

国家社会科学基金(教育学)
“十一五”规划课题研究成果

全国高等职业院校计算机教育规划教材

Java 程序设计项目教程

JAVA PROGRAMMING PROJECT TUTORIAL

郭庚麒 周 江 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

国家社会科学基金（教育学科）“十一五”规划课题研究成果
全国高等职业院校计算机教育规划教材

Java 程序设计项目教程

郭庚麒 周 江 编著

内 容 简 介

本书以 Java SE 技术为背景介绍 Java 程序设计的方法。全书共分为 14 章,全面而翔实地介绍了 Java 程序设计语言的编程方法和技巧,内容包括 Java 的基本语法、Java 异常处理机制、基本输入/输出、Java GUI 编程、Java 数据库编程等。

本书是作者多年教学经验的总结,主要内容以“班主任小助手”项目系统为载体,采用任务驱动的教学方法,按照学生的认知规律,逐层深入,抽丝剥茧地进行介绍,语言通俗而不失严谨。同时,本书还选配有大量的习题和实训项目,实训项目的选择注重一致性、趣味性和实用性相结合的原则,重在培养学生的应用能力。

本书适合作为高等职业院校、应用型本科院校面向对象编程语言的入门教程,也可作为 Java 编程爱好者的参考用书。

图书在版编目(CIP)数据

Java 程序设计项目教程 / 郭庚麒, 周江编著. --北京: 中国铁道出版社, 2011. 7

国家社会科学基金(教育学科)“十一五”规划课题研究成果 全国高等职业院校计算机教育规划教材

ISBN 978-7-113-10117-6

I. ①J… II. ①郭… ②周… 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(2011)第 118616 号

书 名: Java 程序设计项目教程

作 者: 郭庚麒 周 江 编著

策划编辑: 翟玉峰 王春霞

责任编辑: 翟玉峰

编辑助理: 贾淑媛

封面设计: 付 巍

责任印制: 李 佳

读者热线: 400-668-0820

封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街 8 号 邮政编码: 100054)

印 刷: 北京新魏印刷厂

版 次: 2011 年 7 月第 1 版 2011 年 7 月第 1 次印刷

开 本: 787mm×1092mm 1/16 印张: 16 字数: 379 千

印 数: 3 000 册

书 号: ISBN 978-7-113-10117-6

定 价: 26.00 元

版权所有 侵权必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与教材研究开发中心批销部调换。

国家社会科学基金（教育学科）“十一五”规划课题研究成果 全国高等职业院校计算机教育规划教材

主 任：邓泽民

副主任：（按姓氏笔画排序）

吕一中	李 雪	严晓舟	汪燮华	张洪星
张晓云	武马群	赵凤芝	段银田	宣仲良
姚卿达	聂承启	徐 红	彭 勇	蒋川群

委 员：（按姓氏笔画排序）

王浩轩	邓安远	邓璐娟	白延丽	包 锋
朱 立	任益夫	刘志成	刘晓川	孙街亭
延 静	杨立峰	杨永娟	杨志茹	杨俊清
李 洪	李 洛	李 新	李学相	李洪燕
连卫民	吴晓葵	沈大林	宋海军	张 伦
张世正	张晓蕾	张新成	欧阳广	周国征
赵传慧	赵轶群	段智毅	贺 平	秦绪好
袁春雨	徐人凤	徐布克	梅创社	黄丽民
崔永红	梁国浚	蒋腾旭	蔡泽光	翟玉峰

序

PREFACE

国家社会科学基金(教育学科)“十一五”规划课题“以就业为导向的职业教育教学理论与实践研究”(课题批准号 BJA060049)在取得理论研究成果的基础上,选取了高等职业教育十个专业类开展实践研究。高职高专计算机类专业是其中之一。

本课题研究发现,高等职业教育在专业教育上承担着帮助学生构建起专业理论知识体系、专业技术框架体系和相应职业活动逻辑体系的任务,而这三个体系的构建需要通过专业教材体系和专业教材内部结构得以实现,即学生的心理结构来自于教材的体系和结构。为此,这套高职高专计算机类专业系列教材的设计,依据不同教材在其构建知识、技术、活动三个体系中的作用,采用了不同的教材内部结构设计和编写体例。

