

- ✓ 总结了作者长期教学培训成果，难易适中，实用性强
- ✓ 系统全面介绍Java 6技术要点
- ✓ 围绕丰富实例讲解Java编程实践知识
- ✓ 精心编写大量“实验指导”，引导学生深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持



■ 刘万军 郑少京 王松波 梁清华 等编

Java 6 程序设计

实践教程

清华大学出版社

- ✓ 总结了作者长期教学培训成果，难易适中，实用性强
- ✓ 系统全面介绍Java 6技术要点
- ✓ 围绕丰富实例讲解Java编程实践知识
- ✓ 精心编写大量“实验指导”，引导学生深入学习编程实践
- ✓ 课后提供丰富习题，巩固学习成果
- ✓ 网站提供代码下载和课件支持

■ 刘万军 郑少京 王松 梁清 等

Java 6 程序设计

实践教程

清华大学出版社
北 京

内 容 简 介

本书根据 Java 语言程序面向对象的本质特征,以及面向对象程序设计课程的基本教学要求,详细介绍 Java 语言及其面向对象的基本特性、基本技术。全书共分为 12 章,介绍了 Java 语言的产生和发展,Java 语言基础,面向对象概念和面向对象开发过程,面向对象高级特征,字符串和日期类,异常处理机制,Java Swing,Java 事件,多媒体技术,输入输出流,访问数据库等内容。最后一章使用 Java 开发了一个综合实例,介绍如何设计专卖店管理系统的数据库以及如何通过 Swing 实现专卖店管理系统的知识。

本书可作为普通高等院校计算机专业和非计算机专业的 Java 程序设计教材,也可作为读者的自学用书。

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 6 程序设计实践教程 / 刘万军等编著. —北京:清华大学出版社, 2009.5
ISBN 978-7-302-19449-1

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

中国版本图书馆 CIP 数据核字(2009)第 016236 号

责任编辑:夏兆彦

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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:三河市春园印刷有限公司

经 销:全国新华书店

开 本:185×260 印 张:27 字 数:635 千字

版 次:2009 年 5 月第 1 版 印 次:2009 年 5 月第 1 次印刷

印 数:1~5000

定 价:39.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:031602-01

现在, 计算机网络正在以前所未有的速度发展。在网络程序设计领域, Java 面向对象编程技术已经成为计算机网络应用开发的主流, 从电子商务、远程教学到网络游戏, 都在纷纷使用 Java 技术。Java 手机编程和基于 Java 技术的各种芯片的应用等, 在日常生活中也随处可见。

Java 是一种既面向对象又可跨平台的语言, 具有简单、解释型、动态、多线程、安全等特点。Java 是一种网络编程语言, 最大限度地利用了网络。Java 的小应用程序(applet)可在网络上传输而不受 CPU 和环境的限制。Java 还提供了丰富的类库, 使程序设计者可以很方便地建立自己的系统。JDK 6.0 在旧版本的基础上提高了性能和 GUI 应用, 是 Java 程序员的最佳选择。

1. 本书主要内容

本书根据 Java 语言程序面向对象的本质特征, 以及面向对象程序设计课程的基本教学要求, 在阐述面向对象程序设计基本理论和方法的基础上, 详细介绍了 Java 语言及其面向对象的基本特性、基本技术。全书共分为 12 章, 其详细介绍如下。

第 1 章介绍 Java 语言的产生和发展。首先介绍 Java 语言的概念、特点、工作原理和 API。接下来, 介绍 Java 开发环境, 即 JDK 的安装和配置。并以实例方式介绍使用记事本开发 Java 程序的过程。最后, 介绍 Java 程序的两种常用开发工具: Eclipse 和 NetBeans。

第 2 章介绍 Java 语言的基本组成元素, 包括 Java 基本数据类型、控制语句、数组和向量类等内容。熟练掌握此章, 对于 Java 开发有着很重要的作用。

第 3 章首先从面向对象与面向过程的比较入手, 介绍对象的概念以及面向对象程序的开发过程。然后, 介绍如何在 Java 中创建 Java 类、方法和变量。接着, 介绍 Java 对象、方法和继承。最后, 介绍了内部类和抽象类。

第 4 章介绍面向对象的一些高级特性, 如包、接口、访问控制修饰符和封装类等。

第 5 章介绍 Java 中字符串和日期类, 包括字符串类 String、可改变字符串类 StringBuffer、Date 类、Calendar 类、格式化日期类等内容。

第 6 章讲述 Java 异常处理机制, 包括 Java 异常概念、异常处理机制和自定义异常等内容。

第 7 章讲述 Java Swing。首先, 介绍 Java GUI 图形界面发展历程。然后, 介绍 Java 常用容器, 如 JFrame、JDialog 和 JPanel 等。接下来, 介绍 Swing 常用组件, 如按钮、文本框、组合框和菜单等。最后, 介绍 Java GUI 中常用布局管理器 and 高级组件。

第 8 章讲述 Java 事件。首先, 介绍 Java 授权模型处理机制和 Java 事件中常用事件监听器接口。接下来, 介绍 Java 中低级事件和高级事件。最后, 介绍 Swing 中的典型

事件。

第 9 章讲述 Java 多媒体技术。首先, 介绍 Java 图形处理技术。如绘制各种常见基本图形。接下来, 介绍 Java 小应用程序 Applet 和 Japplet。最后, 介绍如何在 Applet 中处理声音。

