



高等职业教育“十三五”规划教材（网络工程课程群）

Java

程序设计

主 编 谢先伟 梅青平
副主编 杨 娟 万 青 毛 红



中国水利水电出版社
www.waterpub.com.cn

高等职业教育“十三五”规划教材（网络工程课程群）

Java 程序设计

主 编 谢先伟 梅青平

副主编 杨 娟 万 青 毛 红



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书全面介绍了Java语言的发展、Java基础知识、面向对象程序设计、集合、文件与输入输出流、图形图像处理、JDBC数据库技术、多线程、网络编程等内容,在基于工作过程的基础上,打破传统的教程模式,采用任务驱动的方式,在每一个学习情境中设计了一个源自真实或接近真实的工作过程的学习任务。学习任务在相关知识的支撑下,详细介绍了案例分析、解决步骤、能力拓展等相关环节。本书内容丰富、重点突出、简明易懂,采用任务驱动的形式设计,图文并茂,循序渐进,并包含丰富的能力拓展练习内容,具有很强的实用性。本书可作为高职高专院校和应用型本科院校的计算机及网络相关专业学生学习Java程序设计的教材和参考书。

本书配有电子教案和案例源码,读者可以从中国水利水电出版社网站和万水书苑免费下载,网址为:<http://www.waterpub.com.cn/softdown/>和<http://www.wsbookshow.com>。

图书在版编目(CIP)数据

Java程序设计 / 谢先伟, 梅青平主编. — 北京 :
中国水利水电出版社, 2016. 8
高等职业教育“十三五”规划教材. 网络工程课程群
ISBN 978-7-5170-4546-5

I. ①J… II. ①谢… ②梅… III. ①JAVA语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第162426号

策划编辑: 祝智敏

责任编辑: 陈 洁

封面设计: 李 佳

书 名	高等职业教育“十三五”规划教材(网络工程课程群) Java 程序设计
作 者	主 编 谢先伟 梅青平 副主编 杨 娟 万 青 毛 红
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网 址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电 话: (010) 68367658 (发行部)、82562819 (万水) 北京科水图书销售中心(零售) 电 话: (010) 88383994、63202643、68545874 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	铭浩彩色印装有限公司
规 格	184mm×260mm 16开本 28.25印张 615千字
版 次	2016年8月第1版 2016年8月第1次印刷
印 数	0001—3000册
定 价	68.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社发行部负责调换

版权所有·侵权必究

丛书编委会

主 任：杨智勇 李建华

副主任：王璐烽 武春岭 乐明于 任德齐 邓 荣

黎红星 胡方霞

委 员：万 青 王 敏 邓长春 冉 婧 刘 宇

刘 均 刘海舒 刘 通 杨 垠 杨 娟

杨 毅 吴伯柱 吴 迪 张 坤 罗元成

罗荣志 罗 勇 罗脂刚 周 桐 单光庆

施泽全 宣翠仙 唐礼飞 唐 宏 唐 林

唐继勇 陶洪建 麻 灵 童 杰 曾 鹏

谢先伟 谢雪晴

前言

与传统程序不同，Sun 公司（2010 年被 Oracle 公司收购）在推出 Java 之际就将其作为一种开放的技术，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。在最受欢迎的程序设计语言排行榜上，Java 语言已经连续数年位列榜首。“Write once, Run anywhere（一次编写，到处可行）”，这是一种很有效率的编程方式。跨平台、完全面向对象、既适合于单机编程也适合于 Internet 编程等特点，给 Java 语言注入了强大的生命力。而 Java 语言也取得了举世瞩目、全球公认的地位。

本书从教学实际和市场对 Java 人才的需求出发，合理安排知识结构，由浅入深，循序渐进，每个章节都用生动的案例引领，便于提高学生的学习趣味和动手实践能力，旨在拉近高等院校在软件人才培养和软件公司在人才需求上的距离，缩小差距。

1. 本书的特点与优势

（1）由浅入深，结构清晰。本书内容以学生为第一视角，本着由浅入深、循序渐进的原则和先易后难的规律合理安排各个章节，便于学生接收所需知识，符合学习规律和教学规律，学生上手快，老师易教学。

（2）理论联系实际，注重实践能力的提升。本书在教学方法上采用的是案例驱动与综合实训相结合的方式，每个章节是以案例引领，引出知识点，还配有案例和项目，并对知识点进行拓展训练，有一个完整的基于任务的认知过程。最后一章提供了一个综合案例，并配有详细的介绍和步骤演示，便于指导学生实践，提高学生的动手能力。

