

计算机热门软件入门与提高丛书

掌握Java，拥有更大舞台！

◎ 基于Java SE 6最新的内容，快速升级到Java SE 6平台。

◎ 基于每个知识点的编程实例，帮助初学者快速掌握。

◎ 流行开发工具Eclipse、JDeveloper和Ant等框架的详细说明。

◎ 帮助读者轻松完成从Java的基础理论到高级应用的学习旅程。

◎ 书中程序代码可以从www.bhp.com.cn免费下载。

Java

入门与提高

徐建华 李玉林 马军 王岩 编著



科学出版社
www.sciencep.com

计算机热门软件入门与提高丛书

Java

入门与提高



内 容 简 介

本书是专门介绍 Java 语言使用方法与技巧的图书。本书由浅入深、循序渐进地介绍了 Java 语言的程序开发的知识体系。

全书共分 18 章, 内容涵盖了 Java 语言基础、Java 控制语句、数组、字符串处理、面向对象设计、Java 的常用类、Java 异常处理机制、IO 及文件操作、Java 5.0 的新特性、开发工具 Eclipse、单元测试框架 JUnit、JDeveloper 开发、XML 开发、网络编程、Java 线程和 Java 图形编程等。

本书最大的特色在于每一节的例子都是经过精挑细选, 具有很强的针对性, 力求让读者通过简单的例子, 学习尽可能多的知识。

本书适用于初、中级 Java 开发人员, 同时也可用作高校相关专业师生和社会培训班的教材。

本书源代码可从 www.bhp.com.cn 免费下载。

需要本书或技术支持的读者, 请与北京清河 6 号信箱 (邮编: 100085) 发行部联系, 电话: 010-82702660 (发行), 82702675 (邮购), 62978181 (总机), 传真: 010-82702698 E-mail: tbd@bhp.com.cn。

图书在版编目 (CIP) 数据

Java 入门与提高 / 徐建华, 李玉林, 马军, 王岩编写. —

北京: 科学出版社, 2008.10

(计算机热门软件入门与提高丛书)

ISBN 978-7-03-022790-4

I. J... II. ①徐...②李...③马...④王... III. Java 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字 (2008) 第 124097 号

责任编辑: 周凤明 / 责任校对: 马 君

责任印刷: 密 东 / 封面设计: 青青果园

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京市密东印刷有限公司

科学出版社发行 各地新华书店经销

*

2008 年 10 月第 一 版 开本: 787×1092 1/16

2008 年 10 月第一次印刷 印张: 25.75

印数: 1-3 000 字数: 602 000

定价: 39.00 元

前言

在当今的各种编程语言辈出的时代,Java 语言从产生至今已经走过了十几个年头,并且还释放着强大的生命力。Java 语言不仅是一个正在被广泛应用的编程语言,而且许多新的技术领域都涉及到了 Java 语言。Java 已经成为软件开发人员需要掌握的一门基础语言。

Java 的最初版本是 Sun 公司于 1995 年发布的 Java 1.0,不久 Sun 公司在 Java 1.0 的基础上又发布了 Java 1.1,功能进一步增强。直到现在,Java 语言已经发布到了最新的 J2SE 6.0。随着 Java 语言的功能的不断增强,Java 语言越来越多地受到软件开发人员和公司的青睐。

为了帮助众多软件开发人员提高 Java 语言的编程水平,笔者精心编著了本书。本书依照读者的学习规律,首先介绍 Java 语言的基本概念和常用的 API,在读者掌握了这些基本概念和常用的 API 的基础上,再对 Java 面向对象编程的内容进行深入的讲解,严格遵循由浅入深、循序渐进的原则。

本书在内容的编排和目录组织上都十分讲究,争取让读者能够快速掌握 Java 编程,比如在每一个知识点,都配有相关的小例子,并且有每一个步骤的详细说明,读者即使是刚接触 Java 语言也可以很快的熟悉。另外,除了常用 Java 类的使用外,还讲述了 Java 流行开发工具 Eclipse、JDeveloper 和 Ant 等框架的使用。读者在学习完本书之后,不仅可以学习到 Java 的基本理论,也可以快速地掌握 Java 的高级应用部分,为真正开发软件项目打好坚实的基础。

和其他书籍相比,本书有何特点

1. 实例经典,内容丰富

本书不像其他书籍一样,只单纯地讲解如何使用 Java 的理论知识,而是针对每个知识点,都配有精心设计的程序实例,将需要注意的问题和在实际编程中常遇到的困惑融入到程序当中进行讲解,这样读者能更快地掌握相应的知识。

2. 取材广泛,内容充实

本书不仅涉及 Java 基本理论和常用的类的使用,还涵盖了 Java 最新版本中新增加的一些特性。例如基本数据类型的自动装箱 (Auto Boxing) 与拆箱 (Unboxing)、新的 for/in 循环形式、可变长参数方法 (Varargs Method)、静态导入 (Import Static)、范型 (Generics) 以及注释 (Annotation) 等。

3. 讲解通俗,步骤详细

每个示例程序都以通俗易懂的语言阐述,并穿插编写步骤和运行结果,在阅读时就像听课一样详细而贴切。读者只需要按照步骤操作,就可以学习到 Java 的各种知识,并体会到编程的乐趣。

