

清华
电脑学堂

超值多媒体光盘
完整实例源代码
12段教学视频文件

- ✓ 合理的知识结构
- ✓ 真实的案例教学
- ✓ 阶梯式实践环节
- ✓ 理论实践一体化

Java 程序开发

基础教程与实验指导

■ 王占中 崔志刚 杨记超 李乃文 等编著

清华大学出版社

清华
电脑学堂



Java 程序设计

基础教程与实验指导

■ 王占中 崔志刚 杨记超 李乃文 等编著

清华大学出版社
北 京

内 容 简 介

本书由浅入深地讲述 Java 编程知识。全书共分 14 章,介绍 Java 面向对象编程的基础知识和技能,从基本的 Java 组成元素、数据类型和控制结构,逐步过渡到方法、类、对象及面向对象程序设计的高级特征;深入学习 Java 事件、Java Swing 图形界面设计、Java 输入/输出和数据库等高级应用;最后,通过旺旺服饰管理系统的综合应用,将对前面学习的知识进行巩固和提高。本书附有配套光盘,提供了书中实例的源代码和视频教学文件。

本书体现了作者在软件技术教学改革过程中形成的“项目驱动、案例教学、理论实践一体化”教学方法,可以作为 Java 职业培训教材和各级院校 Java 授课培训教程,也可以作为 Java 自学资料和参考资料。

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 程序开发基础教程与实验指导 / 王占中等编著. —北京:清华大学出版社, 2012.3
(清华电脑学堂)

ISBN 978-7-302-26895-6

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

中国版本图书馆 CIP 数据核字(2011)第 193225 号

责任编辑:夏兆彦

封面设计:柳晓春

责任校对:徐俊伟

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:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:30.75 字 数:768 千字

附光盘 1 张

版 次:2012 年 3 月第 1 版

印 次:2012 年 3 月第 1 次印刷

印 数:1~4000

定 价:45.00 元

产品编号:042199-01

Java 语言是一种简单的、面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异、多线程的动态编程语言。Java 还提供了丰富的类库，使开发人员可以方便地构建系统。目前常用的 Java 平台基于 Java 1.6，最近即将发布的版本为 Java 1.7。

Java 语言已经广泛应用于生活中的各个领域，无论是网络编程、Web 编程还是数据库编程，甚至是嵌入式开发都有 Java 语言的身影。

另外，在开发 Java 系统过程中，Eclipse 开发工具是众多 Java 开发工具之一。Eclipse 是一个流行的针对 Java 编程的集成开发环境（IDE）。它还可以用作编写其他语言（比如 C++ 和 Ruby）的环境，合并各种种类工具的框架，以及创建桌面或服务器应用程序的富客户端平台。

1. 本书定位与特色

□ 面向职业技术教学

本书是作者在总结了多年开发经验与成果的基础上编写的，以实际项目为中心，全面、翔实地介绍了 Java 开发所需的各种知识和技能。通过本书的学习，读者可以快速、全面地掌握使用 Java 进行面向对象开发的方法。本书体现了作者“项目驱动、案例教学、理论实践一体化”教学理念，是一本真正面向职业技术教学的教材。

□ 合理的知识结构

本书面向程序员职业培训市场，结合程序开发实践介绍 Java 编程知识，突出职业实用性。本书各章都有实例分析，带领读者经历程序开发全过程，是一本真正的实训性案例教程。

□ 真实的案例教学

针对每个知识点，本书设计了针对性强的教学案例，这些小案例既相对独立，又具有一定的联系，是综合性开发实例的组成部分。读者在制作这些小案例的过程中可以掌握每个知识点。

□ 理论实践一体化

在每个案例中，本书都有机融合了知识点讲解和技能训练目标，融“教、学、练”于一体。每个案例的讲解都首先提出功能目标，然后是实例制作演示和学生模仿练习，让读者掌握案例的完成过程，体现“在练中学，学以致用”的教学理念。

□ 阶梯式实践环节

本书精心设置了 3 个教学环节：实验指导、扩展练习和综合实训，让学生通过不断练习实践，实现编程技能的逐步推进，最终职业能力的接轨。

2. 本书主要内容

本书主要针对初学者,由浅入深地学习 Java 语言。全书共为 14 章,首先带领读者了解 Java,认识 Eclipse 开发工具。

然后,从基本的 Java 组成元素、数据类型和控制结构,逐步过渡到方法、类、对象及面向对象程序设计的高级特征。

其次,深入学习程序的异常处理、Java Swing、Java 事件、Java 绘图与音频播放、Java I/O(输入/输出)和 Java 与数据库等。

最后,通过旺旺服饰管理系统,来全面介绍 Java 与项目开发的相关内容,并通过该章巩固及提高开发能力。

本书附有配套光盘。光盘提供了书中实例的源代码,全部经过精心调试,在 Windows XP/Windows 2000/Windows 2003 Server 下全部通过,能够保证正常运行。