（3）学以致用，注重能力。以“案例驱动—实用知识点—任务实施—拓展训练”为主线进行编写，注重能力的培养，重视动手能力的培养，达到学以致用的目的。

（4）提供立体化教学资源。本书除了提供教材外，还提供了下载的教学所用的 PPT 课件、课程案例代码等资源，方便老师授课和学生学习。

2. 本书内容

绪论：要求了解 Java 的诞生和发展历史，掌握 JDK 及 Eclipse 的安装和部署。

第 1 章 Java 编程基础：要求掌握常用的 8 种基本数据类型、运算符和表达式、if 语句和 switch 语句、3 种循环语句、2 个流程跳转语句、一维数组和二维数组。

第 2 章类和对象：要求掌握面向对象编程中类的概念和特征，包括类的定义和对象的生成、封装的概念和实现、构造方法的重载、this 关键字、继承的实现、super 关键字、多态。

第 3 章抽象类、接口和包：要求掌握抽象类的概念、抽象方法的定义、抽象类与抽象方法的关系、子类继承抽象类并实现抽象方法、接口的概念、子类实现接口、抽象类与接口的联系与区别、包的概念和定义、包的导入、Java 权限修饰符。

第 4 章常用类：要求掌握常用类，特别是字符串（String、StringBuffer、StringBuilder）类、基本类型封装类、装箱和拆箱、自动装箱和拆箱、数学类 Math、日

期类 (Date、Calendar) 的使用。

第5章集合: 要求掌握 Java 集合框架、List 接口、泛型、Iterator (迭代器) 接口、Set 接口、Map 接口、LinkedHashMap 类、Collections 类、Comparable 与 Comparator 接口。

第6章异常: 要求掌握 Java 语言异常的概念和类型、异常处理机制、异常的抛出和捕获。

第7章 JDBC: 要求掌握 JDBC 技术基础知识和标准 API、数据库连接、数据库访问和数据的增删改查操作、DAO 模式。

第8章 GUI: 要求掌握 GUI 容器、GUI 组件、常用布局、常用控件、GUI 事件处理模型、监听机制和常用事件。

第9章文件和输入输出流: 要求掌握 Java 文件操作系统、字节流、字符流、转换流等。

第10章多线程: 要求掌握进程和线程的概念、多线程的优缺点、线程的创建方法、线程的状态和常用方法、线程安全问题、线程的同步和死锁、生产者 and 消费者模型。

第11章网络编程: 要求掌握网络模型、IP 地址和端口、UDP 和 TCP 通信协议、基于 UDP 的 Socket 通信、基于 TCP 的 Socket 通信、URL、URLConnection 等。

第12章项目——图书管理系统 V1.0: 要求掌握 GUI 布局、GUI 常用控件、事件处理机制、数据库的创建、表的创建、数据的增删改查等。

3. 读者定位

本书是 Java 语言的入门级教材, 适合于 Java 语言的初学者和零编程经验的读者, 如果读者具备 C 语言的基础知识更佳。

本书主要面向高等职业院校, 既可作为大中专院校的 Java 程序设计与开发的教材, 也可作为读者自学的参考书。

本书的作者团队由经验丰富的一线骨干教师组成, 不但教学经验丰富, 而且参与了大量的 Java 项目开发, 实战经验丰富。在长期的 Java 教学中, 将项目开发的经验融入教学中。

本书由谢先伟、梅青平任主编, 杨娟、万青、毛红任副主编。其中, 绪论和第3、10、11、12章由谢先伟编写; 第2、4、5、9章由梅青平、万青编写; 第1、6、7、8章由万青、杨娟编写, 由谢先伟进行统一审稿。另外, 参与本书部分编写工作的还有: 邵亮、唐林、唐丽均、李崇、刘宇、任亮、李新、周桐、李青野、孙小恒、邱雷等。此外在编写过程中重庆工程职业技术学院吴再生副校长、重庆工程职业技术学院信息工程学院院长杨智勇和书记李建华给予了大力支持和关心。本书在编写过程中还参考了“传智播客”讲师毕向东老师的讲课案例, 在此谨表示感谢。

