



UOOC 联盟指定参考书
深圳大学教材出版基金资助

JAVA 语言程序设计教程

张 席 主编◎
陈国良 主审◎

JAVA YUYAN CHENGXU SHEJI
JIAOCHENG



西安电子科技大学出版社
<http://www.xduph.com>

UOOC 联盟指定参考书
深圳大学教材出版基金资助

Java 语言程序设计教程

张 席 主编
陈国良 主审

西安电子科技大学出版社

内 容 简 介

本书共 12 章,第 1 章为 Java 语言概述;第 2 章是基本数据类型、运算符、表达式以及语句的介绍;第 3 章主要介绍类与对象;第 4 章介绍继承与接口的概念;第 5 章为字符串及其应用;第 6 章讲述泛型与集合;第 7 章介绍 Java 异常处理机制;第 8 章是输入、输出类介绍;第 9 章为图形用户界面设计;第 10 章介绍线程的概念;第 11 章为 Java 网络编程知识;第 12 章是 Java 数据库编程。每章后面均给出习题,以巩固学习的效果,加深学生对相关知识点的理解。

本书适合作为高等学校计算机相关专业 Java 类课程的教材,也适合作为相关工程技术人员的参考用书。

图书在版编目(CIP)数据

Java 语言程序设计教程/张席主编.

—西安:西安电子科技大学出版社,2015.8

ISBN 978-7-5606-3792-1

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

中国版本图书馆 CIP 数据核字(2015)第 183404 号

策 划 马晓娟

责任编辑 马 琼 马晓娟

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 17

字 数 397 千字

印 数 1~3000 册

定 价 30.00 元

ISBN 978-7-5606-3792-1 / TP

XDUP 4084001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

序

Java 语言从上世纪九十年代中期面世，至今已整整二十年，如同一个孩子，渡过了嗷嗷待哺的婴儿期，成长为一个功能健全、魅力十足的青年。从国内外调查机构给出的行业计算机语言应用比例统计数据(比如来自国内 CSDN 的调查或者国外 TIOBE 的调查)来看，从事 Java 开发的程序员在开发者中的比例一直高居榜首。高校作为大学生和社会之间的桥梁，有责任、有义务把最切合时代前沿的知识教授给学生，为全面提升我国信息化水平做出自己的贡献。

Java 作为二十世纪九十年代中期间世的编程语言，天生就带有这个时代的痕迹，如跨平台、分布式、可靠性、安全性、多线程等。它是网络开发与应用的最好语言平台，目前已被世界各大著名公司支持并持续开发研究，比如 IBM、谷歌、微软等。我国的学术界、产业界和应用界同样一直在研究 Java，腾讯、华为、金蝶等国内知名企业也在 Java 平台上研制开发了大量的软件产品。因此，本书的出版发行，是与网络的发展相适应的，必将受到高校师生和广大 Java 从业人员的欢迎。

本书作者张席副教授长期从事“Java 程序设计”教学工作(该门课程自 1998 年便成为深圳大学计算机专业的选修课)，并在 2006 年出版《Java 语言程序设计教程》一书。经过多年发展，目前深圳大学计算机系已经形成一支老中青结合的 Java 教学团队，他们根据自己的教学经验、教学积累，对原书进行了大篇幅的修订校正，形成本书。相信他们在结合自己教学、科研工作经验的基础上编写的教材，能把 Java 的新特点、新功能体现出来，更好地促进高校计算机教学的发展。

陈国良

2015 年 5 月

前 言

俗话说，工欲善其事，必先利其器。对于 IT 行业从业者来说，扎实的编程功底就是手中最锋利的武器。从编程语言的普及程度来看，我们可以参考 CSDN 在 2014 年 1 月份做的“2013 年中国软件开发人员薪资大调查”在线调查活动的统计结果——使用 Java 的开发者比例高达 45.39%，位居第一，而使用 C# 和 C++ 的开发者比例分别仅为 17.63% 和 13.37%。国外编程语言社区排行榜 TIOBE 统计结果表明，Java 语言相对其他编程语言，使用人数一直也是遥遥领先的。作者从事高校 Java 教学工作十多年，深感 Java 语言对计算机专业学生的重要性，其相对于 C++，Java 语言更高效、灵活、健壮，既没有图形开发 API 的预封装，可提高程序员进行图形界面开发的灵活性，又取消了指针的使用，使得系统的安全性大大提高。截止到 2014 年，Java 已经陪伴广大的程序员走过了十九年，其版本从 1996 年的 JDK1.0 发展到目前的 Java SE8，功能越来越完善，使用也越来越广泛，企业、公司、高校处处都能见到它的身影。

本书凝聚了作者多年的 Java 编程教学经验，在 2006 年出版的教材《Java 语言程序设计教程》基础上重新编纂而成。全书内容由浅入深、循序渐进，既有基础类的知识介绍，也包含了泛型、GUI、网络编程与数据库连接等高级应用编程；既可作为计算机专业教材，也可作为非计算机专业的编程参考教材。书中结合 Java 编程概念，给出了大量的实例，所有实例均已在 Myeclipse 9.0 版本下调试通过。读者可在学习本书过程中参考使用实例，边思考边动手，做到融会贯通，为竞争日趋激烈的就业增加一份有力的筹码。

