

Broadview[®]
www.broadview.com.cn

全面覆盖 500 多个技术要点，随查随用，带你勇攀 Java 技术高峰！

Java 开发手册

桂颖 谷涛 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Java 开发手册

桂颖 谷涛 编著

电子工业出版社

Publishing House of Electronics Industry

内 容 简 介

本书由浅入深、全面系统地介绍 Java 编程和开发的方方面面。书中各章提供了大量有针对性的实例，供读者实战演练。本书针对初学和自学读者的特点，按照实际开发流程对 Java 语言的程序设计进行讲解，简明易懂、实用性强。书中以通俗易懂的语言介绍 Java 语言，力求做到深入浅出。

全书共 30 章，分为 5 篇。Java 基础篇介绍 Java 的概念、数据类型、运算符、流程控制语句及数组等内容；面向对象篇介绍对象和类、访问控制符、继承、接口、封装类及异常的处理；Java 常用技术篇介绍字符串、集合、内部类、线程和相关算法的应用；界面编程篇介绍 Java 的 Swing 编程、相关布局使用的控件和图像处理等知识；高级技术篇介绍网络编程的基本知识、输入/输出流的使用和 JDBC 数据库开发技术。

无论对于 Java 的初学者，还是有一定基础的高级用户，本书都是一本难得的参考书。本书非常适合使用 Java 进行编程和开发的本科生、研究生及教师阅读，还适合广大科研和工程技术人员研读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Java 开发手册 / 桂颖, 谷涛编著. —北京: 电子工业出版社, 2013.7
ISBN 978-7-121-20916-1

I. ①J… II. ①桂… ②谷… III. ①JAVA 语言—程序设计—技术手册 IV. ①TP312-62

中国版本图书馆 CIP 数据核字 (2013) 第 146169 号

责任编辑: 李云静

印 刷: 涿州市京南印刷厂

装 订: 涿州市京南印刷厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 53 字数: 1416 千字

印 次: 2013 年 7 月第 1 次印刷

定 价: 99.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

Java 代表了当今国际科学计算软件的先进水平，应用领域非常广泛。很多人都希望将 Java 语言应用于自己的项目和实践中，从而可以直观方便地进行开发、计算和设计工作。编写这本书的目的，就是为了向读者全方位介绍 Java 编程和开发技术。

本书由浅入深、全面系统地介绍 Java 基础、面向对象、Java 常用技术、界面编程及高级技术应用等相关知识。书中的每一章都提供了大量的实例程序，以方便读者进行练习和学习。每个例程都经过精挑细选，具有很强的针对性，适合各个阶段的读者学习。本书既注重基础知识，又非常注重实践，读者可以快速上手并迅速提高。通过学习本书内容，读者不仅可以全面掌握 Java 编程和开发技术，还可以获得快速分析和解决实际问题的能力，从而能够在较短的时间内，以较好的效果来解决实际的工程和学习问题，并提升工作、学习效率。

本书的特点

1. 结构合理，内容全面、系统

本书详细介绍 Java 编程基础、面向对象思想、常用开发技术、界面编程及高级技术应用，将实际项目开发经验贯穿于全书，项目内容非常丰富。在内容的安排上，则根据读者的学习习惯和内容的梯度合理安排，更加适合读者学习。

2. 叙述翔实，例程丰富

本书有详细的例程，每个例子都经过精挑细选，有很强的针对性。书中的程序都有完整的代码，而且代码非常简洁和高效，便于读者学习和调试。读者也可以直接重用这些代码来解决问题。

3. 结合实际，编程技巧贯穿其中

Java 编程非常灵活，所以本书写作时特意给出了大量的实用编程技巧，这些技巧的灵活使用，将会让你事半功倍。

4. 语言通俗，图文并茂

对于程序的运行结果，本书给出了大量的图片。本书不仅注重基础知识，而且也非常注重实践，以便让读者快速上手，迅速掌握 Java 知识。

本书内容体系

本书共 30 章，分为 5 篇，各篇对应的章节和具体内容介绍如下。

第 1 篇包括第 1~5 章，主要介绍 Java 的基础知识，讲解 Java 的安装和基本操作，数据类型、运算符及数组的操作。

第 2 篇包括第 6~12 章，主要介绍 Java 面向对象思想，详细讲解继承、接口、构造器、封装类及怎样处理异常等重要知识。

第 3 篇包括第 13~19 章，主要介绍 Java 开发中经常用到的相关技术，包括字符串的应用、

集合的使用、内部类的使用、线程的技术，以及相关算法。这些都是学好 Java 的必备知识点。

第 4 篇包括第 20~27 章，主要介绍 Java 中的 Swing 编程，包括布局的设计、常用的控件、树状列表的使用，以及对相关图像进行处理技术。

第 5 篇包括第 28~30 章，主要介绍 Java 中经常用到的一些高级技术应用。例如，网络编程的应用、输入/输出流的应用及 JDBC 数据库开发技术等。

本书读者对象

- Java 初学者
- 想全面、系统地学习 Java 的人员
- Java 技术爱好者
- 利用 Java 进行编程和开发的技术人员
- 大中专院校的学生
- 相关培训学校的学员