由于编者水平有限, 书中若有不当之处, 敬请读者指正。

编者

2016年4月

C 目录

CONTENTS

前言

绪论

【主要内容】	001
【学习目标】	001

*.1 认识 Java 语言..... 002

* .1.1 项目 (1) 描述	002
* .1.2 项目知识准备	002
* .1.3 项目实施	012
* .1.4 能力拓展	015

*.2 Eclipse 下 Java 程序开发 017

* .2.1 项目 (2) 描述	017
* .2.2 项目知识准备	017
* .2.3 项目实施	023
* .2.4 能力拓展	026

第 1 章 Java 基础编程

【主要内容】	027
【学习目标】	027

1.1 变量、运算符和表达式 028

1.1.1 项目 (1-1) 描述	028
1.1.2 项目知识准备	028
1.1.3 项目实施	038
1.1.4 能力拓展	039

1.2 分支流程 039

1.2.1 项目 (1-2) 描述	039
1.2.2 项目知识准备	040
1.2.3 项目实施	044
1.2.4 能力拓展	046

1.3 循环流程 046

1.3.1 项目 (1-3) 描述	046
1.3.2 项目知识准备	046
1.3.3 项目实施	053
1.3.4 能力拓展	054

1.4 数组 054

1.4.1 项目 (1-4) 描述	054
1.4.2 项目知识准备	054
1.4.3 项目实施	056
1.4.4 能力拓展	058

第 2 章 类和对象

【主要内容】	059
【学习目标】	059

2.1 类和对象 060

2.1.1 项目 (2-1) 描述	060
2.1.2 项目知识准备	060
2.1.3 项目实施	064
2.1.4 能力拓展	065

2.2 封装、继承和多态 066

2.2.1 项目 (2-2) 描述	066
2.2.2 项目知识准备	066
2.2.3 项目实施	077
2.2.4 能力拓展	079

第 3 章 抽象类、接口和包

【主要内容】	081
【学习目标】	081

3.1 抽象类 082

3.1.1 项目 (3-1) 描述	082
3.1.2 项目知识准备	082
3.1.3 项目实施	085
3.1.4 能力拓展	087

3.2 接口 087

3.2.1 项目 (3-2) 描述	087
3.2.2 项目知识准备	087
3.2.3 项目实施	096
3.2.4 能力拓展	098

3.3 包 100

3.3.1 项目 (3-3) 描述	100
3.3.2 项目知识准备	100
3.3.3 项目实施	104
3.3.4 能力拓展	106

第 4 章 常用类

【主要内容】	107
【学习目标】	107

4.1 字符串类 108

4.1.1 项目 (4-1) 描述	108
4.1.2 项目知识准备	108
4.1.3 项目实施	112

4.1.4 能力拓展	114
4.2 Math 类	115
4.2.1 项目 (4-2) 描述	115
4.2.2 项目知识准备	115
4.2.3 项目实施	116
4.2.4 能力拓展	118
4.3 数据类型转换	119
4.3.1 项目 (4-3) 描述	119
4.3.2 项目知识准备	119
4.3.3 项目实施	124
4.3.4 能力拓展	126

第5章 集合

【主要内容】	128
【学习目标】	128

5.1 List 接口	129
5.1.1 项目 (5-1) 描述	129
5.1.2 项目知识准备	129
5.1.3 项目实施	139
5.1.4 能力拓展	141
5.2 Set 和 Map 接口	142
5.2.1 项目 (5-2) 描述	142
5.2.2 项目知识准备	142
5.2.3 项目实施	146
5.2.4 能力拓展	148

第6章 异常

【主要内容】	150
【学习目标】	150

6.1 异常概述	151
6.1.1 项目 (6-1) 描述	151
6.1.2 项目知识准备	151
6.1.3 项目实施	153
6.1.4 能力拓展	154
6.2 异常处理	155
6.2.1 项目 (6-2) 描述	155
6.2.2 项目知识准备	155
6.2.3 项目实施	161
6.2.4 能力拓展	161

第7章 JDBC

【主要内容】	162
【学习目标】	162

7.1 JDBC 连接	163
7.1.1 项目 (7-1) 描述	163
7.1.2 项目知识准备	163
7.1.3 项目实施	166
7.1.4 能力拓展	167

7.2 数据库的增删改查	167
7.2.1 项目 (7-2) 描述	167
7.2.2 项目知识准备	168
7.2.3 项目实施	170
7.2.4 能力拓展	173

7.3 数据库操作的分层处理	173
7.3.1 项目 (7-3) 描述	173
7.3.2 项目知识准备	173
7.3.3 项目实施	180
7.3.4 能力拓展	181

