

Java面向对象程序设计 实训教程

邵欣欣 蒋晶晶 主 编
王 倩 高 爽 副主编



高等学校计算机专业教材精选·算法与程序设计

Java面向对象程序设计 实训教程

邵欣欣 蒋晶晶 主 编
王 倩 高 爽 副主编

清华大学出版社

内 容 简 介

本书从 Java 的基本概念入手,介绍了面向对象的程序设计思想以及 Java 最基础、最主要的核心技术。

本书既注重理论的介绍,又强调实际的应用,从实用的角度出发,精心设计知识结构及代码实例,并配以大量的习题和实验,让读者在阅读的过程中很轻松地既能掌握枯燥的计算机语言知识,又锻炼了实践能力。通过最后的项目实训,进一步加强了学生对 Java 知识的全面掌握,提高综合应用的能力。

本书既可以作为高校本、专科相关专业学生的课程用书,也可作为自学人员的参考资料。

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 面向对象程序设计实训教程/邵欣欣,蒋晶晶主编. —北京:清华大学出版社,2013

高等学校计算机专业教材精选·算法与程序设计

ISBN 978-7-302-33782-9

I. ①J… II. ①邵… ②蒋… 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(2013)第 211375 号

责任编辑:张 玥 顾 冰

封面设计:傅瑞学

责任校对:白 蕾

责任印制:刘海龙

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

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:清华大学印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:18.25

字 数:445 千字

版 次:2013 年 10 月第 1 版

印 次:2013 年 10 月第 1 次印刷

印 数:1~2000

定 价:34.50 元

产品编号:054554-01

前 言

根据 Java 课程的能力要求和学生的认知规律,编者精心组织了《Java 面向对象程序设计实训教程》一书的教材内容,融“教、学、练”三者于一体,适合“项目驱动、案例教学、理论实践一体化”的教学模式。本书主要以实现酒店房间管理系统为目标,展开了对 Java 基本概念和原理的介绍,最后实现了项目的各个功能。区别于其他传统的教程,本书在每个章节介绍本章知识点在最后的项目实现中的应用情况,这样使读者带着目的去学习,能够有效地提高学习效率。全书共分为 14 章,其中第 1~3 章介绍 Java 基础知识;第 4~6 章介绍 Java 技术核心,是 Java 应用程序编程的基础;第 7 章介绍 Java 中数组和字符串知识的应用;第 8 章介绍 Java 语言中的异常处理机制;第 9 章介绍 Java 常用的图形用户界面组件;第 10~12 章介绍 Java 中高级知识的应用,包括输入输出流、多线程和网络编程;第 13 章介绍 Java 连接各类数据库的知识;第 14 章通过酒店房间管理系统实现对前面知识的综合应用。

本书内容全面,通俗易懂,结构合理,简洁明了,实例丰富,图文并茂,书中所列程序易于读者理解和掌握。本书既注重理论的介绍,又强调实际的应用,从实用的角度出发,精心设计知识结构及代码实例,并配以大量的习题和实验,让读者在阅读的过程中很轻松地既能掌握枯燥的计算机语言知识,又锻炼了实践能力。书中所有例题及相关代码都已在 JDK 1.6 和 Eclipse 开发环境中测试通过。

本书的编者都是长期工作在教育第一线的教师,教学经验丰富,同时具有实际开发项目的经验。本书在编写过程中,充分发挥了各位教师所长,第 1、13 章由蒋晶晶编写,第 2、3 章由蒋晶晶和王倩编写,第 7、10、11、12 章由高爽和王倩编写,第 8 章由王倩和邵欣欣编写,第 4~6、9、14 章由邵欣欣编写,全书最后由邵欣欣统一修改定稿。