第 10 章介绍 Java 输入输出流, 主要包括 File 类、流基本概念、读取数据和写入数据等内容。

第 11 章讲述 Java 数据库技术。介绍 JDBC 组件基本概念和接口。同时介绍不同的数据库访问方式, 以及如何使用 JDBC 组件提供的接口操作数据库等内容。

第 12 章介绍汇智专卖店管理系统。对专卖店管理系统需求和功能进行分析, 对系统总体结构进行介绍。并且, 还介绍如何设计专卖店管理系统的数据库以及在 Swing 中实现专卖店管理系统中的重点和难点模块。

2. 本书主要特色

本书通过实例介绍 Java 面向对象程序开发的相关知识, 具有实用性教程的特色。

- 本书汇总了作者多年的程序员职业教学培训经验, 内容组织合理, 实例丰富全面。
- 本书通过 Java 语言开发了大量实例。读者可以通过这些丰富实例学习 Java 编程技术。
- 本书编写了大量“实验项目”, 指导读者应用相应知识独立练习编程项目。
- 每章课后练习题帮助学生检查对 Java 开发理论知识的掌握程度。
- 清华大学出版社网站提供本书实例完整源代码和教学视频。

3. 本书读者对象

书中采用大量的实例进行讲解, 力求通过实例使读者更形象地理解面向对象思想, 快速掌握 Java 编程技术。本书难度适中, 内容由浅入深, 实用性强。每章附有精心编写的实验和习题, 便于读者实践和巩固所学知识。本书可作为普通高等院校 Java 程序设计课程的教材, 也可作为读者的自学用书。

参与本书编写的除了封面署名人员外, 还有李乃文、肖新峰、宋强、马海军、许勇光、王泽波、孙江玮、田成军、刘俊杰、李海庆、王树兴、朱俊成、王敏、张瑞萍、王黎、安征、亢凤林、康显丽、李海峰、崔群法、孙岩、祁凯、倪宝童、王立新、吴越胜、何方、张银鹤等。

由于时间仓促, 水平有限, 疏漏之处在所难免。欢迎读者朋友登录清华大学出版社的网站 www.tup.com.cn 与我们联系, 帮助我们改进提高。

第 1 章 Java 语言概述	1
1.1 Java 历史	1
1.2 Java 概述	4
1.2.1 Java 介绍	4
1.2.2 Java 语言特点	6
1.2.3 Java 工作原理	10
1.2.4 Java API 介绍及使用	12
1.3 Java 运行环境	13
1.3.1 JDK 安装以及配置	13
1.3.2 Java 应用程序开发	17
1.3.3 Java 小应用程序开发	19
1.4 Java 开发工具简介	21
1.4.1 Eclipse 介绍及使用	21
1.4.2 NetBeans 介绍及使用	26
1.5 实验指导	29
1.5.1 安装并配置 JDK	29
1.5.2 使用 Eclipse 创建 Java 程序	30
1.5.3 使用 NetBeans 创建 Java 程序	31
1.6 习题集	33

第 2 章 Java 基本语法	35
2.1 Java 语言基本组成元素	35
2.1.1 关键字与标识符	35
2.1.2 运算符与表达式	37
2.2 Java 数据类型	40
2.2.1 数据类型	41
2.2.2 数据类型转换	44
2.3 控制语句	45
2.3.1 分支语句	45
2.3.2 循环语句	52
2.3.3 跳转语句	57

2.4 数组	59
2.4.1 一维数组与多维数组	60
2.4.2 一些与数组操作相关函数	64
2.4.3 向量类	65
2.4.4 应用举例	66
2.5 实验指导	68
2.5.1 猜字游戏	68
2.5.2 使用冒泡排序	69
2.5.3 输出图形	70
2.5.4 打印九九乘法表	71
2.6 习题集	72
第 3 章 面向对象程序设计	77
3.1 面向过程与面向对象	77
3.1.1 编程思路的发展	77
3.1.2 面向对象设计	79
3.2 对象入门	79
3.3 面向对象开发过程	81
3.4 类	83
3.4.1 类定义	84
3.4.2 属性和方法	85
3.4.3 访问类	87
3.4.4 构造方法	88
3.5 对象	89
3.5.1 Java 对象定义	89
3.5.2 对象引用	92
3.6 基本方法	93
3.6.1 this	93
3.6.2 方法重载	94
3.6.3 静态变量和方法	96
3.6.4 常量和最终方法	98
3.7 继承	99
3.7.1 Java 中实现继承	99