承担专业理论知识体系构建任务的教材,强调了专业理论知识体系的完整与系统,不强调专业理论知识的深度和难度;追求的是学生对专业理论知识整体框架的把握,不追求学生只掌握某些局部内容,而求其深度和难度。

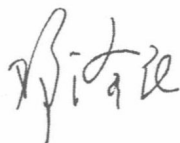
承担专业技术框架体系构建任务的教材,注重让学生了解这种技术的产生与演变过程,培养学生的技术创新意识;注重让学生把握这种技术的整体框架,培养学生对新技术的学习能力;注重让学生在技术应用过程中掌握这种技术的操作,培养学生的技术应用能力;注重让学生区别同种用途的其他技术的特点,培养学生职业活动过程中的技术比较与选择能力。

承担职业活动体系构建任务的教材,依据不同职业活动对所从事人特质的要求,分别采用了过程驱动、情景驱动、效果驱动的方式,形成了“做学”合一的教材结构与体例,诸如项目结构、案例结构等。过程驱动培养所从事人的程序逻辑思维;情景驱动培养所从事人的情景敏感特质;效果驱动培养所从事人的发散思维。

本套教材从课程标准的开发、教材体系的建立、教材内容的筛选、教材结构的设计,到教材素材的选择,均得到了信息技术产业专家的大力支持,他们根据信息技术行业职业资格标准和各类技术在我国应用的广泛程度,提出了十分有益的建议;国内知名职业教育专家和一百多所高职院校参与本课题研究,他们对高职高专信息技术类人才培养提出了可贵意见,对高职高专计算机类专业教学提供了丰富的素材和鲜活的教学经验。

这套教材是我国高职教育近年来从只注重学生单一职业活动逻辑体系构建,向专业理论知识体系、技术框架体系和职业活动逻辑体系三个体系构建的转变的有益尝试,也是国家社会科学研究基金课题“以就业为导向的职业教育教学理论与实践研究”研究成果的具体应用之一。

如本套教材有不足之处,敬请各位专家、老师和广大同学不吝赐教。希望通过本套教材的出版,为我国高等职业教育和信息技术产业的发展做出贡献。



2009年8月

前言

FOREWORD

随着 Internet 的迅速发展和应用的逐步深入, Java 语言得到普遍应用, 已成为首选的“网络”开发工具。Java 是由 Sun 公司提出的面向对象的跨平台编程语言, 其精髓是“一次编程, 随处运行”, 有效地满足了网络环境对编程语言的需求。在网络时代, 掌握 Java 程序设计语言的基本知识和编程技巧, 已成为 IT 职场的一块“敲门砖”。

本书共分为 14 章, 全面而翔实地介绍了 Java 程序设计语言的编程方法和技巧, 内容包括 Java 的基本语法、Java 异常处理机制、基本输入/输出 Java GUI 编程、Java 数据库编程等。

本书是作者多年从事计算机教学和系统开发经验的总结, 通俗易懂、深入浅出, 适合作为高等职业院校计算机专业及其相关专业的教材, 也可作为软件开发人员及其他有关人员的参考用书。本书具有如下特色:

(1) 以项目为载体, 采用任务驱动的教学方法。案例按照“任务描述”、“任务分析”、“任务实现”三个步骤来展示一个实际编程任务的完成。

(2) 按照学生的认知规律, 逐层深入, 抽丝剥茧地进行介绍, 讲解过程注意培养学生的思维能力, 激发他们的创新能力。

(3) 降低学习的难度, 概念和知识的引入多采用案例说明和类比的方法, 语言通俗而不失严谨。

(4) 注重理论和实践相结合, 配有大量难易程度不同的实训题, 实训题的选择注重一致性、趣味性和实用性相结合的原则, 重在培养学生的应用能力, 同时满足分层次教学的需要。

