

信息科学与技术丛书 程序设计系列

文杰书院 编著

Java 程序设计

完全自学手册

一本帮初学者轻松掌握 Java 编程的书

- Java 语法及面向对象编程基础
- Java Web 开发环境与技巧
- Servlet 和 Web 应用体系
- JSP 开发详解
- Java 数据库开发案例——远程教学系统
- Struts 框架开发案例——在线投票系统
- Hibernate 框架开发案例——精美留言板



随书配套源代码下载网址
www.cmpbook.com



机械工业出版社
CHINA MACHINE PRESS

信息科学与技术丛书·程序设计系列

Java 程序设计完全自学手册

文杰书院 编著



机械工业出版社

Java 语言是目前最流行、应用最广泛的程序设计语言。本书采用“基础知识+实践案例”的教学模式,以通俗易懂的语言、精挑细选的实用技巧、翔实生动的操作案例,全面介绍了 Java 编程知识及案例。其主要内容包括 Java 语法基础、算法和流程控制语句、数组、字符串处理、面向对象编程、Servlet 和 Web 应用体系、JSP 语法、访问数据库、Struts 应用、Hibernate 应用等。

本书面向 Java 编程的初、中级用户,既适合零基础的 Java 程序设计初学者入门使用,也适合 Java 编程爱好者及人员提高编辑技巧使用,还可作为 Java 培训班、计算机类成人教育、中职教育、自学考试教材使用。

图书在版编目(CIP)数据

Java 程序设计完全自学手册/文杰书院编著. —北京:机械工业出版社, 2010.1

(信息科学与技术丛书·程序设计系列)

ISBN 978-7-111-29305-7

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

中国版本图书馆 CIP 数据核字(2009)第 231574 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策划编辑:丁 诚 吴鸣飞

责任编辑:丁 诚

责任印制:李 妍

北京振兴源印务有限公司印刷

2010 年 1 月第 1 版·第 1 次印刷

184mm×260mm·36.5 印张·903 千字

0001—4000 册

标准书号:ISBN 978-7-111-29305-7

定价:65.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010) 88361066

门户网:<http://www.cmpbook.com>

销售一部:(010) 68326294

教材网:<http://www.cmpedu.com>

销售二部:(010) 88379649

读者服务部:(010) 68993821

封面无防伪标均为盗版

出版说明



随着信息科学与技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。由于信息科学与技术在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中已经得到充分、广泛的应用，所以这些专业领域中的研究人员和工程技术人员越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员不可或缺的图书。

现今，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社



前言

1995 年,美国 Sun 公司正式推出了 Java 语言。Java 语言是目前最流行的编程语言之一,该语言具有安全、跨平台、面向对象、简单和适用于网络等显著特点。

本书以 MyEclipse 6.0 为开发平台,通过对不同实例的讲解,介绍使用 MyEclipse 6.0 进行 Java 编程的方法和应用技巧。全书共分 25 章,具体内容包括初识 Java、简单了解 Java 程序、Java 语法基础、算法和流程控制语句、开发工具 MyEclipse 6.0、数组和字符串处理等,本书最后详细剖析了远程教学系统、在线投票系统和制作精美留言板等综合实例。

本书主要内容包括以下 8 个部分:

(1) Java 语言的基础知识

第 1~2 章介绍了 Java 语言的历史、Java 语言的开发环境、编写 Java 程序的语法规则等内容。

(2) Java 语法基础

第 3~7 章全面介绍了 Java 语法基础方面的知识,如 Java 语言基础、流程控制语句和数组等,并在每章最后列举 2 个实例,使读者能够在了解本章知识点的同时学会利用本章的知识编写 Java 应用程序。

(3) Java 面向对象编程

第 8~15 章全面介绍了 Java 面向对象编程方面的知识,包括类与对象、继承与多态、包与接口和泛型与集合等,讲述如何以面向对象的思想进行编写 Java 应用程序。

(4) 开发 Java Web 应用基础

第 16 章介绍了 Tomcat 6.0、Dreamweaver CS3 和 MySQL 5.1 数据库的应用方法,帮助读者搭建 Java Web 开发环境。

(5) Servlet 与 JSP

第 17 章,介绍了 Servlet 和 Web 应用体系方面的知识,包括 C/S 模式与 B/S 模式、Http 请求和响应模式、HTML 简介、Servlet 简介、Web 应用体系和 Servlet 高级应用,通过本章学习读者可以掌握 Servlet 方面的知识。第 18 章和第 19 章,则全面介绍了 JSP 方面的知识,主要包括 JSP 和 JSP 语法方面的知识,帮助读者运用 JSP 进行 Java Web 编程。