第8章 GUI

【主要内容】	182
【学习目标】	182

8.1 组件	183
8.1.1 项目 (8-1) 描述	183
8.1.2 项目知识准备	183
8.1.3 项目实施	202
8.1.4 能力拓展	204
8.2 布局管理器	205
8.2.1 项目 (8-2) 描述	205
8.2.2 项目知识准备	205
8.2.3 项目实施	211
8.2.4 能力拓展	213
8.3 事件处理	214
8.3.1 项目 (8-3) 描述	214
8.3.2 项目知识准备	214
8.3.3 项目实施	224
8.3.4 能力拓展	227

第9章 文件和输入输出流

【主要内容】	229
【学习目标】	229

9.1 文件	230
9.1.1 项目 (9-1) 描述	230
9.1.2 项目知识准备	230
9.1.3 项目实施	233
9.1.4 能力拓展	235
9.2 字节流	235
9.2.1 项目 (9-2) 描述	235
9.2.2 项目知识准备	235
9.2.3 项目实施	238
9.2.4 能力拓展	239
9.3 字符流	240
9.3.1 项目 (9-3) 描述	240
9.3.2 项目知识准备	240
9.3.3 项目实施	245
9.3.4 能力拓展	246

第 10 章 多线程

【主要内容】	248
【学习目标】	248

10.1 进程与线程

10.2 线程的实现

10.2.1 项目 (10-1) 描述	250
10.2.2 项目知识准备	250
10.2.3 项目实施	260
10.2.4 能力拓展	262

10.3 线程的状态与线程常用方法 ...

10.3.1 项目 (10-2) 描述	263
10.3.2 项目知识准备	263
10.3.3 项目实施	277
10.3.4 能力拓展	279

10.4 同步与死锁

10.4.1 项目 (10-3) 描述	280
10.4.2 项目知识准备	280
10.4.3 项目实施	286
10.4.4 能力拓展	289

10.5 生产者与消费者模型

10.5.1 项目 (10-4) 描述	290
10.5.2 项目知识准备	290
10.5.3 项目实施	298
10.5.4 能力拓展	302

第 11 章 网络编程

【主要内容】	303
【学习目标】	303

11.1 网络编程的基本概念

11.1.1 项目 (11-1) 描述	304
11.1.2 项目知识准备	304
11.1.3 项目实施	308
11.1.4 能力拓展	309

11.2 UDP 程序设计

11.2.1 项目 (11-2) 描述	310
11.2.2 项目知识准备	310
11.2.3 项目实施	318
11.2.4 能力拓展	321

11.3 TCP 程序设计

11.3.1 项目 (11-3) 描述	321
11.3.2 项目知识准备	321
11.3.3 项目实施	333
11.3.4 能力拓展	336

11.4 URL 与 URLConnection

11.4.1 项目 (11-4) 描述	337
11.4.2 项目知识准备	337
11.4.3 项目实施	340
11.4.4 能力拓展	342

第 12 章 项目——图书管理系统 V1.0

【主要内容】	343
【学习目标】	343

12.1 项目概述

12.2 系统设计

12.2.1 总体结构设计	345
12.2.2 构建开发环境	346
12.2.3 项目工程目录	346

12.3 项目实施

12.3.1 项目目录搭建及导包	347
12.3.2 数据库的设计与创建	348
12.3.3 公共模块设计	351
12.3.4 登录模块的实现	364
12.3.5 主界面模块的实现	369
12.3.6 基础数据维护模块的实现	377
12.3.7 借阅管理模块的实现	411
12.3.8 查询管理模块的实现	426
12.3.9 系统维护模块的实现	433
12.3.10 小结	438

附录 A 常用字符与 ASCII 代码对照表 ...

附录 B Java 语言中的关键字

附录 C 运算符和结合性

附录 D 类 Component 的常用方法 ...

附录 E Eclipse 的常用快捷键

参考文献

绪论

【主要内容】

Java 语言的发展；
Java 语言的特点；
Java 程序的工作过程；
Java 程序的类型；
JDK 的安装与配置；
第一个简单的 Java 程序；
Eclipse 的安装与配置；
使用 Eclipse 开发简单的程序。

【学习目标】

能了解 Java 语言的发展历程；
能理解 Java 语言的特点；
能正确理解 Java 程序的工作过程；
能理解 Java 程序的分类，以及两类程序的区别；
能正确安装和配置 JDK；
能正确安装和使用 Eclipse 集成开发环境；
能编写、编译和运行简单的程序。

*.1 认识 Java 语言

*.1.1 项目 (1) 描述

通过 Java 语言编写一个控制台应用程序，通过 Java 编译器进行编译，运行后在控制台输出“Hello World”。

*.1.2 项目知识准备

(1) Java 语言的诞生

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言，它是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台（即 Java SE、Java EE、Java ME）的总称。