(5) 作为面向对象程序设计课程的教材, 建议授课学时为 112 学时, 着重培养读者面向对象程序设计的思想和方法。

本书配有电子教案和实例源代码, 可以到 <http://www.edusources.net> 网站下载。

本书第 1~5、8、13 章由郭庚麒编写, 第 6、7、9~12、14 章由周江编写。

本书的内容虽然经过多次讲授, 实例也经过了多次测试, 但由于编者水平有限, 难免会存在疏漏和不足之处, 恳请读者批评指正。

编者

2011.6

目录

CONTENTS

第 1 章 Java 概述及开发环境的建立	1
1.1 Java 背景及特点	1
1.2 Java 的工作机制	2
1.3 Java 开发环境简介	3
1.4 HelloWorld 应用程序和 HelloWorld 小应用程序	4
1.4.1 Java 应用程序 Application	4
1.4.2 Java 小应用程序 Applet	6
1.5 项目系统简介	7
1.5.1 项目应用背景	7
1.5.2 系统结构和功能设计	8
1.5.3 系统功能任务清单	8
1.6 思考与练习	10
1.7 综合实训	10
小结	15
第 2 章 Java 语言基础	16
2.1 Java 语言基本语法单位	16
2.1.1 标识符	16
2.1.2 关键字（保留字）	17
2.1.3 语句、空白、注释、分隔符	17
2.2 Java 基本数据类型	18
2.2.1 常量和变量	18
2.2.2 数据类型概述	19
2.2.3 数据类型的转换	22
2.2.4 封装类及其转换方法	22
2.3 运算符和表达式	23
2.3.1 算术运算符与算术表达式	24
2.3.2 关系运算符与关系表达式	24
2.3.3 逻辑运算符与逻辑表达式	25
2.3.4 位运算符	25
2.3.5 赋值运算符与赋值表达式	25
2.3.6 其他运算符	25
2.3.7 运算符优先级	26

2.4 打印语句解析	28
2.5 思考与练习	28
2.6 综合实训	30
小结	31
第3章 Java 流程控制	32
3.1 顺序结构	32
3.2 选择结构	33
3.2.1 if...else 语句	33
3.2.2 switch 语句	36
3.3 循环结构	38
3.3.1 for 循环语句	38
3.3.2 while 循环语句	41
3.3.3 do...while 循环语句	42
3.4 break 和 continue 语句	43
3.5 方法调用和 return 语句	45
3.5.1 方法定义	45
3.5.2 方法调用	45
3.6 思考与练习	47
3.7 综合实训	49
小结	51
第4章 数组和字符串	52
4.1 数组	52
4.1.1 一维数组	52
4.1.2 二维数组	59
4.2 字符串	61
4.2.1 字符数组与字符串的区别	61
4.2.2 字符串常量	62
4.2.3 String 类构造函数和常用方法	62
4.2.4 StringBuffer 类	66
4.2.5 字符串数组及 main()方法的参数	67
4.2.6 字符串比较	68
4.2.7 字符串和其他类型数据的相互转换	69
4.3 思考与练习	69
4.4 综合实训	70
小结	71
第5章 类和对象	72
5.1 面向对象的概念	72

5.1.1	类的定义	73
5.1.2	对象的生命周期	75
5.1.3	由类的定义产生对象	76
5.1.4	类的构造函数	77
5.2	封装性	81
5.2.1	类变量和类方法	82
5.2.2	使用 private 将变量封装起来	84
5.3	扩展应用	85
5.3.1	数组与对象	85
5.3.2	对象作为方法的参数和返回值	87
5.3.3	对象作为类的成员	88
5.4	思考与练习	89
5.5	综合实训	90
	小结	91
第 6 章	继承和多态机制	92
6.1	继承性	92
6.1.1	extends 关键字	93
6.1.2	子类对父类的扩展	94
6.1.3	子类与父类属性同名: 屏蔽	95
6.1.4	子类与父类方法同名: 覆盖	96
6.1.5	子类与父类构造函数之间的关系	98
6.1.6	Java 的单继承结构和 Object 类	102
6.1.7	this 和 super	102
6.2	多态性	103
6.2.1	静态多态	103
6.2.2	动态多态	105
6.3	类型转换	106
6.4	思考与练习	108
6.5	综合实训	109
	小结	110
第 7 章	抽象方法和抽象类、接口和包	111
7.1	抽象方法和抽象类	111
7.1.1	抽象类	111
7.1.2	用抽象类实现运行时多态	114
7.2	接口	115
7.2.1	接口的定义	115
7.2.2	接口的使用	115
7.2.3	接口和抽象类的比较	117