4. 附带源码,内容超值

本书中所涉及到的所有源代码可以从 www.bhp.com.cn 免费获得,有助于读者快速学习。

本书包括的内容

第1章首先带领读者了解Java的应用领域,了解Java的历史、Java语言的特点和优势、Java的版本、Java语言的基本编译和运行原理,最后简单介绍了Java虚拟机的基本知识。

第2章,介绍Java语言的编程基础。包括Java标识符、关键字、变量和常量、Java的基本数据类型以及类型的转换、Java的运算符、运算符的优先级和Java编码规范等。

第3章进一步介绍Java的程序流程控制,包括条件语句、循环语句的使用,这些都是掌握Java语言所最基础的内容。

第4章详解介绍了Java的基本数据类型数组,包括一维数组、二维数组、数组的初始化以及数组的复制等基本原理。

第5章详细讲解Java的基本字符串类型,内容包括字符串类(String)的常用方法、字符串优化类(StringBuffer)、字符类(Character)和字符串标记类(StringTokenizer)的使用。

第6章主要介绍Java的面向对象编程,包括类和对象的概念、类的继承、覆盖和多态等方面的知识,以及包、接口、抽象类、内部类等方面的知识,并详细介绍了Java中的一些常用的关键字final、static、this、super的使用。

第7章讲述了作为Java中的最常用的几个类的使用,包括数学计算类(Math类)、日期类(Date类和Timer类)、定时器类(Timer类)、属性类(Properties类)以及集合类(Collections类)的使用。

第8章介绍了Java异常处理机制的相关内容,包括异常的分类、如何捕捉一个异常、finally子句的使用、如何自定义异常类型。

第9章全面讲述了Java语言中进行文件读写IO操作的相关内容,包括文件类(File类)的使用、输入输出流的操作、随机访问文件类(RandomAccessFile类)、可序列化接口(Serializable)和文件压缩类。

第10章介绍了Java 5.0高级特性,内容涉及基本数据类型的自动装箱(Auto Boxing)与拆箱(Unboxing)、新的for/in循环形式、可变长参数方法(Varargs Method)、静态导入(Import Static)、范型(Generics)以及注释(Annotation)等。

第11章讲述流行的开发工具Eclipse,从而读者可以进行快速开发。内容包括Eclipse的下载和安装、Eclipse基本开发环境、如何编译、运行和调试Java程序,最后介绍了Eclipse开发Java时的一些编辑功能。

第12章讲述单元测试框架JUnit,内容包括JUnit的下载和安装、如何编写测试用例、如何进行单元测试等。

第13章讲述了如何使用开源框架Ant进行Java开发,内容包括Ant的下载和安装、如何编写Ant任务、如何使用Java编译、打包和运行Java程序。

第14章讲述了另一种比较流行的开发工具JDeveloper,它与Eclipse相比也有其独特的优势。内容包括JDeveloper的下载和安装,如何使用JDeveloper开发、编译、调试和运行Java程序以及作为IDE其特有的编辑功能。

第15章介绍了Java中XML的开发,主要内容有XML的基本语法,在Java中如何读取和解析XML。

第16章介绍了Java中网络编程的相关知识,主要内容有网络的基本术语、网络协议、

IP 地址、基于 TCP 协议的 Socket 通信、基于 UDP 协议的 Socket 通信以及支持多客户连接的 Socket 通信等。

第 17 章介绍了 Java 中有关线程的开发，主要内容有线程 (Thread) 的基本概念、运行线程的两种基本方法、对线程的控制、线程的同步以及如何防止死锁等问题。

第 18 章介绍了 Java 图形编程的相关知识，主要内容有 Swing 的常用标准控件 (按钮 JButton、输入框 JTextField、密码输入框 JPasswordField、文本域 JTextArea、标签 JLabel、单选按钮 JRadioButton、按钮组 ButtonGroup、多选按钮 JCheckBox、下拉列表 JComboBox、列表 JList、菜单 JMenu 和右键弹出菜单 JPopupMenu)、布局管理器以及事件等。

适合阅读本书的读者

本书具有知识全面、实例精彩、指导性强的特点，力求以全面的知识性及丰富的实例来指导读者透彻学习 Java 各方面的技术。本书可以作为初次学习 Java 的入门教材，也可以帮助中级读者提高 Java 技能，对高级读者也有一定的启发意义。

本书由徐建华和洛阳理工学院的李玉林、马军、王岩主持编写，同时参与编写的还有周科杰、唐翔、戴珊珊、方安乐、曲先发、李宗琅、李继攀、杨水清、徐磊、吝晓宁、陈杰、栗春阳、梁璐、毛磊达、王太刚、王春中、罗名兰、谢广文、赖玫玫、邓剑钦、陈见勇、雷天均、韦一、廉秀霞、吝晓宁、江丽、项宇峰、张启玉等，在此一并表示感谢。