本书由桂颖、谷涛编写。其中东华理工大学的桂颖负责编写第 1~19 章，天津工业大学的谷涛负责编写第 20~30 章。

另外，本书的相关源代码可以通过 <http://www.broadview.com.cn/20916> 下载。

目 录

第 1 篇 Java 基础

第 1 章 初步认识 Java	1	2.4.2 强制转换类型	31
1.1 Java 的概念	1	2.5 基本数据类型的对象包装类	34
1.1.1 Java 是什么	1	2.6 注释	35
1.1.2 Java 平台	1	2.6.1 单行注释	36
1.1.3 Java 的操作系统	2	2.6.2 区域注释	36
1.1.4 Java 的优点	2	2.6.3 文档注释	36
1.2 Java 的历史	3	2.7 小结	38
1.2.1 Java 的由来	3	第 3 章 运算符详解	39
1.2.2 Java 的发展	4	3.1 算术运算符	39
1.3 如何配置 Java 开发环境	4	3.1.1 加法运算符	39
1.3.1 开发工具包	4	3.1.2 减法运算符	42
1.3.2 运行环境	4	3.1.3 乘法运算符	43
1.3.3 JDK 的安装和下载	5	3.1.4 除法运算符	43
1.3.4 环境变量的设置	6	3.1.5 取余运算符	44
1.3.5 环境变量的测试	7	3.1.6 自增、自减运算	44
1.4 第一个应用程序——HelloWorld	7	3.2 关系与逻辑运算	46
1.4.1 HelloWorld 程序	7	3.2.1 等于/不等于运算	46
1.4.2 编译与运行	8	3.2.2 比较大小运算	47
1.4.3 编译和运行命令	9	3.2.3 “与”运算	48
1.5 小结	12	3.2.4 “或”运算	49
第 2 章 基本数据类型	13	3.2.5 “非”运算	50
2.1 数据类型	13	3.3 三元运算符	50
2.1.1 整型	13	3.4 位运算	51
2.1.2 浮点型	16	3.5 移位运算	53
2.1.3 字符型	19	3.5.1 “<<”左移运算	53
2.1.4 布尔型	20	3.5.2 “>>”右移运算	53
2.2 常量	21	3.5.3 “>>>”无符号右移运算	54
2.2.1 标识符	21	3.5.4 具体实例	55
2.2.2 常量	22	3.6 赋值运算	55
2.3 变量	24	3.6.1 普通赋值运算	56
2.3.1 声明变量及初始化	24	3.6.2 运算赋值运算	56
2.3.2 变量的作用域	26	3.6.3 括号及运算符间的 优先级关系	57
2.3.3 数据类型的自动转换	28	3.7 java.lang.Math 类	57
2.4 基本数据类型之间的转换	29	3.7.1 数学常量	57
2.4.1 自动转换类型	29		



- 3.7.2 常用数学函数方法 58
- 3.8 小结 60
- 第 4 章 流程控制语句** 61
 - 4.1 if 语句 61
 - 4.1.1 常用语句形式 61
 - 4.1.2 完整语句形式 62
 - 4.2 if 语句的嵌套 63
 - 4.3 switch 语句 65
 - 4.3.1 switch 判断语法 65
 - 4.3.2 判断表达式 67
 - 4.3.3 case 表达式 68
 - 4.4 switch 实例 69
 - 4.5 while 循环语句 71
 - 4.6 do-while 循环语句 73
 - 4.7 for 循环语句 74
 - 4.7.1 for 循环语法 74
 - 4.7.2 for 循环实例 76
 - 4.7.3 for 实现其他循环 77
 - 4.8 break 语句 78
 - 4.9 continue 语句 79
 - 4.10 小结 81

- 第 5 章 不可或缺的数组** 82
 - 5.1 数组的概念 82
 - 5.2 一维数组 83
 - 5.2.1 数组的声明 83
 - 5.2.2 创建数组及初始化 83
 - 5.2.3 数组的结构 84
 - 5.2.4 数组长度及越界 84
 - 5.2.5 数组程序实例 85
 - 5.3 多维数组 86
 - 5.3.1 多维数组的声明 86
 - 5.3.2 创建多维数组 86
 - 5.3.3 多维数组结构及初始化 87
 - 5.4 访问多维数组 88
 - 5.5 数组的相关用法 90
 - 5.5.1 复制数组 90
 - 5.5.2 数组修饰符 92
 - 5.5.3 搜索数组 93
 - 5.5.4 排序数组 95
 - 5.5.5 Arrays 类的 sort 方法 98
 - 5.5.6 数组的比较 99
 - 5.6 小结 100