由于编者的水平和时间有限,本书存在错误和不足之处在所难免,恳请同行专家和广大读者批评指正。

本书配有完整课件、实例代码以及授课进度表,可从清华大学出版社网站下载。

编 者

2013 年 8 月于大连

目 录

第 1 章 Java 概述	1
1.1 什么是 Java	1
1.2 Java 的特点	2
1.3 Java 开发工具	3
1.3.1 编辑工具	3
1.3.2 JDK	4
1.4 第一个程序	5
1.5 学习效果评估	7
第 2 章 Java 开发环境	8
2.1 Eclipse 简介	9
2.2 Eclipse 的安装	9
2.3 Eclipse 界面	10
2.3.1 选择工作空间界面	10
2.3.2 Eclipse 的主界面	10
2.4 使用 Eclipse 创建 Java 项目	11
2.4.1 创建项目	11
2.4.2 创建 Java 文件	11
2.4.3 编辑 Java 文件	13
2.4.4 运行 class 文件	14
2.5 实训任务——Java 开发及运行环境的搭建	14
任务 1: JDK 的安装	14
任务 2: Eclipse 的安装及使用	14
2.6 学习效果评估	15
第 3 章 Java 语言基础	17
3.1 基本数据类型	18
3.1.1 字符数据类型	18
3.1.2 布尔数据类型	19
3.1.3 数值数据类型	19
3.2 变量	20
3.2.1 变量的声明	20
3.2.2 标识符	20
3.2.3 变量的赋值	21

3.2.4	常量	21
3.3	数据类型的转换	22
3.3.1	自动数据类型转换	22
3.3.2	强制数据类型转换	22
3.4	运算符和表达式	22
3.4.1	算术运算符和表达式	22
3.4.2	关系运算符和表达式	23
3.4.3	逻辑运算符和表达式	23
3.4.4	其他运算符	24
3.4.5	运算符的优先级	27
3.5	条件语句	28
3.5.1	if 语句	28
3.5.2	if-else 语句	29
3.5.3	switch 语句	31
3.6	循环语句	32
3.6.1	for 语句	32
3.6.2	while 语句	33
3.6.3	do-while 语句	35
3.7	跳转语句	36
3.7.1	break 语句	36
3.7.2	continue 语句	38
3.8	控制语句的应用	39
3.9	实训任务——控制语句的使用	45
	任务 1: 使用条件语句实现程序流程控制	45
	任务 2: 使用循环语句实现程序流程控制	46
	任务 3: 使用跳转语句实现程序流程控制	46
3.10	学习效果评估	47

第 4 章	类和对象	55
4.1	面向对象	56
4.1.1	什么是面向对象	56
4.1.2	面向对象的特征	57
4.2	类的结构	58
4.2.1	属性	59
4.2.2	方法	59
4.2.3	构造方法	60
4.3	类与对象的关系	61
4.4	对象的创建	62
4.5	方法的调用	63

4.6	给方法传递对象参数	65
4.7	变量的作用域	66
4.8	this 关键字	67
4.9	static 关键字	68
4.9.1	类属性	68
4.9.2	类方法	69
4.10	类与对象的应用	70
4.11	实训任务——类和对象的使用	73
	任务 1: 方法的使用	73
	任务 2: 类的编写	73
	任务 3: 构造方法的编写	73
	任务 4: 对象的创建	73
4.12	学习效果评估	74

第 5 章	封装、继承与多态	80
5.1	可见性修饰符	81
5.1.1	类的可见性修饰符	81
5.1.2	类的成员的可见性修饰符	81
5.2	访问器方法	82
5.3	包	85
5.3.1	包的声明	85
5.3.2	包的引入	86
5.4	封装的应用	86
5.5	继承	87
5.5.1	继承的实现	88
5.5.2	属性的隐藏	91
5.5.3	方法的覆盖	92
5.6	多态	93
5.6.1	重载	93
5.6.2	重载与覆盖	95
5.7	super 关键字	96
5.8	继承关系中的构造方法	97
5.9	final 关键字	99
5.10	继承与多态的应用	101
5.11	实训任务——继承与多态设计与实现	107
	任务 1: 可见性修饰符的应用	107
	任务 2: 继承的应用	107
	任务 3: 多态的应用	107
5.12	学习效果评估	108

