



教育部高等学校电子信息类专业教学指导委员会推荐教材

Practical Guide of Java Programming

Java程序设计 案例教程

孙一林 主编 彭波 等参编

Sun Yilin

Peng Bo



清华大学出版社



Practical Guide of Java Programming

Java程序设计案例教程

孙一林 主编
Sun Yilin

彭波 等参编
Peng Bo



清华大学出版社
北京

内 容 简 介

本书适合作为学习 Java 语言及编写 Java 应用程序的实用型教科书或教学参考书,书中主要阐述了 Java 语言的关键字、语句、语法规则、类库等的使用方法,其内容包括 Java 语言开发工具的介绍、面向对象编程、Java 基础语句、Java 类、Java 类的继承与多态、Java 接口、Java 包、Java 异常、Java 基础类库的应用,以及通过应用型案例掌握 Java 语言的输入和输出操作、图形用户界面操作、Applet 小程序的编写、图形绘制操作、多线程处理、图像动画制作、网络编程、数据库操作等相关类库的使用。

本书可作为大学、大专等理工科院校与计算机相关专业的本科或专科生使用的教材,也可作为学习 Java 语言的初、中级读者的参考用书。

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 程序设计案例教程/孙一林主编. --北京:清华大学出版社,2016

ISBN 978-7-302-42875-6

I. ①J… II. ①孙… III. ①JAVA 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2016)第 030577 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:时翠兰

责任印制:杨 艳

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:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24.75

字 数:600 千字

版 次:2016 年 6 月第 1 版

印 次:2016 年 6 月第 1 次印刷

印 数:1~2000

定 价:49.00 元

产品编号:061416-01

前言

PREFACE

本书的作者们都是多年从事 Java 语言教学和使用 Java 开发项目的教育工作者和软件工程师,对 Java 有着深入的理解,了解学生在学习 Java 语言时遇到的难点并知道如何使学生能够更快、更准确地掌握和使用 Java 语言,在编写过程中,能使书的内容尽量通俗、易懂,使其符合循序渐进、由浅入深的学习规则,帮助读者规范、系统地学习 Java 语言,为后续深入学习 Java 语言奠定扎实的基础。

由于笔者多年从事 Java 语言的教学工作,有些学习体会在此愿与读者交流,仅供读者学习参考。计算机语言是人与计算机交流时使用的语言,更确切地说是人命令计算机做某些事情的语言。计算机语言与人类交流使用的自然语言功能是一样的,也有单词(被称为关键字)、词组(被称为类库或函数库)、语法规则等。但是计算机语言的单词是有限的,因为计算机 CPU 中的指令译码器只能理解很少的单词,而语法规则也是规定好的,只可以按照语法规则来编写一些语句,否则计算机不会识别。掌握计算机语言的关键字和语法规则就可以编写计算机识别的可执行语句,而计算机程序就是一些语句遵循语法规则的逻辑组合,它可以让计算机完成特定的功能。顾名思义,计算机好像只能实现计算功能,但它为什么却能在各行各业?因为几乎所有的事物及事物的改变都可以用“数据”来描述,例如最熟悉的阿拉伯数字、ASCII 码(描述文字)、PCM 码(描述视频、音频信息)、表格、图等“广义数据”,处理这些“数据”正是计算机的专长。因此,编写计算机程序首先需要通过“数据”和对数据的操作来建立一些模型(称为数学模型),然后计算机执行程序则是针对数据的操作来实现要达到的目的。目前的面向对象编程的理念就是帮助实现或建立这样的模型,它可以使编写计算机程序变得简单。掌握面向对象编程思想是编写应用程序的基础,而 Java 语言就是适用于面向对象编程的语言。另外,一种计算机语言除了有关键字和规定的语法规则外,还提供了大量的类库或函数库,这些类库或函数库则是针对各种类型的“数据”实施处理的,在编写应用程序时,几乎所有需要处理的数据类型都可以在类库或函数库中找到处理方法,类库或函数库可以说包罗万象。总之学习计算机语言在理解了其使用的编程思想后,剩下的就是学习关键字、语法规则和类库或函数库的使用了。