(6) 访问数据库及实例

第 20 章介绍了 JDBC 技术简介、JDBC 连接方式、JDBC 的 API、连接 MySQL 数据库实例和数据库连接池,帮助读者学习对数据库进行访问。第 21 章详细讲解了远程教学系统的编程方法,这个系统是基于 Servlet/JSP 的 B/S 模式的 Java Web 系统,采用 MySQL 5.1 数据库。通过本章的学习读者可以巩固 Servlet/JSP 和访问数据库方面的知识。

(7) Struts 框架应用及实例

第 22 章,全面介绍了 Struts、为什么要用 Struts 框架、Struts 的核心——MVC、Struts 的工作原理、Struts 的基本配置、Struts 中的控制器、表单处理器 ActionForm 和 Struts 标签库,使读者了解 Struts 框架应用方面的基础知识。第 23 章介绍了在线投票系统,本系统是

基于 Struts 框架的 Java Web 应用, 通过本章的学习读者可以巩固 Struts 框架方面的知识。

(8) Hibernate 框架应用及实例

第 24 章, 全面介绍了 Hibernate、Hibernate 框架概述、Hibernate 框架主要的 API、Hibernate 的持久化技术、Hibernate 框架中的 ORM 技术、Hibernate 框架查询语言、HQL 中的参数绑定和 Hibernate 反向工程, 使读者了解 Hibernate 框架应用方面的基础知识。第 25 章介绍了制作精美留言板, 本系统是基于 Struts 和 Hibernate 框架的 Java Web 应用, 通过本章的学习读者可以巩固 Struts 和 Hibernate 框架方面的知识。

本书面向 Java 编程的初级和中级用户, 既适合 Java 程序设计的初学者, 也可以被广大 Java 编程爱好者及从业人员作为自学手册使用, 同时还可以作为 Java 培训班、计算机类成人教育、中职教育、自学考试教材。

本书由文杰书院组织编写, 参与本书编写工作的有李军、李智颖、张辉、张洋、任新、李强、周军、蔺丹、高桂华、李统财、安国英、蔺寿江、刘义、贾亚军、蔺影、高金环、宋艳辉、贾万学、周莲波、贾亮、安国华、施洪艳、闫宗梅、蔺松权、田园、白素梅等。

我们希望读者在阅读本书之后, 初步掌握 Java 编程方法, 增长实际编程技能, 并从学习和总结操作的经验和规律, 达到灵活运用水平。鉴于编者水平有限, 书中纰漏和考虑不周之处在所难免, 欢迎读者予以批评、指正, 以便我们日后能为您编写更好的图书。

如果您在使用本书时遇到问题, 可以访问网站 <http://www.itbook.net.cn> 或发邮件至 itmingjian@163.com 与我们交流和沟通。

编者



目 录

出版说明

前言

第 1 章 初识 Java	1	2.6 初学者编写 Java 程序应注意的问题	18
1.1 Java 语言的诞生和发展	2	2.7 本章实例	18
1.1.1 Java 语言的诞生	2	2.8 小结	20
1.1.2 Java 语言的发展	2	第 3 章 Java 语法基础	21
1.2 Java 语言的特点	2	3.1 编码规范	22
1.2.1 简单性	2	3.1.1 Java 命名规范	22
1.2.2 面向对象	2	3.1.2 代码编写规则	22
1.2.3 分布式	3	3.2 数据类型	22
1.2.4 解释型	3	3.2.1 整数数据类型	23
1.2.5 健壮安全性	3	3.2.2 浮点数据类型	23
1.2.6 体系结构中立	3	3.2.3 字符数据类型	24
1.2.7 可移植性	3	3.2.4 布尔数据类型	24
1.2.8 高性能	3	3.2.5 基本数据类型的应用	24
1.2.9 多线程	3	3.3 Java 的常量与变量	25
1.3 搭建 Java 语言的开发环境	3	3.3.1 常量	25
1.3.1 JDK 简介	4	3.3.2 变量	26
1.3.2 安装 JDK 6.0	4	3.3.3 变量的作用域	28
1.3.3 配置环境变量	6	3.4 Java 中的运算符	28
1.3.4 测试配置环境变量	7	3.4.1 赋值运算符	28
第 2 章 简单了解 Java 程序	9	3.4.2 算术运算符	29
2.1 认识 Java 程序	10	3.4.3 比较运算符	29
2.2 第一个 Java 程序	10	3.4.4 逻辑运算符	29
2.2.1 解释第一个 Java 程序	10	3.4.5 位运算符	30
2.2.2 在 EditPlus 3 中编写第一个 Java 程序	10	3.4.6 三元（目）运算符	31
2.2.3 在 DOS 窗口中编译与运行	11	3.4.7 自增和自减运算符	31
2.2.4 在 EditPlus 3 中编译与运行	13	3.4.8 表达式	33
2.3 Java 标识符	14	3.4.9 运算符优先级	33
2.4 Java 关键字	15	3.5 基本数据类型之间的转换	34
2.5 为 Java 程序添加注释	16	3.5.1 自动类型转换	34
2.5.1 单行注释	16	3.5.2 强制类型转换	35
2.5.2 多行注释	16	3.5.3 表达式的数据类型自动提升	36
2.5.3 文档注释	17	3.6 本章实例	36