Java 语言的前身被命名为 Oak，第一版 Oak 经历了 18 个月的开发时间，并于 1992 年问世，目标定位是作为家用电器等小型系统的编程语言，用以解决诸如冰箱、洗衣机等家用电器的控制和通信问题。由于在之后的应用过程中发现智能化家电的市场需求没有预期的高，致使 Sun 决定放弃该项计划。就在 Oak 几近失败之时，互联网应用的发展给该语言提供了新的发展契机，Sun 看到了 Oak 在计算机网络上的广阔应用前景，于是改造了 Oak，以“Java”的名称正式发布。

(2) Java 语言的发展和前景

1995 年 5 月 23 日，SunWorld 大会上 Java 和 HotJava 浏览器的第一次公开发布标志着 Java 语言正式诞生。

1996 年 1 月 23 日，Java 1.0 正式发布，第一个 JDK（Java Development Kit，Java 开发工具包）——JDK 1.0 诞生。JDK 是整个 Java 的核心，包括 Java 运行环境、Java 工具和 Java 基础类库。各大知名公司纷纷向 Sun 公司申请 Java 的许可。一时间，Netscape、惠普、IBM、Oracle、Sybase 甚至当时刚推出 Windows 95 的微软都是 Java 的追随者。与此同时，Java 这门新生的语言就拥有了自己的会议——JavaOne。

1997 年 2 月 18 日，JDK 1.1 发布。之后的一年内，下载次数超过 2,000,000 次。

1997 年 4 月 2 日，JavaOne 会议召开，参与者超过一万人，创当时全球同类会议规模之纪录。同年度，JavaDeveloperConnection 社区成员超过十万。

1998 年 12 月 8 日，Java 2 平台正式发布。

1999 年 6 月，Sun 公司发布 Java 的三个版本：标准版（J2SE）、企业版（J2EE）和微型版（J2ME）。以上三个版本构成了 Java 2，它是 Sun 意识到“one size doesn't fit all”之后，把最初的 Java 技术打包成 3 个版本的产物。

2000 年 5 月 8 日，JDK 1.3 发布。

2000 年 5 月 29 日，JDK 1.4 发布。

2001年9月24日, J2EE 1.3 发布。

2002年2月26日, J2SE 1.4 发布, 自此 Java 的计算能力有了大幅提升。

2004年9月30日, J2SE 1.5 发布, 成为 Java 语言发展史上的又一里程碑。为了表示该版本的重要性, J2SE 1.5 更名为 Java SE 5.0。在 Java SE 5.0 版本中, Java 引入了泛型编程 (Generic Programming)、类型安全的枚举、不定长参数和自动装 / 拆箱等语言特性。

2005年6月, JavaOne 大会召开, Sun 公司公开 Java SE 6。此时, Java 的各种版本已经更名, 以取消其中的数字“2”: J2EE 更名为 Java EE, J2SE 更名为 Java SE, J2ME 更名为 Java ME。

2006年12月, Sun 公司发布 JRE 6.0。

2009年4月20日, Oracle 公司以 74 亿美元收购 Sun 公司, 取得了 Java 的版权。

2010年9月, JDK 7.0 发布, 增加了简单闭包功能。

2011年7月, 甲骨文公司发布了 Java 7 的正式版。

2014年3月, 甲骨文公司发布了 Java 的最新版本为 1.8, Oracle 官方称为 Java 8。

Java 平台由 Java 虚拟机和 Java 应用编程接口构成。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。在硬件或操作系统平台上安装一个 Java 平台之后, Java 应用程序就可运行。现在 Java 平台已经嵌入了几乎所有的操作系统。这样 Java 程序只编译一次, 就可以在各种系统中运行。

Java 分为三个体系, 即 Java 平台标准版、Java 平台企业版和 Java 平台微型版。

Java 技术具有卓越的通用性、高效性、平台移植性和安全性, 广泛应用于个人计算机、数据中心、游戏控制台、科学超级计算机、移动电话和互联网, 同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下, Java 更具备了显著的优势和广阔的发展前景。

(3) Java 语言的特点

Java 语言是当前最流行的网络编程语言, 与 C++ 语言相比有着特殊的优点, 如简单性、平台无关性、面向对象、分布式、安全性、支持多线程、健壮性、动态性等。

1) 简单性

与 C++ 相比, Java 不再支持运算符重载、多级继承及广泛的自动强制等易混淆和极少使用的特性, 而增加了内存空间自动垃圾收集的功能。复杂性的省略和实用功能的增加使得程序开发变得简单而又可靠。