本书分为两部分:第一部分包含 6 章,主要是讲解 Java 语言的关键字、由关键字组成的表达式、程序流控制、类定义、接口定义、对象的使用、异常的处理等语句以及 Java 语言的语法规则;第二部分包含 9 章,主要介绍关于 Java 基础类库 JFC 的应用,在 JFC 中每一个类库都是针对一类数据进行的操作,类库提供了许多数据操作方法,每一类操作都制定了操作机制,了解操作机制是更好地使用类库的前提。Sun 公司提供的 J2SDK 中类库使用说明书是最直接的使用类库的学习资料,本书则是通过案例理解类库的操作机制和使用方法,案例涉及的内容有计算机输入和输出操作、图形用户界面操作、Applet 小程序的编写、图形绘

制操作、多线程处理、图像动画制作、网络编程、数据库操作等,为读者深入学习各个应用领域中 Sun 公司提供的 Java API 奠定一些基础。

书中所有案例都是通过编译可运行的 Java 应用程序,并且实现了主要功能,读者稍加修改(添加一些辅助功能)就可以应用到实际项目中。出版社网站上会提供这些案例的源代码和编译后可执行类代码,方便读者学习和使用。

本书由北京师范大学信息科学与技术学院孙一林副研究员主编,参加编写与进行案例程序调试的还有中国农业大学信息与电气工程学院彭波教授、北京师范大学教育技术学院张庆利高级工程师、北京联合大学师范学院电子信息系曾文琪老师,以及胡治国、崔永普、张伟娜、王平等研究生,在此表示感谢。

由于水平有限,书中难免有疏漏之处,敬请读者批评指正。如果读者有问题需要与作者讨论,可发送电子邮件到: yilins@bnu.edu.cn。

作 者

2016 年 1 月

目录

CONTENTS

第一部分 基础原理篇

第 1 章 Java 程序设计基础	3
1.1 Java 程序与运行	3
1.1.1 Java 语言的特征	3
1.1.2 Java 虚拟机	3
1.1.3 Java 字节代码	4
1.1.4 Java 编译单元	5
1.2 Java 程序的开发	5
1.2.1 Java 语言程序开发平台	6
1.2.2 Java 程序基础开发运行平台——J2SDK	6
1.2.3 在 Windows 操作系统中搭建 Java 程序的开发和运行环境	7
1.2.4 Java 程序的编译和运行	7
1.3 Java 语言的标识符和关键字	9
1.3.1 Java 注释语句	9
1.3.2 Java 标识符	10
1.3.3 Java 关键字	11
1.4 Java 基本数据类型	12
1.4.1 数据值的表示法	13
1.4.2 变量	14
1.4.3 变量的数据类型转换	17
1.4.4 常量	18
1.5 Java 运算符和表达式	18
1.5.1 Java 运算符	18
1.5.2 Java 表达式	20
1.5.3 表达式中运算符的使用规则	21
1.6 Java 程序流控制语句	24
1.6.1 分支结构语句	24
1.6.2 循环结构语句	26
1.6.3 辅助流控制语句	29
1.6.4 流控制语句应用示例	32
1.7 小结	33
1.8 习题	33

第 2 章 面向对象编程	35
2.1 面向对象程序设计	35
2.1.1 面向对象的程序设计方法	35
2.1.2 Java 面向对象程序设计	36
2.2 Java 类	37
2.2.1 Java 类的定义	37
2.2.2 方法的定义	38
2.2.3 Java 修饰符及其权限	41
2.3 Java 对象	42
2.3.1 对象的创建	42
2.3.2 构造方法的使用	44
2.3.3 对象的使用	44
2.3.4 对象的清除	47
2.3.5 Java 类和对象的关系	48
2.4 数组对象	48
2.4.1 一维数组的声明和创建	48
2.4.2 多维数组的声明和创建	51
2.4.3 数组的应用	53
2.5 小结	56
2.6 习题	56
第 3 章 Java 类的继承与多态	58
3.1 Java 类的继承	58
3.1.1 概念和语法	58
3.1.2 Java 类继承关系的测试	60
3.1.3 隐藏、覆盖和重载	61
3.1.4 构造方法的重载	64
3.2 abstract 和 final 修饰符	66
3.2.1 abstract 修饰符	66
3.2.2 final 修饰符	67
3.3 this 和 super 变量	68
3.3.1 this 变量	69
3.3.2 super 变量	71
3.4 Java 的多态性	73
3.4.1 多态的概念	73
3.4.2 多态的应用	76
3.4.3 构造方法与多态	77
3.5 小结	78
3.6 习题	79
第 4 章 Java 接口和 Java 包	83
4.1 Java 接口	83
4.1.1 接口的定义	83
4.1.2 接口的实现	84