第 2 篇 面向对象

- 第 6 章 对象和类的“二人转”** 101
 - 6.1 什么是面向对象 101
 - 6.1.1 面向对象的由来 101
 - 6.1.2 面向过程与面向对象思想的碰撞 102
 - 6.1.3 面向对象的背景及特点 104
 - 6.2 类与对象 104
 - 6.2.1 成员变量 105
 - 6.2.2 成员变量的默认值 107
 - 6.2.3 对象引用变量 108
 - 6.3 类的方法 110
 - 6.4 方法和变量引用 111
 - 6.4.1 调用不存在的方法或成员变量 111
 - 6.4.2 用空引用进行调用 112
 - 6.4.3 数组的空引用问题 113
 - 6.5 局部变量 113

- 6.5.1 局部变量的作用域 113
- 6.5.2 局部变量的初始化 114
- 6.6 变长参数 117
- 6.7 this 对象引用 118
- 6.8 系统常用类 120
 - 6.8.1 Date 类 120
 - 6.8.2 GregorianCalendar 类 122
 - 6.8.3 常用系统已有类 124
- 6.9 小结 125
- 第 7 章 Java 中的访问控制符** 126
 - 7.1 类的访问控制 126
 - 7.1.1 public 级别 126
 - 7.1.2 默认访问级别 128
 - 7.2 成员的访问控制 130
 - 7.2.1 公共访问 131
 - 7.2.2 保护访问 132
 - 7.2.3 默认访问 133
 - 7.2.4 私有访问 134



7.2.5 访问级别与数据封装	136	8.8.2 最终的方法	184
7.3 static 关键字	137	8.9 abstract 与继承	185
7.3.1 静态成员变量	138	8.9.1 抽象类	185
7.3.2 静态最终成员变量	139	8.9.2 抽象方法	186
7.3.3 static 方法	140	8.10 继承的多态	189
7.3.4 static 代码块	140	8.11 小结	191
7.4 final 的变量	141	第 9 章 接口的世界	192
7.4.1 final 成员变量	141	9.1 接口的概述	192
7.4.2 final 局部变量	144	9.2 接口中的成员变量	193
7.5 package 的使用	146	9.3 接口中的方法	195
7.5.1 package 的声明与创建	146	9.4 接口引用的使用	198
7.5.2 引入 package 资源	147	9.4.1 可以指向的对象类型	198
7.5.3 静态引入 package	150	9.4.2 接口引用间的赋值	198
7.6 小结	152	9.4.3 类引用与接口引用间的 赋值	200
第 8 章 父与子的继承	153	9.4.4 接口中方法无法 使用的修饰符	201
8.1 什么是继承	153	9.5 接口与抽象类	203
8.1.1 类和类的关系	153	9.6 接口的多态	205
8.1.2 继承性	156	9.7 接口回调	207
8.1.3 类的继承	156	9.8 instanceof 的使用	208
8.2 继承与隐藏	157	9.8.1 基本语法使用	208
8.2.1 成员变量的继承	157	9.8.2 不允许进行测试的 情况	211
8.2.2 成员变量的隐藏	160	9.9 小结	212
8.3 对象	162	第 10 章 构造器应用	213
8.3.1 对象引用类型	162	10.1 构造器的概念	213
8.3.2 强制类型转换	162	10.2 访问限制修饰符与构造器	213
8.3.3 对象调用成员	164	10.2.1 公有类型构造器	213
8.3.4 对象赋值和比较	165	10.2.2 默认与保护类型 构造器	214
8.4 方法的继承	167	10.2.3 私有类型构造器以及 静态工厂方法	215
8.5 方法的重写	168	10.3 构造器与返回类型	216
8.5.1 重写的条件	170	10.4 创建对象	218
8.5.2 返回类型的规则	171	10.5 重载构造器	219
8.5.3 访问级别	173	10.6 级联调用的构造器	220
8.6 重写与继承的关系	174	10.6.1 如何调用构造器	220
8.6.1 静态方法没有重写	175	10.6.2 级联调用	221
8.6.2 重写扩展父类功能	177	10.6.3 默认构造器	223
8.6.3 替代性	178	10.7 自定义构造器	224
8.7 方法的重载	178	10.8 构造器不能继承	225
8.7.1 重载的规则	178		
8.7.2 重载方法的匹配	179		
8.7.3 重写与重载	183		
8.8 final 与继承	183		
8.8.1 最终的方法	183		