本书共分 12 章，第 1 章为 Java 语言概述，第 2 章是数据类型、运算符、表达式和语句，第 3 章为类与对象，第 4 章是继承与接口，第 5 章是字符串及其应用，第 6 章是泛型与集合，第 7 章为 Java 异常处理，第 8 章是 File 类与输入输出流，第 9 章为图形用户界面设计，第 10 章为线程，第 11 章为 Java 网络编程，第 12 章是 Java 数据库编程。同时每章后面均给出练习题，巩固学习的效果，加深对相关知识点的理解。

本书第 1、3、4 章由张席、谭舜泉老师编写，第 2 章由杨芳老师编写，第 5、6、9 章由于仕琪老师编写，第 7、8 章由张鹏、王平老师编写，第 10 章由林少聪老师编写，第 11、12 章由蔡树彬老师编写。在本书的编写过程中，深圳大学计算机与软件学院的陈国良院士、明仲教授给予了大力支持和帮助，同时陈国良院士百忙之中抽出时间为本书作序，在此表示最诚挚的感谢！

鉴于编者水平有限，书中不足之处在所难免，恳请广大读者批评指正。

编 者

2015 年 5 月

目 录

第 1 章 Java 语言概述.....1	2.5.5 位运算符.....25
1.1 Java 语言平台.....1	2.5.6 条件运算符.....26
1.1.1 Java 平台的版本类别.....2	2.5.7 赋值运算符和赋值表达式.....26
1.1.2 Java 语言的特点.....2	2.5.8 运算符的优先级.....27
1.2 Java 的简短历史.....4	2.6 语句.....28
1.3 Java 开发环境概述.....6	2.6.1 语句概述.....28
1.4 浅尝 Java 程序开发.....10	2.6.2 分支语句.....29
1.4.1 使用 JDK 开发 Java 应用程序.....10	2.6.3 循环语句.....32
1.4.2 使用 JDK 开发 Java Applet.....12	2.6.4 跳转语句.....34
1.4.3 使用 Eclipse 开发 Java 应用程序.....14	习题.....36
习题.....16	
第 2 章 数据类型、运算符、表达式和语句.....17	第 3 章 类与对象.....39
2.1 标识符和关键字.....17	3.1 面向对象编程概念的介绍.....39
2.2 基本数据类型.....17	3.2 类声明和类体.....40
2.2.1 整型数据.....17	3.3 构造方法与对象的创建和使用.....43
2.2.2 浮点型数据.....18	3.4 域/成员变量.....46
2.2.3 布尔型数据.....19	3.5 成员方法.....48
2.2.4 字符型数据.....19	3.5.1 “按值传递”基本数据类型参数.....50
2.3 基本数据类型之间的转换.....20	3.5.2 “按值传递”对象数据类型参数.....51
2.3.1 自动转换.....20	3.6 this 关键字.....52
2.3.2 强制类型转换.....20	3.6.1 在实例方法中使用 this.....52
2.4 数组.....21	3.6.2 在构造方法中使用 this.....53
2.4.1 数组的概念.....21	3.7 访问权限.....54
2.4.2 数组的声明和创建.....21	3.7.1 public 访问权限修饰符.....54
2.4.3 数组的初始化和赋值.....21	3.7.2 private 访问权限修饰符.....54
2.5 运算符与表达式.....22	3.7.3 protected 访问权限修饰符.....55
2.5.1 算术运算符和算术表达式.....23	3.7.4 无修饰符.....55
2.5.2 关系运算符和关系表达式.....23	3.8 嵌套类和内部类.....56
2.5.3 逻辑运算符和逻辑表达式.....24	3.9 包.....57
2.5.4 移位运算符.....25	3.9.1 创建包.....58
	3.9.2 使用包.....59
	习题.....59

第 4 章 继承与接口	61	6.5 Map 实现	93
4.1 子类与父类	61	6.6 ArrayList<E>泛型类	93
4.2 子类对象的构造过程	63	6.7 LinkedList<E>泛型类	95
4.3 成员变量隐藏与方法覆盖	64	6.8 HashSet<E>泛型类	95
4.4 super 关键字	66	6.9 TreeSet<E>泛型类	97
4.5 上转型对象	69	6.10 HashMap<K, V>泛型类	100
4.6 抽象类	72	习题	102
4.7 接口	73	第 7 章 Java 异常处理	103
4.8 接口的回调	76	7.1 异常处理概述	103
习题	78	7.1.1 异常处理基础	103
第 5 章 字符串及其应用	79	7.1.2 异常的分类	105
5.1 String 类	79	7.1.3 异常的描述	106
5.1.1 创建字符串	79	7.2 异常处理机制	107
5.1.2 字符串的长度	79	7.2.1 捕获和处理异常	107
5.1.3 字符串连接	80	7.2.2 声明抛出异常	109
5.1.4 字符串比较	81	7.3 finally 子句	110
5.1.5 常量字符串的引用	82	7.4 自定义异常	112
5.1.6 字符串的查询	82	习题	113
5.1.7 字符串的操作	83	第 8 章 File 类与输入输出流	115
5.1.8 将字符串转为数值	83	8.1 File 类	115
5.1.9 将数值转为字符串	84	8.2 输入输出流概述	117
5.1.10 创建格式化字符串	84	8.3 字节流类	119
5.2 StringBuilder 类	84	8.3.1 字节输入输出流	119
5.2.1 长度和容量	85	8.3.2 文件字节流	120
5.2.2 构造方法	85	8.3.3 管道流	123
5.2.3 StringBuilder 常用方法	86	8.3.4 数据流	126
5.3 StringBuffer 类	87	8.4 字符流类	128
习题	87	8.4.1 字符流类层次	128
第 6 章 泛型与集合	88	8.4.2 文件字符流	129
6.1 泛型	88	8.4.3 缓冲流	130
6.1.1 泛型的作用	88	习题	132
6.1.2 泛型类	88	第 9 章 图形用户界面设计	133
6.1.3 泛型接口	90	9.1 AWT、Swing 和 SWT	133
6.2 集合类概述	91	9.1.1 AWT	133
6.3 List 实现	92	9.1.2 Swing	133
6.4 Set 实现	92	9.1.3 SWT	134

