



新世纪高职高专实用规划教材·计算机系列

# Java 程序设计教程 (第2版)

迟丽华 主编



清华大学出版社

新世纪高职高专实用规划教材·计算机系列

# Java 程序设计教程

(第2版)

迟丽华 主 编

清华大学出版社

北 京

## 内 容 简 介

本书通过大量程序实例由浅入深地讲述了 Java 程序设计语言的相关知识。内容包括 Java 语言基础、面向对象编程、基础类库、包和接口、图形界面设计、异常处理、多线程技术和网络编程等。全书不仅介绍理论,更强调实际运用。书中给出的所有例题都通过了调试。这些例题可以帮助读者理解 Java 语言的基本概念和编程技巧。每章还附有各种类型的习题,方便读者练习所学知识和总结提高。

本书可以作为学习 Java 语言的教材,也可以作为自学用书。

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 13501256678 13801310933

### 图书在版编目(CIP)数据

Java 程序设计教程/迟丽华主编. —2 版. —北京:清华大学出版社,2008.3

(新世纪高职高专实用规划教材·计算机系列)

ISBN 978-7-302-17003-7

I. J… II. 迟… III. JAVA 语言—程序设计—高等学校:技术学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2008)第 017452 号

责任编辑:朱颖 宋延清

封面设计:子时文化

版式设计:北京东方人华科技有限公司

责任印制:孟凡玉

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

· <http://www.tup.com.cn> 邮 编:100084

[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

社总机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市清华园胶印厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:21.5 字 数:515 千字

版 次:2008 年 3 月第 2 版 印 次:2008 年 3 月第 1 次印刷

印 数:1~4000

定 价:29.50 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026965-01

# 《新世纪高职高专实用规划教材》序

## 编写目的

目前,随着教育的不断深入,高等职业教育发展迅速,进入到一个新的历史阶段。学校规模之大,数量之众,专业设置之广,办学条件之好和招生人数之多,都大大超过了历史上任何一个时期。然而,作为高职院校核心建设项目之一的教材建设,却远远滞后于高等职业教育发展的步伐,以至于许多高职院校的学生缺乏适用的教材,这势必影响高职院校的教育质量,也不利于高职教育的进一步发展。

目前,高职教材建设面临着新的契机和挑战:

(1) 高等职业教育发展迅猛,相应教材在编写、出版等环节需要在保证质量的前提下加快步伐,跟上节奏。

(2) 新型人才的需求,对教材提出了更高要求,要求出版一批科学性、先进性和实用性的教材。

(3) 高职高专教育自身的特点是强调学生的实践能力和动手能力,教材的取材和内容设置必须满足不断发展的教学需求,突出理论和实践的紧密结合。

(4) 新教材应充分考虑一线教师的教学需要和教学安排。

鉴于此,清华大学出版社在相关主管部门的大力支持下,组织部分高等职业技术学院的优秀教师以及相关行业的工程师,推出了一系列切合当前教育改革需要的、高质量的、面向就业的职业技术实用型教材。

## 系列教材

本系列教材主要涵盖以下领域:

- 计算机基础及其应用
- 计算机网络
- 计算机图形图像处理与多媒体
- 电子商务
- 计算机编程
- 电子电工
- 机械
- 数控技术及模具设计
- 土木建筑
- 经济与管理
- 金融与保险

另外,系列教材还包括大学英语、大学语文、高等数学、大学物理、大学生心理健康等基础教材。所有教材都有相关的配套用书,如实训教材、辅导教材、习题集等。

## 教材特点

为了完善高等职业技术教育的教材体系,全面提高学生的动手能力、实践能力和职业技术素质,特意聘请有实践经验的高级工程师参与系列教材的编写,采用了一线工程技术人员与在校教师联合编写的模式,使课堂教学与实际操作紧密结合。本系列丛书的特点如下:

- (1) 打破以往教科书的编写套路,在兼顾基础知识的同时,强调实用性和可操作性。
- (2) 突出概念和应用,相关课程配有上机指导及习题,帮助读者对所学内容进行总结和提高。
- (3) 设计了“注意”、“提示”、“技巧”等带有醒目标记的特色段落,使读者更容易得到有益的提示与应用技巧。
- (4) 增加了全新的、实用的内容和知识点,并采取由浅入深、循序渐进、层次清楚、步骤详尽的写作方式,突出实践技能和动手能力。

## 读者定位

本系列教材针对职业教育,主要面向高职高专院校,同时也适用于同等学力的职业教育和继续教育。本丛书以三年制高职为主,同时也适用于两年制高职。

本系列教材的编写和出版是高职教育办学体制和运作机制改革下的产物,在后期的推广使用过程中将紧紧跟随职业技术教育发展的步伐,不断汲取新型办学模式、课程改革的思路和方法,为促进职业培训和继续教育的社会需求奉献我们的力量。

我们希望,通过本系列教材的编写和推广应用,不仅有利于提高职业技术教育的整体水平,而且有助于加快改进职业技术教育的办学模式、课程体系和教学培训方法,形成具有特色的职业技术教育的新体系。

新世纪高职高专实用规划教材编委会

# 前 言

近年来,越来越多的高校开设了 Java 语言程序设计课程。随着网络应用的普及和开发网络应用程序的需要,以网络功能为特点的 Java 语言已成为编写网络程序的首选工具之一。学习 Java 语言是时代发展的需要。

Java 语言诞生于 20 世纪 90 年代初,由 Sun 公司推出。Java 语言是一种面向对象的新兴网络编程语言,其强大的网络功能和其他语言难以比拟的,Java 语言特别适合开发 Internet 应用程序。由于 Java 程序具有不依赖于机器结构、与平台无关、可靠性高、安全稳定和支持多线程等特点,因此在当今的计算机和网络中占据关键性的地位。