3.7 小结	39	5.3.5 运行 Java 类	76
第 4 章 算法和流程控制语句	41	5.4 MyEclipse 6.0 资源管理	77
4.1 算法	42	5.4.1 导入与删除 Java 类	77
4.1.1 什么是算法	42	5.4.2 导入 Java 项目	77
4.1.2 算法的特性	42	5.4.3 导出 Java 项目	78
4.1.3 算法的描述	42	5.4.4 删除 Java 项目	79
4.1.4 程序的 3 种控制结构	45	5.5 用断点调试程序	80
4.2 分支语句	46	5.6 小结	81
4.2.1 简单的 if 条件语句	46	第 6 章 数 组	83
4.2.2 if...else 条件语句	47	6.1 数组的概念	84
4.2.3 if...else if 多分支语句	49	6.2 一维数组	84
4.2.4 switch 多分支语句	51	6.2.1 一维数组的声明与创建	84
4.2.5 if 语句和 switch 语句的区别	53	6.2.2 一维数组的初始化	85
4.3 循环语句	53	6.2.3 一维数组的访问	85
4.3.1 for 循环语句	54	6.2.4 一维数组的操作	86
4.3.2 while 循环语句	55	6.2.5 foreach 语句	88
4.3.3 do...while 循环语句	56	6.2.6 内存中的一维数组	89
4.3.4 循环的嵌套	58	6.2.7 一维数组的应用举例	89
4.3.5 各种循环语句的区别	59	6.3 二维数组	91
4.4 跳转语句	60	6.3.1 二维数组的声明与创建	91
4.4.1 break 跳转语句	60	6.3.2 二维数组的初始化	92
4.4.2 continue 跳转语句	61	6.3.3 二维数组的访问	92
4.4.3 return 跳转语句	62	6.3.4 内存中的二维数组	93
4.5 本章实例	63	6.3.5 二维数组的应用举例	94
4.6 小结	65	6.4 多维数组	95
第 5 章 开发工具 MyEclipse 6.0	67	6.4.1 多维数组的声明与创建	95
5.1 安装与启动 MyEclipse 6.0	68	6.4.2 多维数组的初始化	96
5.1.1 安装 MyEclipse 6.0	68	6.4.3 多维数组的访问	96
5.1.2 启动 MyEclipse 6.0	70	6.4.4 使用多维数组	96
5.2 MyEclipse 6.0 工作平台	71	6.4.5 内存中的三维数组	97
5.2.1 常用工具栏介绍	71	6.5 本章实例	97
5.2.2 视图	72	6.6 小结	100
5.2.3 透视图	72	第 7 章 字符串处理	101
5.2.4 编辑器	73	7.1 字符串	102
5.3 使用 MyEclipse 6.0	73	7.1.1 声明字符串	102
5.3.1 认识 Java 项目	73	7.1.2 创建字符串	102
5.3.2 新建 Java 项目	74	7.2 连接字符串	103
5.3.3 创建与编写一个 Java 类	75	7.2.1 字符串与字符串的连接	103
5.3.4 编译 Java 类	75	7.2.2 字符串与其他类型数据的	