7.3	包	117
7.3.1	包的创建和使用	117
7.3.2	类及类成员的访问权限	118
7.4	思考与练习	119
7.5	综合实训	120
	小结	121
第 8 章	异常	122
8.1	异常的概念和分类	122
8.1.1	什么是异常	122
8.1.2	异常与错误的区别	123
8.1.3	异常类的继承结构	123
8.2	异常处理方式	123
8.2.1	捕获、处理异常	124
8.2.2	throws 和 throw 语句	127
8.3	自定义异常	128
8.4	对异常的进一步讨论	130
8.5	思考与练习	130
8.6	综合实训	131
	小结	132
第 9 章	输入/输出流	133
9.1	流	133
9.1.1	输入/输出流概念	133
9.1.2	字节流	134
9.1.3	字符流	139
9.2	System 类及标准输入/输出	142
9.3	文件类 File 和文件 I/O 操作	143
9.4	思考与练习	146
9.5	综合实训	147
	小结	147
第 10 章	图形界面设计	148
10.1	AWT 概述	148
10.1.1	AWT 简介	148
10.1.2	容器、组件的关系	149
10.1.3	AWT 容器介绍	151
10.1.4	AWT 组件介绍	152
10.2	版面配置	154
10.2.1	BorderLayout	154

10.2.2	FlowLayout	155
10.2.3	GridLayout	156
10.2.4	Null 布局管理器	158
10.3	事件处理机制	158
10.3.1	事件处理概述	158
10.3.2	委托事件模型	158
10.3.3	事件类型	161
10.3.4	事件适配器	166
10.4	思考与练习	167
10.5	综合实训	168
小结		169
第 11 章	Swing 组件及应用	170
11.1	Swing 概述	170
11.2	Swing 组件分类	172
11.3	顶层容器	174
11.4	中间层容器	177
11.5	基本组件	180
11.6	思考与练习	183
11.7	综合实训	184
小结		185
第 12 章	数据库编程	186
12.1	JDBC 驱动程序	186
12.2	用 Java 程序连接 Access 数据库	187
12.3	数据库元数据	191
12.4	思考与练习	198
12.5	综合实训	199
小结		200
第 13 章	多线程	201
13.1	线程概述	201
13.2	线程的创建	202
13.2.1	Thread 类创建多线程应用程序	203
13.2.2	使用 Runnable 接口创建多线程应用程序	204
13.3	线程的生存周期	206
13.3.1	线程的优先级	206
13.3.2	线程的控制方法	208
13.4	线程的同步控制	210
13.4.1	线程间通信概述	210

13.4.2	方法的同步	210
13.4.3	wait...notify 信号量同步	212
13.5	线程的分组	216
13.6	思考与练习	220
13.7	综合实训	221
小结		222
第 14 章	Java 的网络功能	223
14.1	概述	223
14.2	URL 类	224
14.2.1	URL 基本知识	224
14.2.2	URL 类	224
14.2.3	URLConnection 类	225
14.3	Socket 套接口编程	230
14.3.1	Socket 基本知识	230
14.3.2	ServerSocket 类和 Socket 类	230
14.4	InetAddress 类和 UDP 编程	237
14.5	思考与练习	238
14.6	综合实训	239
小结		239
附录	Java Applet	240

第1章

Java 概述及开发环境的建立

Java 语言是一种非常优秀的编程语言，具有面向对象、与平台无关、安全、稳定和多线程等优良特性。Java 语言不仅可以用来开发大型的应用程序，而且特别适用于 Internet 的应用开发，具备“一次编写，随处运行”的特点，它是网络时代最重要的语言之一。

要学习 Java 语言，先要了解其相关知识，读者通过本章的学习，应该达到以下目标：

学习目标

- ☑ 了解 Java 语言的背景及其特点、工作机制、开发环境；
- ☑ 了解 HelloWorld 程序和 HelloWorld 小应用程序的基本框架和代码组成；
- ☑ 熟悉本教材所用的项目案例；
- ☑ 掌握安装 JDK、设置环境变量、用记事本编写 Java 程序并运行的方法；
- ☑ 了解变量作用域与存储类别；
- ☑ 掌握安装 JCreator 软件、用 JCreator 编辑 Java 程序并运行的方法。

