



普通高等教育“十一五”国家级规划教材



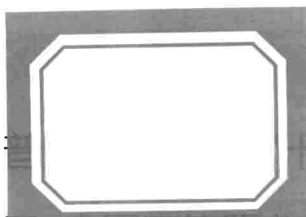
中国高等职业技术教育研究会推荐
高职高专系列规划教材

Java 程序设计 (第三版)

陈圣国 王葆红 编著



西安电子科技大学出版社
<http://www.xduph.com>



“九五”国家级规划教材

中国高等职业技术教育研究会推荐

高职高专系列规划教材

Java 程序设计

(第三版)

陈圣国 王葆红 编著

西安电子科技大学出版社

内 容 简 介

本书首先简要介绍了 Java 语言开发平台的使用方法以及 Java 语言的基本语法和面向对象程序设计的基本概念,然后重点介绍了 Swing 组件库构建图形用户界面应用程序的方法及常用组件的使用,还对多线程、异常处理、网络和数据库编程等内容逐一进行了介绍。

本书采用案例导入方式,给出大量实例,便于学生模仿学习,适合作为应用型本科计算机相关专业 Java 语言程序设计课程的教材,也可供有一定程序设计语言基础而想学习 Java 语言的读者使用。

★ 本书配有电子教案,需要者可登录出版社网站,免费下载。

图书在版编目(CIP)数据

Java 程序设计/陈圣国,王葆红编著. —3 版. —西安:西安电子科技大学出版社,2014.7

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5606-3413-5

I. ① J… II. ① 陈… ② 王… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312

中国版本图书馆 CIP 数据核字(2014)第 140152 号

策 划 马乐惠

责任编辑 马乐惠 高丽萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2014 年 7 月第 3 版 2014 年 7 月第 4 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 18.5

字 数 435 千字

印 数 14 001~17 000 册

定 价 29.00 元

ISBN 978-7-5606-3413-5/TP

XDUP 3705003-4

如有印装问题可调换

本社图书封面为激光防伪覆膜,谨防盗版。

前 言



本书第二版自 2008 年出版以来,在几年来的使用中暴露出一些不足。本次修订除了对原书的一些错误作了修正外,主要从以下几个方面做了修改:

(1) 实验平台的改变。本书采用目前主流的 **Java** 开源开发平台 **Eclipse**,它有助于学生较快接触实际的工作环境。另外,通过安装相应插件 **Eclipse** 也可以支持 **JavaEE**、**Android** 移动应用的开发,可为学生进一步学习 **JavaEE**、**Android** 移动应用的开发打好基础。

(2) 介绍了 **Java5** 之后引入的泛型语法,并简单介绍了泛型语法在集合类库的典型应用,介绍了链表等集合类型的使用方法。

(3) 图形用户界面的使用以 **Swing** 组件库的使用为主,不再详细介绍 **AWT** 的使用。另外,随着 **Web** 开发技术的发展,**Java Applet** 的重要性越来越低,因此图形用户界面以应用程序为主,**Java Applet** 的开发不再作为主要内容。

本书在修订的过程中得到了西安电子科技大学出版社马乐惠老师的关心与支持,在此谨表谢意。

由于时间紧张和作者水平有限,本书还存在不足,希望广大读者批评指正。

作 者

2014 年 2 月

第二版前言



本书第一版自 2003 年出版以来,被众多高职院校选为“Java 程序设计”课程的教材,但在几年来的使用中也暴露出一些不足;另外,随着 Java 语言的发展,原书部分内容也不再适合需要。本次修订除了对原书的一些错误进行了修正外,主要从以下几个方面做了修改:

(1) 实验平台的改变。本书第一版采用的 Java 集成开发环境为微软公司的 Visual J++ 6.0,该平台目前已不再是主流的 Java 开发平台,微软也不再将其后继版本作为 Java 应用的开发平台。因此本次修订实验平台不再采用 Visual J++,而是选用了 Borland 公司的 Jbuilder 2006,该平台是目前主流的商业 Java 开发平台之一,被许多软件公司采用。选用该平台,有助于学生较快适应实际工作环境。

(2) 对 Java 面向对象基本语法部分内容的顺序进行了调整,围绕面向对象方法的三个基本特征——封装、继承和多态循序渐进地进行介绍,并强化案例的作用,使学生更易于接受。