9.2 一个简单例子.....	134	9.12 并发编程与线程安全.....	202
9.3 顶层容器.....	135	9.12.1 初始化线程.....	202
9.4 JFrame 窗体.....	137	9.12.2 事件调度线程.....	202
9.4.1 窗口关闭事件.....	138	9.12.3 工作线程.....	203
9.4.2 JFrame 中的常用方法.....	138	习题.....	204
9.4.3 内部窗体.....	140		
9.5 菜单.....	140	第 10 章 线程	205
9.5.1 创建菜单.....	140	10.1 线程概述.....	205
9.5.2 弹出式菜单.....	144	10.1.1 并行概念的引入.....	205
9.5.3 菜单事件处理.....	146	10.1.2 程序、进程与线程.....	206
9.6 布局管理.....	148	10.1.3 线程的状态.....	207
9.6.1 布局管理器的设置.....	149	10.2 创建线程.....	208
9.6.2 FlowLayout.....	149	10.2.1 继承 Thread 类创建线程.....	208
9.6.3 BorderLayout.....	151	10.2.2 实现 Runnable 接口创建线程.....	209
9.6.4 GridLayout.....	152	10.2.3 Thread 类的主要方法.....	210
9.6.5 BoxLayout.....	153	10.3 线程的同步机制.....	216
9.7 常用组件.....	153	10.3.1 线程的异步与同步.....	216
9.7.1 按钮.....	153	10.3.2 synchronized 关键字.....	219
9.7.2 标签.....	156	10.3.3 线程间的协作.....	221
9.7.3 单选按钮.....	157	10.3.4 线程的挂起.....	224
9.7.4 复选框.....	159	10.4 线程调度的优先级别与调度策略.....	226
9.7.5 下拉列表.....	161	习程.....	227
9.7.6 文本框与密码框.....	164		
9.7.7 文本区.....	167	第 11 章 Java 网络编程	228
9.7.8 进度条组件.....	169	11.1 网络地址 InetAddress.....	228
9.7.9 树组件.....	177	11.2 UDP 数据报.....	229
9.8 常用对话框.....	179	11.2.1 端口与数据报套接字.....	230
9.8.1 消息对话框.....	180	11.2.2 发送 UDP 数据报.....	230
9.8.2 确认对话框.....	181	11.2.3 接收 UDP 数据报.....	231
9.8.3 输入对话框.....	182	11.3 TCP 连接.....	232
9.8.4 自定义对话框.....	183	11.3.1 连接.....	232
9.8.5 文件对话框.....	183	11.3.2 套接字 Socket.....	232
9.8.6 颜色对话框.....	184	11.3.3 Socket 连接到服务器.....	232
9.9 在 Swing 组件中使用 HTML.....	185	11.3.4 ServerSocket 实现服务器.....	233
9.10 事件处理.....	188	11.3.5 服务器多线程处理套接字连接.....	235
9.10.1 窗口事件.....	188	11.3.6 Socket 关闭与半关闭.....	237
9.10.2 鼠标事件.....	192	11.4 URL 链接.....	238
9.10.3 键盘事件.....	195	11.4.1 统一资源定位符 URL.....	238
9.11 界面外观.....	199	11.4.2 获取 URL 对应的资源.....	238

11.4.3 超链接事件	239	12.3 在 Java 中执行 SQL 语句	249
习题	241	12.3.1 JDBC 和数据库连接	249
第 12 章 Java 数据库编程	242	12.3.2 Java 对数据库的增删改查操作	251
12.1 MySQL 简介	242	12.3.3 预处理语句的应用	254
12.2 MySQL 的控制台操作	242	12.3.4 结果集的选择	254
12.2.1 数据库的连接与使用	242	12.4 JTable 组件的操作	255
12.2.2 表的创建、修改和删除操作	244	习题	261
12.2.3 数据的增删改查操作	246		