1.1 Java 背景及特点

Java 是 1995 年由 Sun 公司发布的编程语言，它的出现源于对独立于平台语言的需要，希望用这种语言能编写出嵌入到各种家用电气设备的芯片上并易于维护的程序。1990 年 12 月，Sun 公司成立了由 James Gosling 等人组成的一个叫 Green Team 的小组，该小组的主要目标是开发一种能够在 PDA、手机、信息家电等消费性电子产品操作平台上运行的分布式系统，James Gosling 为此研发了名为 oak（一种橡树的名字）的编程语言。oak 具备安全性、网络通信、面向对象、垃圾收集和多线程等特点，由于注册商标时已有 oak 这个名字，遂将其改名为 Java。2009 年 4 月，Oracle（甲骨文）公司宣布 74 亿美元收购 Sun 公司。

Java 语言的发展受益于 Internet 和 Web 的出现，因为 Internet 上有各种不同的计算机，它们可能使用完全不同的操作系统和 CPU 芯片，但却希望运行相同的程序。Java 的出现满足了这种要求，标志着真正分布式系统的到来。

Java 是目前使用最为广泛的网络编程语言之一。它具有简单、面向对象、平台无关、多线程、安全性、健壮性等特点。

1. 简单性

Java 语言具有简单性,是指这门语言比较容易学习而且好用。Java 是从 C++ 演变而来的,保留了许多 C++ 的优点,但废除了许多容易产生错误的功能,并提出相应加强或替代的方案。

2. 面向对象

面向对象的编程更符合人的思维模式,使人们更容易编写程序。Java 语言引入了类的概念,是彻底的面向对象编程语言。类是用来创建对象的模板,它包含对被创建对象的状态描述和行为的定义。

3. 平台无关

与平台无关是 Java 语言最大的优势,其他语言编写的程序面临的一个主要问题是:操作系统的变化、处理器的升级以及核心系统资源的变化,都可能使程序出现错误或者无法运行。Java 的虚拟机成功地解决了这个问题,用 Java 编写的程序可以在任何安装了 Java 虚拟机(Java Virtual Machine, JVM)的计算机上运行, Sun 公司实现了自己的目标:让用 Java 语言编写的程序可以“一次编写,随处运行”。

4. 多线程

多线程是指程序中同一时间内执行多项工作的能力,例如从网络上下载一个影片的同时也可以播放它。多线程功能在图形用户界面和网络程序设计上特别有用,例如在设计网络程序时,一个服务器可以同时为多个客户端服务。

5. 安全性

Java 中没有指针,不会直接指向本地机器的内存,它自身的安全机制不会允许程序出现这种致命性的错误。当从网络上下载一个程序时,最担心的问题是该程序中是否含有恶意代码,如试图读取或者删除本地机器上的重要文件等。当客户使用支持 Java 的浏览器时,可以放心地运行 Java 的小应用程序 Applet,因为 Applet 是被限制在 Java 运行环境中,不允许它访问计算机的其他部分。

6. 健壮性

Java 编译器提供了很好的错误检测功能,可检测出许多在执行阶段才显示出来的问题。Java 也不使用那些比较容易出现错误的程序功能,例如指针等。同时,Java 语言的异常机制进一步提供了在程序执行阶段的可靠性保障。

1.2 Java 的工作机制

Java 编译器先将源程序翻译为与平台无关的字节代码(byte code),然后由在特定平台下运行的 Java 解释器来解释执行字节代码文件,其工作原理如图 1-1 所示。解释器对 Java 程序屏蔽了底层的操作系统和硬件平台的差异,因此同一个 Java 程序代码可以不加修改地运行在不同的硬件平台和操作系统上。可以说,Java 程序代码是在一个 Java 虚拟机上运行的。

JVM 是运行 Java 程序必不可少的机制。编译后的 Java 程序指令并不直接在硬件系统的 CPU 上执行,而是由 JVM 执行。JVM 是编译后的 Java 程序和硬件系统之间的接口,程序员可以把 JVM 看作一个虚拟的处理器。它不仅解释执行编译后的 Java 指令,而且还进行安全检查。JVM 是 Java 程序能在多平台间进行无缝移植的可靠保证,同时也是 Java 程序的安全检验引擎。