3. 读者对象

本书体现了作者在软件技术教学改革过程中形成的“项目驱动、案例教学、理论实践一体化”教学方法,读者通过本书可以快速、全面地掌握使用 Java 应用于 Internet 编程的开发经验和技能。本书可以作为 Java 职业培训教材和各级院校 Java 授课培训教程,也适合作为 Java 自学资料和参考资料。

除了封面署名人员之外,参与本书编写的人员还有孙岩、马海军、张仕禹、夏小军、赵振江、李振山、李文采、吴越胜、李海庆、何永国、李海峰、陶丽、吴俊海、安征、张巍屹、崔群法、王咏梅、康显丽、辛爱军、牛小平、贾栓稳、王立新、苏静、赵元庆、郭磊、徐铭、李大庆、王蕾、张勇、郝安林等。由于编者水平有限,在编写过程中难免会有漏洞,欢迎读者通过清华大学出版社网站 www.tup.tsinghua.edu.cn 与编者联系,帮助编者改正提高。

第 1 章 了解 Java	1
1.1 Java 简介	1
1.1.1 Java 语言的发展	1
1.1.2 Java 平台	2
1.1.3 Java 的特点	4
1.2 深入熟悉 Java	7
1.2.1 Java 运行流程	7
1.2.2 JVM 介绍	8
1.2.3 Java 中 Application 与 Applet	9
1.3 Java API 介绍	10
1.4 什么是 JDK 7	10
1.4.1 JDK 7 介绍	10
1.4.2 编译和解释命令	11
1.4.3 生成帮助命令 javadoc	13
1.4.4 压缩命令 jar	16
1.5 扩展练习	18
第 2 章 熟悉 Eclipse 开发工具	21
2.1 Eclipse 概述	21
2.1.1 Eclipse 简介	21
2.1.2 Eclipse 工作台	23
2.1.3 Eclipse 视图	28
2.1.4 【编辑器】视图	30
2.2 使用 Eclipse	35
2.2.1 创建 Java 项目	35
2.2.2 创建 Java 包与类文件	37
2.2.3 编写及运行 Java 代码	39
2.2.4 添加 Java 注释	41
2.3 自定义开发环境	42
2.3.1 设置工作空间	42
2.3.2 Java 代码样式	43
2.3.3 设置颜色和字体	46
2.4 Eclipse 调试程序	48

2.4.1 调试器	48
2.4.2 添加断点	48
2.4.3 调试的变量值	52
2.4.4 【调试】视图	52
2.5 扩展练习	53
第 3 章 Java 语言基础	56
3.1 基本数据类型	56
3.1.1 整数类型	56
3.1.2 浮点类型	58
3.1.3 字符类型	59
3.1.4 布尔类型	61
3.2 变量与常量	61
3.2.1 标识符和关键字	61
3.2.2 声明变量	62
3.2.3 声明常量	64
3.2.4 变量的有效范围	65
3.3 运算符	66
3.3.1 赋值运算符	66
3.3.2 算术运算符	66
3.3.3 比较与逻辑运算符	70
3.3.4 位运算符	71
3.3.5 三元运算符	72
3.3.6 运算符优先级	73
3.4 数据类型转换	74
3.4.1 隐式类型转换	74
3.4.2 显式类型转换	75
3.5 扩展练习	75
第 4 章 程序控制与数组	78
4.1 流程控制语句	78
4.1.1 if 条件语句	78
4.1.2 switch 多分支语句	82
4.1.3 for 循环语句	85
4.1.4 while 循环语句	86