(3) 使用了少量 Java 5 的语法。Java 语言在不断发展,每个新版本的 JDK 的推出都会引入一些新的特征。本书根据实际需要,引入了少量新的特征。

(4) 适当地介绍了 Web 应用开发。随着 Internet 的发展,Java 语言逐渐成为一个主流的 Web 开发工具,本书第 10 章简要介绍了使用 JSP 开发动态 Web 应用的方法。

本书在修订的过程中得到了西安电子科技大学出版社马乐惠老师的关心与支持,在此谨表谢意。

由于时间和作者水平有限,书中可能还存在一些不足之处,敬请读者批评指正。

作 者

2007 年 10 月

第一版前言



Java 语言作为一种基于网络的程序设计语言,自 Sun 公司 1995 年推出以来,得到了越来越多软硬件厂商的支持。Java 语言具有面向对象、跨平台、安全性高等特点,已成为开发互联网应用的首选语言之一。

本书第 1 章简要介绍了 Java 语言的特点以及 Sun 公司的 Java 开发工具包 (JDK) 的最新版本——J2 SDK 1.4.2,还介绍了利用 Microsoft 公司的 Visual J++ 6.0 调试 Java 应用程序和 Java 小程序的步骤。

第 2 章介绍了 Java 语言的数据类型、表达式和流程控制语句。Java 语言吸收了 C/C++ 语言的基本语法,它在这几个方面的内容与 C 语言基本相同。大部分高校目前都将 C/C++ 作为计算机专业首选的程序设计语言,因此本书未作详细介绍,读者在学习时应特别注意 Java 语言与 C/C++ 的区别。

从第 3 章开始详细介绍了 Java 语言面向对象程序设计的语言成分,然后重点介绍了 Java 小程序的编写方法,特别是图形界面组件的使用,还对多线程、异常处理、网络和数据库编程等内容逐一进行了介绍。

关于图形用户界面,本书仅介绍了基本的抽象窗口工具包——AWT 图形组件的使用方法。由于 Swing 图形用户界面组件实际上是在 AWT 组件的基础上发展起来的,使用上有一定的相似之处,因此本书对 Swing 编程技术未作介绍。读者只要熟练掌握了 AWT 图形用户界面的使用方法,再学习 Swing 编程技术并不困难。

由于 Java 语言是一种基于网络的编程语言,特别是 Java Applet 与网络的联系更加密切,因此在学习 Java 语言之前,最好有一定的网络课程的基础或使用互联网的经验。限于篇幅,本书对相关的背景知识没有作详细的介绍。

另外,学习本书第 9 章 JDBC 编程接口前,应对关系数据库系统的基本原理有一定的了解,也要了解客户机/服务器结构数据库应用的基本结构。

本书在每一章都安排了实训内容，有关操作的具体步骤均以 Windows 98 平台为蓝本介绍。第 4 章的实训中安排了在 Web 服务器上发布 Java Applet 的内容，介绍了个人 Web 服务器 PWS 的使用，如果读者使用其他的实验平台应参考相关平台的使用手册。在进行该项实训时，如果条件允许，应安排在局域网的多台计算机上进行。