作者

目 录

第1章 Java 概述.....1	
1.1 Java 产生的背景.....1	
1.2 Java 语言的特点.....2	
1.2.1 简单性.....2	
1.2.2 面向对象.....2	
1.2.3 平台无关性.....2	
1.2.4 分布式.....2	
1.2.5 安全性.....2	
1.2.6 健壮性.....3	
1.2.7 多线程.....3	
1.2.8 解释性和高性能.....3	
1.3 Java 2 SDK 版本.....3	
1.3.1 J2ME.....3	
1.3.2 J2SE.....3	
1.3.3 J2EE.....3	
1.4 Java 开发工具.....4	
1.4.1 Applet 浏览器 appletviewer.....4	
1.4.2 Java 编译器 javac.....4	
1.4.3 Java 解释器 java.....4	
1.4.4 Java 文档生成器 javadoc.....5	
1.4.5 Java 打包工具 Jar.....5	
1.4.6 Java 类分解器 javap.....5	
1.5 Java 虚拟机.....6	
1.5.1 Java 虚拟机 JVM.....6	
1.5.2 Java 虚拟机工作原理.....6	
1.6 上机实战——两个简单的 Java 程序	
HelloWorld.....7	
1.6.1 Java 应用程序	
Application-HelloWorld.....7	
1.6.2 Java 小应用程序——HelloWorld.....9	
1.7 小结.....10	
第2章 Java 编程基础.....11	
2.1 注释规则.....11	
2.2 标识符和关键字.....12	
2.2.1 标识符.....12	
2.2.2 关键字.....13	

2.3 变量和常量.....13	
2.3.1 变量.....13	
2.3.2 常量.....14	
2.4 基本数据类型.....14	
2.4.1 整数类型.....14	
2.4.2 浮点类型.....15	
2.4.3 字符类型.....15	
2.4.4 逻辑类型.....16	
2.5 数据类型之间的转换.....17	
2.5.1 自动类型转换.....17	
2.5.2 强制类型转换.....17	
2.6 Java 中的运算符.....18	
2.6.1 算术运算符.....18	
2.6.2 关系运算符.....19	
2.6.3 逻辑运算符.....19	
2.6.4 赋值运算符.....20	
2.6.5 其他运算符.....20	
2.7 运算符的优先级.....22	
2.8 Java 语言的编码规范.....23	
2.8.1 语法规范.....23	
2.8.2 代码编写规范.....23	
2.9 小结.....24	
第3章 程序控制语句.....25	
3.1 条件语句.....25	
3.1.1 if 语句.....25	
3.1.2 switch 语句.....27	
3.2 循环语句.....29	
3.2.1 for 循环结构.....29	
3.2.2 while 循环结构.....30	
3.2.3 do-while 循环结构.....31	
3.3 关键字 break 和 continue.....32	
3.3.1 break 语句.....32	
3.3.2 continue 语句.....33	
3.4 小结.....34	
第4章 数组.....35	
4.1 一维数组.....35	

4.1.1 一维数组的定义和创建.....	35	5.4.3 取字符对象的值.....	66
4.1.2 数组元素的使用.....	36	5.5 字符串标记类 (StringTokenizer)	66
4.1.3 数组的初始化.....	36	5.5.1 常用构造函数.....	66
4.1.4 一维数组的使用.....	37	5.5.2 常用方法.....	67
4.1.5 数组元素的缺省值.....	37	5.6 小结.....	68
4.2 二维数组.....	38	第6章 面向对象程序设计.....	69
4.2.1 二维数组定义和创建.....	39	6.1 面向对象的基本概念.....	69
4.2.2 二维数组初始化.....	39	6.1.1 面向过程与和面向对象的 程序设计.....	69
4.2.3 二维数组的使用.....	40	6.1.2 对象和封装.....	69
4.3 数组的操作.....	41	6.1.3 类和实例.....	70
4.3.1 数组的长度.....	41	6.2 类.....	71
4.3.2 数组的复制.....	42	6.2.1 类的定义.....	71
4.4 数组的排序.....	43	6.2.2 成员变量和局部变量.....	72
4.5 小结.....	45	6.2.3 成员方法.....	73
第5章 字符串处理.....	46	6.2.4 构造方法.....	75
5.1 字符串类 (String)	46	6.2.5 main 方法.....	76
5.1.1 常用构造函数.....	46	6.2.6 类对象的创建、使用和清除.....	76
5.1.2 字符串的连接.....	48	6.2.7 方法中参数的传递.....	79
5.1.3 获取字符串的长度.....	48	6.3 包.....	82
5.1.4 获取字符串某位置的字符.....	49	6.3.1 包的作用.....	82
5.1.5 字符串的比较.....	49	6.3.2 包的定义.....	82
5.1.6 字符串的查找与截取.....	51	6.3.3 运行包中的类.....	82
5.1.7 字符串大小写转换.....	52	6.3.4 引入包中的类.....	83
5.1.8 字符串内容的替换.....	53	6.4 类中变量和方法的访问权限.....	84
5.1.9 分割字符串.....	54	6.4.1 公有类型变量和方法.....	84
5.1.10 字符串的其他操作.....	55	6.4.2 受保护类型变量和方法.....	85
5.1.11 格式化字符串.....	57	6.4.3 私有类型变量和方法.....	86
5.2 字符串数组 (String[])	60	6.5 类的继承.....	87
5.2.1 字符串数组的构造函数.....	60	6.5.1 继承的概念.....	87
5.2.2 字符串数组的属性及常用方法.....	61	6.5.2 继承的实现——创建子类.....	88
5.3 字符串缓存类 (StringBuffer)	61	6.5.3 继承数据成员和方法.....	88
5.3.1 常用构造函数.....	62	6.5.4 覆盖.....	90
5.3.2 字符串缓存与字符串的转化.....	62	6.5.5 继承中要注意的问题.....	92
5.3.3 字符串缓存的扩充.....	63	6.6 类的多态性.....	92
5.3.4 字符串缓存的插入.....	63	6.7 抽象类.....	94
5.4 字符类 (Character)	65	6.7.1 理解抽象类.....	94
5.4.1 常用构造函数.....	65	6.7.2 抽象类的定义.....	94
5.4.2 字符的比较.....	65		