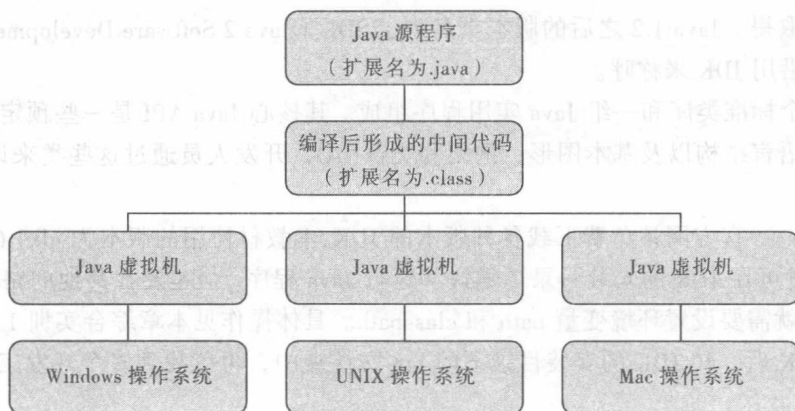


图 1-1 Java 程序处理过程

JVM 的定义：JVM 是在一台真正的机器上用软件方式实现的假想机器。JVM 使用的代码存储在扩展名为.class 的文件中。JVM 的某些指令很像真正的 CPU 指令，包括算术运算、流控制和数组元素访问等。

Java 虚拟机规范提供了编译所有 Java 代码的平台。因为编译是针对假想机的，所以该规范能让 Java 程序独立于平台。它适用于每个具体的硬件平台，以保证为 JVM 编译的代码的运行。JVM 不但可以用软件实现，而且可以用硬件实现。

Java 虚拟机规范对运行时数据区域的划分及字节码的优化并不做严格的限制，它们的实现依平台的不同而有所不同。JVM 的实现称做 Java 运行时系统，简称为运行时 (Java Runtime Environment, JRE)，其作用是把编译过的字节代码“翻译”成所在硬件平台可以辨别的机器码并执行。

Java 运行时必须遵从 Java 虚拟机规范，这样，Java 编译器生成的类文件才可被所有 Java 运行时系统下载。嵌入了 Java 运行时系统的应用程序，就可以执行 Java 程序。目前，有许多操作系统和浏览器都嵌入了 Java 运行时环境。

Sun Microsystems 于 1995 年发布 Java 1.0 版本，1997 年发布 Java 1.1 版本，1998 年发布 Java 1.2，Java 1.2 及其后的版本有一个统一的名称：Java 2 Platform。

Sun 将最近所开发的 Java 运行环境区分成四大版本，分别是：

- 应用于服务器 (Server) 上的 Java EE (Java Platform, Enterprise Edition)；
- 应用于一般个人计算机 (Personal Computer) 上的 Java SE (Java Platform, Standard Edition)；
- 应用于小型设备 (Mobile Device) 上的 Java ME (Java Platform, Micro Edition)；
- 应用在 Smart Card 上的 Java Card。

本书介绍的运行环境为 Java SE (Java Platform, Standard Edition)。

1.3 Java 开发环境简介

要想学好一门编程语言，选择一个好的运行环境非常重要。JDK 是“Java Development Kit (Java 开发工具包)”的缩写，由 Sun 公司开发。最初的版本是 JDK 1.0，现在已经发展到 JDK 1.6。

JDK 是所有 Java 开发工具的基础，JDK 可以到 www.oracle.com 网站上免费下载，或从其他

相关的网站中取得。Java 1.2 之后的版本更名为 J2SDK (Java 2 Software Development Kit), 不过大部分的人仍沿用 JDK 来称呼。

JDK 由一个标准类库和一组 Java 实用程序组成。其核心 Java API 是一些预定义类库, 包括一些重要的语言结构以及基本图形、网络 and 文件 I/O, 开发人员通过这些类来访问 Java 语言的功能。

可以从 Oracle 官方网站免费下载各种版本的 JDK, 本教材使用的版本为 jdk1.6.0-10。在 JDK 安装成功后, 就可在 JDK 的安装目录下编译和运行 Java 程序, 但是要在其他的路径下编译和运行 Java 程序, 就需要设置环境变量 path 和 classpath。具体操作见本章综合实训 1。