连接	105	8.6 本章实例	147
7.3 字符串常用操作	106	8.7 小结	150
7.3.1 比较字符串是否相等	106	第9章 Java 的继承与多态	151
7.3.2 获取字符串的长度	106	9.1 继承	152
7.3.3 字符串的大小写转换	107	9.1.1 继承的概念与特点	152
7.3.4 查找字符串	108	9.1.2 访问父类的成员变量和方法	153
7.3.5 获取字符串的子字符串	110	9.1.3 子类的构造函数	154
7.3.6 替换字符串与除去字符串	111	9.1.4 子类的构造过程	155
中的空格	111	9.2 方法的重写和字段隐藏	157
7.3.7 分割字符串	112	9.2.1 方法的重写	157
7.4 格式化字符串	113	9.2.2 字段的隐藏	159
7.4.1 常规类型的格式化	113	9.3 多态	160
7.4.2 日期和时间字符串格式化	115	9.3.1 多态的含义	160
7.5 StringBuilder 类	119	9.3.2 抽象方法与抽象类	161
7.5.1 创建字符串生成器	119	9.4 final 关键字的使用	164
7.5.2 字符串生成器的应用	120	9.4.1 最终方法	164
7.6 本章实例	122	9.4.2 最终类	165
7.7 小结	125	9.5 本章实例	165
第8章 面向对象编程	127	9.6 小结	172
8.1 面向对象程序设计	128	第10章 类的高级应用	173
8.1.1 面向对象程序设计的概述	128	10.1 接口	174
8.1.2 面向对象程序设计的特点	129	10.1.1 接口的含义	174
8.2 类	129	10.1.2 创建接口	174
8.2.1 类的结构	129	10.1.3 接口的使用	175
8.2.2 成员变量	130	10.1.4 接口的继承	176
8.2.3 成员方法	130	10.1.5 接口的多态	177
8.2.4 访问修饰符	131	10.2 内部类	180
8.3 类的对象	132	10.2.1 在类中定义内部类	180
8.3.1 创建与使用对象	132	10.2.2 内部类的使用	181
8.3.2 实现类的封装	134	10.2.3 在方法中定义内部类	183
8.3.3 构造方法(函数)	136	10.3 包	184
8.3.4 this、static 关键字	138	10.3.1 包的描述	184
8.4 方法	141	10.3.2 创建包	184
8.4.1 方法的参数传递机制	141	10.3.3 包的使用	184
8.4.2 方法的重载	144	10.4 本章实例	188
8.5 垃圾回收	145	10.5 小结	191
8.5.1 垃圾回收原理	145	第11章 Java 中的异常处理	193
8.5.2 finalize()方法	146	11.1 异常处理概述	194
8.5.3 垃圾回收的本质	146	11.2 Java 常见异常及产生原因	194

11.2.1 空指针异常	194	13.3 File 类	250
11.2.2 没有找到具有指定名称的类	195	13.3.1 文件类的构造方法	250
11.2.3 算术异常	195	13.3.2 获取文件信息	251
11.2.4 数组越界异常	195	13.3.3 复制文件中的内容	253
11.2.5 非法参数异常	196	13.3.4 压缩文件	254
11.3 异常的处理机制	196	13.3.5 分行写入文件	255
11.3.1 使用 try...catch 捕获异常	197	13.4 本章实例	257
11.3.2 使用 throws 声明抛出异常	198	13.5 小结	261
11.3.3 自定义异常与 throw 关键字	199	第 14 章 线程	263
11.4 本章实例	201	14.1 线程概述	264
11.5 小结	209	14.1.1 线程的定义	264
第 12 章 泛型和集合	211	14.1.2 线程与进程	264
12.1 引出泛型的原因	212	14.2 线程的创建	264
12.2 理解泛型	213	14.2.1 通过继承 Thread 类来创建	264
12.2.1 泛型的定义 1	213	14.2.2 通过实现 Runnable 接口来	266
12.2.2 泛型的定义 2	216	14.3 线程的控制	268
12.2.3 泛型的定义 3	218	14.3.1 线程的让步	268
12.3 集合概述	220	14.3.2 线程优先级的改变	269
12.4 List 集合	220	14.4 线程的同步	270
12.4.1 List 集合的性质	220	14.4.1 线程同步所引发的安全	271
12.4.2 ArrayList 类的使用	221	14.4.2 线程的同步机制	272
12.4.3 使用泛型	223	14.5 线程之间的通信	274
12.5 Set 集合	225	14.5.1 监视器的概念	274
12.5.1 Set 集合的性质	225	14.5.2 一些基本的线程通信方法	274
12.5.2 HashSet 类的使用	226	14.6 本章实例	276
12.5.3 TreeSet 类的使用	228	14.7 小结	278
12.6 Map 集合	230	第 15 章 自动装箱、拆箱与反射	279
12.6.1 Map 集合的性质	230	15.1 自动装箱与拆箱	280
12.6.2 HashMap 类的使用	230	15.1.1 自动装箱	280
12.7 本章实例	232	15.1.2 拆箱	281
12.8 小结	239	15.2 装箱与拆箱所带来的问题	282
第 13 章 I/O 处理	241	15.2.1 “=” 的问题	282
13.1 Java I/O 体系结构	242	15.2.2 拆箱带来的异常问题	283
13.2 Java 提供的流类	242	15.3 反射	284
13.2.1 I/O 中的输入字节流	242	15.3.1 Java 反射机制概述	284
13.2.2 I/O 中的输出字节流	244	15.3.2 加载类	285
13.2.3 字节流与字符流	246	15.3.3 通过反射查看类信息	285
13.2.4 I/O 中的输入字符流	246		
13.2.5 I/O 中的输出字符流	248		