本书在编写的过程中，特别是在内容选择和结构安排等方面得到了西安电子科技大学出版社马乐惠老师等的大力帮助，在此谨表谢意。

由于时间和作者水平限制，书中可能还存在一些不足之处，敬请广大读者不吝赐教。

作 者

2003 年 8 月

目 录

第 1 章 Java 语言概述	1	2.4.6 位运算符	27
1.1 计算机与程序设计语言	1	2.4.7 条件运算符	28
1.2 Java 的发展历史	2	2.4.8 类型转换	28
1.3 跨平台的 Java	3	2.5 案例的初步实现	29
1.4 面向对象的 Java	4	2.6 流程控制语句	29
1.5 Java 程序的开发环境	4	2.6.1 分支语句	30
1.5.1 Java 程序的开发过程	4	2.6.2 循环语句	33
1.5.2 Java 开发工具	5	2.6.3 break 语句	34
1.6 最简单的 Java 程序	5	2.6.4 continue 语句	35
1.6.1 最简单的 Java 应用程序	5	2.6.5 其他流程控制语句	36
1.6.2 最简单的 Java Applet	6	2.7 案例的完整实现	36
实训一 安装与熟悉 Java 开发工具	8	2.8 程序举例	37
A. JDK 开发工具	8	实训二 结构化程序设计	38
B. Eclipse 集成开发环境	11	习题二	41
习题一	16		
第 2 章 Java 语言基础	17	第 3 章 类与对象	45
2.1 案例：课程成绩统计程序	17	3.1 面向对象的基本思想和基本概念	45
2.2 变量和常量	17	3.1.1 面向对象的基本思想	45
2.2.1 变量的定义	17	3.1.2 对象与类	45
2.2.2 常量的定义	19	3.1.3 封装性、继承性与多态性	46
2.3 数据类型	19	3.2 案例：员工工资计算程序	48
2.3.1 整型	19	3.3 类的声明与对象的创建	48
2.3.2 实型	20	3.3.1 类声明的基本语法	48
2.3.3 字符型	20	3.3.2 类的构造方法与对象的初始化	53
2.3.4 布尔型	21	3.3.3 对象的使用	56
2.4 运算符与表达式	22	3.3.4 案例的初步实现	57
2.4.1 概述	22	3.4 封装性	61
2.4.2 算术运算符	22	3.4.1 成员的访问权限	61
2.4.3 赋值运算符	24	3.4.2 包的使用	63
2.4.4 关系运算符	25	3.5 继承性	65
2.4.5 逻辑运算符	25	3.5.1 子类的定义	65
		3.5.2 super	66

3.5.3 子类对象的构造	68	4.2 菜单与对话框	136
3.5.4 final 方法与 final 类	69	4.2.1 案例 2: 简易文本编辑器	136
3.5.5 改进的案例	69	4.2.2 建立主菜单	137
3.6 多态性	73	4.2.3 快捷菜单的使用	142
3.6.1 类内方法的重载	73	4.2.4 自定义对话框	142
3.6.2 类继承中的多态性	75	4.2.5 JOptionPane 标准对话框	144
3.6.3 接口	79	4.3 Swing 常用组件简介	147
3.6.4 案例的进一步改进	80	4.3.1 Swing 组件分类	147
3.7 静态成员	82	4.3.2 JFrame 与 JDialog	147
3.7.1 静态变量成员	82	4.3.3 文本显示和编辑组件	148
3.7.2 静态方法成员	84	4.3.4 命令按钮	152
3.8 字符串	85	4.3.5 复选框与单选按钮	153
3.8.1 创建 String 类对象	85	4.3.6 下拉列表	154
3.8.2 获取字符串的信息	85	4.3.7 列表框	157
3.8.3 字符串的操作	86	4.3.8 工具栏	159
3.8.4 StringBuffer 类	87	4.3.9 面板	160
3.9 数组	88	4.4 Applet 与图形界面	162
3.9.1 一维数组	89	4.4.1 Applet 程序结构	162
3.9.2 多维数组	91	4.4.2 HTML 中使用 Applet	164
3.9.3 案例的完整实现	93	实训四 图形用户界面的实现	166
3.10 包装类	95	习题四	168
3.10.1 包装类的概念	95		
3.10.2 字符串与基本类型的转换	95		
3.11 编程实例	96		
3.12 泛型与集合类简介*	99		
3.12.1 泛型的作用	99		
3.12.2 泛型的基本语法	100		
3.12.3 集合类的使用	102		
实训三 面向对象程序设计	107		
习题三	113		
第 4 章 图形用户界面	117	第 5 章 多线程	170
4.1 进入图形用户界面	117	5.1 线程的概念	170
4.1.1 案例 1: 图形界面的简易 计算器	117	5.1.1 线程与多线程	170
4.1.2 容器与组件	117	5.1.2 进程与线程	170
4.1.3 组件的布局	120	5.1.3 线程的优先级与类别	170
4.1.4 响应组件的事件	129	5.1.4 线程的状态与生命周期	171
		5.2 多线程的实现方法	171
		5.2.1 线程类 Thread	171
		5.2.2 继承 Thread 类	173
		5.2.3 实现 Runnable 接口	174
		5.3 采用多线程实现动画效果	176
		5.4 线程的同步与死锁	177
		5.4.1 同步的概念	177
		5.4.2 synchronized 方法	178
		5.4.3 synchronized 块	180
		5.4.4 线程的死锁	181