2) 平台无关性

平台无关性是 Java 最吸引人的地方。Java 是一种网络语言, 而网络上充满了各种不同类型的机器和操作系统。Java 采用了解释执行而不是编译执行的运行环境, 首先编译成字节码, 然后装载与校验, 再解释成不同的机器码来执行, 即“Java 虚拟机”的思想, 屏蔽了具体的平台环境的要求。

3) 面向对象

面向对象的技术具有继承性、封装性和多态性等多种优点, Java 在保留这些优点的基础上, 又具有动态编程的特性, 更能发挥出面向对象的优势。

4) 分布式

Java 建立在扩展 TCP/IP 网络平台上。库函数提供了用 HTTP 和 FTP 协议传送和接受信息的方法。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。

5) 安全性

作为网络开发语言,Java 有建立在公共密钥技术基础上的确认技术,提供了足够的安全保障。Java 在运行应用程序时,严格检查其访问数据的权限,如不允许网络上的应用程序修改本地的数据。同时,Java 程序运行稳定,轻易不会出现死机现象。

6) 支持多线程

多线程机制使应用程序能同时进行不同的操作,处理不同的事件。Java 有一套成熟的同步语言,保证了对共享数据的正确操作。通过使用多线程,程序设计者可以分别用不同的线程来完成特定的任务。

7) 健壮性

健壮性反映出程序的可靠性。Java 的几个内置的特性使程序的可靠性得到改进。

① Java 是强类型语言。编译器和类载入器保证所有方法调用的正确性,防止隐式类版本的不兼容性。

② Java 没有指针,不能引用内存指针,避免了内存或数组越界访问。

③ Java 进行自动内存回收,编程人员无法意外释放内存,不需要判断应该在何处释放内存。

④ Java 在编译和运行时,都要对可能出现的问题进行检查,以消除错误的产生。另外,在编译的时候,还可揭示出可能出现但尚未被处理的异常,以防止系统的崩溃。

8) 动态性

Java 在类库中可以自由地加入新方法和实例变量,而不影响用户程序的执行;同时,Java 通过接口来支持多重继承,使其具有更灵活的方式和扩展性。

Java 语言除了具有上述主要特点外,还具有高性能、解释性和可移植性等特点。

(4) Java 程序的分类

根据程序结构和运行环境的不同,Java 程序可以分为两类:Java 应用程序 (Java Application) 和 Java 小应用程序 (Java Applet)。

应用程序以 main() 方法作为程序入口,由 Java 解释器加载执行。Java 应用程序是完整的程序,能够独立运行。

而 Java Applet 小应用程序不使用 main() 方法作为程序入口,需要嵌入到 HTML 网页中运行,由 Appletviewer 或其他支持 Java 的浏览器加载执行,不能独立运行。

无论是哪种 Java 源程序,都用扩展名为“.java”的文件来保存。

1) Java Application 程序

Java Application 程序是可独立运行的 JVM 程序,它由一个或多个类组成,其中必须有一个类中定义了 main() 方法,main() 方法就像 C 语言的 main() 方法一样,是 Java Application 程序运行的入口。

编写和运行 Java Application 程序需要按下列步骤进行。

① 创建一个 Java Application 源程序（.java 文件）。

创建一个名为 HelloWorldApp.java 的文件，可在任何字符编辑器上输入并保存下列 Java 源程序代码：

```
public class HelloWorldApp {  
    public static void main(String args[]){  
        System.out.println("Hello Java!"); // 在控制台上输出字符串 "Hello Java!"  
    } //main 方法结束  
} //class 定义结束
```

其实质是创建一个名为 HelloWorldApp 的类，并把它保存在与它相同名字的文件中（即 HelloWorldApp.java 文件）。

一个 Java 源程序是由若干个类组成的，本程序中只有一个类。class 是 Java 的关键字，用来定义类。public 也是关键字，用来声明一个类是公共类。

源文件的命名规则如下：如果一个 Java 源程序中有多个类，那么只能有一个类是 public 类；如果有一个类是 public 类，那么 Java 源程序的名字必须与这个类的名字完全相同，扩展名是“.java”；如果源文件中没有 public 类，那么源文件的名字只要与某个类的名字相同，而且扩展名是“.java”就可以了。