15.3.4 使用反射生成与操作对象	287
15.4 本章实例	289
15.5 小结	292
第 16 章 Java Web 应用程序开发	
环境	293
16.1 Tomcat 6.0 的使用	294
16.1.1 Tomcat 6.0 简介	294
16.1.2 安装与启动 Tomcat 6.0	294
16.1.3 在 MyEclipse 6.0 中配置与测试 Tomcat 6.0	296
16.2 Dreamweaver CS3 的使用	299
16.2.1 Dreamweaver CS3 简介	299
16.2.2 Dreamweaver CS3 安装	299
16.3 MySQL 5.1 数据库基础	301
16.3.1 MySQL 5.1 数据库简介	301
16.3.2 安装 MySQL 5.1	301
16.3.3 安装 MySQL Administrator 1.1 插件	306
16.3.4 安装 MySQL Query Browser 1.1 插件	308
16.4 MySQL 5.1 数据库的 基本操作	310
16.4.1 创建数据库	310
16.4.2 在数据库中创建表	311
16.4.3 插入数据	313
16.4.4 MySQL 5.1 数据库的备份	314
16.4.5 MySQL 5.1 数据库还原	315
第 17 章 Servlet 和 Web 应用 体系	317
17.1 C/S 模式与 B/S 模式	318
17.1.1 C/S 模式	318
17.1.2 B/S 模式	318
17.2 Http 请求和响应模式	319
17.2.1 超文件传输协议 Http	319
17.2.2 Http 请求	319
17.2.3 Http 响应	321
17.3 HTML 简介	322
17.4 Servlet 简介	324
17.4.1 什么是 Servlet	324
17.4.2 认识 Servlet	324
17.5 Web 应用体系	325
17.5.1 Servlet 容器的概念	326
17.5.2 编写一个简单的 Servlet	326
17.5.3 在 MyEclipse 6.0 中编写 Servlet	327
17.5.4 配置 web.xml	329
17.5.5 部署到 Tomcat 6.0 服务器上 运行	330
17.6 Servlet 高级应用	333
17.6.1 Servlet 生命周期	333
17.6.2 ServletContext 与 ServletConfig 对象	336
17.6.3 在 MyEclipse 中创建 JSP	339
17.6.4 Servlet 重定向	340
17.7 小结	344
第 18 章 学习 JSP	345
18.1 JSP 概述	346
18.1.1 JSP 简介	346
18.1.2 JSP 的优势	346
18.1.3 JSP 的处理过程	346
18.1.4 了解 JSP 页面基本结构	346
18.1.5 JSP 注释	351
18.2 HttpSession 接口	352
18.2.1 JSP Session 机制	352
18.2.2 HttpSession 的使用	353
18.3 监听器 Listener	353
18.3.1 什么是监听器	353
18.3.2 ServletContext 事件监听器	353
18.3.3 HttpSession 事件监听器	357
18.3.4 ServletRequest 的事件 监听器	361
18.4 过滤器	364
18.4.1 过滤器的原理	364
18.4.2 过滤器的创建	365
18.4.3 过滤器的顺序	367
18.5 小结	369
第 19 章 JSP 语法	371
19.1 Cookie 的应用	372