4.1.5 do...while 循环语句.....	87	5.5.3 常规类型格式化.....	130
4.1.6 跳转语句.....	89	5.6 使用正则表达式.....	131
4.2 数组概述.....	91	5.7 扩展练习.....	133
4.2.1 一维数组.....	92	第 6 章 了解面向对象与类.....	137
4.2.2 二维数组.....	93	6.1 了解面向对象.....	137
4.3 数组的基本操作.....	95	6.1.1 什么是面向对象.....	137
4.3.1 遍历数组.....	95	6.1.2 面向对象的基本特征.....	139
4.3.2 填充替换数组元素.....	96	6.1.3 消息、接口和方法.....	141
4.3.3 对数组进行排序.....	97	6.2 类.....	142
4.3.4 复制数组.....	98	6.2.1 类的定义.....	142
4.3.5 数组查询.....	99	6.2.2 类的属性和方法.....	143
4.4 数组排序算法.....	100	6.2.3 访问类.....	144
4.4.1 冒泡排序.....	100	6.2.4 类的构造方法.....	146
4.4.2 直接选择排序.....	102	6.2.5 包装类.....	147
4.4.3 插入排序.....	102	6.3 对象.....	157
4.5 扩展练习.....	104	6.3.1 创建对象.....	157
第 5 章 字符串.....	107	6.3.2 访问对象的属性和行为.....	157
5.1 String 类概述.....	107	6.3.3 引用对象.....	159
5.1.1 字符集与字符序列.....	107	6.3.4 销毁对象.....	160
5.1.2 String 类的基本操作.....	108	6.4 基本方法.....	161
5.2 字符串检索.....	109	6.4.1 this 关键字.....	161
5.2.1 获取字符串信息.....	109	6.4.2 super 关键字.....	162
5.2.2 获取子字符串.....	112	6.4.3 方法重载.....	163
5.2.3 判断字符串的开始与 结尾.....	113	6.5 访问控制修饰.....	165
5.2.4 按字典顺序比较 两个字符串.....	115	6.5.1 访问控制修饰符.....	165
5.3 字符串其他操作.....	116	6.5.2 static 修饰符.....	169
5.3.1 去除空格.....	116	6.5.3 final 修饰符.....	171
5.3.2 字符串替换.....	117	6.6 扩展练习.....	172
5.3.3 判断字符串是否相等.....	118	第 7 章 Java 核心技术.....	177
5.3.4 字母大小写转换.....	119	7.1 继承.....	177
5.3.5 字符串分割.....	120	7.1.1 类的继承.....	177
5.4 可改变字符串类.....	121	7.1.2 Object 类.....	180
5.4.1 创建 StringBuffer.....	121	7.2 接口.....	181
5.4.2 StringBuffer 应用.....	122	7.2.1 接口的声明.....	181
5.5 格式化字符串.....	125	7.2.2 接口的特征.....	182
5.5.1 字符串生成器.....	125	7.2.3 接口的实现.....	183
5.5.2 日期和时间字符串 格式化.....	127	7.3 Java 程序包.....	184
		7.3.1 包的导入与访问.....	184
		7.3.2 管理程序包.....	185

7.4 对象类型的转换.....	186	9.1.2 Swing 与 JFC 关系.....	232
7.4.1 对象类型转换.....	186	9.1.3 Swing 的类层次结构.....	232
7.4.2 强制对象类型转换.....	188	9.2 窗口与面板.....	234
7.4.3 instanceof 判断对象类型.....	189	9.2.1 容器概述.....	234
7.5 内部类.....	190	9.2.2 JFrame 主窗口.....	235
7.5.1 内部类概述.....	190	9.2.3 JDialog 对话框窗口.....	237
7.5.2 实例内部类.....	192	9.2.4 JPanel 面板.....	240
7.5.3 局部内部类.....	194	9.3 Swing 常用组件.....	242
7.5.4 匿名类.....	196	9.3.1 标签组件.....	242
7.5.5 静态内部类.....	197	9.3.2 按钮类组件.....	245
7.6 扩展练习.....	198	9.3.3 文本组件.....	250
第 8 章 异常处理机制	204	9.3.4 JSlider 滑块.....	257
8.1 异常概述.....	204	9.3.5 JMenu 菜单.....	259
8.1.1 什么是异常.....	204	9.3.6 JToolBar 工具栏.....	263
8.1.2 Java 异常处理结构.....	205	9.4 常用布局管理器.....	264
8.2 Java 异常处理机制.....	206	9.4.1 绝对布局.....	265
8.2.1 try...catch 语句捕获异常.....	207	9.4.2 流动布局.....	265
8.2.2 try...catch 语句嵌套.....	209	9.4.3 边界布局.....	267
8.2.3 多个 catch 语句.....	210	9.4.4 网格布局.....	267
8.2.4 try...catch...finally 语句.....	213	9.5 扩展练习.....	268
8.3 自定义异常.....	216	第 10 章 Java 事件	271
8.4 在方法中抛出异常.....	217	10.1 Java 事件处理机制.....	271
8.4.1 使用 throws 关键字抛出异常.....	217	10.1.1 事件处理模型.....	271
8.4.2 使用 throw 关键字抛出异常.....	218	10.1.2 按钮单击事件.....	273
8.5 异常的使用原则.....	219	10.2 事件类和事件监听器.....	275
8.5.1 异常只能用于非正常情况.....	219	10.3 常见事件处理.....	279
8.5.2 尽可能地避免异常.....	221	10.3.1 鼠标事件.....	279
8.5.3 保持异常的原子性.....	222	10.3.2 适配器实现鼠标事件.....	281
8.5.4 catch 子句中指定具体的异常类型.....	223	10.3.3 焦点事件.....	283
8.5.5 不要在 catch 代码块中忽略被捕获的异常.....	224	10.3.4 选项事件.....	285
8.6 扩展练习.....	225	10.3.5 窗口事件.....	287
第 9 章 Java Swing	230	10.3.6 单击事件.....	290
9.1 Java Swing 概述.....	230	10.3.7 键盘事件.....	293
9.1.1 什么是 Swing.....	230	10.3.8 弹出菜单事件.....	296
		10.3.9 树组件事件.....	298
		10.4 扩展练习.....	300
		第 11 章 Java 绘图与音频播放	307
		11.1 图形处理技术.....	307
		11.1.1 Graphics2D 图形类.....	307