10.9	调用兄弟构造器	226	12.1.1	捕获异常的 try 和 catch	252
10.9.1	调用语法	226	12.1.2	finally 语句	256
10.9.2	注意事项	227	12.1.3	try、catch 和 finally 语句	259
10.10	单列模式	229	12.2	捕获异常	261
10.11	程序的加载	230	12.3	未捕获异常	263
10.12	小结	232	12.4	再次抛出异常	264
第 11 章	迷惑的封装类	233	12.4.1	异常的再抛出	264
11.1	封装类是什么	233	12.4.2	显性再抛出	265
11.2	字符串与基本数据类型值的转换	235	12.4.3	隐性再抛出	268
11.3	封装类与基本数据类型的转换	238	12.5	方法重写对抛出异常的限制	269
11.4	字符串与封装类的转换	239	12.6	自定义异常	271
11.5	其他常用方法	239	12.6.1	自己创建异常类	271
11.5.1	静态工厂方法	239	12.6.2	使用自定义异常	273
11.5.2	isNaN()方法	241	12.6.3	显性再抛出异常的作用	274
11.5.3	equals()方法	242	12.7	异常的匹配	277
11.5.4	toString()方法	244	12.8	断言	279
11.5.5	进制转换方法	244	12.8.1	为什么要引入断言	279
11.6	自动打包/解包	245	12.8.2	断言的语法规则	280
11.7	特殊数值计算	247	12.8.3	断言与编译	280
11.7.1	计算特大整数	248	12.8.4	启动断言执行代码	281
11.7.2	精确计算浮点型	250	12.8.5	防止滥用断言	282
11.8	小结	251	12.9	小结	283
第 12 章	如何处理异常	252			
12.1	异常处理	252			

第 3 篇 Java 常用技术

第 13 章	字符串应用	284	13.7	正则表达式	301
13.1	String 类	284	13.7.1	正则表达式的语法	302
13.2	String 对象的内存机制	289	13.7.2	Pattern 与 Matcher 类的综合实例	304
13.3	String 内存机制的优势	290	13.8	在 String 类中使用正则表达式	308
13.4	String 对象特殊机制的劣势	294	13.8.1	匹配检查	308
13.5	StringBuffer 类	295	13.8.2	查找替换	310
13.5.1	StringBuffer 类的优势	295	13.8.3	对字符串进行分析	311
13.5.2	编写方法链	297	13.9	小结	312
13.5.3	StringBuffer 类的重要方法	298	第 14 章	集合的使用	313
13.6	StringBuilder 类	300	14.1	集合框架的结构	313



14.2	Object 类	314	第 15 章	内部类应用	369
14.2.1	toString 方法的重写	314	15.1	非静态内部类	369
14.2.2	equals 方法	315	15.2	在外部类中创建内部类	369
14.2.3	hashCode 方法	317	15.3	在外部类外创建内部类	371
14.3	equals 与 hashCode 方法的 重写	318	15.4	内部类与外部类之间的 成员互访	373
14.4	Ordered 与 Sorted 的接口	322	15.4.1	在内部类中访问 外部类的成员	373
14.5	列表	323	15.4.2	在外部类中访问 内部类的成员	374
14.5.1	List 接口	323	15.5	预定义对象引用 this	375
14.5.2	数组实现列表	324	15.6	局部内部类	376
14.6	向量	326	15.6.1	定义局部内部类及 创建	376
14.7	链接实现列表	327	15.6.2	局部变量与局 部内部类	377
14.7.1	依赖性倒置原理	328	15.6.3	静态方法中的 局部内部类	380
14.7.2	将数组转换为列表	329	15.7	静态内部类	381
14.8	集合	330	15.8	静态/非静态内部类的区别	383
14.8.1	Set 接口	330	15.9	匿名内部类	383
14.8.2	HashSet 类	331	15.10	内部类的理解	389
14.8.3	equals 与 hashCode 方法重写的作用	332	15.11	内部接口	390
14.8.4	LinkedHashSet 类的 使用	335	15.12	小结	393
14.9	SortedSet 接口与 TreeSet 类	336	第 16 章	多线程的功力	394
14.10	自定义 Sorted 集合的类	338	16.1	线程的概念	394
14.10.1	定制 SortedSet 的 排序规则	340	16.2	自定义线程	395
14.10.2	集合的遍历	343	16.2.1	继承 Thread 类	395
14.10.3	使用 for-each 循环遍历集合	345	16.2.2	实现 Runnable 接口	395
14.11	映射	346	16.2.3	两种方式的比较	396
14.11.1	Map 接口	346	16.3	创建线程对象	396
14.11.2	HashMap 类	347	16.3.1	继承 Thread 类方式	396
14.11.3	Hashtable 类	348	16.3.2	实现 Runnable 接口方式	397
14.11.4	LinkedHashMap 类	349	16.4	启动线程	398
14.12	SortedMap 接口与 TreeMap 类	351	16.5	同时使用多个线程	400
14.13	映射的遍历实例	353	16.6	线程的状态	401
14.14	栈的实现	355	16.7	线程的使用	403
14.14.1	Stack 类	355	16.7.1	睡眠使用	403
14.14.2	Deque 接口	356	16.7.2	线程的优先级	405
14.14.3	栈计算数学表达式	358	16.7.3	线程的让步	407
14.15	集合元素的常用操作	361	16.7.4	守护线程	410
14.16	小结	368			