19.1.1 什么是 Cookie	372	20.3.3 SQL 声明类 Statement	413
19.1.2 Cookie 的好处	372	20.3.4 SQL 声明类 PreparedStatement	413
19.1.3 创建 Cookie	372	20.3.5 查询结果集类 ResultSet	414
19.2 JSP 的内置对象	372	20.4 连接 MySQL 数据库实例	414
19.2.1 request 对象	373	20.4.1 创建数据库连接实例	415
19.2.2 response 对象	374	20.4.2 操作 MySQL 数据库实例	416
19.2.3 out 对象	374	20.5 数据库连接池	421
19.2.4 session 对象	375	20.5.1 多用户数据库连接方法	421
19.2.5 application 对象	380	20.5.2 在 Tomcat 6.0 下配置 MySQL 5.1 连接池	422
19.2.6 page 和 pageContext 对象	381	20.5.3 测试连接池设置是否生效	422
19.2.7 exception 对象	381	20.6 小结	424
19.2.8 config 对象	382	第 21 章 远程教学系统的实现	425
19.3 JavaBean 简介	387	21.1 远程教学系统的系统分析	426
19.3.1 什么是 JavaBean	388	21.1.1 可行性分析	426
19.3.2 为什么使用 JavaBean	389	21.1.2 需求分析	426
19.4 JSP 动作元素	389	21.2 远程教学系统的总体设计	427
19.4.1 <jsp:param>动作	389	21.2.1 以教师身份登录的 功能模块	428
19.4.2 <jsp:include>动作	389	21.2.2 以学生身份登录的功能 模块	428
19.4.3 <jsp:forward>动作	390	21.2.3 以管理员身份登录的功能 模块	428
19.4.4 <jsp:useBean>动作	390	21.3 远程教学系统的数据库 设计	429
19.4.5 <jsp:setProperty>与 <jsp:getProperty>动作	391	21.3.1 远程教学系统的数据库的 E-R 图设计	429
19.5 EL 表达式	396	21.3.2 数据表设计	430
19.5.1 EL 表达式的形式	396	21.4 Java Web 应用配置文件 web.xml	431
19.5.2 EL 表达式功能	397	21.5 学生身份登录的模块设计 与实现	436
19.6 JSTL 基础知识	397	21.5.1 学生模块 JavaBean 的设计与 实现	436
19.6.1 JSTL 简介	397	21.5.2 学生模块操作数据库方法的 设计与实现	438
19.6.2 JSTL 的核心标签库	397	21.5.3 学生模块 Servlet 的设计与 实现	445
19.7 小结	405		
第 20 章 访问数据库	407		
20.1 JDBC 技术简介	408		
20.2 JDBC 连接方式	408		
20.2.1 JDBC 驱动介绍	408		
20.2.2 安装 MySQL 驱动	409		
20.2.3 JDBC 连接数据库的步骤	409		
20.3 JDBC 的 API	411		
20.3.1 驱动程序管理类 DriverManager	411		
20.3.2 数据库连接类 Connection	412		

21.5.4 学生模块主页面的设计与实现	448	23.1.3 数据字典分析	491
21.6 教师身份登录的模块设计与实现	450	23.2 总体设计	491
21.6.1 教师模块操作数据库方法的设计与实现	450	23.2.1 前台功能设计	491
21.6.2 教师模块 Servlet 的设计与实现	458	23.2.2 后台功能设计	491
21.7 管理员身份登录的后台模块设计与实现	461	23.2.3 系统的数据库设计	492
21.7.1 管理员模块数据库底层方法的设计与实现	461	23.3 系统架构设计	492
21.7.2 管理员模块 Servlet 的设计与实现	464	23.4 创建基于 Struts 框架的项目包	493
第 22 章 Struts 应用	469	23.5 创建数据库连接	494
22.1 Struts 简介	470	23.6 编写系统配置文件	495
22.2 为什么要用 Struts 框架	470	23.7 后台模块的设计与实现	497
22.3 Struts 的核心——MVC	470	23.7.1 标题及内容管理模块 JavaBean 的设计与实现	497
22.4 Struts 的工作原理	471	23.7.2 标题及内容管理模块 FormBean 的设计与实现	498
22.5 Struts 的工作流程	471	23.7.3 标题及内容管理模块底层方法的设计与实现	499
22.6 Struts 的基本配置	472	23.7.4 标题及内容管理模块 Action 的设计与实现	502
22.6.1 web.xml 配置文件	472	23.7.5 后台登录页面和主页面的设计与实现	504
22.6.2 struts-config.xml 配置文件	473	23.8 前台用户投票模块的设计与实现	506
22.7 表单处理器 ActionForm	475	23.8.1 用户投票模块底层方法的设计与实现	506
22.7.1 ActionForm 类	475	23.8.2 用户投票模块 Action 的设计与实现	507
22.7.2 编写 FormBean	477	23.8.3 用户投票模块的页面设计	508
22.8 Struts 中的控制器 Action	478	第 24 章 Hibernate 应用	513
22.8.1 Action 类	478	24.1 Hibernate 简介	514
22.8.2 Action 的代码实现	478	24.2 Hibernate 框架概述	514
22.9 Struts 标签库	479	24.2.1 Hibernate 的优势	514
22.9.1 导入 Struts 的标签库	479	24.2.2 Hibernate 架构	514
22.9.2 Struts 的 HTML 标签	479	24.3 Hibernate 框架主要的 API	514
22.9.3 Struts 的 Bean 标签	483	24.3.1 Configuration 类	515
22.9.4 Struts 的 Logic 标签	485	24.3.2 SessionFactory 接口	515
22.10 小结	487	24.3.3 Session 接口	516
第 23 章 在线投票系统的实现	489	24.3.4 Query 接口	517
23.1 需求分析	490	24.3.5 Transaction 接口	517
23.1.1 用例分析	490		
23.1.2 数据流分析	490		