11.1.2	Shape 图形绘制对象	308	13.2.1	详细了解 JDBC	372
11.2	绘制图形操作	309	13.2.2	JDBC 驱动程序的类型	373
11.2.1	移动原点以及图形旋转	310	13.2.3	JDBC 中常用的类和接口	375
11.2.2	图形填充	314	13.3	数据库连接及操作	379
11.2.3	颜色渐变	316	13.3.1	连接不同的数据库	380
11.2.4	绘制图片	319	13.3.2	MySQL 数据库的连接步骤	381
11.3	文字与笔画设置	325	13.3.3	获取数据	383
11.3.1	设置笔画属性	325	13.4	在 Swing 中操作数据	384
11.3.2	文字处理	327	13.4.1	通过 JTable 显示数据	384
11.4	声音处理技术	329	13.4.2	通过界面修改数据	387
11.5	扩展练习	333	13.5	扩展练习	389
第 12 章	输入输出流	341	第 14 章	旺旺服饰管理系统	392
12.1	管理文件或目录	341	14.1	系统概述	392
12.1.1	创建 File 对象	341	14.1.1	需求分析	392
12.1.2	使用 File 类的方法	343	14.1.2	系统流程图	393
12.1.3	目录的操作方法	345	14.1.3	系统设计	394
12.2	字节输入输出流	346	14.2	数据库设计	395
12.2.1	Java 流式 I/O 的基本概念	346	14.2.1	数据表	395
12.2.2	InputStream 和 OutputStream	347	14.2.2	视图	399
12.2.3	FileInputStream 和 FileOutputStream	350	14.2.3	存储过程	400
12.2.4	标准输入输出流	353	14.2.4	系统开发前的准备工作	403
12.3	字符的输入输出流	355	14.3	基础类	404
12.3.1	Reader 和 Writer	355	14.3.1	数据库连接类	405
12.3.2	FileReader 和 FileWriter	358	14.3.2	构造实例类	406
12.4	RandomAccessFile 类	359	14.3.3	数据库操作类	408
12.4.1	创建 RandomAccessFile 对象	359	14.4	登录	408
12.4.2	RandomAccessFile 对象的读写	360	14.4.1	登录界面的背景界面	408
12.5	扩展练习	361	14.4.2	登录界面	409
第 13 章	数据库操作	369	14.5	主界面	413
13.1	数据库基础知识	369	14.5.1	主页面中的组件与布局	414
13.1.1	什么是数据库	369	14.5.2	创建主界面的导航标签面板	415
13.1.2	数据库管理系统的类型及功能	370	14.6	基础信息管理	420
13.2	JDBC 概述	372	14.6.1	服饰信息管理	420
			14.6.2	服饰查询	421
			附录	单元练习	426
			第 1 单元		426

练习 1-1 输入图形	426
练习 1-2 反编译程序文件	427
第 2 单元	429
练习 2-1 导入外部项目	429
练习 2-2 使用 Eclipse 生成 Javadoc 文件	430
第 3 单元	432
练习 3-1 制作青少年健康表	432
练习 3-2 数学计算题	433
第 4 单元	436
练习 4-1 九九乘法表	436
练习 4-2 对数值进行排序	437
第 5 单元	440
练习 5-1 对字符串加密解密	440
练习 5-2 产生随机验证码	442
第 6 单元	444
练习 6-1 实现交通工具类	444
练习 6-2 制作日历	446
第 7 单元	447
练习 7-1 实现员工工资类	447

练习 7-2 实现树类	450
第 8 单元	453
练习 8-1 实现验证码类	453
练习 8-2 实现计算类	456
第 9 单元	458
练习 9-1 制作树型目录	458
练习 9-2 选项卡面板	460
第 10 单元	462
练习 10-1 彩色列表框	462
练习 10-2 自定义光标	464
第 11 单元	466
练习 11-1 图像的锐化与模糊	466
练习 11-2 应用图像滤镜	470
第 12 单元	471
练习 12-1 列出目录下文件	471
练习 12-2 显示目录文件信息	472
第 13 单元	474
练习 13-1 删除指定学号的 学生信息	474
练习 13-2 用户登录界面	477

第1章 了解 Java

Java 是由 Sun 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台的总称。

Java 语言是当今非常流行的一种面向对象编程语言，具有跨平台、可移植、分布式、简单、可扩展等诸多特性。Java 发展至今不仅代表一门程序语言，更多的是一种软件开发平台，目前已经演化出了 Java SE、Java EE 和 Java ME 等 3 个版本。

Java 凭借其易学易用，以及功能强大等特点，深受开发人员所青睐。它不仅可以进行桌面应用、Web 应用、分布式系统及嵌入式系统等应用程序开发，并且应用领域也较为广泛。

本章学习目标

- 了解 Java 语言
- 了解 Java 软件平台
- Java 运行流程
- 介绍 JVM
- 介绍 Java API
- 了解 JDK 7

1.1 Java 简介

初识 Java 的用户，可能感觉蒙着一层浓浓的面纱，甚至只知道 Java 是一种编程语言，而对其他一无所知。那么，下面将揭开这一层面纱，并深入地了解 Java 语言及平台内容等。