实训五 多线程程序设计	183	8.1.2 网络编程的基本方法	214
习题五	184	8.2 URL 编程	214
第 6 章 异常处理	185	8.2.1 URL 的概念	214
6.1 异常的概念	185	8.2.2 URL 类	215
6.1.1 案例: 异常处理方法演示	185	8.2.3 URLConnection 类	225
6.1.2 异常处理	186	8.2.4 URL 编程实例	226
6.2 Java 语言异常的处理	187	8.3 Socket 编程简介	233
6.2.1 try-catch 块	187	8.3.1 TCP Socket 编程	233
6.2.2 异常的抛掷	189	8.3.2 UDP Socket 编程	238
6.2.3 实例	189	8.3.3 Socket 编程实例	241
6.3 异常的类型	191	实训八 用 Java 实现网络通信	245
6.3.1 Java 异常类层次	191	习题八	246
6.3.2 创建自己的异常类	192	第 9 章 JDBC 编程技术	247
实训六 处理并创建异常	193	9.1 JDBC 概述	247
习题六	193	9.1.1 JDBC 的概念	247
第 7 章 输入/输出	195	9.1.2 JDBC URL	248
7.1 流和文件	195	9.1.3 JDBC 驱动程序	248
7.1.1 流	195	9.2 使用 JDBC 开发数据库应用	249
7.1.2 文件	196	9.2.1 一个完整的例子	249
7.2 基本输入/输出类	196	9.2.2 一般步骤	252
7.2.1 InputStream 类	196	9.2.3 JDBC 相关类介绍	254
7.2.2 OutputStream 类	197	实训九 数据库应用程序开发	260
7.2.3 PrintStream 类	197	习题九	267
7.2.4 其他常用流类	198	第 10 章 Web 应用入门	269
7.3 文件的输入/输出	201	10.1 Web 服务器与 Web 应用	269
7.3.1 FileInputStream 类	201	10.2 Tomcat Web 服务器	269
7.3.2 FileOutputStream 类	201	10.3 JSP 简介	273
7.3.3 RandomAccessFile 类	203	10.4 案例: 网上书店查询页面	274
7.3.4 File 类	205	10.4.1 功能需求	274
7.4 编程实例	208	10.4.2 创建 books.jsp 页面	276
实训七 输入/输出的实现	209	10.4.3 创建 bookInf.jsp 页面	278
习题七	210	实训十 简易 Web 应用	280
第 8 章 网络编程概述	213	习题十	285
8.1 概述	213	参考文献	286
8.1.1 网络技术基础	213		

第1章 Java 语言概述

1.1 计算机与程序设计语言

1946年2月15日，第一台通用电子数字计算机 ENIAC 在美国研制成功。它由 1.8 万个电子管组成，重达 30 多吨。ENIAC 无论功能还是运算速度都无法跟今天的家用电脑相比，但它的出现却开启了一个新的时代。

计算机作为一种通用工具，与人类历史上发明的各种工具相比，一个突出的不同就是可以编程控制，通过执行不同的程序，计算机可以实现不同的功能。程序是计算机能够执行的指令序列，程序员可以使用不同的程序设计语言来编写程序，其中有一些程序设计语言计算机能够直接识别，而另一些程序设计语言需要经过翻译才能为计算机所执行。

程序设计语言的发展经历了从机器语言、汇编语言到高级语言的历程。

1) 机器语言

计算机所使用的是由“0”和“1”组成的二进制数，计算机发明之初，人们只能使用一串串由“0”和“1”组成的指令序列来编写程序，这种语言称为机器语言。机器语言难以使用，程序的调试和修改十分困难。由于不同型号计算机的指令系统往往不同，在一台计算机上执行的程序，要想在另一台计算机上执行，必须另编程序，造成了重复工作。

2) 汇编语言

针对机器语言的缺点，人们进行了一种有益的改进，采用一些类似于英语单词的缩写代替计算机的各种二进制指令，如“ADD”代表加法，“MOV”代表数据传递等。这样一来，程序变得易于理解和维护。这些缩写便构成了基本的汇编语言。用汇编语言编写的程序不能直接由计算机执行，需要一个翻译程序将这些符号翻译成二进制的机器语言，这种翻译程序被称为汇编程序。

3) 高级语言

虽然利用汇编语言编写程序的效率远高于其他语言，但是对于一项简单的任务，仍需要大量的指令才能完成，所以使用汇编语言编程调试来实现一个复杂应用依然是一件很痛苦的事情。