4.1.3	接口的继承	86
4.1.4	Java 类同时继承父类并实现接口	87
4.1.5	接口与 Java 抽象类	88
4.1.6	接口的应用	89
4.2	Java 包	93
4.2.1	package 语句	93
4.2.2	Java 包与路径	94
4.2.3	import 语句	95
4.2.4	直接引用 Java 包中的类和接口	97
4.2.5	Java 包的应用	98
4.3	小结	100
4.4	习题	101
第 5 章	Java 异常处理	105
5.1	Java 异常处理机制	105
5.1.1	异常的类型	105
5.1.2	异常处理机制	105
5.1.3	Java 的异常处理	106
5.2	Java 异常的捕获与处理	107
5.2.1	try-catch 语句	107
5.2.2	finally 语句	111
5.3	Java 异常的抛出	113
5.3.1	从方法体中抛出异常对象	113
5.3.2	针对被抛出的异常对象的处理	114
5.4	Java 基础包中定义的常用异常类	115
5.4.1	异常类的根类与直接子类	115
5.4.2	java.lang 包中定义的具体异常类	118
5.5	自定义异常类	120
5.5.1	异常类定义规则及抛出	121
5.5.2	捕获自定义异常对象	121
5.6	小结	125
5.7	习题	125
第 6 章	Java 基础类的应用	129
6.1	java.lang 包	129
6.2	Object 类	130
6.3	基本数据类型类	132
6.3.1	整型类	132
6.3.2	浮点类	134
6.3.3	其他常用类	136
6.4	字符串 String 类	138
6.4.1	String 类	138
6.4.2	创建 String 对象并对其进行操作	140
6.4.3	StringBuffer 类	141
6.4.4	创建 StringBuffer 对象并对其进行操作	143

6.5	Math 类	144
6.6	Runtime 类	146
6.7	System 类	148
6.8	小结	150
6.9	习题	150

第二部分 案例应用篇

第 7 章	Java 输入和输出操作案例	155
7.1	Java 的输入、输出机制	155
7.1.1	Java 数据流传输模式	155
7.1.2	Java 数据流的主要操作类	156
7.2	控制台输入、输出操作案例	156
7.3	文件输入、输出操作案例	159
7.3.1	字节流文件输入、输出操作	160
7.3.2	字符流文件输入、输出操作	163
7.4	文件随机读写操作案例	170
7.5	对象序列化传输案例	172
7.6	小结	174
7.7	习题	174
第 8 章	Java 图形用户界面设计案例	178
8.1	构成 GUI 的组件	178
8.1.1	Java 组件类	178
8.1.2	组件属性控制	179
8.1.3	GUI 的组成	179
8.2	组件事件处理	179
8.2.1	Java 组件事件监听处理机制	179
8.2.2	Java 组件事件监听标准程序代码	181
8.3	java.awt 包中组件应用案例	183
8.3.1	鼠标操作应用案例	183
8.3.2	键盘操作应用案例	187
8.4	javax.swing 包中组件应用案例	190
8.4.1	修改组件属性案例	190
8.4.2	记事本应用程序案例	192
8.4.3	Excel 表格文件内容显示案例	195
8.5	小结	199
8.6	习题	199
第 9 章	Applet 小程序案例	201
9.1	Applet 类及 Applet 小程序	201
9.1.1	Applet 小程序类	201
9.1.2	Applet 小程序编程框架	201
9.2	Applet 小程序的运行机制	204
9.3	Java 程序 Application 和 Applet	205

