靠性，这不仅减少了维护应用系统的成本，而且 Java 支持面向对象技术的特性和 Java 平台内嵌的 API 能够减少应用系统的开发时间和成本。Java 编译一次随处可运行的特性使它能够提供一种随处可用的开放结构，并低成本地在多平台之间传递信息。特别是 Java 的企业应用编程接口(Java Enterprise APIs)为企业计算及电子商务应用系统提供了有关技术和丰富的类库。

1.4 Java 的应用领域

Java 主要应用于以下几个方面。

1. 各种手机应用程序 APP(Application)

目前，Android 手机具有非常高的市场占有率，绝大部分 Android 手机的 APP 都是用 Java 编写的。如图 1-2 所示是 Android 手机界面，里面安装了支付宝、QQ 等 APP。

2. 行业和企业信息化

由于 Sun、IBM、Oracle、BEA 等国际厂商相继推出各种基于 Java 技术的应用软件和应用服务器，从而使 Java 在金融、电信、制造、互联网等领域日益得到广泛的应用。例如，Oracle 公司就基于 Java 开发了 Oracle 数据库系统。

3. 电子政务及办公自动化

Java 同样也在电子商务和网站开发上有着广泛的运用。

开发者利用架构框架 Spring MVC、Struts 2.0 或者相似框架开发的网站项目在政府、金融、医疗、保险、教育、国防等领域得到了广泛应用。

4. 嵌入式设备及消费类电子产品

目前，在手机等无线手持设备、通信终端、医疗设备、信息家电(如数字电视、机顶盒、电冰箱)、汽车电子设备上面的应用开发，一直是 Java 比较热门的应用领域。例如，几兆字节大小的 Java 程序能够在一片小芯片、传感器上运行。Java 天生就是为嵌入式设备而设计的，这也体现了 Java “立即编写，随处运行”的思想。

5. 大数据技术

目前，Java 在大数据技术领域得到了更大的发展空间。Hadoop 是基于 Java 编写的开源软件框架，用于分布式存储。大数据用户可以在不了解分布式底层细节的情况下，开发出分布式应用程序，充分利用集群进行高速运算和存储。Hadoop 实现了一个分布式文件系统(Hadoop Distributed File System, HDFS)。Hadoop 框架最核心的设计就是 HDFS 和 MapReduce。HDFS 为海量的数据提供了存储，MapReduce 则为海量的数据提供了计算。

6. 高频交易领域

Java 平台已经大大提高了即时编译技术和性能，并且 Java 也拥有 C++ 级别的传输性能。因此，Java



图 1-2 Android 手机界面

也大量用于编写高频并发系统。虽然 Java 的传输性能不如 C++，但开发者可以不用考虑 Java 的安全性、可移植性以及可维护性等问题。

7. 软件工具

目前，很多有用的软件和开发工具都是运用 Java 编写和开发的，如 Eclipse、Netbeans IDE 都是编程者经常要使用的 Java 应用开发平台。甲骨文公司还基于 Java 语言开发了 Oracle 数据库软件。

当然，Java 还有其他方面的应用，在此省略不表。

1.5 Java 开发环境

虽然开发者可以通过记事本编辑 Java 源程序，并在 Windows 中的 MS-DOS 窗口中调试 Java 程序，但是这种编写 Java 程序的方法效率非常低下。目前，主要采用的集成开发环境有 Eclipse、MyEclipse 以及 NetBeans 等。建议使用 NetBeans 平台开发 Java 程序，这使开发 Java 应用程序非常高效、方便。本节重点讲解 NetBeans 开发平台的搭建过程。

1. 安装软件包

Java 是一种跨平台的编程语言，即用 Java 编写的程序可以在不同的操作系统(如 Windows、Linux、UNIX 等)上运行，而 Java 平台是指运行在各种终端(如 PC、服务器、移动设备、嵌入式设备)上的系统软件。开发者可以利用 Java SE、Java EE、Java ME 分别开发出 Java 应用程序、服务器端 Java 程序、移动设备 Java 程序。

JDK 是 Java 的开发工具包，主要包括 Java 的运行环境、Java 的基础类库以及 Java 工具。由于 Java 被 Oracle 收购，开发者可以从 Oracle 网站下载最新的 JDK 安装包。也可以从 www.netbeans.org 网站上下载 `netbeans-8.1-windows.exe` 软件。