“痛”则思变，高级语言应运而生。1954年，第一个完全脱离机器硬件的高级语言——FORTRAN 问世了，它采用接近于数学语言或人的自然语言的语法形式，同时又不依赖于计算机硬件，编程效率和程序的通用性得到很大的提高。50多年来，已出现的高级语言有几百种，影响较大的也有几十种。

高级语言程序本身不能直接为计算机所执行，必须由专门的编译程序将高级语言编写

的程序转换为一个或多个包含了机器语言的文件。高级程序经过编译后，链接程序将包含了机器语言的文件链接成一个计算机可以运行的程序，采用这种方式工作的高级语言称为编译型语言。

高级语言的另一种工作方式是由解释器直接执行高级语言程序，与编译型语言方式相比，解释器方式的执行效率比较低。

1.2 Java 的发展历史

1995 年 5 月，Sun 公司在“SunWorld 95”大会上推出了 Java 语言。Java 语言作为一种网络编程语言，随着国际互联网的飞速发展，很快得到了广泛的支持和实际的应用。

其实，Java 语言最初并不是为互联网设计的，它来自于 Sun 公司的一个叫 Green 的项目，目的是为家用消费电子产品开发一个分布式代码系统，这样用户可以把 E-mail 发给电冰箱、电视机等家用电器，对它们进行控制和信息交流。该项目小组开始准备采用 C++，但他们很快意识到 C++ 太复杂，安全性差，后来基于 C++ 开发了一种新的语言——Oak(Java 的前身)。

Oak 是一种用于网络的精巧而安全的语言，Sun 公司曾以此投标一个交互式电视项目，但未获成功，使得 Oak 几乎夭折。Marc Andreessen 开发的 Mosaic 和 Netscape 启发了 Oak 项目组成员，他们用 Java 编制了 HotJava 浏览器，得到了 Sun 公司首席执行官 Scott McNealy 的支持，Java 得以进军互联网。为此，Sun 公司引入了一个称为 Applet 的 Java 程序创建机制，用以在 Web 页上执行并通过 Web 浏览器进行显示，使得原本静态的网页“活跃起来”。

Java 的取名也有一段趣闻。有一天，几位 Oak 项目组成员正在讨论给这个新的语言取什么名字，当时他们正在咖啡馆喝着 Java 咖啡，有一个人灵机一动说，就叫 Java 怎么样，结果得到了其他人的赞同，于是 Java 这个名字就这样传开了。

目前通常所说的 Java 有三层含义：先是指一种编程语言；其次是一种开发环境；再者是一种应用环境。如今的 Java 语言不再只是将 Web 网页“活跃起来”的一门语言，它已成为许多机构编程时的首选语言。

作为新一代面向对象程序设计语言，Java 特别适合于 Internet 应用程序开发，它的平台无关性使 Java 作为软件开发的一种革命性技术地位已被确立。计算机产业的许多大公司购买了 Java 的许可证，包括 IBM、Microsoft、Apple、Oracle 等。Java 开发工具软件日渐丰富，数据库厂商如 Sysbase、Versant、Oracle 都提供对 Java 平台的支持。

Sun 公司于 1996 年年初发布了 Java 的第一个版本，在 1998 年召开的 JavaOne 大会上发布了 Java 1.2 版。1999 年，Sun 公司发布了以 Java 2 平台为核心的 J2SE、J2EE 和 J2ME 三大平台；2002 年 2 月，Sun 公司发布了 J2SE 1.4 版，是 Java 语言早期影响最大的版本；2004 年 10 月，J2EE 1.5 版的发布引入了一些新的特性，Sun 公司将其正式称为 Java 5，相应的三个平台分别改称为 Java SE、Java EE、Java ME。

(1) Java SE(Java Platform, Standard Edition)。Java SE 含有基本的 Java SDK 工具和运行时 API，开发者可以用它们来编写、部署和运行 Java 应用程序和 Applet(在 Web 浏览器如 IE 中运行的 Java 小程序)。

(2) Java EE(Java Platform, Enterprise Edition)。Java EE 建立在 Java SE 的基础上,它是 JSP(Java Server Page)、Servlet、EJB、JTS(Java Transaction Service)、Java mail 以及 JMS(Java Message Service)等多项技术的混合体,主要用于开发分布式的、服务器端的多层结构的应用系统,如电子商务网站。

(3) Java ME(Java Platform, Micro Edition)。Java ME 主要用于开发电子产品,如移动电话、数字机顶盒、汽车导航系统等。