24.4 Hibernate 的持久化技术	517
24.4.1 什么是持久化	517
24.4.2 持久化的意义	517
24.4.3 持久化类	517
24.4.4 Hibernate 中的对象状态	518
24.5 Hibernate 框架中的 ORM 技术	519
24.5.1 持久化类	519
24.5.2 映射文件	519
24.5.3 Hibernate 的配置文件	520
24.6 Hibernate 框架查询语言	521
24.6.1 HQL 中的 from 子句	521
24.6.2 HQL 中的属性查询	521
24.6.3 HQL 中的更新和删除	522
24.6.4 HQL 中的 order by 子句	522
24.6.5 HQL 中的 group by 子句	522
24.6.6 HQL 中的连接	523
24.7 HQL 中的参数绑定	523
24.7.1 按参数名称绑定	523
24.7.2 按参数位置绑定	523
24.7.3 使用 setParameter() 方法来 绑定参数	524
24.7.4 使用 setProperties() 方法来 绑定参数	524
24.8 Hibernate 反向工程	524
24.8.1 在 MyEclipse 6.0 中创建 数据库连接	524
24.8.2 在 Java Web 项目中加入 Hibernate 架包	526
24.8.3 在 MyEclipse 6.0 中生成 代码	528
24.9 小结	529
第 25 章 制作精美的留言板	531
25.1 需求分析	532
25.1.1 用例分析	532
25.1.2 数据流分析	532
25.1.3 数据字典分析	533
25.2 详细设计	533
25.2.1 前台功能设计	533
25.2.2 后台功能设计	533
25.2.3 系统的类图设计	534
25.2.4 系统的数据库设计	534
25.3 系统总体设计	535
25.3.1 系统的设计思想	535
25.3.2 系统中的过滤器	535
25.3.3 后台管理留言模块设计	536
25.4 创建数据库连接	537
25.4.1 系统 O/R Mapping 映射文件的 设计与实现	537
25.4.2 Hibernate.cfg.xml 的配置	539
25.5 系统 struts-config.xml 配置 文件	540
25.6 系统 web.xml 设计与实现	542
25.7 系统 JavaBean 设计与实现	543
25.8 后台模块的设计与实现	546
25.8.1 管理员登录模块的设计与 实现	547
25.8.2 回复留言模块的设计与 实现	551
25.8.3 查询留言模块的设计与 实现	554
25.8.4 删除留言模块的设计与 实现	559
25.9 前台模块的设计与实现	564
25.9.1 用户留言模块 FormBean 的 设计与实现	564
25.9.2 用户留言模块底层方法的设计 与实现	565
25.9.3 用户留言模块 Action 的设计 与实现	566
25.9.4 用户留言模块的页面设计与 实现	567

第1章

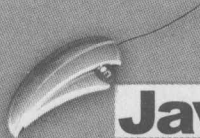
初识 Java

知识要点

- Java 语言的诞生和发展
- Java 语言的特点
- 搭建 Java 语言的开发环境

本章主要内容

本章主要介绍了 Java 语言的诞生和发展、Java 语言的特点和搭建 Java 语言的开发环境，通过本章的学习，读者可以大致了解 Java 语言和其环境，为以后学习 Java 语言打下基础。