16.8	同步线程	411	17.7	队列	453
16.8.1	使用同步	412	17.8	阻塞栈	459
16.8.2	同步调度方法	415	17.8.1	BlockingDeque 接口与 LinkedBlockingDeque 类	459
16.9	“生产者—消费者”案例	415	17.8.2	LinkedBlockingDeque 类的使用实例	460
16.10	notify 方法	420	17.9	线程安全的单变量操作	461
16.11	同步语句块	421	17.9.1	atomic 包	461
16.12	线程的死锁	424	17.9.2	atomic 包中类的 使用实例	462
16.13	注意事项	426	17.10	障碍物	463
16.14	获取运行线程	427	17.10.1	CyclicBarrier 类	464
16.15	volatile 关键字	428	17.10.2	CyclicBarrier 类的 使用实例	464
16.16	小结	429	17.11	小结	466
第 17 章	高级线程	430	第 18 章	内存管理技术	467
17.1	线程池	430	18.1	程序中的“垃圾”是什么	467
17.2	固定尺寸线程池	430	18.2	“垃圾”收集器	470
17.2.1	自定义尺寸固定线 程池实例	431	18.3	如何收集“垃圾”	472
17.2.2	单任务线程池实例	433	18.3.1	finalize 重写	472
17.2.3	可变尺寸线程池 实例	434	18.3.2	finalize 安全问题	474
17.2.4	延迟线程池的实例	436	18.4	最终守护者实例	476
17.3	使用自定义线程池	437	18.5	非线程“垃圾”	477
17.4	有返回值的线程调用	440	18.6	线程“垃圾”	479
17.4.1	Callable 接口	440	18.7	三种特殊的引用	481
17.4.2	Future 接口	440	18.7.1	弱引用实例	481
17.4.3	Callable 与 Future 接口的使用实例	441	18.7.2	软引用实例	483
17.5	封锁资源	442	18.7.3	幻影引用实例	485
17.5.1	Lock 接口与 ReentrantLock 类	443	18.8	小结	485
17.5.2	ReentrantLock 锁 使用实例	443	第 19 章	数据结构和算法	486
17.5.3	ReadWriteLock 接口与 ReentrantReadWriteLock 类	446	19.1	栈和队列	486
17.5.4	ReentrantReadWriteLock 读/写锁使用实例	447	19.1.1	栈的应用	486
17.6	信号量	450	19.1.2	队列的应用	489
17.6.1	Semaphore 类	450	19.2	三种排序	492
17.6.2	Semaphore 类 使用实例	451	19.2.1	冒泡排序	492
			19.2.2	选择排序	495
			19.2.3	插入排序	498
			19.3	小结	501



第 4 篇 界面编程

第 20 章 Swing 编程	502	21.9 弹簧布局	551
20.1 Swing 是什么	502	21.9.1 弹簧布局简介	551
20.2 第一个 Swing 小程序	502	21.9.2 控件与容器边框之间 使用支架	551
20.3 JFrame 类	504	21.9.3 控件与容器边框之间 使用弹簧	552
20.4 事件处理	508	21.10 SpringLayout 类	554
20.4.1 事件处理模型	509	21.11 SpringLayout.Constraints 内部类	555
20.4.2 事件层次结构	510	21.12 Spring 类	556
20.5 窗体事件	511	21.13 弹簧布局使用实例	557
20.6 事件适配器	514	21.14 弹簧布局描述法使用实例	558
20.7 容器	516	21.15 小结	561
20.8 JPanel 类	517	第 22 章 各种 Swing 控件	562
20.9 JLabel 类	518	22.1 控件类	562
20.10 JButton 类	521	22.2 文本框与密码框的使用实例	565
20.11 监听器与事件源对应 关系的研究	524	22.3 文本区使用实例	569
20.12 小结	527	22.4 开关按钮使用实例	574
第 21 章 布局流行设计	528	22.5 单选按钮与复选框	577
21.1 布局管理器概述	528	22.5.1 JRadioButton 类	577
21.2 常用布局管理器	528	22.5.2 ButtonGroup 类	578
21.3 FlowLayout 流布局的使用	529	22.5.3 JCheckBox 类	578
21.3.1 流布局简介	529	22.6 ItemEvent 事件	579
20.3.2 流布局使用实例	531	22.7 ItemEvent 事件实例	579
21.4 GridLayout 网格布局	532	22.8 单选按钮与复选框的 综合实例	582
21.4.1 网格布局简介	532	22.9 小结	585
21.4.2 网格布局使用实例	533	第 23 章 菜单、工具栏和对话框的 使用	586
21.5 BorderLayout 边框布局	535	23.1 菜单	586
21.5.1 边框布局简介	536	23.2 JMenuBar 类	587
21.5.2 边框布局使用实例	537	23.2.1 使用窗体提供的 setJMenuBar 方法	588
21.6 空布局	539	23.2.2 使用传统的 add() 方法	588
21.6.1 空布局简介	539	23.3 JMenuItem 类	589
21.6.2 空布局使用实例	540	23.3.1 为菜单项添加助记符	590
21.7 CardLayout 卡片布局	541	23.3.2 为菜单项设置快捷键	590
21.7.1 卡片布局简介	541	23.4 JMenu 类	591
21.7.2 卡片布局使用实例	543	23.5 JRadioButtonMenuItem 类	593
21.8 BoxLayout 箱式布局	545		
21.8.1 箱式布局简介	545		
21.8.2 Box 容器简介	547		
21.8.3 Box 容器与 BoxLayout 布局管理器使用实例	548		