3.7.2 方法重写和变量隐藏	101	5.1.4 字符串检索	150
3.7.3 使用 super	103	5.1.5 字符串其他操作	151
3.8 内部类	105	5.2 可改变字符串类	153
3.9 抽象类	106	5.2.1 创建 StringBuffer 类	154
3.10 实验指导	109	5.2.2 StringBuffer 应用	154
3.10.1 实现银行账户类	109	5.3 字符串拆分类 StringTokenizer	157
3.10.2 实现秒表类	111	5.4 常用日期类	158
3.10.3 实现校园 201 电话卡类	113	5.4.1 Date 类	158
3.10.4 实现交通工具类	114	5.4.2 Calendar 类	159
3.11 习题集	116	5.5 格式化	162
第 4 章 面向对象高级特征	122	5.5.1 格式化日期	162
4.1 包	122	5.5.2 格式化数字	164
4.1.1 包声明	122	5.6 实验指导	165
4.1.2 包引用	123	5.6.1 检索字符串	165
4.1.3 创建并引用包求素数	124	5.6.2 对字符串加密解密	166
4.2 访问控制修饰符	125	5.6.3 利用 3 种格式输出时间	167
4.2.1 控制修饰符介绍	125	5.7 习题集	168
4.2.2 public 修饰符	126	第 6 章 异常处理机制	174
4.2.3 protected 修饰符	127	6.1 异常	174
4.2.4 默认修饰符和 private 修饰符	128	6.1.1 异常概念	174
4.3 接口	130	6.1.2 Java 异常类组织结构	175
4.3.1 接口概念和意义	130	6.2 异常处理机制	176
4.3.2 接口声明和继承	130	6.2.1 try...catch 语句捕获异常	176
4.3.3 接口实现	131	6.2.2 多个 catch 语句	178
4.3.4 接口回调	132	6.2.3 try...catch 语句嵌套	180
4.4 数据封装类	133	6.2.4 finally 语句	181
4.5 实验指导	136	6.2.5 抛出和声明异常	183
4.5.1 包管理	136	6.3 自定义异常	185
4.5.2 模拟主板上的 PCI 插槽	137	6.4 实验指导	187
4.5.3 登录程序	139	6.4.1 数字转换异常处理	187
4.6 习题集	140	6.4.2 多异常处理	189
第 5 章 字符串和日期类	146	6.5 习题集	190
5.1 字符串 String 类	146	第 7 章 Java Swing	194
5.1.1 创建字符串	146	7.1 Java GUI 概述	194
5.1.2 字符串常用方法介绍	148	7.1.1 发展历史	194
5.1.3 字符串比较	149	7.1.2 Swing 概念	198
		7.1.3 Swing 类库层次结构	199

7.2	窗口与面板	200
7.2.1	容器概述	200
7.2.2	JFrame 主窗口	201
7.2.3	JDialog 对话框	203
7.2.4	JPanel 面板	205
7.3	Swing 常用组件	207
7.3.1	按钮	207
7.3.2	文本组件	210
7.3.3	组合框	214
7.3.4	进度条及 Timer 组件	215
7.3.5	菜单	219
7.3.6	工具栏	224
7.4	布局管理器	227
7.4.1	边框布局	227
7.4.2	流式布局	229
7.4.3	网格布局	230
7.4.4	null 布局	231
7.5	高级组件	232
7.5.1	表格组件	232
7.5.2	树组件	235
7.5.3	选项卡组件	237
7.6	实验指导	240
7.6.1	用户注册界面	240
7.6.2	选项卡界面	242
7.6.3	计算器界面	245
7.6.4	记事本界面	248
7.7	习题集	250
第 8 章	Java 事件	256
8.1	Java 事件处理机制	256
8.1.1	事件处理模型	256
8.1.2	为组件注册事件	259
8.2	事件监听器接口	261
8.3	GUI 低级事件	264
8.3.1	键盘事件	264
8.3.2	鼠标事件	267
8.3.3	焦点事件	270
8.3.4	窗口事件	272
8.4	GUI 高级事件	275

8.4.1	选项事件	275
8.4.2	单击事件	277
8.5	Swing 常用事件	279
8.5.1	弹出菜单事件	279
8.5.2	TreeSelectionEvent 树事件	281
8.6	实验指导	284
8.6.1	实现图形绘画	284
8.6.2	实现按钮事件	286
8.6.3	实现资源管理器	287
8.6.4	实现计算器	290
8.7	习题集	292
第 9 章	多媒体技术	294
9.1	图形处理技术	294
9.1.1	Graphics 图形类	294
9.1.2	基本图形绘制	297
9.1.3	图形效果处理	300
9.2	Java JApplet	303
9.2.1	JApplet 概述	303
9.2.2	JApplet 编写	304
9.2.3	HTML 文件的参数设定	305
9.3	声音处理技术	308
9.4	实验指导	309
9.4.1	绘制钟表	309
9.4.2	信号灯	312
9.5	习题集	314
第 10 章	输入输出流	318
10.1	File 类	318
10.1.1	File 类概述	318
10.1.2	获取文件属性信息	320
10.1.3	文件重命名和删除	321
10.1.4	显示目录文件	322
10.2	流基本概念	323
10.2.1	什么是流	323
10.2.2	输入输出流机制	324
10.3	流的使用	326
10.4	文件读写处理	327