1.1.1 Java 语言的发展

1991 年，Sun 公司的 James Gosling、Bill Joe 等人，为一些消费性电子产品而设计的一个名为 Oak（橡树的意思）软件的通用环境。他们最初的目的只是为了开发一种独立于平台的软件技术，而在网络出现之前，没有引起人们的注意。

直到 1994 年，随着互联网和万维网（World Wide Web，WWW。）的飞速发展，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的支持，Hot Java 浏览器得以研发和发展。因为促销和法律的原因，1995 年 Oak 更名为 Java。从此，Java 被广泛接受并推动了 Web 的迅速发展，常用的浏览器现在均支持 Java Applet。

Sun 公司虽然推出了 Java，但这只是一种语言，若要开发出复杂的应用程序，必须要有一个的强大的开发库支持才行。因此，Sun 公司在 1996 年发布了 JDK1.0。这个版本包括了两部分：运行环境（即 JRE）和开发环境（即 JDK）。运行环境包括了核心 API、集成 API、用户界面 API、发布技术和 Java 虚拟机（JVM）5 个部分。

随后,计算机产业的各大公司,如 IBM、Apple、DEC、Adobe、HP、Oracle 和 Microsoft 等,相继从 Sun 公司购买了 Java 技术许可证,开发相应的产品。

1998 年, Sun 公司发布了 JDK 1.2,也是 Java 的历史上最重要的一个 JDK 版本。这个版本标志着 Java 已经进入 Java2 时代。

Java2 在语言体系上,分成了 J2EE、J2SE 和 J2ME 3 大块,还增加了很多新的特性。其中, Swing 类(图形库)最吸引眼球,另外还在多线程、集合类和非同步类上做了大量的改进。除此之外,Java2 不仅兼容智能卡和小型消费类设备,还兼容大型服务器系统等等。

Java2 对 Java 进行了技术革命性的变化,并且这些变化一直沿用到现在,对 Java 的发展影响深远。

2000 年, Sun 公司推出 JDK 1.3 版本,主要表现在一些类库上(如数学运算、新的 Timer API 等)、JNDI (Java Naming and Directory Interface, Java 命名和目录接口)接口方面增加了一些 DNS、JNI (Java Native Interface, Java 本地调用)的支持,使 Java 可以访问本地资源、支持 XML,以及使用了新的 Hotspot 虚拟机。

2002 年, Sun 公司发布了 JDK 1.4 版本, JDK 历史上最为成熟的版本。在 JDK 1.4 版本中, Sun 公司针对 Hotspot 虚拟机的锁机制进行了改进,使 JDK 1.4 的性能有了质的飞跃。

2004 年, Sun 公司发布了 JDK 1.5 版本,并改名为 J2SE 5.0。该版本更注重其易用性,增加了诸如泛型、增强 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 (Enterprise Java Bean, 译服务器端组件模型)的复杂性,并推出了 EJB 3.0 规范。同时又针对 JSP 的前端界面设计而推出了 JSF (Java Server Faces)。

2005 年, JavaOne 大会召开, Sun 公司公开 Java SE 6,也就是 JDK 1.6 版本,也称为 Java 6.0 (或者 JDK 6.0)。此时, Java 的各种版本被更名,并取消其中的数字 2,如 J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。2006 年, Sun 公司发布 JRE 6.0

2009 年, Oracle (甲骨文)宣布收购 Sun 公司。Oracle 公司预计 2011 年发布 JDK 7.0 (无 Lambda, Jigsaw 和部分 Coin), 2012 年下半年发布 JDK 8.0 (包括 Lambda, Jigsaw 和 Coin 剩下的部分)。

1.1.2 Java 平台

Java 平台由 Java 虚拟机 (Java Virtual Machine) 和 Java 应用编程接口 (Application Programming Interface, API) 构成。Java 应用编程接口为 Java 应用提供了一个独立于操作系统的标准接口,可分为基本部分和扩展部分,其结构如图 1-1 所示。

在硬件或操作系统中,安装一个 Java 平台之后,其 Java 应用程序就可运行,这样 Java 程序可以只编译一次,就可以在各种系统中运行。目前,常用的 Java 平台基于 JDK 1.5 版本,可分为如下 3 个平台。