- 23.6 JCheckBoxMenuItem 类 593
- 23.7 菜单使用实例 594
- 23.8 弹出式菜单实例 597
- 23.9 鼠标事件使用实例 598
- 23.10 工具栏的开发使用实例 602
- 23.11 对话框使用实例 606
 - 23.11.1 JDialog 类简介 606
 - 23.11.2 JOptionPane 类简介 607
 - 23.11.3 JOptionPane 对话框使用实例 609
- 23.12 文件选择器 611
- 23.13 颜色选择器 614
- 23.14 文件、颜色对话框使用实例 615
- 23.15 小结 617
- 第 24 章 经典的树状列表 618**
 - 24.1 树的专有名词 618
 - 24.2 JTree 类 619
 - 24.3 树模型 623
 - 24.4 树的节点 625
 - 24.5 树的路径 628
 - 24.6 树的相关事件 629
 - 24.6.1 TreeSelectionEvent 事件 629
 - 24.6.2 TreeExpansionEvent 事件 630
 - 24.6.3 TreeModelEvent 事件 631
 - 24.7 树节点的绘制 632
 - 24.7.1 TreeCellRenderer 接口 632
 - 24.7.2 DefaultTreeCellRenderer 类 632
 - 24.7.3 自定义绘制器实例 633
 - 24.8 树状列表的综合实例 635
 - 24.8.1 实例概述 635
 - 24.8.2 搭建界面 636
 - 24.8.3 信息提示功能 638
 - 24.8.4 节点增删功能 640
 - 24.8.5 图标更改功能 643
 - 24.9 小结 646
- 第 25 章 表格制造 647**
 - 25.1 表格简介 647
 - 25.2 JTable 类 648
 - 25.2.1 使用 JTable 的简单实例 650
 - 25.2.2 表格的数据模型 652
 - 25.2.3 表格模型的使用实例 655
 - 25.3 表格列 657
 - 25.3.1 TableColumnModel 接口 658
 - 25.3.2 DefaultTableColumnModel 类 659
 - 25.4 表格的相关事件 661
 - 25.4.1 TableColumnModelEvent 事件 661
 - 25.4.2 TableModelEvent 事件 662
 - 25.4.3 ListSelectionEvent 事件简介 662
 - 25.4.4 表格事件使用实例 663
 - 25.5 表格绘制器 665
 - 25.5.1 TableCellRenderer 接口 666
 - 25.5.2 DefaultTableCellRenderer 类简介 666
 - 25.5.3 自定义表格绘制器 667
 - 25.6 表格编辑器 668
 - 25.6.1 TableCellEditor 接口 668
 - 25.6.2 AbstractCellEditor 类 668
 - 25.6.3 DefaultCellEditor 类 669
 - 25.6.4 为特定的类型指定编辑器 669
 - 25.7 自定义表格编辑器与绘制器的使用实例 670
 - 25.7.1 实例概述 670
 - 25.7.2 界面框架的搭建 670
 - 25.7.3 自定义表格以及表格控件的添加 671
 - 25.7.4 自定义绘制器 673
 - 25.7.5 自定义编辑器的添加 674
 - 25.8 表格中的排序 677
 - 25.8.1 RowSorter 类 678
 - 25.8.2 DefaultRowSorter 类 678
 - 25.8.3 TableRowSorter 类简介 679



25.9	表格中的过滤	680	26.10	异或模式绘图	708
25.10	表格排序与过滤的 使用实例	682	26.11	抗锯齿	709
25.10.1	实例概述	682	26.12	动画	710
25.10.2	搭建界面框架	683	26.12.1	编写动画的原理	710
25.10.3	添加表格	684	26.12.2	重新绘制	710
25.10.4	添加排序器	685	26.13	Timer 类	711
25.10.5	添加设置过滤条件的 控件	686	26.14	简单动画使用实例	712
25.10.6	表格设置过滤器	688	26.15	小结	714
25.11	小结	691	第 27 章	处理图像	715
第 26 章	图形绘制与动画	692	27.1	加载与绘制	715
26.1	绘制简单图形	692	27.2	图标	718
26.1.1	可以充当画布的控件	692	27.2.1	Icon 接口	719
26.1.2	绘图的 paint 方法	692	27.2.2	ImageIcon 类	721
26.2	画笔	693	27.3	处理图像编码	723
26.2.1	Graphics 对象的获取	694	27.3.1	JPEG 编码器	723
26.2.2	坐标系统	694	27.3.2	GifEncoder 编码器	724
26.3	调配颜色	696	27.4	图像抓取	726
26.4	图形绘制的使用实例	697	27.4.1	createScreenCapture 方法	726
26.5	绘制文本	698	27.4.2	抓屏功能使用实例	726
26.5.1	drawString 方法	698	27.5	图像滤镜的开发	730
26.5.2	控制字体	698	27.6	图像灰度处理使用实例	731
26.5.3	文本绘制使用实例	700	27.7	RGB 色彩通道过滤	734
26.6	Java 2D	701	27.8	RGB 色彩通道过滤 使用实例	735
26.7	线条的粗细控制	702	27.9	卷积滤镜	738
26.8	渐变颜色	704	27.10	卷积滤镜使用实例	740
26.9	图形变换	706	27.11	小结	743