10.4.1	FileInputStream 与 FileOutputStream 类	327	11.4	JDBC 元数据	369
10.4.2	FileReader 和 FileWriter 类	331	11.4.1	使用 DatabaseMetaData	369
10.4.3	RandomAccessFile 类	333	11.4.2	使用 ResultSetMetaData	370
10.5	应用举例——文件信息 读取与写入	336	11.5	GUI 中显示 MySQL 数据	372
10.6	实验指导	339	11.5.1	添加数据	373
10.6.1	利用输入流求阶乘	339	11.5.2	显示数据	375
10.6.2	过滤符合条件的文件	340	11.6	实验指导	376
10.6.3	文件内容的复制	342	11.6.1	JDBC 应用	376
10.6.4	随机访问文件	343	11.6.2	成绩管理—— 添加、查询	379
10.7	习题集	344	11.6.3	成绩管理—— 删除、修改	382
第 11 章	访问数据库	348	11.7	习题集	386
11.1	JDBC 基础	348	第 12 章	汇智专卖店管理系统	390
11.1.1	JDBC 概述	348	12.1	系统概述	390
11.1.2	JDBC API 介绍	350	12.1.1	需求分析	390
11.1.3	JDBC 连接数据库方式	354	12.1.2	系统用例图	391
11.2	数据库常见操作	356	12.1.3	系统设计	394
11.2.1	连接数据库	357	12.2	数据库设计	395
11.2.2	显示数据库数据	360	12.3	通用模块实现	397
11.2.3	更新数据库数据	361	12.4	系统前台模块	404
11.2.4	查询数据库数据	362	12.4.1	系统主界面	404
11.3	JDBC 高级操作	363	12.4.2	用户注册模块	406
11.3.1	使用 PreparedStatement 接口执行 SQL 语句	364	12.5	系统后台模块	409
11.3.2	使用 CallableStatement 接口执行 SQL 语句	366	12.5.1	管理员登录模块	409
			12.5.2	商品管理模块	411
			12.5.3	订单管理模块	416

第1章 Java 语言概述



内容摘要 | Abstract

在经历了以大型机为代表的集中计算模式和以 PC 为代表的分散计算模式之后, 互联网的出现使得计算模式进入了网络计算时代(异构时代)。网络计算模式的一个特点是计算机是异构的, 即计算机的类型和操作系统是不一样的。网络计算模式的另一个特点是代码可以通过网络在各种计算机上进行迁移。这就迫切需要一种跨平台的编程语言, 使得用其编写的程序能够在网络中的各种计算机上正常运行, Java 就是在这种需求下产生的。正是因为 Java 语言符合了互联网时代的发展要求, 才使它获得了巨大的成功。

Java 是一种网络编程语言, 是一种既面向对象又可跨平台的语言, 具有简单、解释型、动态、多线程、安全等特点。本章首先介绍 Java 的产生和发展, 然后介绍 Java 语言的概念、特点、工作原理和 API, 接下来介绍 Java 的开发环境, 即 JDK 的安装和配置, 并以实例的方式介绍使用记事本开发 Java 程序的方法。最后介绍 Java 程序两种常用的开发工具: Eclipse 和 NetBeans。



学习目标 | Objective

- 理解 Java 产生的必然性
- 掌握 Java 概念
- 了解 Java 特点
- 熟练掌握 Java 工作原理
- 掌握使用 Java API
- 掌握 JDK 的安装和配置
- 熟练掌握开发 Java 应用程序和小应用程序
- 熟练使用 Eclipse 和 NetBeans 开发 Java 程序

1.1 Java 历史

从 JDK 诞生到现在已经有 13 年了。到目前为止, JDK 已经发布了 6 个版本。在这 13 年里, 诞生了无数和 Java 相关的技术和标准。现在, 从 1995 年开始, 回顾一下 Java 的产生和发展过程。

1. Java 产生背景

1991 年 4 月, Sun 公司的 James Gosling 领导的绿色计划 (Green Project) 开始着力发展一种分布式系统结构, 使其能够在各种消费性电子产品上运行。Green 项目组的成员一开始使用 C++ 语言来完成这个项目。由于 Green 项目组的成员都具有 C++ 背景, 所

以他们首先把目光锁定在 C++ 编译器上。Gosling 首先改写了 C++ 编译器。但是,很快他发现 C++ 存在很多不足,需要研发一种新的语言来替代它。

这种语言最初被命名为 Oak,目标设定为家用电器等小型系统的编程语言,用来解决诸如电视机、电话、闹钟、烤面包机等家用电器的控制和通讯问题。17 个月之后,整个系统完成,这个系统是更类似机顶盒式的操作系统。不过,在当时市场不成熟的情况下,他们的项目没有获得成功。但 Oak 语言却得到了 Sun 公司总裁的赏识。

2. Java 产生

直至 1994 年下半年,由于 Internet 的迅猛发展和环球信息网 WWW 的快速增长,第一个全球信息网络浏览器 Mosaic 诞生了。此时,工业界对适合在网络异构环境下使用的语言有一种非常急迫的需求。James Gosling 决定改变绿色计划的发展方向,对 Oak 进行了小规模改造。就这样,Java 在 1995 年 3 月 23 日诞生了。Java 的诞生标志着互联网时代的开始。Java 能够被应用在全球信息网络平台上编写互动性极强的 Applet 程序。1995 年的 Applet 给人们无穷的视觉和脑力震荡。其实,Java 的诞生颇有一股“有心栽花花不开,无心插柳柳成荫”的味道。就像 Unix 和它的前身 MULTICS 系统一样。

Sun 在 1995 年 5 月 23 日的 SunWorld'95 会议上正式发布 Java 和 HotJava 浏览器。同年,有很多公司先后获得了 Java 许可证。如 Netscape 在 1995 年 8 月,Oracle 在 1995 年 10 月分别获得了 Java 许可证。Sun 在 1995 年发布第一个 Java 版本后,于 1996 年 1 月宣布成立新的业务部门——JavaSoft 部。这个部分主要负责开发、销售并支持基于 Java 技术的产品。