6.7.3	抽象类的应用	94	7.5.1	向量表(ArrayList)	130
6.8	接口	96	7.5.2	游标(Iterator)	132
6.8.1	接口的定义	96	7.5.3	向量(Vector)	133
6.8.2	接口的使用	97	7.5.4	哈希集合(HashSet)	134
6.8.3	接口的继承	98	7.5.5	哈希映射(HashMap)	136
6.9	内部类	99	7.5.6	哈希表(Hashtable)	138
6.9.1	匿名类(anonymous class)	99	7.6	小结	139
6.9.2	静态类(static class)	100	第8章	异常的处理	140
6.9.3	成员类(member class)	100	8.1	错误跟异常的区别	140
6.9.4	局部类(local class)	101	8.2	异常的分类	141
6.10	和类相关的其他关键字	101	8.3	捕捉一个异常	142
6.10.1	final 修饰符	102	8.4	使用 finally 子句	144
6.10.2	static 修饰符	103	8.5	抛出一个异常	146
6.10.3	this 关键字	105	8.6	自定义异常类型	147
6.10.4	super 关键字	105	8.7	小结	151
6.11	小结	106	第9章	I/O 及文件操作	152
第7章	Java 常用类的使用	107	9.1	文件类(File)	152
7.1	数学计算(Math)类	107	9.1.1	常用构造函数	152
7.1.1	Math 的数学常量	107	9.1.2	常用方法	152
7.1.2	Math 的常用方法	108	9.1.3	列出文件信息	153
7.2	日期(Date)类型	110	9.1.4	使用 FilenameFilter	155
7.2.1	Date 的构造函数	111	9.1.5	查找文件实例	157
7.2.2	Date 的常用方法	112	9.1.6	文件的创建	160
7.2.3	java.sql.Date 与 java.sql.Time	115	9.1.7	文件的删除和更名	162
7.3	计时器(Timer)类	116	9.2	流(Stream)	164
7.3.1	Timer 的构造函数	116	9.2.1	输入流与输出流	164
7.3.2	Timer 的常用方法	116	9.2.2	标准输入输出流	165
7.3.3	用 TimerTask 描述一个任务	117	9.2.3	流的常用方法	166
7.3.4	一个简单的 Timer 实例	117	9.2.4	读取文件内容的操作	167
7.4	属性(Properties)类	118	9.2.5	将内容保存到文件的操作	170
7.4.1	Properties 类的构造函数	119	9.2.6	I/O 读写时可能遇到的异常	173
7.4.2	Properties 的常用方法	119	9.2.7	文件的锁定操作	174
7.4.3	使用 Properties 配置发送电子邮件	119	9.3	随机访问文件类(RandomAccessFile)	176
7.4.4	Properties 文件的读写	122	9.3.1	常用构造函数	176
7.4.5	Properties 资源的国际化(Internationalize)	127	9.3.2	常用方法	176
7.5	集合工具类(Collections)	130	9.3.3	随机访问文件实例	177
			9.4	可序列化接口(Serializable)	182
			9.5	文件的压缩	184

9.5.1	文件压缩实例	185
9.5.2	文件解压缩实例	187
9.6	小结	188
第 10 章	Java 高级特性	189
10.1	自动装箱与拆箱	189
10.1.1	适用范围	189
10.1.2	自动装箱与拆箱示例	189
10.1.3	注意事项	191
10.2	新的 for/in 循环形式	192
10.2.1	遍历数组	192
10.2.2	遍历集合	193
10.2.3	for/in 循环示例	194
10.3	可变长参数方法	194
10.3.1	定义可变长参数方法	195
10.3.2	可变参数方法示例	195
10.4	协变式返回类型	196
10.4.1	概述	196
10.4.2	一个实例	196
10.5	静态导入	197
10.5.1	概述	197
10.5.2	一个实例	197
10.6	范型	198
10.6.1	概述	198
10.6.2	使用多个范型	199
10.6.3	范型通配符	200
10.6.4	范型化类与方法	201
10.7	枚举类型	203
10.7.1	概述	203
10.7.2	枚举类型的用处	204
10.7.3	扩展枚举类型	205
10.8	注解	207
10.8.1	概述	207
10.8.2	Java 标准注解	208
10.8.3	自定义注解	209
10.9	小结	210
第 11 章	开发工具 Eclipse	211
11.1	Eclipse 概述	211
11.1.1	Eclipse 平台体系结构	211