第 6 章 抽象类与接口	117
6.1 抽象类	118
6.1.1 创建抽象类	118
6.1.2 继承抽象类	119
6.2 接口	120
6.2.1 创建接口	120
6.2.2 实现接口	122
6.3 抽象类和接口的应用	123
6.4 实训任务——抽象类和接口的应用	126
任务 1: 抽象类的应用	126
任务 2: 接口的应用	126
6.5 学习效果评估	126
第 7 章 基础类库	129
7.1 数组	130
7.1.1 声明数组	130
7.1.2 创建数组	131
7.1.3 访问数组	131
7.1.4 对象数组	132
7.1.5 二维数组	132
7.2 向量	134
7.3 字符串	135
7.3.1 String 类	135
7.3.2 StringBuffer 类	137
7.3.3 String 与其他数据类型间的转换	138
7.4 Math 类	138
7.5 实训任务——基础类库的使用	140
任务 1: 数组和向量的使用	140
任务 2: 字符串的使用	140
7.6 学习效果评估	140
第 8 章 异常及其处理	145
8.1 什么是异常	146
8.1.1 异常与错误	146
8.1.2 异常的分类	148
8.1.3 异常是如何产生的	149
8.2 捕获异常	150
8.2.1 使用 try/catch 子句	150
8.2.2 多重 catch 子句	151

8.2.3	finally 子句	152
8.3	声明异常	153
8.4	抛出异常	154
8.5	创建自己的异常	154
8.6	实训任务——异常处理	156
任务 1:	异常的捕获	156
任务 2:	自定义异常	156
8.7	学习效果评估	157
第 9 章	图形用户界面	160
9.1	认识 GUI	161
9.1.1	什么是 GUI	161
9.1.2	第一个 GUI 程序	162
9.2	框架	163
9.3	布局管理器	164
9.3.1	流水布局管理器	164
9.3.2	网格布局	165
9.3.3	边界布局	166
9.4	面板	167
9.5	组件	169
9.5.1	按钮	169
9.5.2	文本框和标签	170
9.5.3	复选框和单选按钮	171
9.5.4	列表框和组合框	173
9.5.5	菜单	175
9.5.6	对话框	178
9.6	GUI 事件处理	181
9.6.1	窗口事件	182
9.6.2	动作事件	183
9.6.3	键盘事件	185
9.7	绘制图形	187
9.8	辅助类	190
9.9	实训任务——GUI 编程	193
任务 1:	布局管理器的使用	193
任务 2:	组件的使用	194
任务 3:	事件处理的应用	194
任务 4:	绘制图形和辅助类的应用	195
9.10	学习效果评估	195

第 10 章	输入输出流	200
10.1	输入输出流	201
10.1.1	输入流和输出流	201
10.1.2	字节流和字符流	201
10.2	标准输入输出流	202
10.2.1	标准输出流	203
10.2.2	标准输入流	204
10.3	文件管理	206
10.3.1	File 类	206
10.3.2	获取文件属性	207
10.3.3	获取文件夹中的文件列表	208
10.3.4	创建、删除文件	209
10.4	读写文件	209
10.4.1	读取文件内容	209
10.4.2	向文件写入内容	210
10.5	实训任务——输入输出实践	211
	任务 1: 记事本	211
10.6	学习效果评估	212
第 11 章	多线程	214
11.1	什么是线程	214
11.1.1	线程与进程	214
11.1.2	线程 Thread 类	215
11.2	创建自己的线程	216
11.2.1	通过 Thread 类创建线程	216
11.2.2	通过 Runnable 接口创建线程	217
11.3	线程的控制与状态	218
11.3.1	线程的控制	218
11.3.2	线程的状态	220
11.4	线程的优先级	220
11.5	线程的同步问题	221
11.5.1	什么是同步问题	221
11.5.2	同步锁	223
11.5.3	死锁问题	224
11.6	实训任务——多线程实践	225
	任务 1: 赌马游戏	225
11.7	学习效果评估	227