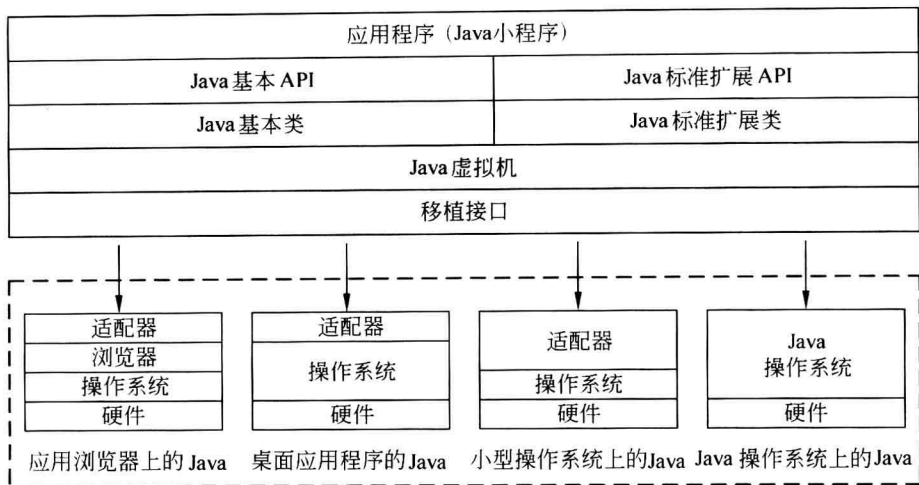


图 1-1 Java 平台结构

1. Java SE 平台

Java SE (Java Platform, Standard Edition, Java 平台标准版) 允许开发和部署在桌面、服务器、嵌入式环境和实时环境中使用的 Java 应用程序。同时, 为 Java EE 提供了 Java Web 服务开发的类, 也是 Java EE 的基础。

Java SE 可分为 JVM (虚拟机)、JRE (运行环境)、JDK 和 Java 语言 4 个主要部分, 其中 JVM 和 JDK 都包含在 Java 的运行环境 (JRE) 中。而 Java 语言只是 Java SE 的一部分, 除此之外, Java 还提供了庞大的 API 类库, 如数字类处理、字符串处理、网络组件、用户窗口等等。Java SE 的组成结构, 如图 1-2 所示。

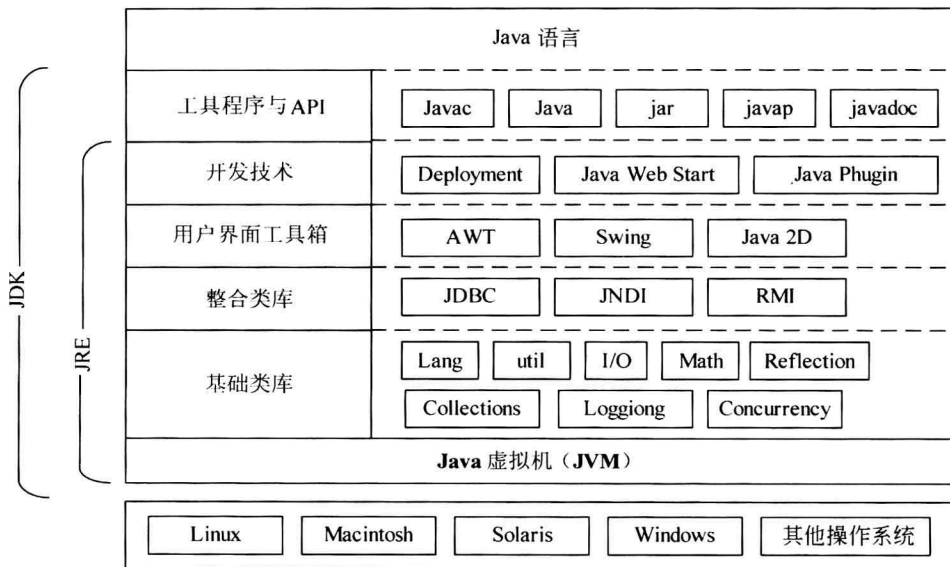


图 1-2 Java SE 组成结构图

2. Java EE 平台

Java EE (Java Platform, Enterprise Edition, Java 平台企业版), 帮助开发和部署可移植、健壮、可伸缩且安全的服务器端 Java 应用程序。Java EE 是在 Java SE 的基础上构建的, 它提供对 EJB (Enterprise JavaBeans)、Java Servlets API、JSP (Java Server Pages) 以及 XML 技术的全面支持。Java EE 可以用来实现企业级的 SOA (service-oriented architecture, 面向服务体系结构) 和 Web 2.0 应用程序, 图 1-3 所示为 Java EE 5.0 系统架构。