11.1.2	Eclipse 的下载和安装	212
11.1.3	Eclipse 不同版本的选择	213
11.2	认识工作台 (Workbench)	213
11.2.1	什么是工作台	213
11.2.2	认识视图 (View)	214
11.2.3	认识编辑器 (Editor)	215
11.2.4	认识透视图 (Perspective)	216
11.2.5	视图、编辑器与透视图的关系	217
11.3	开发 Java 程序	217
11.3.1	查看 JDK 版本	217
11.3.2	创建 Java 项目	218
11.3.3	创建 Java 包	219
11.3.4	创建 Java 类	219
11.4	编译和运行 Java 程序	220
11.4.1	编译后文件存放目录	221
11.4.2	导入项目使用的包	222
11.4.3	设置编译方式	223
11.4.4	编译程序	224
11.4.5	运行程序	225
11.5	调试 Java 程序	226
11.5.1	理解断点 (Breakpoint)	226
11.5.2	调式程序	227
11.5.3	调试透视图	228
11.5.4	断点后的处理	229
11.5.5	查看调试中的变量值	229
11.5.6	条件断点	230
11.6	常用的代码编辑功能	231
11.6.1	代码提示	231
11.6.2	错误更正	232
11.6.3	格式化代码	232
11.6.4	查看代码结构	232
11.6.5	覆盖父类方法	233
11.6.6	代码重构	234
11.7	设置首选项	235
11.8	小结	236
第 12 章	单元测试框架 JUnit	237
12.1	JUnit 概述	237
12.1.1	什么是 JUnit	237

12.1.2	为什么要进行 JUnit 测试	237	13.3.7	复制文件	255
12.1.3	JUnit 的术语介绍	238	13.3.8	设置文件集合	256
12.1.4	JUnit 与 Eclipse 的集成	238	13.4	编译、打包和运行 Java 程序	256
12.2	编写第一个 JUnit 测试	238	13.5	小结	258
12.2.1	编写一个 Java 类	239	第 14 章	JDeveloper 开发	259
12.2.2	创建测试类	239	14.1	Developer 概述	259
12.2.3	运行 JUnit	240	14.1.1	JDeveloper 界面一览	259
12.3	JUnit 中常用的接口和类	241	14.1.2	JDeveloper 的功能	260
12.3.1	Test 接口	241	14.2	JDeveloper 的下载和安装	260
12.3.2	TestCase 抽象类	242	14.2.1	JDeveloper 的下载	260
12.3.3	Assert 断言类	242	14.2.2	JDeveloper 的安装	260
12.3.4	TestSuite 测试包类	242	14.3	编写第一个 Java 程序	261
12.3.5	TestResult 结果类	243	14.3.1	新建应用程序 (Application)	261
12.3.6	其他类与接口	243	14.3.2	新建项目工程 (Project)	262
12.4	进行 JUnit 测试的基本步骤	243	14.3.3	新建 Java 类	263
12.4.1	编写要测试的类	243	14.3.4	运行 Java 程序	264
12.4.2	编写测试用例	245	14.4	JDeveloper 的自定义设置	264
12.4.3	运行 JUnit 测试用例	246	14.4.1	项目属性设置	264
12.5	小结	246	14.4.2	导入项目使用的类库	265
第 13 章	使用 Ant 进行 Java 开发	247	14.4.3	首选项设置	266
13.1	Ant 概述	247	14.5	常用的代码编辑功能	266
13.1.1	什么是 Ant	247	14.5.1	添加注释	266
13.1.2	Ant 的下载	247	14.5.2	自定义格式化代码	267
13.1.3	Ant 的安装与配置	248	14.5.3	自动生成 getter 和 setter 代码	268
13.1.4	第一个 Ant 构建文件	249	14.5.4	代码的重构	269
13.1.5	运行 Ant	250	14.5.5	代码的展开和折叠	270
13.2	编写 build.xml 文件	251	14.5.6	代码比较	270
13.2.1	project 元素	251	14.5.7	子类覆盖父类的方法	271
13.2.2	target 元素	252	14.5.8	代码模板	272
13.2.3	task 元素	253	14.6	代码错误提示	273
13.2.4	property 元素	253	14.6.1	如何定位错误	273
13.3	常用 Ant 任务	254	14.6.2	自动修正错误	274
13.3.1	输出信息	254	14.7	文件查找	274
13.3.2	编译 Java 源文件	254	14.7.1	文件内部查找	274
13.3.3	打包任务	255	14.7.2	项目内查找	275
13.3.4	运行 Java 程序	255	14.8	使用快捷键	275
13.3.5	删除文件夹或文件	255	14.8.1	查看快捷键	275
13.3.6	创建文件夹	255	14.8.2	编辑快捷键	276