第 1 章 Java 语言概述

1.1 Java 语言平台

Java 是 Sun Microsystems 公司(现在已经被 Oracle 公司兼并)的 James Gosling 领导的开发小组开发的一种编程语言,在 1995 年正式推出后,大受欢迎,近年来变得非常流行。Java 的快速发展以及被广泛接受都归功于它的设计特点。Java 不仅仅是一门编程语言,它是一整套自成体系的平台,这个平台的核心是一个运行时环境、庞大的 Java 开发类库和一个 Java 语言编译器。

Java 提供了包括安全性、跨平台性以及自动内存垃圾回收在内的一整套系统服务。Java 运行环境由一组实现了 Java 应用程序编程接口(API)的标准类库和一个 Java 虚拟机(JVM)组成。Java 源代码经过编译后,生成的是一种中间代码,称之为字节码,Java 字节码运行在 Java 运行环境中。Java 虚拟机是一个执行 Java 字节码指令的应用程序,一般运行在现有的操作系统,如 Windows 和 Linux 上,也可以在硬件裸机上直接运行。虽然 Java 应用程序是跨平台的,但是执行 Java 应用程序的 Java 虚拟机不是跨平台的,每一种支持 Java 应用程序的硬件平台和操作系统都拥有它自己的 Java 虚拟机。

大部分的现代操作系统都提供庞大的操作系统 API 库帮助程序员重用代码,减轻编程负担。这样的 API 库通常都以动态链接库的形式提供,应用程序在运行时可以按需调用它们。但是由于 Java 平台是自成体系的,不依赖于任何特定的操作系统,因此 Java 应用程序也不能依赖于任何预先存在的操作系统 API 库。为了解决这个问题,Java 平台提供了自己的一套庞大复杂的标准开发类库,这个标准类库提供了和现代操作系统 API 库相似的可重用的编程接口。大部分的 Java 标准开发类库都是用 Java 编写的,比如 Java 的图形界面编程类库 Swing,它自己绘制用户界面和处理用户事件,从而消除了在不同的操作系统中图形界面元素特征的不确定性。

Java 语言是 Java 平台的一个核心组成部分,但是除了 Java 语言以外,Java 平台,特别是 Java 运行环境还支持其他的语言。一些第三方组织已经开发出了许多基于其他语言的可运行在 Java 运行环境上的解释器,这些解释器包括 JRuby(一种基于 Java 虚拟机的脚本语言 Ruby 解释器)以及 Jython(一种基于 Java 虚拟机的脚本语言 Python 解释器和字节码编译器)。当然,在这本教程中,我们集中讨论基于 Java 语言的 Java 平台。使用 Java 编程语言编写的源代码,需要预先编译为平台中立的 Java 字节码才能够在 Java 虚拟机中运行。Java 语言编译器在编译过程中一般不对代码进行优化,因此在 Java 发展的早期,它的性能相对于其他编程语言生成的可执行程序是较差的,不过现在这个问题已经在新一代的 Java 虚拟机中通过运行时的预编译优化解决了。

1.1.1 Java 平台的版本类别

根据应用的不同, Java 平台被分为四个版本。

1. Java SE(Java 标准版)

Java SE 是使用 Java 语言进行编程开发时最常使用的一个平台。利用该平台可以开发一般用途的桌面应用程序和 Java Applet。Java SE 主要包含以下几个具有一般用途的 Java 类库。

- (1) java.lang: 提供和 Java 语言及 Java 运行环境紧密相关的基础类和接口。
- (2) java.io: 提供输入输出类, 这些类一般是面向流的, 但是其中也包含了一个用于随机访问文件的类。

(3) java.math: 提供用以进行高精度运算的数学类以及高精度的素数生成器。

(4) java.net: 提供和网络相关的输入输出类, 为基本的应用层协议提供封装类。

(5) java.text: 提供与文本处理相关的算法支持类。

(6) java.util: 主要提供与复杂数据结构算法处理相关的算法支持类。

Java SE 还包含一些有特殊用途的 Java 类库, 其中包括:

- (1) java.applet: 用以支持运行在浏览器上的 JavaApplet 小应用程序的开发。
- (2) java.beans: 用以支持 JavaBeans 架构定义的 beans 可重用小组件的开发。
- (3) java.awt: 也称之为抽象窗口工具包, 提供了支持和本地操作系统紧密相关的重组件开发的基础类, GUI 事件系统的核心以及窗口系统的基础编程接口。
- (4) java.sql: 用以支持基于 JDBC API 的数据库应用开发。
- (5) javax.swing: 在 Java.awt 基础上, 提供和平台无关的图形界面应用程序的开发。

2. Java EE(Java 企业版)

Java EE 是企业级的 Java 计算平台, 它是 Java SE 平台的扩展。这个平台为开发和运行企业级应用软件提供了 API 和运行环境。Java EE 可用于开发网络和 WEB 应用服务, 以及其他的大规模、多层次、可伸缩、高可靠性和高度安全的网络应用程序。Java EE 应用程序主要使用 Java 语言进行开发, 使用 XML 进行配置。

3. Java ME(Java 微型版)

Java ME 是专用于嵌入式系统的 Java 平台, 它的目标设备包括工控平台、移动电话、掌上电脑和机顶盒。目前世界上有超过 21 亿台支持 Java ME 的移动电话和掌上电脑。虽然目前一些最新的移动设备(iPhone、Windows Phone 和 Android 等)并没有使用它, 但它仍然在低端的移动设备上非常流行。