2. 设置环境变量

在 Windows 7 中，开发者按“控制面板”→“系统和安全”→“系统”→“高级系统设置”顺序进行操作，弹出如图 1-3 所示的“系统属性”对话框。

单击图 1-3 “高级”选项卡中的“环境变量”按钮，设置 JAVA_HOME 变量，弹出如图 1-4 所示的对话框。

然后，重新设置 path 变量，即在原值的基础上增加;%JAVA_HOME%\bin，如图 1-5 所示。



图 1-3 “系统属性”对话框

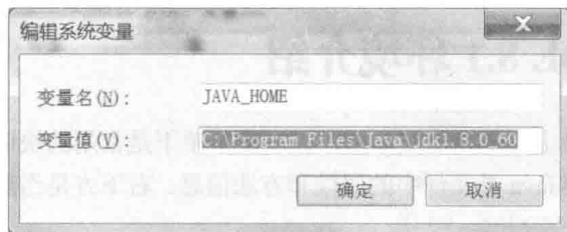


图 1-4 设置 JAVA_HOME 环境变量

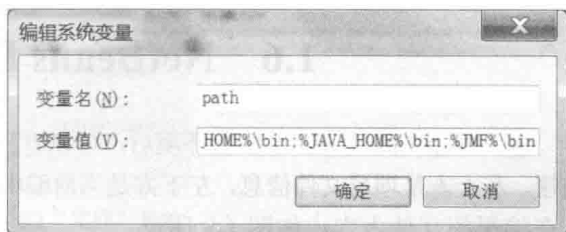


图 1-5 设置 path 环境变量

最后，还需要设置 CLASSPATH 环境变量，其值是 .;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar，如图 1-6 所示。

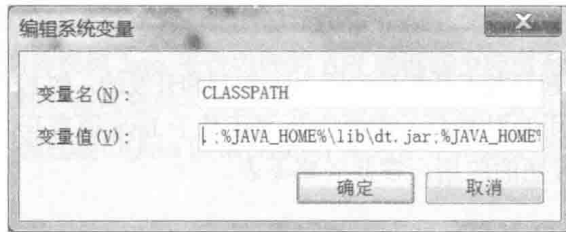


图 1-6 设置 CLASSPATH 环境变量

安装完 JDK 之后，可以安装 NetBeans 集成开发环境。

3. Java 运行环境

JRE(Java Runtime Environment)就是 Java 程序运行环境，如图 1-7 所示。通常情况下，利用 JDK 编写 Java 源程序，然后经过 javac 编译形成字节码文件(文件扩展名是.class)，该文件只能传给 JVM(即 Java 的虚拟机)，由 JVM 进行解释后被计算机执行。而 JVM 就在 JRE 中，也就是说 Java 程序的运行要由 JRE 负责。在大部分常用操作系统中，Java 都提供了相应的 JRE，绝大多数的 Java 字节码文件都不需要做任何修改，就可以在不同的 JRE 中运行。

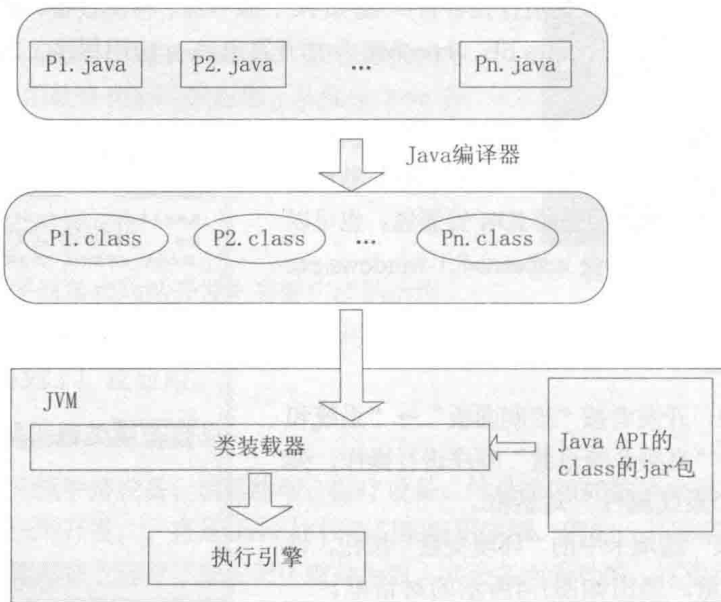


图 1-7 Java 程序运行环境

1.6 NetBeans IDE 8.1 环境介绍

进入 NetBeans IDE 8.1 编程环境后，可以看到最上方是功能菜单区，在功能菜单下是常用的快捷按钮。左上方是项目文件信息，左下方是当前编辑的 Java 类文件中的属性和方法信息。右下方是当前正在编辑的文件内容，如图 1-8 所示。