9.4	Applet 小程序应用案例	206
9.4.1	显示外部参数 Applet 小程序	206
9.4.2	显示时间 Applet 小程序	208
9.4.3	播放声音 Applet 小程序	210
9.4.4	Applet 小程序界面添加菜单	212
9.5	小结	213
9.6	习题	214
第 10 章	Java 基础绘制图形案例	216
10.1	Java 基础图形绘制功能	216
10.1.1	Graphics 图形类	216
10.1.2	绘图坐标体系	216
10.1.3	Graphics 类中主要绘图操作方法	217
10.2	Java 图形绘制案例	219
10.2.1	绘制各种图形和图像	219
10.2.2	绘制数学函数图形	221
10.2.3	绘制直方图	223
10.2.4	绘制文字	225
10.2.5	简单绘图程序	226
10.3	小结	231
10.4	习题	231
第 11 章	Java 高级图像处理案例	233
11.1	Java 2D 绘制图形案例	233
11.1.1	二维图形的绘制机制	233
11.1.2	绘制二维图形案例	234
11.2	Java 2D 图形、文字处理案例	236
11.2.1	二维图形后期处理案例	236
11.2.2	二维文字后期处理案例	239
11.3	Java 2D 图像处理案例	241
11.3.1	二维图像处理机制	241
11.3.2	二维图像边缘检测案例	244
11.3.3	二维图像综合处理案例	246
11.4	小结	256
11.5	习题	256
第 12 章	Java 多线程应用案例	261
12.1	线程	261
12.1.1	Runnable 接口和 Thread 类	261
12.1.2	创建启动线程对象	262
12.1.3	创建具有多线程功能的 Applet 小程序	264
12.2	Java 多线程机制	264
12.2.1	线程对象的生命周期和状态	265
12.2.2	线程对象的基本控制	265
12.2.3	多线程问题	266

12.2.4	线程间的同步控制机制	267
12.3	多线程应用程序案例	269
12.3.1	Thread 类中的 sleep() 方法	269
12.3.2	Object 类中的线程控制方法	270
12.3.3	账户数据操作问题	274
12.3.4	实时时钟显示 Applet 小程序	275
12.3.5	滚动显示文字信息 Applet 小程序	277
12.4	小结	280
12.5	习题	281
第 13 章	Java 动画制作案例	283
13.1	简单图形动画制作案例	283
13.2	文字动态显示案例	287
13.3	图像动态显示案例	289
13.3.1	动态显示多幅图像	289
13.3.2	单幅图像变形动态显示	291
13.4	图像缓冲技术动态显示案例	293
13.4.1	缓冲技术	293
13.4.2	利用缓冲技术实现动态显示图像案例	295
13.5	小结	298
13.6	习题	298
第 14 章	Java 网络应用案例	301
14.1	URL 通信	301
14.1.1	创建并连接 URL 对象	301
14.1.2	获取网络资源案例	303
14.1.3	Web 服务器提供 HTTP 服务案例	308
14.2	Socket 通信	310
14.2.1	建立服务器和客户机 Socket 通信程序框架	310
14.2.2	Socket 通信案例	316
14.2.3	网络聊天室程序案例	323
14.3	UDP 通信	326
14.3.1	建立 UDP 通信程序框架	327
14.3.2	UDP 通信案例	330
14.4	小结	336
14.5	习题	336
第 15 章	Java 数据库应用案例	341
15.1	JDBC 概述	341
15.1.1	JDBC API	341
15.1.2	JDBC 的组成	342
15.1.3	JDBC 的任务	343
15.2	数据库操作命令 SQL	343
15.2.1	创建、删除数据库	344
15.2.2	创建、删除、修改基本表格	344

15.2.3	创建、删除索引	345
15.2.4	创建、删除视图	346
15.2.5	数据查询	346
15.2.6	数据更新	346
15.3	创建 Java 数据库应用模型	347
15.3.1	创建数据源	347
15.3.2	加载数据库驱动程序	350
15.3.3	连接数据库	351
15.3.4	操作数据库	352
15.3.5	获取数据结果集	353
15.4	JDBC API 应用案例	354
15.4.1	显示查询数据库结果	354
15.4.2	向数据库中追加记录	358
15.4.3	SQL 命令操作数据库	363
15.4.4	Applet 数据库应用案例	375
15.5	小结	382
15.6	习题	382

第一部分

PART I

基础原理篇

计算机语言是人与计算机沟通的一种形式,更确切地说是人命令计算机做一些事的语言,但是要实现沟通的目的,需要程序设计员遵循计算机程序的语法结构来设计。Java 语言源程序由 Java 语言的基本语句构成,其程序结构与其他计算机语言的程序结构一样,都具有有一些基本特性,例如计算能力、运算优先顺序、类型转换及选择和循环等程序流的控制等。