第 12 章 网络编程	228
12.1 网络基础知识	228
12.1.1 网络标识	228
12.1.2 端口	229
12.2 URL 类	229
12.3 InetAddress 类	230
12.4 Socket 编程	231
12.4.1 ServerSocket 服务器端	231
12.4.2 Socket 客户端	233
12.4.3 多客户端通信的实现	234
12.5 实训任务——Socket 编程实践	235
任务 1: 聊天程序	235
12.6 学习效果评估	236
 第 13 章 Java 连接数据库编程	 239
13.1 JDBC 简介	240
13.1.1 JDBC 驱动器	240
13.1.2 JDBC 访问数据库的流程	241
13.2 数据库连接实例	243
13.2.1 SQL Server 数据库的访问	243
13.2.2 MySQL 数据库的访问	246
13.3 实训任务——数据库编程	248
任务 1: 数据库基本操作	248
13.4 学习效果评估	248
 第 14 章 酒店房间管理系统项目实训	 250
14.1 系统功能和流程分析	250
14.2 数据库设计	250
14.3 酒店房间管理系统实施	252
14.3.1 数据库连接	252
14.3.2 登录模块	253
14.3.3 用户管理模块	261
14.3.4 管理员模块	273
 参考文献	 280

第 1 章 Java 概述

学习要求

通过本章的学习,读者能够了解 Java 语言在实际应用中的优势,了解 Java 的常用开发工具,掌握 Java 应用程序的简单实例,掌握如何运行 Java 应用程序。

知识要点

- Java 语言的发展史;
- Java 语言的特点;
- Java 的常用开发工具;
- 简单的 Java 应用程序的编写。

教学重点与难点

(1) 重点:

- Java 语言的特点;
- JDK 的使用;
- 简单的 Java 应用程序。

(2) 难点: 简单的 Java 应用程序的结构组成。

Java 是最具代表性的高级程序设计语言之一。本章将介绍 Java 语言的发展史、Java 的特点,以及常用的 Java 开发工具,并以一个简单的应用程序为例,介绍 Java 程序执行的步骤。

1.1 什么是 Java

Java 是一种面向对象的程序设计语言,由 Sun Microsystems 公司的 James Gosling 等人于 20 世纪 90 年代初开发。它最初被命名为 Oak,用来开发消费类电子产品,解决诸如电话、电视机等家用电器的控制和通信问题。后来随着互联网的发展,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,在 1995 年 5 月正式命名为 Java,如图 1.1 所示。

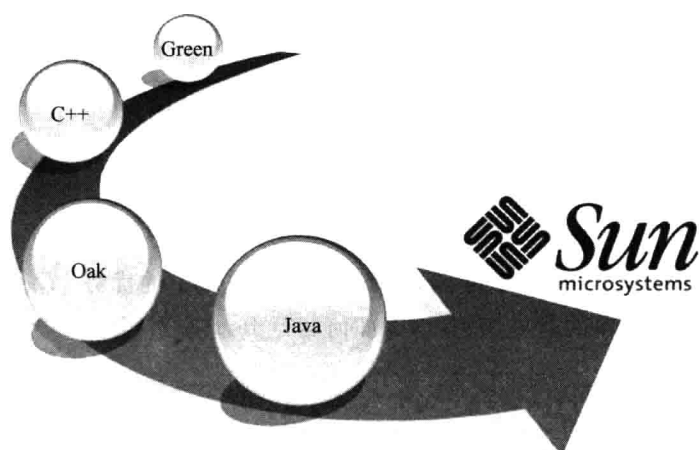


图 1.1 Java 简史

1.2 Java 的特点

Java 伴随着互联网的蓬勃发展已逐渐成为重要的网络编程语言,是一种简单、动态、面向对象、分布式、解释执行、健壮、安全、结构中立、可移植、高效能、具有多线程能力的新一代语言。

1. 面向对象

对象可以是人们研究的任何实体,小到一个原子大到整个宇宙均可看作对象,它不仅能表示具体的事物,还能表示抽象的规则、概念等。对象之间通过消息相互作用,用公式表示,面向对象编程语言可以表示为:程序=对象+消息。现实世界中的对象均有属性和行为,映射到计算机程序上,属性就表示对象的数据(用来表示对象的状态),行为就表示对象的方法(用来处理数据或同外界交互)。