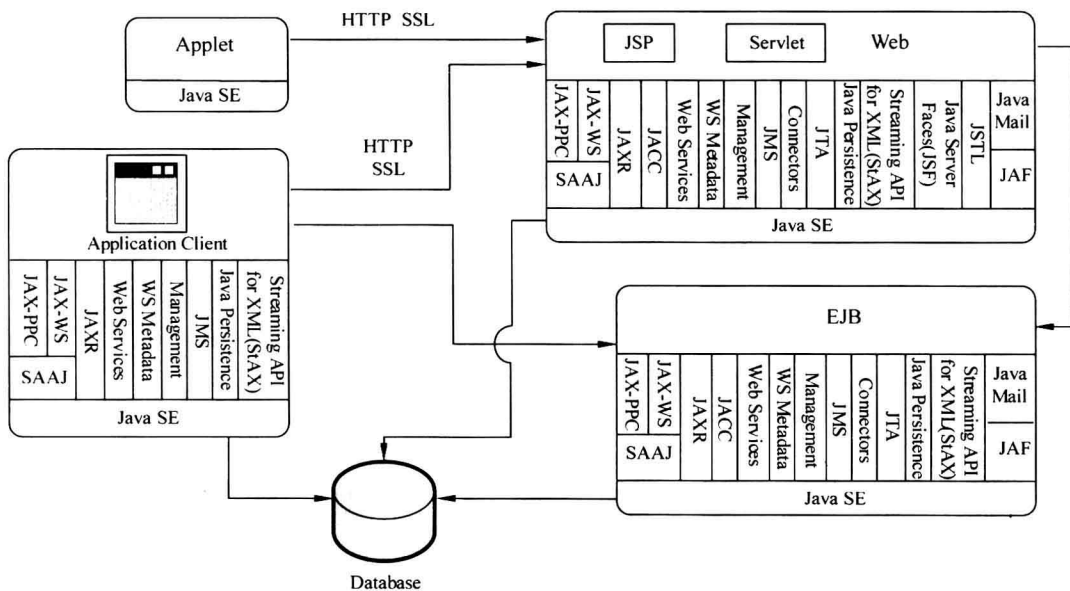


图 1-3 Java EE 5.0 系统架构

3. Java ME 平台

Java ME (Java Platform, Micro Edition, Java 平台微型版) 一般为移动设备和嵌入式设备提供一个健壮且灵活的应用程序运行环境, 如手机、PDA、电视机顶盒和打印机等等。

Java ME 包括灵活的用户界面、安全模型、许多内置的网络协议以及对可以动态下载的联网和离线应用程序的丰富支持。基于 Java ME 规范的应用程序只需编写一次, 就可以用于许多设备, 而且可以利用每个设备的本机功能。

1.1.3 Java 的特点

Java 是 Sun 公司推出的新一代面向对象程序设计语言, 由于具有移植性、安全性、分布式等诸多特点, 其应用范围比较广泛。并且, Java 特别适合于 Internet 应用程序开发, 下面将介绍 Java 的特点。

1. 简单性

从某种意义上讲, Java 语言是 C 及 C++ 语言的一个变种, 非常类似于 C++ 风格。因此, C++ 程序员可以很快就掌握 Java 编程技术。

另外, Java 语言对 C++ 语言进行了简化和提高, 如摒弃了 C++ 中容易引发程序错误的地方, 如指针和内存管理。Java 还增加了自动内存收集功能, 虽然让系统变得稍微复杂了一些, 但简化了 Java 程序的工作。

Java 为协助开发独运行在小型机器上的软件, 提供了丰富的类库。Java 的基本解释器和对象类支持约占 40KB 空间, 若增加基本的标准程序库和线程支持则需增加约 175KB。

2. 面向对象性

Java 吸取了 C++ 面向对象的概念, 将数据封装于类中, 实现了程序的简洁性和便于维护性。面向对象编程提供了很大的灵活性、模块化和可以重复使用性。

Java 语言的程序设计集中于对象及接口上, 并提供了简单的类机制以及动态的接口模型。Java 类的继承关系是单一的非多重的, 一个子类只有一个父类, 子类的父类又有一个父类。但是, 通过继承机制, 子类可以使用父类所提供的方法, 实现程序的复用性。

3. 分布式

若要想与多个用户或客户机共享一个中央资源(如一个数据库)时, 就会使用分布式计算。分布式计算用来利用多个系统的组合计算能力, 比在单个系统上更有效或更快地解决问题。

Java 包括一个支持 HTTP 和 FTP 等基于 TCP/IP 协议的子库。Java 可以通过 RMI 和 JNDI 等技术, 是建立分布式 Java 应用程序的方便途径。任意两台计算机之间的通讯完全由 RMI 负责, 调用远程计算机上的对象就像调用本地对象一样方便。

4. 健壮性

通过 Java 编写的程序具有多方面的可靠性, 如程序前期的问题检测、后期动态的(运行时)检测, 并消除了有出错倾向的状态等。Java 与 C++ 最大的区别, 在于 Java 采用的指针模型可以消除重写内存和损坏数据的可能性。

一般的语言程序都是在运行过程中, 才能检测一些程序错误的内容。而 Java 编译器能够在程序编写过程中, 检查程序在编译和运行时的错误, 如类型检查帮助检查出许多开发早期出现的错误。

Java 还实现了真数组, 避免了覆盖数据的可能。这些功能特征大大缩短了开发 Java 应用程序的周期。

5. 结构中立性

Java 编译器通过生成与特定的计算机体系结构无关的字节码指令来实现这一特性。

编译生成一个体系结构中立的字节码 (Byte-Code) 目标文件格式, 这是一种编译过的代码, 只要有 Java 运行时系统, 就可以在许多处理器上运行。

字节码 (Byte-Code) 格式语言不仅可以很容易地在任何机器上解释执行, 而且还可以迅速地翻成本地机器的代码。

6. 安全性

Java 不需要使用指针构造诸如字符串、数组这样的结构。如果必要的话, 它也能够具有指针的能力, 如链表。Java 永远不会存取一个“坏的”指针, 造成内存分配的错误, 也不必防范内存泄漏。