14.9 程序调试.....	276	16.1.5 TCP/IP 协议.....	299
14.9.1 设置断点.....	276	16.2 URL.....	300
14.9.2 跟踪调试程序.....	276	16.2.1 URL 构成.....	300
14.10 小结.....	277	16.2.2 URL 类.....	300
第 15 章 XML 开发概述.....	278	16.3 InetAddress 类.....	303
15.1 XML 概述.....	278	16.4 基于 TCP 协议的 Socket 通信.....	304
15.1.1 XML 的特点.....	278	16.4.1 理解 Socket.....	304
15.1.2 XML 的用处.....	278	16.4.2 Socket 和 ServerSocket.....	304
15.1.3 XML 的标签.....	279	16.4.3 Java 网络通信模型.....	306
15.2 XML 语法.....	279	16.4.4 一个基于 TCP 的 C/S 程序举例.....	306
15.2.1 XML 文档的例子.....	279	16.5 基于 UDP 协议的 Socket 通信.....	308
15.2.2 XML 标签必须结束.....	280	16.5.1 DatagramSocket 和 DatagramPacket 类.....	308
15.2.3 XML 标签大小写敏感.....	280	16.5.2 基于 UDP 的 C/S 程序举例.....	309
15.2.4 XML 元素必须合理嵌套.....	280	16.6 支持多客户连接的 Socket 通信方式.....	311
15.2.5 XML 有且仅有一个根元素.....	280	16.6.1 支持多客户连接的 Socket 通信模型.....	311
15.2.6 XML 属性必须使用引号.....	281	16.6.2 创建服务器线程.....	312
15.2.7 XML 中的空白将被保留.....	281	16.6.3 服务器主程序.....	313
15.2.8 XML 中的注释.....	281	16.6.4 客户机程序.....	314
15.3 XML 元素.....	281	16.7 小结.....	315
15.3.1 元素命名规则.....	281	第 17 章 线程.....	316
15.3.2 元素间的关系.....	282	17.1 线程(Thread).....	316
15.4 XML 属性.....	282	17.1.1 线程简介.....	316
15.4.1 属性命名规则.....	282	17.1.2 线程状态.....	316
15.4.2 属性的一些缺点.....	282	17.2 创建线程.....	316
15.5 XML 解析.....	283	17.2.1 使用 Thread 类创建线程.....	317
15.5.1 DOM 解析.....	283	17.2.2 使用 Runnable 创建线程.....	318
15.5.2 SAX 解析.....	289	17.3 线程的属性.....	319
15.6 使用 JDOM 简化 XML 解析.....	293	17.3.1 优先级(Priority).....	319
15.6.1 JDOM 简介.....	293	17.3.2 守护线程(Daemon Thread).....	321
15.6.2 免费获取 JDOM.....	293	17.4 线程的操作.....	323
15.6.3 JDOM 的 XML 解析.....	294	17.4.1 让步(yield).....	323
15.7 小结.....	297	17.4.2 沉睡(sleep)与唤醒(interrupt) ...	324
第 16 章 Java 网络技术.....	298	17.4.3 停止(stop).....	326
16.1 网络相关知识.....	298	17.4.4 连接(join).....	327
16.1.1 IP 地址.....	298	17.5 线程间的同步.....	328
16.1.2 端口.....	298		
16.1.3 客户机和服务器.....	299		
16.1.4 面向连接和非连接.....	299		

17.5.1	线程并发的例子.....	328	18.5.4	文本域(JTextArea).....	359
17.5.2	线程间的同步.....	331	18.5.5	标签(JLabel).....	363
17.5.3	等待(wait)与通报(notify).....	332	18.5.6	单选按钮(JRadioButton)与 按钮组(ButtonGroup).....	366
17.6	线程死锁(dead lock).....	335	18.5.7	多选按钮(JCheckBox).....	368
17.6.1	一个死锁的例子.....	335	18.5.8	下拉列表(JComboBox)与 列表(JList).....	371
17.6.2	死锁的解决方法.....	337	18.5.9	菜单(JMenu).....	374
17.7	小结.....	339	18.5.10	右键弹出菜单(JPopupMenu)....	378
第 18 章	图形编程.....	340	18.6	布局管理器(LayoutManager).....	381
18.1	AWT 概述.....	340	18.6.1	不使用布局.....	381
18.1.1	小应用程序(Applet).....	340	18.6.2	流式布局(FlowLayout).....	382
18.1.2	应用程序(Frame).....	342	18.6.3	边框布局(BorderLayout).....	383
18.2	Swing 概述.....	343	18.6.4	网格布局(GridLayout).....	384
18.2.1	什么是 Swing.....	343	18.7	事件处理(Event).....	385
18.2.2	一个简单的 Swing.....	344	18.7.1	事件(Event)与监听器(Listener)...	386
18.3	JApplet.....	345	18.7.2	ActionEvent 与 ActionListener.....	386
18.3.1	JApplet 的常用方法.....	345	18.7.3	MouseEvent 与 MouseListener.....	388
18.3.2	JApplet 的一个实例.....	345	18.7.4	KeyEvent 与 KeyListener.....	390
18.3.3	在网页里运行 JApplet.....	347	18.7.5	WindowEvent 与 WindowListener.....	391
18.3.4	在 Eclipse 中运行 JApplet.....	349	18.8	Swing 高级组件.....	393
18.4	应用程序(JFrame).....	350	18.8.1	简单对话框(JOptionPane).....	393
18.4.1	JFrame 的构造函数.....	350	18.8.2	文件选择框(JFileChooser).....	394
18.4.2	JFrame 的常用方法.....	350	18.8.3	表格(JTable).....	396
18.4.3	JFrame 实例.....	351	18.8.4	树(JTree).....	398
18.5	常用组件(JComponent).....	352	18.9	小结.....	401
18.5.1	常用组件的共性操作.....	352	附录	JDK 命令详解.....	402
18.5.2	按钮(JButton).....	353			
18.5.3	输入框(JTextField)与密码 输入框(JPasswordField).....	355			

第1章 Java 概述

Java 语言是 Sun 公司推出的具有革命性的、优秀的编程语言，它具有面向对象、与平台无关、安全、稳定和多线程等优良特性，是目前程序设计中较为健壮的语言。Java 语言不仅是一个正在被广泛应用的编程语言，而且许多新的技术领域都涉及到了 Java 语言。Java 已经成为软件开发人员必须掌握的一门基础语言。