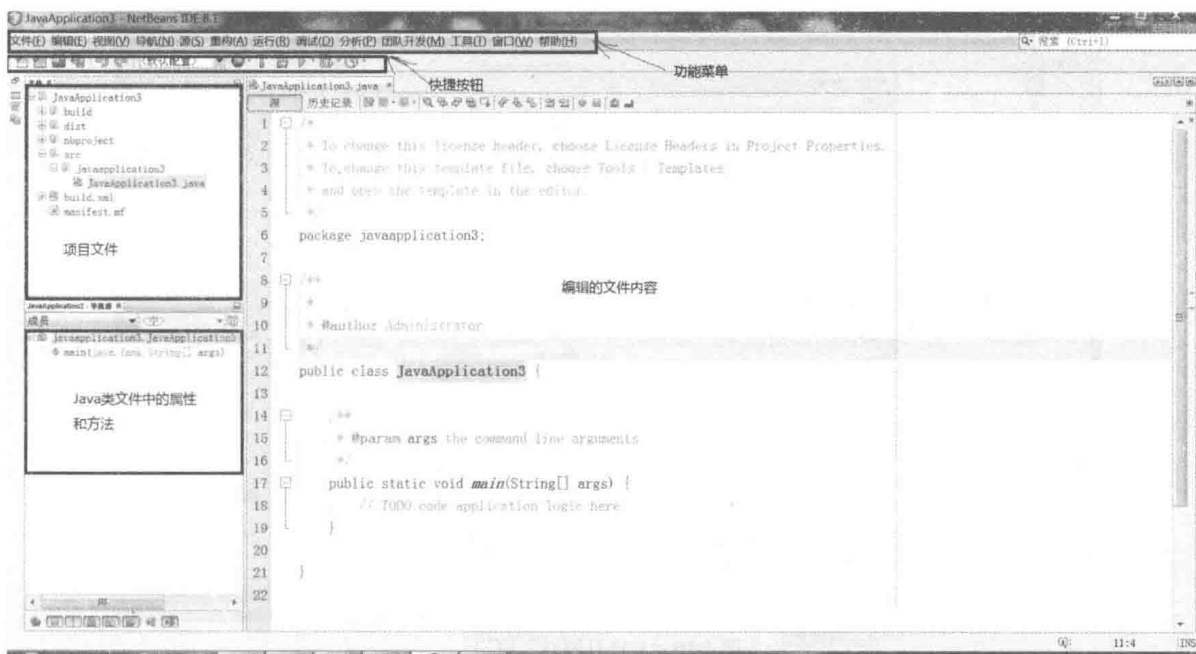


图 1-8 NetBeans IDE 8.1 界面

在“文件”菜单中，不仅可以创建、打开、关闭项目和文件，还可以设置项目属性，如图 1-9 所示。在“项目属性”设置窗口中，可以设置源包文件夹、Java 库、构建、运行、应用程序等信息，如图 1-10 所示。编程者可以添加外部存放 Java 源文件的文件夹，并选定文件的编码方式。如果选错了编码方式，打开的 Java 源文件内容就会出现乱码。