4. Java CARD

Java CARD 是一种使得基于 Java 的 Applet 应用程序能在智能卡片上运行的技术。Java CARD 是最小的针对嵌入设备的 Java 平台。Java CARD 被广泛应用于 SIM 卡和 ATM 卡上。

本书主要讲解 Java SE, 也即 Java 标准版相关的基于 Java 语言的编程开发。

1.1.2 Java 语言的特点

Java 语言具有以下的特点。

1. Java 是简单的

Java 比它的前辈 C 语言以及在它之前另一种占统治地位的面向对象的编程语言 C++ 都简单得多。Java 可被看作是 C++ 的一种很大程度上的改进和简化的版本，比如，C++ 中最让程序员头疼的两种特性是指针和多重继承，在 Java 中，指针被取消掉了，多重继承也被替换成了另一种更简单的语言结构——接口。另外 Java 还拥有自动的内存垃圾回收机制，使得 Java 程序员不需要像 C++ 程序员那样手动申请内存和回收内存垃圾，从而 Java 程序员的编程工作变得更为简单。

2. Java 是面向对象的

Java 是一种完全的面向对象的语言。很多面向对象的编程语言都是从早期的模块化编程语言改进过来的，而 Java 从它的原型设计时开始就遵循面向对象的原则。进行 Java 软件开发就是创建不同的对象、操作不同的对象以及使这些对象能够在一起工作。软件开发的一个关键要素就是如何重用代码。面向对象的编程思想通过封装、继承以及多态保证了代码的灵活性、模块化以及简洁性，从而确保了代码可被重用。而 Java 正是通过它简洁的面向对象的语法，使得面向对象的编程思想成为了编程界的主流。

3. Java 是分布式的

分布式计算把网络中的计算机联合起来。Java 的其中一个设计思想就是使得分布式计算变得更容易，因此分布式计算能力深深地植根在 Java 中。在 Java 中编写网络应用程序就像利用其他语言编写一个从文件中读写数据的程序那么简单。

4. Java 是解释型的

Java 字节码和机器硬件架构无关，可以运行在任何安装了 Java 解释器的计算机中。Java 解释器是 Java 虚拟机的一部分，它负责解释和执行 Java 字节码，把 Java 字节码翻译成目标机器能够运行的机器码。相反，C/C++ 的编译器将基于高级语言的源代码直接编译成和硬件架构相关的机器码，因此如果把运行在一种硬件架构上的 C/C++ 程序搬到另外一种硬件架构上，需要重新编译，才能生成适用于新的硬件架构的机器码。

5. Java 是健壮的

Java 的编译器能够报告很多其他编程语言生成的程序在编译完成后第一次运行时才能够发现的问题，因此可以帮助程序员在程序开发过程的早期就发现潜在的问题。Java 取消了在 C/C++ 中存在的容易引起错误的编程元素，如指针的取消就保证了不会出现内存泄露的问题。

Java 支持运行时的异常捕捉。它能确保程序在运行时即使出现了错误，也还能够继续执行并顺利地完成任务。

6. Java 是安全的

作为一种和因特网紧密结合的编程语言，Java 被广泛应用于网络和分布式环境。它的安全性基于一个前提：没有任何东西是可以被信任的。因此 Java 采用了多种安全机制来确保运行着 Java 虚拟机的宿主系统不会受到恶意软件的损害。

7. Java 是架构中立的

Java 是一种解释型的语言，这种特性确保了 Java 是架构中立的，或者换句话说，是与

平台无关的。Java 编译生成的字节码能够运行在任何安装了 Java 虚拟机的平台上，当今世界上主流的操作系统平台(Windows、MAC、Linux 等)都能运行 Java 虚拟机。通过使用 Java，应用程序的开发者能够真正做到“一次编写，处处运行”。

8. Java 是可移植的

正因为 Java 是架构中立的，因此用 Java 编写的程序是可移植的。Java 应用程序不需经过重新编译，就可以运行在任何可以运行 Java 虚拟机的平台上。很多编程语言，它们的语言特性是平台相关的，如 C 语言中，整型和浮点型的长度就是由 CPU 的字长决定的。但是在 Java 语言中没有针对特定平台的特性，例如 Java 整型和浮点型的长度在所有的平台上都是一样的。这就确保了 Java 的可移植性。

Java 虚拟机本身也具备可移植性，可以很容易地移植到新的硬件和操作系统上。实际上，Java 编译器本身就是用 Java 写的。

9. Java 是高性能的

在 Java 发展的早期，它的性能备受诟病，这是因为 Java 是一种解释型的语言，Java 字节码并不是由系统直接执行，它的运行速度当然不能和本地机器码的运行速度相提并论。但是近年来 Java 虚拟机的运行速度已经得到了显著的提升，这得归功于 JIT(JUST-IN-TIME)技术的引入。采用了 JIT 技术的 Java 虚拟机可以在某段字节码开始运行之前，就抢先把它编译成对应的本地机器代码，在这段字节码开始运行的时候，Java 虚拟机用对应的本地机器代码替换掉它，从而大大提升了 Java 字节码的运行速度。此外，新的 Java HotSpt 引擎还可以对经常使用的字节码进行优化，进一步提升 Java 虚拟机的运行速度。