Sun 虽然在 1995 年推出了 Java,但这只是一种语言,要想开发复杂的应用程序,必须要有一个强大的开发库支持才行。Sun 在 1996 年 1 月 23 日发布了 JDK1.0。这个版本包括运行环境(即 JRE)和开发环境(即 JDK)两部分。运行环境包括了核心 API、集成 API,用户界面 API,发布技术,Java 虚拟机(JVM)5 个部分。而开发环境还包括编译 Java 程序的编译器(即 javac)。在 JDK1.0 版本中,除了 AWT(一种用于开发图形用户界面的 API)外,其他的库并不完整。

在推出 JDK1.0 后,紧跟着,Sun 在 1997 年 2 月 18 日发布了 JDK1.1。JDK1.1 相对于 JDK1.0 最大的改进就是为 JVM 增加了 JIT(即时编译)编译器。JIT 和传统的编译器不同。传统的编译器是编译一条,运行完后再将其扔掉,而 JIT 会将经常用到的指令保存在内容中,在下次调用时就不需要再编译。这样,JDK 在效率上有了非常大的提升。

Sun 在推出 JDK1.1 后,接着又推出了数个 JDK1.x 版本。自从 Sun 推出 Java 后,JDK 的下载量不断飙升。在 1997 年,JDK 的下载量突破了 220 000。而在 1998 年,JDK 的下载量已经超过了 2 000 000。

3. Java 高速发展

虽然在 1998 年之前,Java 已被众多的软件企业所采用,但由于当时硬件环境和 JVM 技术的限制,它的应用却很有限。当时 Java 主要使用在前端的 Applet 以及一些移动设备

中。然而,这并不等于 Java 的应用只限于这些领域。1998 年是 Java 开始迅猛发展的一年。在 1998 年 12 月 4 日, Sun 发布了 Java 历史上最重要的一个版本: JDK1.2。这个版本的发布标志着 Java 已经进入 Java2 时代。这个时期也是 Java 飞速发展的时期。在这一年中 Sun 发布了 JSP/Servlet、EJB 规范以及将 Java 分成了 J2EE、J2SE 和 J2ME。这标志着 Java 已经吹响了向企业、桌面和移动 3 个领域进军的号角。

JDK1.2 被分成了 J2EE、J2SE 和 J2ME3 大块,得到了市场的强烈反响。不仅如此, JDK1.2 还对它的 API 分成了 3 大类。

- **核心 API** 由 Sun 公司制定的基本 API,所有的 Java 平台都应该提供,即为平常所说的 Java 核心类库。
- **可选 API** 是 Sun 为 JDK 提供的扩充 API,因平台的不同而不同。
- **特殊 API** 用于满足特殊要求的 API,如用于 JCA 和 JCE 的第三方加密类库。

Java2 除了上述的一些改进外,还增加了很多新的特性。其中最引人注目的当属 Swing 了。Swing 是 Java 的另一个图形库。它不但有各式各样先进的组件,而且连组件风格都可以替换。Swing 出现后,很快就抢了 AWT 的风头。但 Swing 并不是为取代 AWT 而存在的。事实上,Swing 是建立在 AWT 之上的,就像 JFace 是建立在 SWT 之上一样。另外,Java2 还在多线程、集合类和非同步类上做了大量的改进。

从 JDK 1.2 开始, Sun 以平均两年发布一个版本的速度推出新的 JDK。在 2000 年 5 月 8 日, Sun 对 JDK 1.2 进行了重大升级,推出了 JDK 1.3。Sun 在 JDK 1.3 中同样进行了大量的改进,主要表现在一些类库上(如数学运算、新的 Timer API 等)。在 JNDI 接口方面 JDK 1.3 增加了一些 DNS 的支持,增加了 JNI 的支持。这使得 Java 可以访问本地资源、支持 XML 以及使用新的 Hotspot 虚拟机代替了传统的虚拟机。在 JDK 1.3 时代,相应的应用程序服务器也得到了广泛的应用,如第一个稳定版本 Tomcat 3.x 在这一时期得到了广泛的应用, WebLogic 等商业应用服务器也渐渐被接受。

Sun 在 2002 年 2 月 13 日发布了 JDK 历史上最为成熟的版本: JDK 1.4。在进入 21 世纪以后,在 .NET 平台和 Java 平台之间发生了一次声势浩大的论战,Java 的主要问题就是性能。因此, Sun 将主要精力放到 Java 的性能上。在 JDK 1.4 中, Sun 对 Hotspot 虚拟机的锁机制进行了改进,使 JDK 1.4 的性能有了极大地提高。同时由于 Compaq、Fujitsu、SAS、Symbian、IBM 等公司的参与,使 JDK 1.4 成为发展最快的一个 JDK 版本。到 JDK 1.4 为止,已经可以使用 Java 实现大多数的应用了。