应用程序的入口是 main() 方法，它有固定的书写方式：

```
public static void main(String args[]) {  
    .....  
}
```

main() 方法之后的大括号及括号内的内容叫作方法体。一个 Java 应用程序必须有且仅有一个类含有 main() 方法，这个类称为应用程序的主类。public、static 和 void 用于对 main() 方法进行声明。在一个 Java 应用程序中，main() 方法必须被声明为 public、static 和 void，public 声明 main() 是公有的方法，static 声明 main() 是一个类方法，可以通过类名直接调用，而 void 则表示 main() 方法没有返回值。

在定义 main() 方法时，String args[] 用来声明一个字符串类型的数组 args，它是 main() 方法的参数，用来接收程序运行时所需要的参数。

上例的 main 方法中只有一条语句：

```
System.out.println("Hello Java!"); // 在控制台上输出字符串 "Hello Java!"
```

这个语句是把字符串“Hello Java!”输出到系统的标准输出上，例如系统屏幕。其中 System 是系统类的对象，out 是 System 对象中的一个对象，表示“标准输出”，println 是 out 对象的一个方法，其作用是在系统标准输出上显示形参里指定格式的字符串，并回车换行。“//”代表注释，用来说明这一条语句的功能，注释主要用来提高程序的可读性，不会参与程序的编译。

② 对已创建好的 Java 源程序进行编译。

该步骤是用 Java 编译器对 Java 源程序（.java）进行编译，生成对应的字节码文件（.class）。如果编译成功，会产生一个有相同文件名的带 .class 扩展名的字节码文件。

其命令格式如下：

```
javac HelloWorldApp.java
```

如果编译中不出现错误，将会得到一个名为 HelloWorldApp.class 的文件。编译选项使用缺省方式。

③ 解释执行已编译成功的字节码文件（.class 文件）

用 Java 解释器对 Java 字节码文件（.class 文件）解释执行。

在上例得到的 HelloWorldApp.class 文件，现在可以用 Java 解释器执行：

```
java HelloWorldApp
```

运行的结果，将会在标准输出设备上输出：

```
Hello Java!
```

Java 解释器在解释执行时，解释处理的是类名，而不是文件名，所以在解释器 Java 后面跟随的是类名，而不能写成文件名的形式（HelloWorldApp.class）。其选项也是使用缺省的方式。

2) Java Applet 程序

一个 Java Applet 也是由若干个类组成的，一个 Java Applet 不再需要 main() 方法，但必须有且仅有一个类扩展了 Applet 类，即它是 Applet 类的子类，这个类称为 Java Applet 的主类，Java Applet 的主类必须是 public 的，Applet 类是系统提供的类。

Applet 与 Application 的区别在于其执行方式不同。Application 是从 main() 方法开始运行的，而 Applet 是在浏览器中运行的，必须创建一个 HTML 文件，通过编写 HTML 代码告诉浏览器载入何种 Applet 以及如何运行。

开发 Java Applet 程序的步骤如下：

首先，编写 Applet 源程序，将其保存为“.java”文件。

编译源程序，生成字节码文件“.class”。如果源文件包含了多个类，则会生成多个扩展名为“.class”的文件，都与源文件存放在相同的目录下。如果对源文件进行了修改，那么必须重新进行编译，再生成新的字节码文件。

编写一个 HTML 文件，即含有 applet 标记的 Web 页，嵌入 Applet 字节码文件“*.class”。

最后，运行 Java 小应用程序。

案例 1：开发一个输出“Hello Java !!”的 Java 小应用程序。

本案例的实现步骤主要有：

① 以文件名“HelloWorldApplet.java”保存 Java Applet 源程序。程序代码如下：

```
// 文件 HelloWorldApplet.java
import java.awt.*;      // 引入 java.awt 包中的类
import java.applet.*;   // 引入 java.applet 包中的类
public class HelloWorldApplet extends Applet { // 继承 Applet
    public void paint(Graphics g) { // 重写 paint 方法
        // 在 (50, 40) 位置输出字符串
        g.drawString("Hello Java !!", 50, 40);
    }
}
```

② 编写嵌入字节码文件的 HTML 文件 “HelloWorldApplet.html”：

```
<html><!-- 标识 HTML 文件的开始 -->
<!-- 告诉浏览器将运行一个 Java Applet-->
<appletcode="HelloWorldApplet.class"width="200"height="80"></applet>
</html><!-- 标识 HTML 文件的结束 -->
```