2009 年 4 月 Oracle 公司以 74 亿美元收购 Sun,目前 Oracle 公司网站已正式发布了 Java 7。除了官方的三大平台外,在移动应用领域,Google 公司的 Android 操作系统采用 Java 语言作为应用开发的主要语言。

本书介绍的内容基本都属于 Java SE 的范畴,第 9 章涉及 Java EE 部分内容,语言主要采用 Java 2,少量程序使用了 Java 5 以后的一些新特性。

1.3 跨平台的 Java

与常见的编译型高级程序设计语言不同,Java 语言编译器产生的二进制代码是一种与具体机器指令无关的指令集合,只要有 Java 运行时系统存在,编译后的代码便可在许多处理器上运行。Java 运行时系统被称为 Java 虚拟机(Java Virtual Machine, JVM),Java 编译器产生的代码由 Java 虚拟机解释执行,如图 1.1 所示。由此,Java 语言实现了平台独立性——一次编写,随处运行(Write once, Run anywhere)。

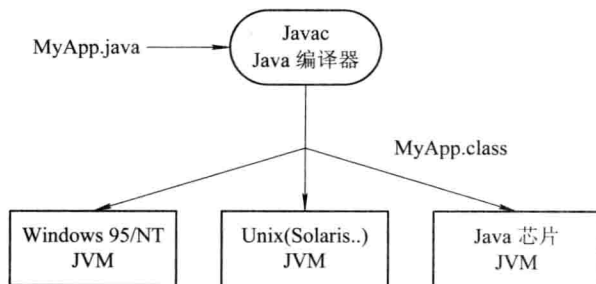


图 1.1 Java 的平台无关性

Java 编译器产生的二进制代码称为字节码(Byte Code),字节码与任何具体计算机结构都无关。实际上,这并不是一个新想法,多年前的 UCSD Pascal 系统就在一个商业产品中做了同样的努力,甚至比这更早, Niklaus Wirth 的 Pascal 的最初实现也使用了同样的方法。

不过这种技术存在着一个很大的缺点,即与通常高级语言采用的编译为机器指令相比,其程序执行效率相差很多。为此,Java 系统采用了一些独特的方法来改善其执行效率。例如,采用即时编译技术(Just In Time, JIT),在程序执行前将字节码转换为本地指令,避免了对某些指令段的重复解释;精心设计 Java 字节码,使其既可容易地在任何机器上解释,又可容易地被快速翻译为本地机器代码。

Java 系统的基本数据类型不依赖于具体实现,在任何硬件平台上均保持一致。而通常的高级语言数据类型是与平台相关的,如 C 语言在 Windows 3.1 中整数(int)为 16 位,在

Windows 95 中整数为 32 位，在 DEC Alpha 中整数为 64 位，在 Intel 486 中整数为 32 位。

与体系结构无关的特性使得 Java 应用程序可以在配备了 Java 虚拟机的任何计算机系统上运行，这成为 Java 应用软件便于移植的良好基础。

1.4 面向对象的 Java

20 世纪 60 年代，软件开发技术发展缓慢，随着硬件技术的发展，出现了“软件危机”。在软件开发过程中，所做的工作常常落后于预先的计划，成本大幅提高，超过预算，而得到的最终产品却并不可靠，不能满足实际的需要。

为了解决“软件危机”，20 世纪 60 年代末 70 年代初人们提出了结构化程序设计的思想，按照结构化程序设计的原则和方法，设计出结构清晰、容易理解、容易修改、容易验证的程序。结构化程序设计方法采用自顶向下、逐步细化的方法，将复杂的过程分解成由三种基本控制结构(顺序结构、选择结构和循环结构)构成的程序。

到了 20 世纪 70 年代末期，随着计算机科学的发展和应用领域的不断扩大，对计算机技术的要求越来越高。结构化程序设计语言和结构化分析与设计已无法满足用户需求的变化，面向对象技术开始崭露头角。

面向对象程序设计方法起源于 Simula 67 语言，它本身虽因比较难学、难用而未能广泛流行，但在它的影响下所产生的面向对象技术却迅速传播开来，并在全世界掀起了一股 OO(Object Oriented)热潮，至今盛行不衰。面向对象程序设计在软件开发领域引起了大的变革，极大地提高了软件开发的效率，为解决软件危机带来了一线光明。