本章将对 Java 语言做一个简单的介绍，并初步了解 Java 语言的工具及使用方法，配置 Java 中的环境变量与 Java 虚拟机 JVM，对 Java 应用程序 (Application) 和 Java 小应用程序 (Applet) 有一个初步的认识。

1.1 Java 产生的背景

Java 语言的前身是 Oak 语言，是 Sun 公司在 1991 年由一个叫 Green 的项目组专门为家用消费电子产品开发的一个分布式代码系统，主要用于与电视机、冰箱、空调等家电产品进行信息交换和控制。由于商业原因，Oak 语言一直没有被推广出去。

1994 年随着 Internet 的快速发展，Sun 公司的 Bill Joe 加入了 Green 小组，决定将 Oak 语言应用于 Internet 的 WWW 开发中，并编制了 HotJava 浏览器。通过 HotJava 浏览器在 Internet 上的应用，充分展现了 Oak 语言的网络方面的特点。同时，Bill Joe 决定通过 Internet 让世界上无数的软件开发人员免费地使用 Java 语言，这更加推动了 Java 语言的发展。

1995 年 Sun 公司将 Oak 语言正式命名为 Java，同时召开了 Java 语言的发布会，推出 Java 和 HotJava 的最初版本。

Java 是 Sun 公司推出的新一代面向对象程序设计语言，特别适用于 Internet 应用程序的开发，其平台无关性是其突出的优势。Java 语言作为软件开发的一项革命性的技术，其地位已经确立，这主要表现在以下几个方面：

- 计算机行业许多大公司购买了 Java 许可证，包括 IBM、Apple、DEC、HP、Oracle 等。这一点说明 Java 语言已经得到了业界的认可。
- 众多软件开发商开始开发支持 Java 的软件产品。例如：Borland 公司的 Jbuilder，IBM 的 Visual Age for Java，Sun 公司的 Forte、NetBeans，IBM 捐献给开放源码社区的 Eclipse，Oracle 公司的 JDeveloper 等，这些都为 Java 的开发者提供了方便、快捷的开发环境。
- 随着 Internet 成为企业信息系统的最佳解决方案，Java 将发挥着重要的作用。

Java 的最初版本是 Sun 公司于 1995 年发布的 Java 1.0，不久 Sun 公司在 Java 1.0 的基础上又发布了 Java 1.1，功能进一步增强。

从 1998 年开始，Sun 公司相继推出了 J2SE 1.2 到现在的 J2SE 6.0，随着 Java 语言功能的不断增强，Java 语言越来越多地受到软件开发人员和公司的青睐。

1.2 Java 语言的特点

Java 语言是当前最流行的网络编程语言。和 C++ 语言相比, 其有着特殊的优点, 如简单、面向对象、平台无关性、分布式、安全性、多线程等。

1.2.1 简单性

Java 语言是一种面向对象的语言, 它通过提供最基本的方法来完成指定的任务, 用户只需理解一些基本的概念, 就可以用其编写出适合于各种情况的应用程序。Java 语言的大多数语法是从 C++ 继承过来的, 但 Java 略去了 C++ 中所用到的一些难以理解而极易混淆的概念, 如 Java 语言中没有了头文件、指针、运算符重载、多重继承等模糊的概念, 并且通过实现自动垃圾收集机制大大简化了程序设计者的内存管理工作。

1.2.2 面向对象

Java 语言的面向对象的概念吸取了 C++ 面向对象的思想。面向对象的最大优点是将数据封装于类中, 利用类的优点, 实现了程序的简洁性和便于维护性。类的封装性、继承性等有关对象的特性, 使程序代码只需一次编译, 然后通过上述特性反复利用。程序员只需把主要精力用在类和接口的设计和应用上。Java 提供了众多的一般对象的类, 通过继承即可使用父类的方法, 实现代码的重用。

在 Java 中, 类的继承关系是单一的非多重的, 一个子类只有一个父类, 子类的父类也只有一个父类。

1.2.3 平台无关性

平台无关性是指 Java 能运行于不同的平台上。Java 引进了虚拟机原理, Java 的数据类型与机器无关, Java 虚拟机 (Java Virtual Machine) 是建立在硬件和操作系统之上的, 实现 Java 二进制代码的解释执行功能, 提供不同平台的接口。Java 语言可以运行在 Windows、Solaris、Unix、Linux 等操作系统上, 使得 Java 程序经过一次编译, 就可以运行在安装了 Java 运行环境的操作系统上。真正实现了“一次编译, 到处运行”的特点。

1.2.4 分布式

Java 建立在扩展的 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议传送和接收信息的方法。Java 应用程序通过 URL 对象访问网络资源, 这使得程序员使用网络上的文件就像使用本地文件一样容易。

1.2.5 安全性

Java 的编程类似 C++, 学习过 C++ 的读者将很快掌握 Java 的精髓。Java 舍弃了 C++ 的指针对存储器地址的直接操作, 程序运行时, 内存由操作系统分配, 这样可以避免病毒通过指针侵入系统。Java 对程序提供了安全管理器, 如 Applet (小应用程序)、互联网程序是不能随意访问客户端的资源的, 这样可防止病毒的侵入和程序的非法访问。