本章重点介绍 Java 语言的关键字、数据类型、表达式、程序流控制,以及由 Java 关键字、表达式等组成的 Java 程序语句和语句的语法格式。

1.1 Java 程序与运行

Java 语言是由 C 语言发展而来的,其继承了 C 语言的所有优良特征,并去除了 C 语言的一些容易引起错误的语句,是一个彻底的纯面向对象的程序设计语言,最重要的是 Java 的可执行代码不依赖于任何现有的操作系统和计算机中的 CPU(中央处理器)。

1.1.1 Java 语言的特征

Java 语言的特征可以高度概括为:Java 是一个简单的、面向对象的、分布的、解释的、健壮的、安全的、独立于平台的、可移植的、可扩展的、高性能的、多线程的及动态的程序设计语言。

Java 是一个编译和解释型语言,Java 的源程序代码通过 Java 编译器(Sun 公司提供的 Java 语言程序开发工具)将 Java 语言文本代码编译为(翻译为)Java 字节代码(称为 Java 的可执行代码),该代码是运行在 Java 虚拟机环境中的。Java 程序的编译和解释执行的特性如图 1-1 所示。

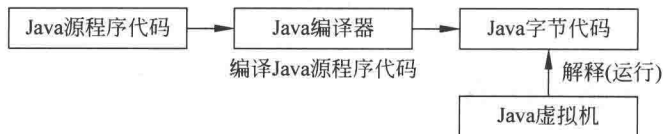


图 1-1 Java 语言的编译和解释执行的特性

1.1.2 Java 虚拟机

Java 虚拟机(Java Virtual Machine,JVM)是运行 Java 程序的支持系统,用于解释 Java 字节代码并执行代码所要完成的操作。Java 虚拟机是与计算机硬件和操作系统相关联的,是由发明 Java 语言的 Sun 公司根据不同的 CPU 或计算机操作系统制作而成的,即针对不同的 CPU 或操作系统解释 Java 字节代码,将 Java 字节代码翻译为指定的 CPU 或操作系

统可识别并能够执行的指令代码,也就是将 Java 字节代码转化成实际硬件设备的调用。因此,Java 虚拟机针对指定的 CPU 或操作系统也被称为 Java 解释器,专门用于支持 Java 程序运行,是面向 Java 语言程序的一个独立运行系统(Java Runtime System)。另外,Java 虚拟机未必非要运行在某种操作系统之上,其下面可以直接是各种 CPU 芯片,对于每一种操作系统或 CPU,Java 解释器是不同的,但实现的 Java 虚拟机功能是相同的,这就是 Java 语言程序与平台无关的关键所在。

Java 语言应用程序分为两种形式:Application 应用程序和 Applet 小程序。Java 源程序通过 Java 编译器生成可执行的 Java 字节代码,Application 应用程序是由 Java 虚拟机支持运行的,Applet 小程序是由嵌入了 Java 虚拟机的 Web 浏览器支持运行的,目前几乎所有的 Web 浏览器都嵌入了 Java 虚拟机,计算机 CPU 和 Web 浏览器支持 Java 应用程序运行的层次关系如图 1-2 所示。

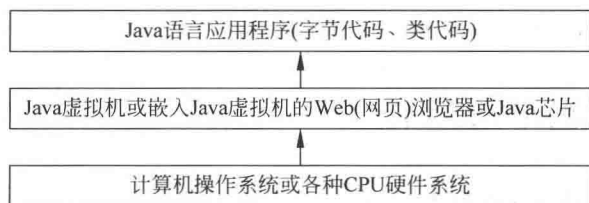


图 1-2 支持 Java 应用程序运行的层次关系图

Java 虚拟机由 5 个部分组成:一组指令集、一组寄存器、一个堆栈、一个无用内存单元收集器、一个方法区域。这 5 部分是 Java 虚拟机逻辑抽象成分,不依赖于任何实现技术或组织,但是它们的功能在某种真实机器上(确定的 CPU 或操作系统)是一致的。Java 虚拟机也有类似微处理器(CPU)中的寄存器,它们用于保存机器的运行状态。Java 虚拟机的寄存器有 4 种:Java 语言程序计数器(pc)、指向操作数栈顶端的指针(optop)、指向当前执行方法的执行环境指针(frame)、指向当前执行方法局部变量区第一个变量的指针(var)。Java 虚拟机是用堆栈来处理数据传递的,它不定义或使用寄存器来传递或接收参数,其目的是为了保证指令集的简洁性和实现时的高效性。Java 虚拟机的堆栈有 3 个区域:局部变量区、执行环境区和操作数区。Java 虚拟机的堆栈是 Java 应用程序运行时动态分配的对象存储区域并是可自动回收的,Java 虚拟机的内部结构和运行机制都是围绕着解释 Java 字节代码而制定的。