另外, 使用 Java 创建浏览器时, 语言功能和浏览器本身提供的功能结合起来, 使它更安全。并且, 在编写 Java 代码时, 还需要过很多次的测试, 经过代码校验、检查代码段的格式、检测指针操作、对象操作等等。

7. 可移植性

Java 与体系结构无关, 如大多数编译器产生的目标代码只能运行在一种 CPU 上, 即使支持多种 CPU 的编译器, 也不能同时适合多种 CPU 的目标代码 (J-Code)。

而 Java 解释产生的目标代码是针对一种并不存在的 CPU, 即 Java 虚拟机。Java 虚拟机避免了不同 CPU 之间的差别, 使 J-Code 能运行于任何具有 Java 虚拟机的机器上。

因此, Java 程序可以方便地移植到不同的计算机中, 并且 Java 的类库也实现了针对不同平台的接口。

8. 解释器

Java 解释器相当于运行 Java 字节码的 CPU, 但该 CPU 不是通过硬件实现的, 而是用软件实现的。Java 解释器实际上是特定的平台下的一个应用程序。只要实现了特定平台下的解释器程序, Java 字节码就能通过解释器程序在该平台下运行, 这是 Java 跨平台的根本原因。

简单地讲, Java 解释器是把 java 虚拟机上运行的目标代码 (字节码) 解释成为具体平台的机器码的程序。

一般用户可以在 JDK 或 JRE 的 Bin 目录中, 看到有 Java.exe 和 Javac.exe 文件。其中, Javac.exe 是编译器, 而 Java.exe 是解释器。运行 Java 程序时, 是先用 Javac.exe 编译器进行编译代码, 然后用 Java.exe 解释器进行代码解释。而代码一经编译成功后, 就可以直接用 Java.exe 随处解释运行。

9. 高效性

前面已经介绍过, Java 程序代码被编译成字节码 (Byte-Code) 格式, 字节码格式在设计上即已考虑了机器码的产生, 因此实际的机器码生成程序相当简单。

另外, Java 已经拥有与 C/C++ 同样, 甚至更好的运行性能, 直接使用解释器 1 秒钟内, 可调用 300 000 个过程。翻译目标代码的速度与 C/C++ 的性能不相上下。

10. 多线程

Java 多线程是 Java 语言中一个非常重要的部分,尤其是在 Java 5.0 之后,Java 对多线程做了很多扩展,大大增强了 Java 多线程编程的能力。

线程是程序执行流的最小单元,也是进程中的一个实体,是被系统独立调度和分派的基本单位。线程是程序中一个单一的顺序控制流程。在单个程序中同时运行多个线程完成不同的工作,称为多线程。

Java 实现的多线程技术,比 C/C++ 更强健。例如,在打开浏览器时,显示网页内容等待时间较长,而多线程技术可以用一个线程来图片,而另外线程显示网页文本信息等,这样显示网页的速度会非常快。

11. 动态性

Java 动态特性是其面向对象设计方法的扩展,允许程序动态地装入运行过程中所需要的类。在 C++ 程序设计过程中,每当在类中增加一个实例变量或一种成员函数后,引用该类的所有子类都必须重新编译,否则将导致程序崩溃。

而 Java 编译器不是将对实例变量和成员函数的引用编译为数值引用,而是将符号引用信息保存在字节码中下传递给解释器,再由解释器在完成动态连接类,然后将符号引用信息转换为数值偏移量。这样,对类中的变量和方法进行更新时就不至于影响现存的代码。解释执行字节码时,这种符号信息的查找和转换过程仅在一个新的名字出现时才进行一次,随后代码便可以全速执行。

如果程序连接了网络中另一系统中的某一类,该类的所有者也可以自由地对该类进行更新,而不会使任何引用该类的程序崩溃。

Java 还简化了使用一个升级的或全新的协议的方法。如果系统运行 Java 程序时遇到无法处理的程序,Java 能自动下载所需要的功能程序。

1.2 深入熟悉 Java

通过上述内容,读者已经对 Java 及 Java 语言有所了解,但在学习 Java 语言之前,先来深入的了解一下 Java 工作原理,以及 Java 虚拟机工作原理。通过对下面内容的学习,可以很清楚的认识到 Java 语言是如何在 Java 平台中运行的。

1.2.1 Java 运行流程

用 Java 可以编写各种各样的程序,并应用于不同的平台及操作系统上,如应用程序、Web 程序等等。而无论开发什么样的程序,Java 的运行流程大致相同,如图 1-4 所示。

从图 1-4 中可以清楚了解到,首先编写的代码为 Java 源程序,以 .java 为扩展名的文件。而在编写 Java 源程序时,除通过一些 Java 开发工具外,还可以直接在文本编辑器中直接编写代码,并将扩展名更改为 .java 即可,如 Windows 操作系统中的记事本。