第 5 篇 高级技术

第 28 章	网络编程的世界	744	28.5.1	客户端	748
28.1	网络编程简介	744	28.5.2	服务器端	749
28.2	网络传输协议	745	28.5.3	程序运行结果	751
28.3	Internet 地址 (InetAddress 类)	746	28.6	UDP 协议网络通信 (Datagram)	752
28.3.1	创建 InetAddress 对象的方法	746	28.7	面向无连接的使用实例	752
28.3.2	InetAddress 的 其他方法	746	28.7.1	发送端程序示例	752
28.4	Socket (套接字)	747	28.7.2	接收端程序示例	754
28.5	Socket 使用实例	748	28.7.3	程序运行结果	755
			28.8	URL 的网络编程	756
			28.8.1	URL 类与 URL 的 解析	756



28.8.2	URLConnection 类	757	29.7.4	字符缓冲区读、写流	783
28.8.3	获取远程图像	758	29.8	小结	784
28.8.4	获取远程文档	759	第 30 章	JDBC 数据库应用技术	785
28.8.5	获取远程音频	761	30.1	数据库应用架构模型	785
28.9	小结	762	30.1.1	两层结构模型	785
第 29 章	你来我往的 I/O 流	763	30.1.2	三层结构模型	786
29.1	流的简介	763	30.2	JDBC 的层次结构	787
29.2	文件 File	764	30.3	JDBC 编程入门	788
29.2.1	文件或目录的生成	764	30.3.1	数据库的创建	788
29.2.2	文件名的处理	764	30.3.2	JDBC-ODBC 连接桥	789
29.2.3	文件属性测试	765	30.3.3	加载 JDBC 驱动	792
29.2.4	普通文件信息和工具	765	30.4	建立数据库连接	793
29.2.5	目录操作	765	30.5	SQL 命令	794
29.3	File 类使用实例	765	30.6	结果集	795
29.4	RandomAccessFile 类	767	30.7	连接数据库使用实例	796
29.4.1	构造方法	767	30.8	预编译	797
29.4.2	文件指针的操作	767	30.9	访问其他数据库	800
29.5	RandomAccessFile 使用实例	767	30.9.1	MySQL 数据库的 访问	800
29.6	字节流 InputStream、 OutputStream	770	30.9.2	Oracle 数据库的访问	802
29.6.1	输入、输出流	770	30.10	事务	804
29.6.2	文件输入、输出流	772	30.10.1	编写事务	804
29.6.3	缓冲输入、输出流	773	30.10.2	批处理	807
29.6.4	数据输入、输出流	775	30.11	可滚动结果集	809
29.6.5	对象输入、输出流	776	30.12	可滚动与不可滚动 结果集的比较	810
29.7	字符流 Reader、Writer	780	30.13	控制游标移动	810
29.7.1	字符读、写流	780	30.14	元数据	813
29.7.2	输入、输出流	781	30.15	数据库综合实例	817
29.7.3	文件读、写字符流	782	30.16	小结	834

第 1 篇 Java 基础

第 1 章 初步认识 Java

随着 Internet 的迅速发展以及 WWW (World Wide Web) 的日益普及和快速增长, 计算机网络正在经历着深刻的变革。1996 年, 原 Sun 公司 (现已由甲骨文公司收购) Java 语言的正式发布, 是 Internet 的一次技术革命。Java 语言的诞生解决了 Internet 程序的安全性、健壮性、平台无关性、可移植性等诸多问题。本章主要讲述 Java 的特点及基本的开发入门知识。

1.1 Java 的概念

Java 是一门面向网络的高级程序设计语言, 程序员可以用它创建应用程序, 这些程序通过网络下载后运行, 对于本地主机是安全的。如果将 Java 和网络融合在一起, 将会组合成一个网络计算环境。Java 分布式的特性, 可使网络上的任何主机运用部署的程序。

1.1.1 Java 是什么

Java 语言作为一种编程语言, 它的语法规则与 C++ 很相似, 但又避免了 C++ 中存在的弊端, 因此有其自身的优点, 如简单、面向对象、分布式、解释性、可靠、安全、可移植性、高性能、多线程、动态性等。所以, 也可以说 Java 是一种解释性、跨平台、通用的编程语言。

Java 也是一种网络程序设计语言。Applet 程序编译器编译成字节码文件, 将被放在 WWW 网页中, 并在 HTML 做出标记, 只要用户的主机安装了 Java 就可以直接运行 Applet。Java 比较适合网络环境, 因此, 其成为 Internet 中最流行的编程语言之一。