1.1 Java 语言的诞生和发展

Java 语言是目前最流行的编程语言之一，下面将分别从 Java 语言的诞生和发展两个方面进行详细介绍。

1.1.1 Java 语言的诞生

Java 语言诞生于 1991 年，起初被称为 Oak 语言，是由 Sun 公司开发的。Java 语言的诞生主要得益于家用电器的芯片设计。最初其开发者想用 C++ 语言来开发家用电器的芯片，但由于芯片种类太多，程序需要进行多次编译，而且 C++ 语言中的指针一旦操作不慎，就会出现错误，因此其开发者去掉了 C++ 语言中的指针操作、运算符重载和多重继承等便得到了 Java 语言。

1.1.2 Java 语言的发展

1995 年，Sun 公司正式向 IT 业界推出了 Java 语言，该语言具有安全、跨平台、面向对象、简单和适用于网络等显著特点，当时以 Web 为主要形式的互联网技术正在迅猛发展，Java 语言的出现迅速引起了程序和软件公司的极大关注，程序员们纷纷尝试使用 Java 语言编写网络应用程序，并利用网络把程序发布到世界各地，IBM、Oracle、微软、Apple、Netscape 和 SGI 等大公司纷纷与 Sun 公司签订合同，获得使用 Java 平台技术的授权。

Java 平台包括三个版本，分别是 Java SE (J2SE)、Java EE (J2EE) 和 Java ME (J2ME)。Java 平台由 JVM (Java Virtual Machine) 和 Java API (Application Programming Interface) 构成，在操作系统中安装了 Java 平台之后，Java 应用程序即可运行，而目前 Java 平台已经被嵌入到了几乎所有的操作系统中，因此 Java 应用程序编译一次即可运行在各种操作系统中。

1.2 Java 语言的特点

Java 语言具有简单、面向对象、分布式、解释型、健壮安全、结构中立、可移植、高性能和多线程 9 大特点。

1.2.1 简单性

Java 语言摒弃了 C++ 语言中的一些难于理解和令人迷惑的特性，如操作符重载、多重继承和自动强制类型转换以及指针操作，并且增加了自动垃圾回收功能，这样使得程序员不必再担忧繁琐的指针操作和复杂的内存管理。

1.2.2 面向对象

面向对象设计是一种以对象为中心的程序设计思想，在 Java 语言中，提供了类、对象、接口和继承等概念，类是对象的模板，对象封装了类的属性和动作，通过不同的属性和动作即可描述出现实世界中的具体事物。

1.2.3 分布式

Java 是面向网络的语言，因此 Java 语言支持 Internet 应用的开发，提供了用于网络应用编程的类库，包括 URL、URLConnection、Socket 和 ServerSocket 等。Java 的 RMI(远程方法激活)机制也是开发分布式应用的重要手段。

1.2.4 解释型

Java 程序在 Java 平台上被编译为字节码格式，然后可以在实现这个 Java 平台的任何系统中运行。在运行时，Java 平台中的 Java 解释器对这些字节码进行解释执行，执行过程中需要的类在连接阶段被载入到运行环境中。

1.2.5 健壮安全性

Java 的类型机制、垃圾自动回收和异常处理等都是 Java 程序健壮性的重要保证。Java 语言的安全性主要是指在 Java 语言中摒弃了指针操作，这样便不需要程序员来操作内存，而由 JVM 来管理，因此不会造成忘记释放内存而带来的麻烦。

1.2.6 体系结构中立

Java 编译器读取 Java 源程序后，生成与体系结构无关的字节码指令，这些指令字节码根据浏览器的要求被发送到客户端的机器上，只要在客户端的机器上安装了 Java 运行时系统即可运行 Java 程序，而且还可以运行在不同的平台中。

1.2.7 可移植性

Java 的可移植性即是与平台无关性，由于 Java 语言中的类库实现了与不同平台的接口，因此 Java 程序可以被移植到网络上的不同机器中。另外，Java 编译器是用 Java 语言实现的，而 Java 运行时系统是用 C 语言实现的，这使得 Java 系统本身也具有可移植性。

1.2.8 高性能

在 Java 语言中，字节码的设计使目标代码能直接转换成对应于特定 CPU 的机器码，如果解释器速度不慢，Java 可以在运行时直接将目标代码翻译成机器指令，从而获得较高的性能。

1.2.9 多线程

在 Java 语言中，支持多线程的同时提供多线程之间的同步机制（关键字为 synchronized）来确保对共享数据的正确操作。

1.3 搭建 Java 语言的开发环境

搭建 Java 语言的开发环境是学习 Java 语言的基础，本节主要讲解 JDK 简介、安装 JDK 6.0、配置环境变量和测试配置环境变量，下面将详细介绍。