如图 1-11 所示，在“运行”菜单内，可以进行运行项目、测试项目、构建项目、清理并构建项目、运行文件等操作。

如果当前需运行的文件是项目主类，可以单击“运行项目”菜单项或者“运行文件”菜单项运行文件。如果需要运行文件而不是项目主类，则需要单击“运行文件”菜单项运行。

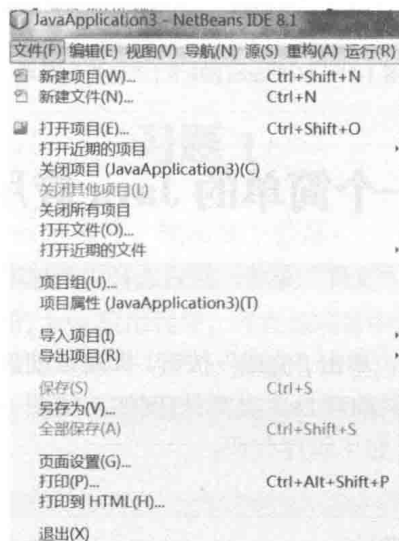


图 1-9 NetBeans IDE 8.1 “文件”菜单

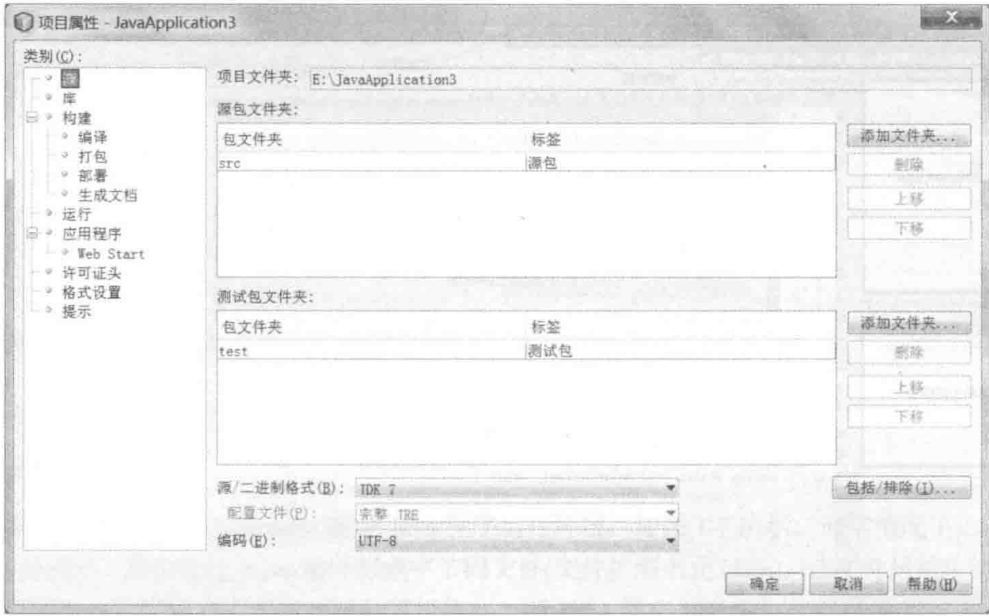


图 1-10 “项目属性”窗口



图 1-11 NetBeans IDE 8.1 “运行”菜单

1.7 一个简单的 Java 应用程序

启动 NetBeans IDE 8.1 并单击“文件”菜单，然后选择“新建项目”，创建一个 Java 应用程序项目，如图 1-12 所示。

输入项目名称和项目位置信息，单击“完成”按钮，即可以创建一个 Java 应用程序项目。此时系统自动创建了一个与项目名称相同的项目主类文件(例如，如图 1-13 中，创建的 Java 源程序是 JavaApplication4.java 文件)。输入如下程序代码：

```
public class JavaApplication4 {
    public static void main(String[] args) {
        System.out.println("Helloworld 程序");
    }
}
```



图 1-12 创建 Java 应用程序项目



图 1-13 输入 Java 应用程序项目的名称和位置

单击“运行”菜单栏，单击如图 1-11 所示的“清理并构建项目”菜单项(或按快捷键 Shift+F11)，该操作执行完成后，可以单击如图 1-11 所示的“运行文件”菜单项(或者按快捷键 Shift+F6)，就可以重新编译并运行程序。

习题 1

(1) 建立一个名为 FirstApp.java 的程序，输出如下信息：

这是第一个 Java 程序！

(2) 创建一个项目名称是 Pro1 的 Java 应用程序，并在该项目中创建 ProApp1.java、ProApp2.java、ProApp3.java、ProApp4.java，允许这些程序后，分别输出如下内容：

Hello,我是 ProApp1 程序！
Hello,我是 ProApp2 程序！
Hello,我是 ProApp3 程序！
Hello,我是 ProApp4 程序！

(3) 在外部文件夹 test 中，存在 Test1.java、Test2.java、Test3.java、Test4.java，将它们调入上面项目 Pro1 中，进行调试并运行。