结构化程序设计方法将要实现的系统描述为一个过程，而一些复杂的系统如 Windows 的窗口系统、一个大型机构的管理信息系统等，我们则很难将其运作规律描述成一个单一的过程。面向对象的基本思想认为系统是由若干对象构成的，每个对象都有各自的内部状态和运动规律，不同对象之间通过消息传送相互作用和联系。使用面向对象的方法，我们可以通过分别描述系统中的每个对象的特性和这些对象之间的联系来描述整个系统。

Java 语言的设计完全是面向对象的。在 Java 语言编写的程序中，使用类(class)来描述一组对象的共同特性，类可以看成是创建对象的蓝图，对象之间通过方法调用来实现相互间的联系。

Java 语言程序的基本单位是类(class)，一个完整的 Java 语言程序一般由多个类构成。Java 语言运行环境还提供了功能强大的类库(或称为应用程序接口(API))，可以帮助应用程序很容易地实现一些复杂的功能。

1.5 Java 程序的开发环境

1.5.1 Java 程序的开发过程

Java 程序的开发过程与其他高级语言相似。首先编写程序，然后采用文本编辑软件输入源程序，再使用 Java 语言编译器进行编译，生成字节码文件。

与 C/C++ 等其他高级语言不同的是, 一个 Java 程序由若干个字节码文件构成, Java 虚拟机直接执行字节码文件, 没有连接生成独立的可执行文件的过程。Java 类库代码也不像 C/C++ 的函数库或类库那样需要连接到执行文件中, 用户要执行 Java 程序必须安装相应的 Java 运行环境(Java Runtime Environment, JRE), 这些类库在安装 Java 运行环境时已安装在用户的系统中了。

1.5.2 Java 开发工具

Java 语言最基本的开发环境是由 Oracle 公司免费提供的 JDK(Java Development Kit), 该开发环境比较简单, 使用命令行编译源代码, 编译、调试程序都不是很方便。

目前已有很多商业或开源的 Java 语言集成开发工具, 这些开发工具集源代码的编辑、编译以及调试于一体, 而且这些开发工具可以帮助程序员生成应用程序框架, 减少程序员的重复劳动, 提高软件开发的效率。

Oracle 公司免费提供的 JDK 虽然开发环境比较简单, 但编译、调试程序都不是很方便, 大部分的 Java 集成开发环境均需要 JDK 的支持, 某些厂商的集成开发环境在安装过程中会自动安装特定版本的 JDK, 并允许用户自行安装其他版本的 JDK, 以使用户能够使用 Java 语言新版本的一些特性。目前 Oracle 公司的 Java 官方网站提供 JDK7 等多个版本的下载, 用户可根据需要选择下载。

在众多集成开发环境中, Eclipse 功能比较强大, 是目前影响最大的开源 Java 集成开放平台之一。Eclipse 提供了一个以插件为基础的框架, 通过安装不同的插件还可以支持 C++ 等开发语言, Google 公司提供了相应的插件, 将其作为 Android 平台应用开发的官方指定平台。Eclipse 的平台是基于 Java 语言编写的, 并包含大量的插件创建工具以及实例。它已经在大范围的开发工作站上得到了应用, 包括 Linux、HP-UX、AIX、Solaris、QNX、Mac OS X, 以及 Windows 平台的系统。

本书的实验环境选择采用 Eclipse, 目前官方网站提供 Eclipse Kepler(4.3.1)、Eclipse Juno(4.2)、Eclipse Indigo(3.7)等版本的下载。

1.6 最简单的 Java 程序

Java 程序主要有 Java 应用程序(Java Application)和 Applet 两种。下面给出两个最简单的 Java 程序, 从中可以了解 Java 语言程序的基本结构。

1.6.1 最简单的 Java 应用程序

【程序 1.1】 最简单的 Java 应用程序。

```
/* HelloWorldApp.java */
class HelloWorldApp
{
    /**
     * Traditional "Hello World!" program.
```

```
*/  
public static void main (String args[])  
{  
    // Write to stdout.  
    System.out.println("Hello World!");  
}  
}
```

程序 1.1 是一个 Java Application，它的功能很简单，向标准输出设备输出字符串“Hello World!”，运行该程序，可以在显示器上看到该字符串。