所有面向对象编程语言都支持三个概念:封装、继承和多态,Java 也不例外。封装、继承和多态的概念将在第 6 章和第 7 章中详细介绍。

2. 健壮性

Java 在编译和运行程序时要对可能出现的异常进行检查。并且,Java 提供自动垃圾收集机制来进行内存管理。同时,Java 在编译时还可捕获类型声明中的许多常见错误。

3. 安全性

Java 不支持指针,避免了由于指针操作所引起的错误和非法入侵。同时,Java 在运行应用程序时,严格检查其访问数据的权限,保证数据的可靠性。

4. 平台无关性

Java 主要靠 Java 虚拟机(JVM)实现平台无关性。首先,由 Java 编译器编译生成了与体系结构无关的字节码文件;然后,由 Java 解释器来解释执行该字节码文件。任何一台机器只要配备了 Java 解释器,就可以运行该程序,而不管这种字节码是在何种平台上生成的。

5. 可移植性

根据 Java 的平台无关性这一特征,可以将 Java 系统移植到不同的处理器和操作系统。

Java 的编译器是由 Java 语言实现的,解释器是由 Java 语言和标准 C 语言实现的,因此可以较为方便地进行移植工作。

6. 分布式

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上,对于数据分布,Java 提供了一个 URL 对象;操作分布是指把一个计算分散在不同主机上处理,Java 的客户端/服务器模式可以把运算从服务器分散到客户一端。另外,Java 提供的一整套网络类库也为开发人员方便地实现 Java 的分布式特性提供了便捷。

1.3 Java 开发工具

1.3.1 编辑工具

Java 的开发工具可以分成三大类,包括文本编辑器和集成开发工具。下面给大家介绍几种常用的开发工具。

1. 文本编辑器

这类工具只提供了文本编辑功能,能进行多种编程语言的开发,如 C、C++ 和 Java 等。除了使用记事本这一工具外,UltraEdit 和 EditPlus 这两种编辑器也是其中的代表。

2. 集成开发工具

Java 的集成开发环境(Integrated Development Environment, IDE)目前已有多种。所谓 IDE,就是把编辑、编译、调试及运行集成在一个开发环境中的软件,并且还增加了许多提高开发效率的实用功能,比如界面设计功能、自动编译、错误提示、设置断点、单步调试、在 IDE 内部显示运行结果等功能。这类工具是 Java 开发工具的发展趋势,不仅为需要集成 Java 和 J2EE 的开发者或开发团队提供了 Java 的集成开发环境,而且提供了对 Web Applications、Servlets、JSPs、EJBs、数据库访问和企业应用的强大支持。

常见的 Java 集成开发环境有:

- Borland 的 JBuilder;
- Oracle 的 JDeveloper;
- Eclipse。

下面主要介绍 Eclipse 和 JBuilder。

1) Eclipse

Eclipse 是一个开放源代码的、可扩展的集成开发环境。IBM 公司捐出价值 4000 万美元的源代码组建了 Eclipse 联盟,并由该联盟负责这种工具的后续开发。虽然 Eclipse 本质上只是一个框架和一组服务,用于通过插件组件构建开发环境,但它的发展目标不仅仅是成为专门开发 Java 程序的 IDE 环境。根据 Eclipse 的体系结构,通过开发插件,它能扩展到任何语言的开发。另外,由于 Eclipse 是一个开源项目,任何人都可以下载 Eclipse 的源代码,并在此基础上开发自己的功能插件,所以 Eclipse 成为业界非常受欢迎的 Java 开发工具(见图 1.2)。