③ 使用 JDK 编译 Java 小应用程序：

```
javac HelloWorldApplet.java
```

④ 使用 JDK 提供的 Appletviewer 运行：

```
appletviewer HelloWorldApplet.html
```



提醒

Java Applet 程序必须创建一个 Applet 或 JApplet 的子类。Applet 程序中不需要有 main() 方法。

(5) Java 程序的工作过程

Java 语言包括 3 种核心机制：Java 虚拟机、垃圾收集机制和代码安全检测。

Java 程序的开发过程大致分为以下 3 个阶段：

① 编写 Java 源文件。将编辑好的源程序以扩展名 “.java” 保存起来，即保存成 “.java” 文件。

② 编译 Java 源程序。使用 Java 编译器编译 “.java” 源程序，从而得到字节码文件 “.class”。

③ 运行 Java 程序。

Java 程序的开发流程如图 1 所示。

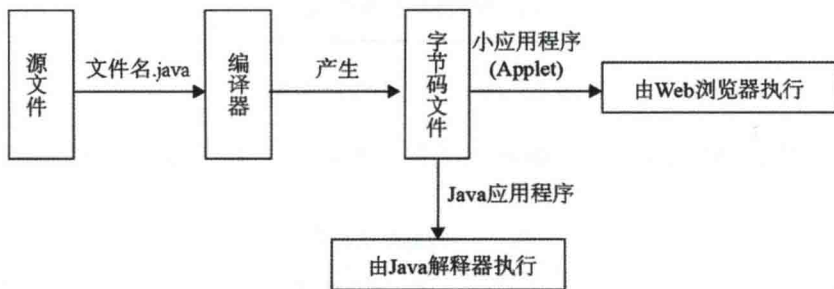


图 1 Java 程序的开发流程

从图 1 中可以看出，一个源文件首先应保存为扩展名为 .java 的文件，通过编译器产生字节码文件。

如果编写的是小应用程序（Java Applet），可以直接由浏览器解释运行，对于 Java 应用程序，就会由 Java 解释器来执行。

在本书中，我们以应用程序为主，也就是说，我们使用的是 Java 解释器。对 Java Applet 小应用程序感兴趣的读者可以参考其他相关书籍进行学习。

(6) JDK 的安装与配置

Java 开发平台最常用的就是 JDK，Java 语言的主流开发环境也是 JDK+Eclipse，先来介绍下载和安装 JDK，并配置 JDK 的环境变量。

JDK（Java Development Kit，Java 开发工具包）是 Sun 公司（现在是 Oracle 公司）提供的 Java 开发环境和运行环境，是所有 Java 应用程序的基础。它包括一组 API 和 JRE（Java 运行时环境），这些 API 是构建 Java 应用程序的基础，而 JRE 是运行 Java 应用程序的基础。JDK 包括 J2ME（微型版）、J2SE（标准版）和 J2EE（企业版）3 个版本，最基本的开发包是 J2SE。

JDK 为免费开源的开发环境，任何开发人员都可以直接从 Oracle 公司的官方网站（<http://www.oracle.com/technetwork/java/javase/downloads/index.html>）下载获得安装程序包。本书使用的 JDK 版本为 JDK 6，操作系统平台为 Windows 系统。

Java 开发平台是一种能够运行 Java 程序并且支撑 Java 程序开发的软件系统，包括 Java 虚拟机和 Java API 两部分，如图 2 所示。

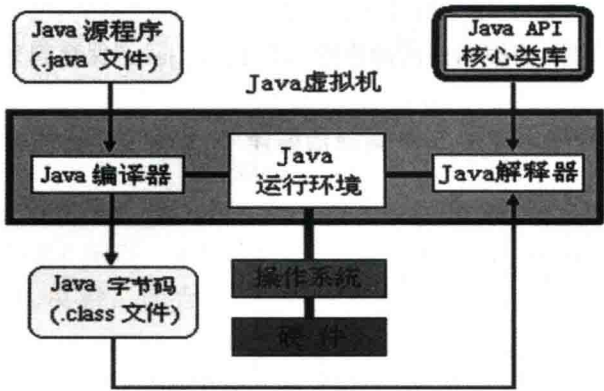


图 2 Java 开发平台结构图

Java™ 2 Platform Standard Edition 6（简称 JDK 6）是 Java 开发平台的一种具体实现，主要内容如下：

表 1 Java 开发平台

英文名称	英文缩写	中文名称
Java™ SE Development Kit 6	JDK™ 6	Java 开发工具
Java™ SE Runtime Environment 6	JRE™ 6	Java 运行环境