1.1.3 Java 字节代码

Java 虚拟机可以解释 240 多个字节代码,每个字节代码的执行相当于一种基础的 CPU 运算或操作。Java 字节代码也可以称为 Java 语言指令集,相当于 Java 语言程序的“汇编语言”。Java 语言指令集的一条指令有一个操作码、0 个或多个操作数,操作码指定实施的操作,操作数是操作码所需的参数,Java 语言的操作码均为一个字节长,操作数没有限制。由 Java 字节代码组成的指令是不依赖于任何 CPU 或操作系统的,而是针对 Java 虚拟机设计的。Sun 公司提供的 Java 语言程序开发环境中的 Java 编译器可将 Java 源程序代码编译成 Java 字节代码指令,这些指令被 Java 的运行系统(Java 虚拟机)有效地解释并运行。另外,Java 字节代码也被称为类(class)代码。

1.1.4 Java 编译单元

Java 语言源程序是由一个或多个 Java 编译单元(compilation unit)组成的。Java 编译单元由 Java 语句组成,每个编译单元只能包含下列语句内容:一个程序包语句(package statement)、一组导入语句(import statements)、一个或多个类的声明语句(class declarations)、一个或多个接口声明语句(interface declarations)。每个 Java 的编译单元可包含多个类或接口,但是每个编译单元最多只能有一个类或者接口是公共的。

Java 编译单元是以公共的类或者接口名为文件名,以 .java 为后缀保存在磁盘中的,Java 编译器则是针对 *.java 文件实施编译的,编译后可生成存放在磁盘中的 Java 字节代码,字节代码是以 .class 为后缀的文件名(*.class)形式存放的,在一个 Java 编译单元中如果定义了多个类和接口,则编译后每个类和接口都是以独立的类文件名(*.class, * 为类或接口名)存放在磁盘中的。

【示例 1-1】 Java 编译单元的程序源代码,包含两个类,但只有一个被修饰为公共(public)类。

```
package helloworld;           //一条包语句
import somepackage. *;       //引用语句
public class HelloWorld {     //类声明语句,文件名为 HelloWorld.java
    ...;                      //程序要实现的功能
}                             //类声明结束
class AnotherClass {         //其他类声明语句
    ...;                      //程序要实现的功能
}                             //类声明结束
```

该编译单元以 HelloWorld.java(公共类名)为文件名存放在磁盘中,通过 Java 编译器编译后在磁盘中将产生 HelloWorld.class 和 AnotherClass.class 两个类文件,它们组成了 HelloWorld.java 编译单元的可执行字节代码。

【示例 1-2】 包含一个公共(public)接口的一个编译单元的程序源代码。

```
package helloworld;           //一条包语句
import somepackage. *;       //引用语句
public interface InterfaceName { //接口声明语句,文件名为 InterfaceName.java
    ...;                      //接口定义的内容
}                             //接口声明结束
```

该编译单元以 InterfaceName.java(公共接口名)为文件名存放在磁盘中,通过 Java 编译器编译后将在磁盘中将产生以 InterfaceName.class 为文件名的类文件。

Java 编译单元中的源程序框架结构是确定的,Java 程序要实现的功能都应该编写在类或接口中,源程序代码将由标识符、关键字、变量定义、表达式、流控制语句等按照 Java 语法规则组成,它们都应编写在 Java 编译单元中类或接口定义的大括号{}中。

1.2 Java 程序的开发

J2SDK 是 Java 2 Software Development Kit 的简称,其前身是 JDK(Java Development Kit),J2SDK 和 JDK 都是 Sun Microsystems 公司推出的一套 Java 语言程序开发工具兼作