虽然从 JDK1.4 开始, Java 的性能有了显著的提高,但 Java 又面临着另一个问题,那就是复杂。虽然 Java 是纯面向对象语言,但它对一些高级的语言特性(如泛型、增强的 for 语句)并不支持。而且,和 Java 相关的技术,如 EJB 2.x,也由于复杂而很少有人使用。也许是 Sun 意识到了这一点。因此,在 2004 年 10 月, Sun 发布了期待已久的版本: JDK 1.5。同时, Sun 将 JDK 1.5 改名为 J2SE 5.0,与 JDK 1.4 不同。JDK 1.4 的主题是性能,而 J2SE 5.0 的主题是易用。Sun 将版本号 1.5 改为 5.0,预示着 J2SE 5.0 较以前的 J2SE 版本有着很大的改过。

4. Java 的今天

Sun 不仅为 J2SE 5.0 增加了诸如泛型、增强的 for 语句、可变数目参数、注释

(Annotations)、自动拆箱 (unboxing) 和装箱等功能。同时,也更新了企业级规范,如通过注释等新特性改善了 EJB 的复杂性,并推出了 EJB 3.0 规范。同时,针对 JSP 的前端界面设计推出了 JSF。JSF 是类似于 ASP.NET 的服务端控件,通过它可以很快地建立起复杂的 JSP 界面。

2006 年底, Sun 推出了 J2SE 6.0 的测试版。2007 年, Sun 公司又推出 J2SE 6.0 正式版。J2SE 6.0 不仅在性能、易用性方面得到了极大的提高,而且还提供了如脚本、全新的 API (Swing 和 AWT 等 API 已经被更新) 的支持。而且, J2SE 6.0 专为 Vista 而设计,它在 Vista 上将会拥有更好的性能。在推出 J2SE 6.0 的同时, J2SE 7.0 项目也已经启动。

本书将会详细介绍 JDK 1.6, 即 J2SE 6.0。

目前,全球 Java 开发人员已经超过 450 万, Java 社区充满活力和创新精神。这都是 Java 下一个 10 年更加繁荣的保障。为了保持 Java 的增长和推进 Java 社区的繁荣, Sun 在 Java One 开发者大会上宣布开放 Java 核心源代码,以鼓励更多的人参与到社团活动中来。这是 Sun 为推进社团发展和维护 Java 技术兼容性而迈出的重要一步。随着 Java 的开源, Java 的未来似乎变得更加明朗。在未来, Java 的应用范围有可能变得更广。

1.2 Java 概述

Java 是一种既面向对象又可跨平台的语言。Java 语言产生于 C++ 语言之后,是完全的面向对象的编程语言。Java 语言充分吸取 C++ 语言的优点,采用程序员所熟悉的 C 和 C++ 语言的许多语法,同时又去掉了 C 语言中指针、内存申请和释放等影响程序健壮性的部分。可以说, Java 语言是站在 C++ 语言这个“巨人的肩膀上”前进的。

1.2.1 Java 介绍

Java 是一个广泛使用的网络编程语言,是一种新的计算概念。首先,作为一种程序设计语言,它简单、面向对象、不依赖于机器的结构、具有可移植性、鲁棒性、安全性、并且提供了并发的机制、具有很高的性能。其次,它最大限度地利用了网络, Java 的小应用程序 (applet) 可在网络上传输而不受 CPU 和环境的限制。另外, Java 还提供了丰富的类库,使程序设计者可以很方便地建立自己的系统。

Java 是一种面向对象程序设计语言。面向对象技术提倡模拟现实世界的思维方式,数据与操作相绑定的程序风格,符合现代大规模软件开发的要求和潮流,正逐步替代面向过程的程序设计技术,成为计算机应用开发领域的主流趋势。同时, Java 也是一种跨平台的语言。在单机时代,程序进程之间靠共享存储进行变量交换。在网络时代,运行在不同宿主机上的程序按网络协议进行无格式的消息 (二进制字符流) 交换,消息的语义由交换程序双方维护。不可否认,现在正处在一个网络异构时代,不同的计算机如果要配合运行一个程序,其编程语言一定是跨平台的。Java 以跨平台、安全性高等特点受到大部分程序员的欢迎。

Java 不同于一般的编译执行计算机语言和解释执行计算机语言。Java 首先将源代码编译成字节码 (bytecode), 然后依赖各种不同平台上的虚拟机来解释执行字节码, 从而实现“一次编译、到处执行”的跨平台特性。不过, 这同时也在一定程度上降低了 Java 程序的运行效率。但在 J2SE 1.4.2 发布后, Java 的执行速度有了大幅提升。

与传统程序不同, Sun 公司在推出 Java 时就将其作为一种开放的技术。全球数以万计的 Java 开发公司被要求所设计的 Java 组件必须相互兼容。“Java 语言靠群体的力量而非公司的力量”是 Sun 公司的口号之一, 并获得了广大软件开发商的认同。这与微软公司所倡导的注重精英和封闭式的模式完全不同。

Java 语言的前身 Oak 在 C++ 的基础上开发, 而 C++ 在 C 的基础上开发。因此, Java 和 C、C++ 具有许多相似之处。Java 继承了 C、C++ 的优点, 增加了一些实用的功能, 使 Java 语言更加精炼。Java 摒弃了 C、C++ 的缺点, 去掉了 C、C++ 的指针运算、结构体定义、手工释放内存等容易引起错误的功能和特征, 增强了安全性, 也使其更容易被接受和学习。