1.1.2 Java 平台

Java 不仅具有跨平台的特性, 而且可以通用于各种计算机网络之间的技术体系。它的特点是简单、便于网络传输、对硬件的依赖性比较低。在开发了 Java 语言之后, 它已经从一门语言演化为一个计算机平台。Java 以其独特的优势, 将给未来的网络世界带来巨大的变革。Java 具有“编写一次, 到处运行”的特点, 完全实现了不同系统之间的互操作。

平台是程序运行的软件和硬件的组合, 大多数平台包括硬件和操作系统两部分。Java 平台 (Java Platform) 广义的概念指可执行 Java 字节码的平台。它由两部分组成: Java 虚拟机 (Java Virtual Machine, JVM) 和 Java 应用程序界面 (Java API)。

图 1-1 就是 Java 程序 (应用程序 (application) 或小程序 (applet)) 在 Java 虚拟机运行的结构框架。

Java 平台在具体实现中, 可以表现出如下三种方式。

- Java 虚拟机 (Java Virtual Machine, JVM)。
- Java 操作系统 (Java Operating System)。
- Java 芯片 (Java Chip)。

Java 虚拟机 指在操作系统上面运行的一种软件程序, 此程序可以解析 Java 字节码, 并在操作系统的配合下执行字节码。这样的执行框架如图 1-2 所示。

由于芯片上面执行的是操作系统, 操作系统上面又直接执行 Java 虚拟机, 而 Java 虚拟机

执行上面的 Java 字节码，因此 Java 字节码的执行是在 Java 虚拟机上面，所以包含 Java 虚拟机的部件才称为 Java 平台。

Java 操作系统 指将 Java 虚拟机与操作系统整合在一起，或者构造一个全新的 Java 操作系统。这样就形成三层的执行架构，如图 1-3 所示。

Java 芯片 指 Java 操作系统和芯片整合到一起，或者制造全新的 Java 专用芯片，因此，这样的执行架构可分为两层，如图 1-4 所示。

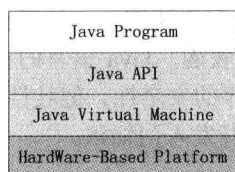


图 1-1 Java 平台结构

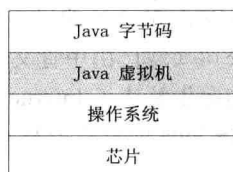


图 1-2 Java 虚拟机

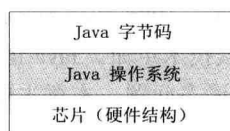


图 1-3 Java 操作系统

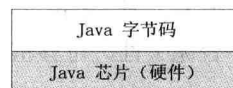


图 1-4 Java 芯片

上面介绍了三种 Java 平台结构，各有其优缺点，不能完全说哪一种方式更具有优势，必须根据实际应用情况才能决定。通常情况下，通过 Java 操作系统执行 Java 字节码效率比虚拟机的方式要好，这种方式可能成为今后的发展趋势。但是设计一套执行 Java 的全新操作系统，不是一件容易的事情。



注意：Java 芯片的方式实现 Java 平台，对于 Java 程序来说对程序的执行速度有很大的提高，这种方式将会成为嵌入式系统的主流。

1.1.3 Java 的操作系统

Java 语言不仅仅是一门编程语言，更是一个网络操作系统，因为：

- Java 语言不依赖于操作系统，它可以运行到计算机硬件或 CPU 上。
- Java 语言可以自己管理内存。
- Java 语言支持多线程运行，可对处理器资源进行分配和管理。
- Java 语言可以管理硬件设备，如鼠标、键盘等。
- Java 语言支持网络文件管理，它的文件概念已经扩展到整个 Internet 网络。

Java 是具有很强吸引力的面向对象的编程语言，又是目前最为流行的网络编程语言。Java 语言的出现引起了整个计算机软件程序开发的重大变革，这给以往的软件设计理念提出了新的挑战。

1.1.4 Java 的优点

Java 语言由 C++ 发展而来，因此其继承了 C++ 面向对象的全部特征，同时它还剔除了 C++ 语言中一些复杂的特性，成为一种纯正的面向对象的程序设计语言。可以说，Java 语言非常适合企业级的应用程序开发。Java 语言是一种简单的、面向对象的、分布式的、健壮的、安全的、平台无关的、可移植的、解释性的、高效率的、多线程的、动态的程序设计语言。

- 简单性是指由于 Java 语言最初是为了消费性电子产品进行集成控制研发的一种程序设计语言，所以它具有简单明了、易于学习的特点。
- 面向对象是对现实世界自然事物的抽象和延伸。现实世界的任何事物都可被归为一类事物，具体事物可被看作某一类事物的一个实例对象。面向对象语言是对象和消息的结合。面向对象程序设计语言的三个基本特征是：封装、多态性和继承。
- 分布式的概念包括数据形式的分布和操作形式的分布。数据分布指将数据存储在不同