1.2.6 健壮性

Java 语言开发的应用程序是非常稳健的, Java 语言没有指针的应用, 在没有授权的情况下是不能访问内存的, 减少了内存出错的可能性。Java 致力于检查程序在编译和运行时的错误。类型检查可帮助用户查出程序开发早期出现的错误。Java 语言对程序的错误采用异常处理机制, 保证了 Java 程序的正常运行。

1.2.7 多线程

Java 支持多线程编程, 允许利用多线程编写开发程序, 其同步机制保证了对共享数据的正确操作。多线程带来的最大好处是更好的交互性能和实时控制性能。

1.2.8 解释性和高性能

Java 语言是解释性语言, 但并不影响其运行的高效率。Java 源程序通过编译器生成字节代码。在运行时, Java 程序依靠 Java 解释器将其转换成机器码, 即直接对 Java 字节码进行解释执行。

1.3 Java 2 SDK 版本

Java 软件开发工具包 Java 2 SDK 是 Java 2 Soft Development Kit 的缩写, 之前的名称叫 JDK (Java Development ToolKit)。目前 Java 2 平台有 3 个版本: J2ME、J2SE 和 J2EE。

1.3.1 J2ME

J2ME 是 Java 2 微型版, 是 Java 2 Micro Edition 的缩写。用于开发嵌入式设备和电子设备产品。例如, 手机、PDA、电视机顶盒、汽车导航系统等。J2ME 在开发面向内存有限的移动终端(如移动电话)的应用时, 显得特别实用。目前很多手机上的游戏均是利用 Java 开发的。

J2ME 是建立在操作系统之上的, 使得应用的开发无须考虑太多特殊的硬件配置类型和操作系统, 因此开发商也无须为不同的终端建立特殊的应用。

1.3.2 J2SE

J2SE 是 Java 2 平台标准版, 是 Java 2 Standard Edition 的缩写。是 Sun 公司为桌面开发和低端商务计算解决方案而开发的版本。例如, 用户可以利用 J2SE 开发 Applet (小应用程序) 和 Application (应用程序) 桌面应用程序。这个版本是基础, 是平时开发和使用最多的技术, Java 的主要技术将在这个版本中体现, 这也是本书讲述的重点。

1.3.3 J2EE

J2EE 是 Java 2 平台企业版, 是 Java 2 Enterprise Edition 的缩写。是一种利用 Java 2 平台来简化企业解决方案的开发、管理和部署相关复杂问题的体系结构。

J2EE 技术的基础是核心 J2SE, 它不仅巩固了 J2SE 的优点, 同时还提供了对 EJB (Enterprise JavaBeans)、Servlet (服务小程序)、JDBC (Java 数据库连接) 和 JSP (Java Server Page) 等技术的全面支持。

总之, J2ME 是针对消费性产品而设计的; J2SE 是针对 PC 机、笔记本电脑而设计的, 是人们接触最多的; J2EE 是针对企业级和服务器端的高端应用而设计的。

1.4 Java 开发工具

Java 开发工具提供了编译、运行、调试 Java 程序的方法, 主要包括 appletviewer、javac、java、javadoc、jar 等。下面分别讲述其用途、语法及应用举例。

1.4.1 Applet 浏览器 appletviewer

appletviewer 是 Java 小应用程序浏览器, 用于浏览 Java 小应用程序。使用 appletviewer 的优点是可以及时反映程序所做的修改, 而通常的 IE 浏览器由于具有保存 Applet 的功能, 所以不能及时反映程序所做的修改。

appletviewer 的语法如下:

```
appletviewer [options] url
```

其中: options 是可选参数, 不用此参数也可以运行 appletviewer 命令; url 指包含 Applet 的网页。

应用举例:

要访问当前目录下的带有 Applet 的 HelloWorld.html 文件, 可以在控制台输入:

```
appletviewer HelloWorld.html
```

即可浏览 Applet 程序。

1.4.2 Java 编译器 javac

javac 是 Java 语言的编译器, 它将后缀名为 .java 的源文件编译成后缀名为 .class 的字节码文件。javac 的语法为:

```
javac [options] [sourcefiles]
```

其中: options 是可选参数, 使用 javac 可以不加参数; sourcefiles 是要编译的 .java 文件, 若要一次编译多个文件, 则多个文件之间用 “,” 分开。

应用举例:

```
javac HelloWorld.java // 只编译 HelloWorld.java 一个文件
```

```
javac HelloWorld.java, Hello.java // 同时编译 HelloWorld.java 和 Hello.java 两个文件
```

1.4.3 Java 解释器 java

Java 是 java 语言的解释器, 实际上是 Java 类文件的解释器, 它直接从类文件执行 Java 应用程序。

java 编译器的语法为:

```
java [options] class [arguments...] // 用来执行 Java 的类文件;
```

其中: options 是可选参数, 使用 java 可以不加参数; class 表示 java 命令要运行的程序; arguments 是可选参数, 程序运行的外部参数。

```
java [options] -jar file.jar [arguments...] // 用来执行一个在 JAR 压缩包中的文件。
```

其中: -jar file.jar 表示运行压缩包中的文件。

应用举例:

```
java HelloWorld // 运行 HelloWorld 程序;
```