2) JBuilder

JBuilder 由 Borland 公司开发,也是目前非常流行的 Java 集成开发工具之一(见图 1.3),在

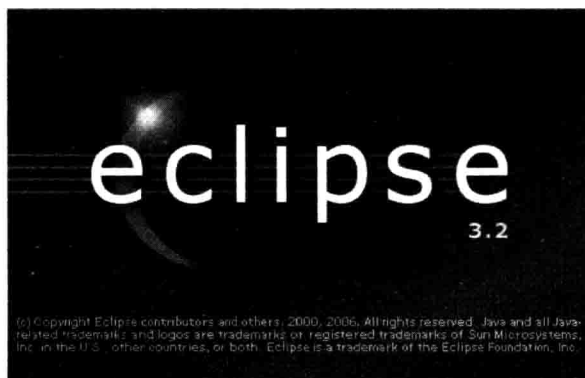


图 1.2 Eclipse 开发工具

协同管理、对 J2EE 和 XML 的支持等方面均走在同类产品的前列。JBuilder 可以集成多种应用服务器,使 Web 开发更容易;拥有专业化的图形调试界面,支持远程调试和多线程调试;提供更简单的程序发布功能,所有的应用都可以打包;同时,提供了团队开发能力,可以集成多种版本控制产品。

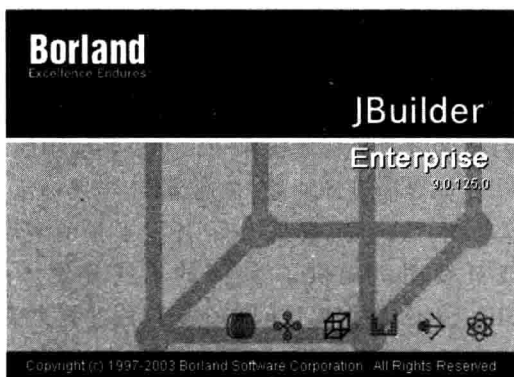


图 1.3 JBuilder 开发工具

1.3.2 JDK

JDK(Java Development Kit)是整个 Java 的核心,包括 Java 运行环境 JRE(Java Runtime Environment)、Java 基础类库和一系列 Java 工具。掌握 JDK 是学好 Java 的第一步。

JDK 中常用工具包括:

- javac: Java 语言编译器,负责将 Java 源代码(. java)文件编译为字节码(. class)文件。
- java: Java 语言解释器,负责执行 Java 字节码(. class)文件。
- javadoc: Java 语言文档生成器,负责将源程序中的注释提取成 HTML 格式文档。
- jdb: Java 调试工具,可以逐行执行代码,设置断点和检查变量,是调试程序、查找错误的有效工具。