虽然 Java 在 C++ 的基础上开发, 但并不是 C++ 的增强版, 也不是用来取代 C++ 的。Java 与 C++ 既不向上兼容, 也不向下兼容, 两者将长时间共存。Java 在理论和实践上都与 C++ 有着重要的区别。Java 是独立于平台, 面向 Internet 的分布式编程语言。Java 对 Internet 编程的影响如同 C 和 C++ 对系统编程的影响。Java 的出现改变了编程方式, 但 Java 并不是孤立存在的一种语言, 而是计算机语言多年演变的结果。

用 Java 语言可以编写两种程序, 一种是应用程序 (Application), 一种是小应用程序 (Applet)。应用程序可以独立运行, 可以用在网络, 多媒体等开发上。小应用程序不可以独立运行, 是嵌入到 Web 网页中由带有 Java 插件的浏览器解释运行, 主要用在 Internet 上。

Java 至今主要发展出 3 个领域的应用平台: Java 2 Platform, Standard Edition (Java SE)、Java 2 Platform, Enterprise Edition (Java EE) 和 Java 2 Platform, Micro Edition (Java ME)。

□ Java SE 平台

该平台是各应用平台的基础。如果要学习其他平台, 必须先了解 Java SE。Java SE 正是本书主要介绍对象。Java SE 可以分为 4 个主要部分: JVM、JRE、JDK 和 Java 语言。为了能运行 Java 程序, 平台上必须安装有 Java 虚拟机 (Java Virtual Machine, JVM)。JVM 包含在 Java 运行环境 (Java SE Runtime Environment, JRE) 中, 所以, 为了要运行 Java 程序, 必须取得 JRE 并进行安装。如果要开发 Java 程序, 则必须取得 JDK (Java SE Development Kits) 并进行安装。JDK 包括了 JRE 以及开发过程中所需要的一些工具程序, 如 javac、java、appletviewer 等。

Java 语言只是 Java SE 的一部分。除了语言之外, Java 最重要的就是提供了庞大且功能强大的 API 类库, 如字符串处理、数据输入输出、网络组件、用户窗口接口等功能。可以使用这些 API 作为基础进行程序开发, 而无须重复开发功能相同的组件。

Java SE 的组成如图 1-1 所示。

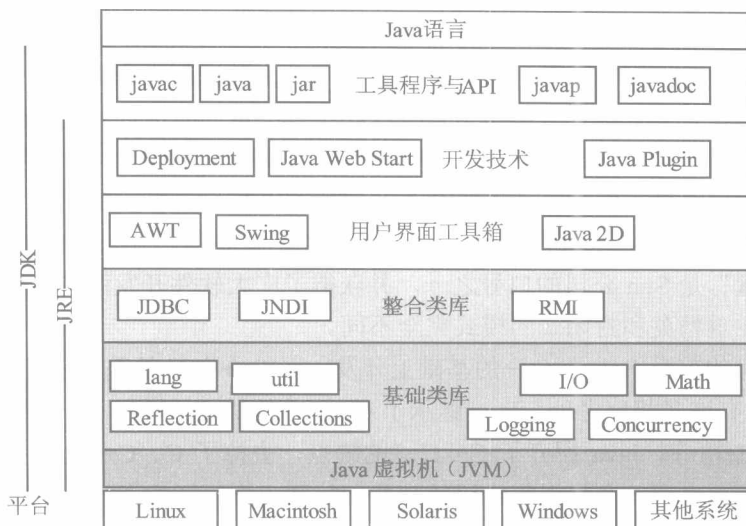


图 1-1 Java SE 的组成概念图

□ Java EE 平台

Java EE 以 Java SE 为基础，定义了一系列的服务、API、协议等，适用于开发分布式、多层式（Multitiered）、以组件为基础、以 Web 为基础的应用程序。整个 Java EE 的体系是相当庞大的，比较为人熟悉的技术有 JSP、Servlet、Enterprise JavaBeans（EJB）、Java Remote Method Invocation（RMI）等。

□ Java ME 平台

Java ME 是 Java 平台版本中最小的一个，目的是作为小型数字设备上开发及部署应用程序的平台，如消费型电子产品或嵌入式系统等。最为人熟悉的设备有手机、PDA、股票机等。

提示

Java SE 指平台名称，全名为 Java Platform, Standard Edition 6，JDK 6 是基于平台的程序开发工具集发行版本，全名为 Java SE Development Kit 6。JRE 6 则是基于平台的执行环境发行版本，全名为 Java SE Runtime Environment 6。

1.2.2 Java 语言特点

Sun 的“Java 白皮书”对 Java 做了如下定义。“Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high-performance, multi-threaded, and dynamic language”。Java 是一种简单的、面向对象的、分布式的、解释执行的、健壮的、安全的、结构中立的、可移植的、高效率的、多线程的和动态的语言。Sun 公司对 Java 的定义充分展示了 Java 的如下几个特点。

1. 简单

Java 是一种简单的语言。Java 在 C、C++ 的基础上开发，继承了 C 和 C++ 的许多特性，同时也取消 C 和 C++ 语言中繁琐的、难以理解的、不安全的内容，如指针、多重继承等。JDK 还提供了丰富的基础类库，具有 C 或 C++ 编程经验的程序员都会对这些基础类库很熟悉，无需经过长时间训练即可掌握它。