10. Java 是多线程的

多线程对于图形用户界面和网络编程都是非常重要的。在图形用户界面编程中，一个应用程序需要同时执行多个任务，比如一个图形界面文件下载工具就需要在后台下载的同时不断更新用户界面，而一个 WEB 应用服务器需要同时为多个客户端服务。在很多编程语言中，使用多线程必须调用特定的操作系统接口，而在 Java 中使用多线程却非常简单，因为 Java 提供了简单易用的多线程类库。

11. Java 是动态的

Java 是一种动态的语言，它能够适应不断变化的应用环境。一个 Java 应用程序无需重新编译就可以在运行的时候读入一个新的类，因此用户不需要重新安装一个新的应用程序版本。在需要的时候，新的功能可以无缝地、透明地植入到 Java 应用程序中。

1.2 Java 的简短历史

Java 的诞生可用“有心栽花花不成，无心插柳柳成荫”来形容。从 1990 年开始，Sun 公司由 James Gosling 领导的开发小组试图开发一种 C/C++ 语言的替代品。他们认为 C++ 需要太多的内存，太过于复杂以至于容易导致编程错误，缺乏垃圾回收机制使得程序员必须自己进行内存管理，对分布式编程以及线程缺乏支持。他们希望新的语言能够克服 C++ 的这些缺点，同时这种语言能应用于内存较小的嵌入式系统，容易被移植到各种各样的设

备上。到了 1992 年夏天,新的语言创建出来了,Gosling 以他办公室外的橡树把它命名为 Oak,并且制作出了一个用于演示 Oak 语言能力的样机。Sun 公司原本预期 Oak 语言会在消费电子设备(如有线电视机顶盒、电话、闹钟、烤面包机)市场上派上用场,但是由于这些智能化家电的市场需求没有预期的高,Oak 语言并没有引起行业巨头的兴趣。最终 Sun 公司被迫放弃了该项计划。

从 1991 年开始直到 1994 年,Oak 语言一直都无人问津。1994 年 6 月,在经历了一场历时三天的激烈讨论后,Sun 公司的高层认为随着图形界面 WEB 浏览器的全面普及,因特网将会是一个比智能化家电更大的市场,同时因特网也需要 Oak 这样一种与平台无关、简单而可靠的编程语言,因此他们决定把 Oak 的技术应用于互联网,重新推出。当 Sun 公司打算注册 Oak 商标时发现“Oak”已被一家显卡制造商抢注,在无奈之下,他们决定把这种新的语言命名为 Java,其寓意是为世人端上一杯热咖啡。“Java”是印度尼西亚一个盛产咖啡的岛屿,中文译名是“爪哇”。

1995 年,Sun 公司正式发布 Java 语言,由于它拥有平台独立、自动垃圾回收等在当年非常先进的特性,在整个 IT 界引起了巨大轰动,当年的许多软件业界巨头,包括 Microsoft、IBM、NETSCAPE、NOVELL、APPLE、DEC、SGI 等公司纷纷购买 Java 语言的使用权,Java 的统治地位随之得到了肯定。此后 Java 的发展一波三折,早期曾经迅猛发展,也在流行几年后,经历了一段低迷期。近十年来,随着企业版的推出,Java 终于以跨平台的企业级应用站稳了脚跟。一方面,通过把 Java 和 WEB 应用服务器结合起来,使得 Java 平台成为了把企业级后台系统和 WEB 整合的首选平台,很多大企业都愿意把部分或全部的以 Java 编写的业务系统迁移到基于 Java 的高交互性的网络平台上;另一方面,Java 在企业级应用上的成熟表现也吸引了大量的开源组织投身于 Java 阵营,这些开源组织为 Java 企业级应用贡献了成熟的开源框架(Spring Framework、Hibernate 等)、开源的标准实现以及开源工具(Apache Tomcat、GlassFish Application Server 等)。这两方面的因素又反过来进一步促进了 Java 在跨平台的企业级应用上的发展。

在移动领域,最近非常受欢迎的 Google Android 操作系统也使用 Java 语言进行应用程序开发。不过 Android 操作系统并不使用传统 Java 平台所提供的类库,因为 Android 上的 Java 字节码使用一种特殊的虚拟机——Dalvik 执行,因此它不能称之为传统的 Java 平台。在 Java 曾经因为其执行效率而大受诟病的桌面应用领域,Java 也得到了长足的发展。根据一项统计,全球大约 8.5 亿台个人电脑安装了 Java 运行环境。一些广泛使用的桌面应用程序使用 Java 编写,这其中包括集成开发环境 NetBeans 和 Eclipse、办公软件 OpenOffice 以及著名的科学计算软件 Matlab。一些高端的基础软件,如 Lotus Notes 和 IBM DB2,也使用基于 Java 开发跨平台的图形用户界面。一项统计表明,现在 Java 是世界上开发人员最常使用的一种语言。世界上大约四分之一的程序员在使用 Java 进行开发工作。