安装完 JDK 后, 在 JDK 的安装目录下的 bin 文件夹中, 可以找到多个开发工具运行文件, 简单介绍如下:

javac: Java 编译器, 用来将 Java 原始文件转成字节码。

Java: Java 解析器, 运行扩展名为.class 的 Java 程序。

javadoc: Java 帮助文件管理器。此工具可产生 HTML 格式的 API 帮助文件。

appletviewer: Java Applet 查看器。查看内部包含 Applet 路径的 HTML 网页, 如果 HTML 网页中没有包含 Applet 路径, 则该查看器将没有反应。

Java 源程序文件是纯文本文件, 可以用 Windows 自带的记事本进行编辑, 也可以用 TextPad、Microsoft Word 等常用工具编辑, 编辑后将其另存为扩展名为.java 的文件, 然后用 JDK 中提供的几种简单工具进行编译、运行及程序的调试。

除了 JDK 这种最基本的开发环境之外, 还有许多第三方团队开发了一些综合性开发环境 (Integrated Developed Enviroment, IDE), 提供图形操作的界面, 但这些 IDE 都是架构在 JDK 上的。

以下是流行的一些综合开发环境工具:

- JBuilder (Borland 公司);
- VisualCafe (Symantec 公司);
- Oracle Jdeveloper (Oracle 公司);
- Visual Age (IBM 公司);
- JCreator (Xinox 公司);
- Eclipse (eclipse.org 协会)。

就笔者的经验, 推荐使用 JCreator, 它的优点是免费使用、所占资源少, 并且适合初学者。

JCreator 的下载、安装及简单使用可以参考本章实训 3 和实训 4。

1.4 HelloWorld 应用程序和 HelloWorld 小应用程序

学习语言的捷径是从简单的任务开始, 本节通过对简单的 Java 应用程序 (Application) 和小应用程序 (Applet) 的介绍, 使读者初步了解用 Java 语言编写的程序的组成和结构。

1.4.1 Java 应用程序 Application

Java 应用程序 (Application) 是一类可以独立运行的程序, 下面先介绍一个简单的任务, 并对其进行分析。

【任务 1-1】 建立一个 HelloWorld 应用程序

(一) 任务描述

编写一个 Java 程序, 输出如下一行信息:

Hello World!

(二) 任务分析

(1) 在程序中, 应首先用保留字 `class` 来声明一个新的类, 其类名为 `HelloWorld`, 它是一个公共类 (由 `public` 关键字声明)。整个类定义由大括号 `{ }` 括起来。

```
public class HelloWorld {
```

```
}
```

(2) 在该类的括号中定义了一个 `main()` 方法如下:

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

(3) 在 `main()` 方法的实现中, 只需要写一条语句:

```
System.out.println("Hello World!");
```

它用来实现字符串的输出。另外, 在程序的第一行可以加一行注释 “//任务 1_1: 打印 HelloWorld!”, 说明该程序的作用。

(三) 知识与技能

(1) 与 `main` 同一行的 `public` 表示访问权限, 指明所有的类都可以使用这一方法。

(2) `static` 指明该方法是一个类方法, 它可以通过类名直接调用。

(3) `void` 则指明 `main()` 方法不返回任何值。

(4) `main()` 方法定义中的 `String[] args` 是传递给 `main()` 方法的参数, 参数名为 `args`, 它是 `String` 的一个数组, 参数可以为 0 个或多个, 多个参数之间用逗号分隔。

(四) 任务实施

任务实现的代码如下:

```
//任务 1_1:打印 Hello World
```

```
public class HelloWorld {
```

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

```
        System.out.println("Hello World");
```

```
    }
```

```
}
```

(五) 扩展内容

对于一个应用程序来说, `main()` 方法是必需的, 而且必须按照以上的格式来定义。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。

一个 Java 程序中可以定义多个类, 每个类中可以定义多个方法, 但是一个 Java 程序最多只有一个公共类, 一个类也只能有一个 `main()` 方法作为程序的入口。

Java 应用程序的运行过程和输出结果详见本章实训 2。