2. 面向对象

Java 是一种纯面向对象的语言。Java 程序面向对象的设计思路不同于 C 语言基于过程的程序设计思路。面向对象程序设计，具备更好地模拟现实世界环境的能力和可重用性。它将待解决的实际问题转换成一组分离的程序对象，这些对象彼此之间可以进行交互。一个对象包含了对应实体应有的信息以及访问和改变这些信息的方法。通过这种设计方式，所设计出来的程序更易于改进、扩展、维护和重用。Java 语言提供类、接口和继承等原语。Java 语言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。

3. 分布式

Java 是一种分布式的语言。传统的基于 C/S（客户端/服务器）架构的程序，均采用客户端向服务器提出服务请求，服务器再根据要求执行适当的程序并将结果返回的方式，所以，服务器负荷较重。Java 采用 Java 虚拟机架构，可将许多工作直接交由终端处理，因此，数据可以被分布式处理。此外，Java 类库的运用，大大减轻了网络传输的负荷。Java 类库包含了支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。Java 应用程序可凭借 URL 打开并访问网络上的对象，其访问方式与访问本地文件系统几乎完全相同。分布式网络环境是 Java 进一步发展的地方。

4. 高效解释执行

Java 是高效解释执行的语言。高级语言程序必须转换为机器语言程序才能在计算机上执行。但是，不同的计算机系统所使用的机器语言不同。为了实现“一次编译，随处运行”的目标，Java 程序在编译时并不直接编译成特定的机器语言程序，而是编译成与系统无关的“字节码（bytecode）”，由 Java 虚拟机（Java Virtual Machine, JVM）来执行。JVM 使得 Java 程序可以“一次编译，随处运行”。任何系统只有安装了 Java 虚拟机后，才可以执行 Java 程序。

JVM 能直接在任何机器上执行，为字节码提供运行环境。当 JVM 解释执行 Java 程序时，Java 实时编译器（Just-In-Time, JIT）会将字节码译成目标平台对应的机器语言的指令代码。

早先的许多尝试解决跨平台问题的方案对性能要求都很高。其他解释执行的语言系统，如 BASIC、TCL、PERL 都有无法克服的性能缺陷。但是，Java 却可以在非常低档的 CPU 上顺畅运行。这是因为 JVM 能够直接使用 JIT 编译技术将经过精心设计的字节

码转换成高性能的本机代码。事实上,随着 JIT 编译器技术的发展,Java 程序的运行速度已接近于 C++。因而,“高效且跨平台”对 Java 来说已不再矛盾。

5. 健壮性

Java 是健壮的语言。为了更好地理解 Java 的健壮性,先讨论一下传统编程环境下程序设计失败的主要原因:内存管理错误和误操作引起的异常或运行时异常。

在传统的编程环境下,内存管理是一项困难、乏味的工作。例如,在 C 或 C++中,必须手工分配、释放所有的动态内存。如果忘记释放原来分配的内存,或是释放了其他程序正在使用的内存时,系统就会出错。同时,在传统的编程环境下,异常情况可能经常由“被零除”、“Null 指针操作”、“文件未找到”等原因引起,必须用既繁琐又难理解的一大堆指令来进行处理。

Java 通过自行管理内存分配和释放的方法,从根本上消除了有关内存的问题。Java 提供垃圾收集器,可自动收集闲置对象占用的内存。Java 提供面向对象的异常处理机制来解决异常处理的问题。通过类型检查、Null 指针检测、数组边界检测等方法,在程序开发早期就发现程序的错误。

6. 安全

Java 是安全的网络编程语言。因为 Java 常被用于网络环境中,为此,Java 提供了一系列的安全机制以防恶意代码攻击,确保系统安全。Java 的安全机制分为多级,包括 Java 语言本身的安全性设计以及严格的编译检查、运行检查和网络接口级的安全检查。

Java 语言是强类型语言,每种类型都要求严格定义。首先,每个变量、表达式都有类型。其次,所有的数值传递,不管是直接的、还是通过方法调用经由参数传递,都要进行类型相容性检查。有些语言没有自动进行数据类型相容性检查,或对冲突的类型进行转换的机制。Java 编译器对所有的表达式和参数都要进行类型相容性的检查,以确保类型是兼容的。任何类型的不匹配都是错误的,在编译完成之前,错误必须全部被纠正。此外,Java 摒弃指针类型和数据类型的隐式转换,对内存访问进行了严格的限制。Java 编译器在编辑期间并不分配内存,而是推迟到运行时由解释器决定。这样,编程人员就无法通过指针来非法访问内存。在运行期间,Java 的运行环境提供了 4 级安全保障机制:字节码校验器、类装载器、运行时内存布局 and 文件访问限制。在网络接口级,用户可按自己的需要来设置网络访问权限。

此外,Java 的未来版本将采用公开密钥法以及其他加密技术,来核实从网络上传输过来的代码的源主机及该代码的完整性。

7. 结构中立

Java 是结构中立的语言。Java 的设计目标是要支持网络应用。一般而言,网络是由许多不同的系统构成,包括各种不同的 CPU 与操作系统。为了让 Java 应用程序能够在网络上任何地方执行,其编译器会产生一种具备结构中立性的对象文件格式,即 Java 字节码文件。Java 字节码可在任何安装了 Java 虚拟机的平台上运行。