以下以 Java 开发包(JDK)的版本号演化为主线,对 Java 发展历程中的大事进行回顾:

- (1) 1995 年 5 月 23 日,Java 语言诞生。
- (2) 1996 年 1 月,第一个 JDK——JDK1.0 诞生。
- (3) 1996 年 4 月,10 个最主要的操作系统供应商声明,将在其产品中嵌入 Java 技术。
- (4) 1996 年 9 月,约 8.3 万个网页使用了 Java 技术来制作。
- (5) 1997 年 2 月 18 日,JDK1.1 发布。

(6) 1997 年 4 月 2 日, Java One 会议召开, 参与者逾一万人, 创当时全球同类会议规模之纪录。

(7) 1997 年 9 月, Java Developer Connection 社区成员超过十万。

(8) 1998 年 2 月, JDK1.1 被下载超过 2 000 000 次。

(9) 1998 年 12 月 8 日, Java2 企业平台 J2EE 发布。

(10) 1999 年 6 月, Sun 公司发布 Java 的三个版本: 标准版(J2SE)、企业版(J2EE)和微型版(J2ME)。

(11) 2000 年 5 月 8 日, JDK1.3 发布。

(12) 2000 年 5 月 29 日, JDK1.4 发布。

(13) 2001 年 6 月 5 日, Nokia 宣布, 到 2003 年将出售 1 亿部支持 Java 的手机。

(14) 2001 年 9 月 24 日, J2EE1.3 发布。

(15) 2002 年 2 月 26 日, J2SE1.4 发布, 自此 Java 的计算能力有了大幅提升。

(16) 2004 年 9 月 30 日 18:00p.m., J2SE1.5 发布, 成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE1.5 更名为 Java SE 5.0。

(17) 2005 年 6 月, JavaOne 大会召开, Sun 公司公开 Java SE 6。此时, Java 的各种版本已经更名, 以取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

(18) 2006 年 12 月, Sun 公司发布 JRE6.0。

(19) 2009 年 12 月, Sun 公司发布 Java EE 6。

(20) 2011 年 7 月 28 日, Oracle 公司发布 Java SE 7。

(21) 2014 年 3 月 26 日, Oracle 公司发布 Java SE 8。

1.3 Java 开发环境概述

学习 Java 的第一步就是掌握 Java SE 平台的使用。如果只想运行别人写好的 Java 程序, 可以只安装 Java 运行环境, 也即 JRE。JRE 由 JVM、Java 核心类库以及一些支持文件组成, 可以登录到 Oracle 公司的网站免费下载。此外, Oracle 公司提供了 Java SE 平台下的开发包, 简称为 JDK(Java Development KIT), 目前的版本号为 7。也可以登录到 Oracle 公司的网站免费下载 JDK7。下载安装后就可以编写 Java 程序并进行编译、运行了。这是因为 JDK 中已经包含了 JRE。

安装好的 JDK 主要目录内容如图 1.1 所示, 分别介绍如下:

(1) bin 目录: 用以开发、执行、调试和打包 Java 程序的实用程序。

(2) db 目录: 由 Oracle 打包的开放源代码数据库 Apache Derby, 这个数据库可以被嵌入 Java 应用程序中。

(3) demo 目录: Java SE 平台的示例源代码, 包括了使用 Swing 和其他 Java 基类以及 Java 平台调试器体系结构的示例。

(4) include 目录: 包括了支持使用 Java 本地接口、JVM 工具接口以及 Java 平台的其



图 1.1 JDK 的目录结构

他功能进行本地代码编程的 C 语言头文件。

(5) jre 目录: 这是包含在 JDK 内部的 Java 运行环境, 也即 JRE, 其中包括了 Java 虚拟机、类库以及其他支持执行 Java 程序的文件。

(6) lib 目录: 附加库目录, 里面包括了 Java 开发过程中所需的其他类库以及相关的支持文件。

(7) sample 目录: 包括了某些 Java API 的编程示例源代码。

(8) src.zip 文件: Java 核心 API 类库中所有类的 Java 源文件(也即 Java.*、Javax.* 等基础包中包含的类的源文件)。

在 JDK 安装完毕后, JDK 平台提供的 Java 编译器(Javac.exe)、Java 解释器(Java.exe)和 Java 调试器(jdb.exe)都位于 Java 安装目录下的 bin 子目录中, 为了能在任何目录中使用 Java 编译器和解释器, 还需要设定系统环境变量 Path。在 Windows 平台中, 右键单击“我的电脑”, 在弹出的右键快捷菜单中选择“属性”, 再单击弹出的“系统属性”对话框中的“高级”选项栏, 然后单击其中的“环境变量”按钮, 添加系统环境变量。假设 JDK 安装在 C:\JDK 中, 则应该在原有的环境变量字符串后添加“C:\JDK”, 如图 1.2 所示。