JDK 安装完成后,会在安装路径下的 bin 目录找到这些工具,它们都是可执行文件,如图 1.4 所示。

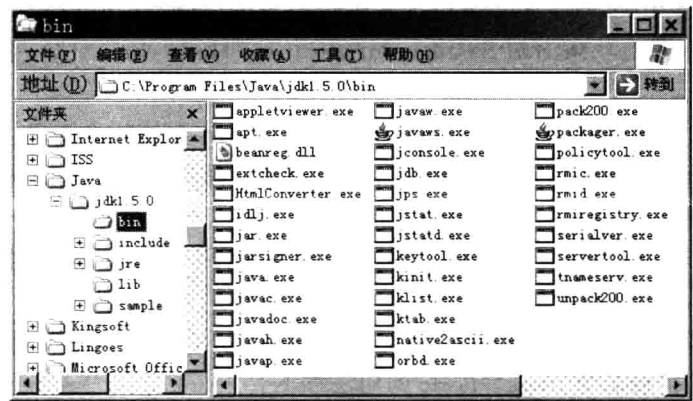


图 1.4 JDK 的 bin 目录

Java 文件编辑、编译并运行的过程如图 1.5 所示。

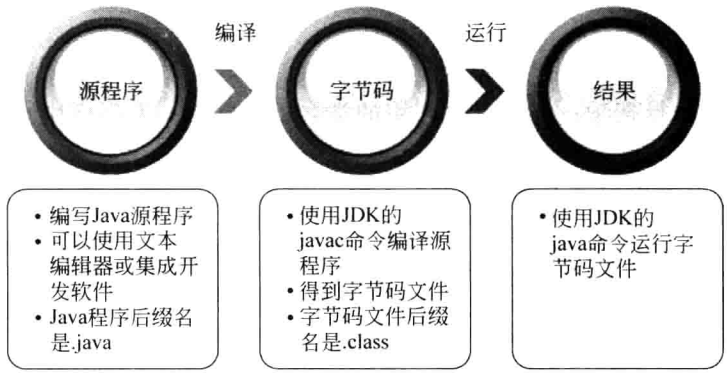


图 1.5 编辑、编译及运行 Java 程序的过程

1.4 第一个程序

Java 程序可以分为 Java 应用程序(Java Application)和 Java 小程序(Applet)两种。这里只介绍简单 Java 应用程序实例,并对其进行分析。

例 1.1 第一个 Java 应用程序 HelloWorld.java。

```
public class HelloWorld {
    public static void main(String args[]){
        System.out.println("Hello World!");
    }
}
```

例 1.1 中定义一个类 HelloWorld,在该类中定义了一个 main 方法。对于一个应用程序来说,main 方法是必需的,而且必须按照如上的格式来定义。有关类和方法的概念将于

第 5 章中详细描述。

Java 解释器在没有生成任何实例的情况下,以 main 方法作为入口来执行程序。Java 程序中可以定义多个类,每个类中可以定义多个方法,但只能有一个类有 main 方法。在该例中,main 方法中只有一条语句 `System.out.println("Hello World!")`,用来实现字符串的输出。

运行 Java 应用程序包括三步:Java 代码的编写、Java 文件的编译和 Java 程序的执行。Java 首先将源代码文件编译成字节码文件,然后依赖各种不同平台上的虚拟机来解释执行字节码,从而实现了“一次编译、处处执行”的跨平台特性。

(1) Java 代码的编写。Java 源程序文件可以使用任何文本编辑器进行编辑,如 Windows 记事本、UltraEdit 和 EditPlus 等,源程序文件的扩展名为 java。

(2) Java 文件编译。编译过程由 `javac` 命令来完成,它将源程序文件编译成字节码文件,字节码文件的扩展名为 class。javac 的用法如下:

```
javac [options] sourcefiles
```

options 为命令行选项;sourcefiles 为一个或多个要编译的 java 源文件。

(3) Java 程序的执行。Java 程序的执行由解释器 java 命令完成,使用方法如下:

```
java [options] class [argument]
```

options 为命令行选项;class 为被调用的类名;argument 为程序中 main 方法使用的参数。

Java 程序运行过程如图 1.6 所示。

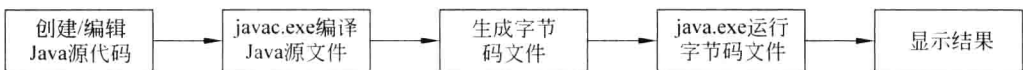


图 1.6 Java 程序创建、编译和运行过程

现在按照图 1.6 演示的过程来运行例 1.1 的程序。首先把代码存放到一个名为 HelloWorld.java 的文件中,并把该文件保存到相应磁盘中(比如在 D 盘根目录下)。这里,文件名应与类名相同,因为 Java 解释器要求公共类必须放在与其同名的文件中。然后对它进行编译:

```
D:\>javac HelloWorld.java
```

编译的结果是生成字节码文件 HelloWorld.class。最后用 Java 解释器来运行该字节码文件:

```
D:\>java HelloWorld
```

本程序的作用是在屏幕上显示“Hello World!”,效果如图 1.7 所示。

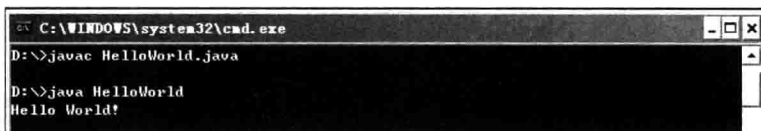


图 1.7 HelloWorld 的运行结果