从程序 1.1 中首先看到的是注释语句，编译器在编译时将跳过该部分内容。Java 语言的注释语句有两种基本形式：

- (1) 以 /* 开始，以 */ 结束，其中间的所有字符在编译时被忽略。
- (2) 行注释，以 // 开始到本行结束。

程序 1.1 定义了一个类 HelloWorldApp，其中定义了一个方法 main。有关类和方法的概念本书将在第 3 章详细描述。

main 方法是程序的入口点，Java 应用程序从 main 方法开始执行，若 main 方法执行结束，则该程序退出运行。因此，如果一个程序由多个类构成，则只能有一个类有 main 方法。

程序 1.1 使用了 Java API 完成字符串的输出功能，System.out 为标准输出流对象，用于访问操作系统的标准输出设备，通常情况下标准输出设备为显示设备。println 为其方法成员，其功能为输出括号中的字符串或其他类型的数据并换行。类似的还有方法成员 print，它与 println 的区别是输出数据后不换行。

程序 1.1 的结构比较简单，只定义了一个类，在后面的章节中读者将会看到包含多个类的程序。

1.6.2 最简单的 Java Applet

Java Applet 的执行环境与 Java 应用程序不同，Applet 不是独立的应用程序，它是嵌入在 HTML 文件中使用的，程序被放置在 Web 服务器上，下载到客户端后，由 Web 浏览器如微软的 Internet Explorer 执行。

【程序 1.2】最简单的 Java Applet。

```
import java.applet.Applet;  
import java.awt.Graphics;  
public class HelloWorld extends Applet  
{  
    public void paint(Graphics g)  
    {  
        g.drawString ("你好,Java 世界!",2,20);  
    }  
}
```

程序 1.2 是一个最简单的 Java Applet。下面是一个发布程序 1.2 的 HTML 文件的内容, 请注意其中斜黑体的部分。

```
<html>
<head><title>我的第一个 JavaApplet 程序</title></head>
<body>
<p><applet code=HelloWorld.class width=300 height=200></applet>
</body>
</html>
```

将上述 HTML 文件和程序 1.2 编译得到的字节码文件 HelloWorld.class 放在 Web 服务器的同一个目录下, 当使用 Web 浏览器浏览该 HTML 文件时, 浏览器将下载 HelloWorld.class, 然后执行。

因为执行环境与 Java Application 不同, Applet 的程序结构与 Java Application 也有所不同, 当然它们有一点是共同的, 即都是由若干个类组成的。

程序 1.2 的第 1 行表示该 Applet 程序需要引用 Java API 类库提供的 Applet 类。学习过 C 语言的读者应注意 import 与 C 语言中的 #include 类似, 但 Java 编译器的处理方法不同于 C 语言, 它并不将该文件读入, 而且它引用的是已经编译过的 Java 字节码文件。在编译阶段, Java 编译器将从该字节码文件中读取有关 Applet 类的信息, 检验程序中对 Applet 的使用是否正确。程序 1.2 编译生成的 HelloWorld.class 文件中也不包含 Applet 类的代码。

第 2 行的作用与第 1 行类似, 由于程序中用 java.awt.Graphics 类的功能来输出字符串, 因此引入该类。

第 3 行开始定义 HelloWorld 类, 注意后面的 extends Applet, 这是 Java 类继承语法。一个 Applet 程序可以由多个类构成, 其中只有一个类继承自 Applet 类, 这是 Applet 程序的入口。

Applet 的执行与 Java Application 不同, 从程序中看不到像 Application 中 main 方法那样的一个明显的执行流程。实际上这些都已经是在 Applet 中实现了, Applet 在执行时一直等待用户的输入或其他的一些事件(如关闭浏览器), 根据不同的事件执行不同的功能。在编写 Applet 时需要做的就是提供各种事件的处理程序, 例如程序 1.2 类 HelloWorld 中定义了方法 paint, 该方法当 Applet 需要绘制界面时被调用。

Applet 类中定义了 paint 方法的调用形式, 它有一个 Graphics 类的对象作参数, 通过它可以在 Applet 的界面上绘制图形和文字。程序 1.2 调用 drawString 方法来输出一个字符串:

```
g.drawString("你好, Java 世界!", 2, 20);
```

drawString 方法有三个参数: 第一个是要输出的字符串; 第二、三个是输出位置, 分别为 x 轴、y 轴的值。

图 1.2 是程序 1.2 在 IE6.0 中执行的画面。



图 1.2 Applet 的运行