此外, Java 安装目录下的 jre 子目录中包含了 Java 应用程序运行时所需的 Java 类库, 这些类库包含在 jre\lib 目录下的压缩文件 rt.jar 中。一般情况下, 使用这些 Java 类库不需要额外的设置, 但是在某些特殊情况下, 系统中可能包含多个彼此不兼容的 Java 应用环境, 这个时候可能会发生冲突, 出现类似“程序要加载的类无法找到”这样的运行时错误。为了显式的指明我们所使用的当前 Java 类库, 需要另外设定系统环境变量 Classpath。在 Windows 平台中, 需要添加如图 1.3 所示的系统环境变量。

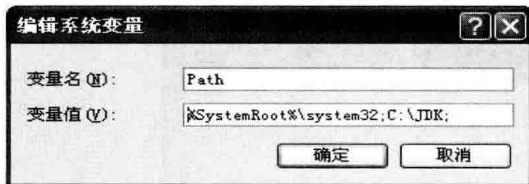


图 1.2 设置环境变量 Path

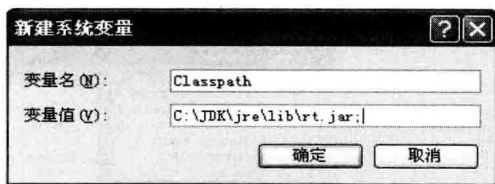


图 1.3 设置环境变量 Classpath

单独安装的 JDK 中是不包含帮助文档的。为了给在 Java SE 平台下的编程开发工作提供便利, 建议下载 Oracle 公司提供的帮助文档, 帮助文档在 Oracle 公司的网站上可以免费下载, 文件名为 jdk-7u2-apidocs.zip。

在 JDK 安装好后, 基本的 Java 开发环境就已经具备了。开发一个 Java 应用程序是一个迭代循环的过程:

- (1) 编写/修改源文件。
- (2) 编译源文件生成字节码, 如果出现编译错误, 则重回第 1 步。
- (3) 运行字节码, 如果一切正常, 则开发过程结束, 如果出现异常, 则进入第 4 步。
- (4) 调试 Java 应用程序, 找出错误所在, 重回第 1 步。

在第 2 步中, 使用的是 JDK 提供的 Java 编译器(Javac.exe), 在 3 步中, 使用的是 JDK 提供的 Java 解释器(Java.exe), 在第 4 步中, 使用的是 JDK 提供的 Java 调试器(jdb.exe)。在第 1 步中, 需要使用一个文字编辑器(如记事本)来编写源文件, 不可以使用文字处理软件, 如 Microsoft Word, 这是因为文字处理软件生成的文档中含有 Java 编译器无法识别的不可

见字符。

因此一个编辑器加上 JDK 就可以构成一个最简单的 Java 开发环境。在 Windows 平台上最简单的编辑器就是记事本，但是记事本的功能非常有限，不适宜于进行应用程序的开发，一般用于进行 Java 应用程序开发的编辑器需要支持语法高亮。在 Windows 平台上一个小而精致的支持语法高亮的编辑器是 Notepad++，Notepad++ 可以在它的主页上免费下载 (<http://notepad-plus-plus.org>)。Notepad++ 提供了一系列编程相关的功能，如自动识别源代码类型，支持自动缩进，语法高亮，支持单词、函数名称自动补全等。通过配置，还可以在 Notepad++ 内运行 Java 命令程序。使用 Notepad++ 开发 Java 应用程序的界面如图 1.4 所示。

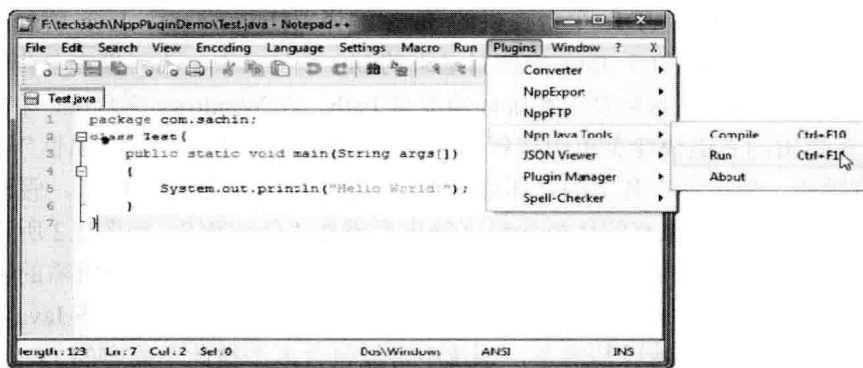


图 1.4 使用 Notepad++ 开发 Java 应用程序的界面

除了 Notepad++ 外，Windows 平台上还有许多免费的编辑器，如 WinEdit、JEdit 等，也都支持开发 Java 应用程序。在 Linux 平台上的两大编辑器 EMACS 和 VIM 均可用于开发 Java 应用程序。使用 EMACS 开发 Java 应用程序的界面如图 1.5 所示。



图 1.5 使用 EMACS 开发 Java 应用程序的界面

但是如果要设计开发较大型的 Java 应用程序项目，还是要使用 IDE(Integrated Development Environment, 集成开发环境)，它是一种辅助程序开发人员开发软件的应用软件。IDE 是集成了代码编写功能、分析功能、编译功能、调试功能等一体化的开发软件集合。市面上有大量的 Java IDE，这些 IDE 产品都集成了 JDK 作为其主要的组成部分，其中