本书针对初学者的特点,由浅入深地讲述 Java 程序设计语言。通过大量程序实例帮助读者理解 Java 语言的基本概念和编程思想。在教材内容的编排和例题的选择方面都注重培养读者分析问题和解决问题的能力,在阐述理论知识的同时强调实用性,使读者能够学以致用。每章后面都附有各种类型的习题,方便读者练习和总结所学知识。

全书共分 8 章。第 1 章简述 Java 语言的发展和特点,通过简单的程序例子说明 Java 程序的组成和结构,还介绍常用的 Java 程序开发工具。通过该章的学习,可以对 Java 程序有一个初步的认识,并能够上机编写和调试简单的 Java 程序。第 2 章介绍 Java 语言的基础知识,包括数据类型、常量、变量、表达式、数组和流程控制语句。第 3 章和第 4 章讨论面向对象程序设计的基本概念,包括对象、类、抽象、封装、继承和多态。详细介绍 Java 语言面向对象编程的机制和方法。第 5 章介绍 Java 提供的常用类,包括 Applet 类、常量字符串类 String、变量字符串类 StringBuffer、数学类 Math、系统类 System、日期类 Date、随机数类 Random 和向量类 Vector 等,并通过程序实例说明如何在 Java 程序中使用这些常用类。第 6 章详细介绍图形用户界面的设计,包括各种图形界面组件、事件类型和事件处理。第 7 章介绍 Java 的多媒体技术,包括绘制图形、设置文字显示效果、显示图像、设计动画和播放声音。第 8 章介绍 Java 的异常处理、多线程编程、输入输出处理和网络通信编程。

全书由迟丽华主编。在编写此书的过程中得到了边殿英、刘梅、王秀芝、王元战等的帮助,在此表示感谢。

由于水平有限,书中难免有不当之处,敬请读者批评指正(chilihua@tju.edu.cn)。

编 者

2008 年 1 月

# 目 录

<b>第 1 章 概述</b> .....1	
1.1 Java 程序简介 .....1	
1.2 Java 程序举例 .....1	
1.2.1 Java Application 程序举例 .....2	
1.2.2 Java Applet 程序举例 .....4	
1.2.3 超文本标记语言 HTML 简介 .....7	
1.3 Java 程序结构 .....7	
1.4 简单的输入和输出.....9	
1.4.1 字符界面输入和输出 .....9	
1.4.2 图形界面输入和输出 .....11	
1.5 Java 语言的特点 .....13	
1.6 SDK 开发工具的下载和安装.....14	
1.6.1 下载 SDK .....14	
1.6.2 安装 SDK .....15	
1.6.3 设置 SDK 的工作环境 .....16	
1.7 上机指导.....17	
1.7.1 SDK 开发工具的使用 .....17	
1.7.2 Java 集成开发工具的使用 .....19	
1.8 习题.....21	
<b>第 2 章 Java 语言基础</b> .....23	
2.1 Java 语言的基本要素 .....23	
2.1.1 标识符 .....23	
2.1.2 数据类型 .....24	
2.1.3 常量和变量 .....26	
2.1.4 数据类型转换 .....30	
2.1.5 注释 .....30	
2.2 运算符与表达式.....31	
2.2.1 算术运算符 .....31	
2.2.2 关系运算符 .....33	
2.2.3 逻辑运算符 .....35	
2.2.4 位运算符 .....36	
2.2.5 赋值运算符 .....39	
2.2.6 字符串运算符 .....40	
2.2.7 其他运算符 .....41	
2.2.8 运算符的优先级和结合性 .....42	
2.3 数组.....42	
2.3.1 一维数组 .....43	
2.3.2 二维数组 .....46	
2.4 流程控制语句.....47	
2.4.1 选择语句 .....47	
2.4.2 循环语句 .....52	
2.4.3 break 语句和 continue 语句.....57	
2.5 上机指导.....60	
2.5.1 Java 基本数据类型的取值范围 .....60	
2.5.2 条件语句和循环语句的使用 ...60	
2.5.3 数组的定义和使用 .....61	
2.6 习题.....61	
<b>第 3 章 Java 面向对象编程</b> .....65	
3.1 面向对象程序设计的基本概念.....65	
3.1.1 对象与类 .....65	
3.1.2 抽象与封装 .....65	
3.1.3 继承与多态 .....66	
3.2 Java 类的定义和使用 .....66	
3.2.1 类的定义 .....66	
3.2.2 对象与类的使用 .....68	
3.2.3 类和类成员的修饰符 .....73	
3.3 方法.....84	
3.3.1 构造方法 .....84	
3.3.2 递归方法 .....86	
3.4 上机指导.....87	
3.4.1 Java 类的定义和使用 .....87	
3.4.2 抽象类的定义和使用 .....88	
3.5 习题.....89	
<b>第 4 章 继承与多态</b> .....93	
4.1 子类与继承.....93	
4.1.1 派生子类与继承关系 .....93	
4.1.2 成员方法的继承与覆盖 .....95	
4.1.3 数据成员的继承与隐藏 .....96	
4.1.4 关键字 this 和 super.....98	
4.2 多态.....100	

4.2.1 通过方法重载实现多态 .....	100	5.7.2 系统类 System 的使用 .....	166
4.2.2 通过方法覆盖实现多态 .....	103	5.7.3 随机数类 Random 的使用 .....	167
4.3 构造方法的继承和重载 .....	103	5.8 习题 .....	167
4.4 接口 .....	108	<b>第 6 章 图形用户界面设计</b> .....	<b>173</b>
4.4.1 接口的定义 .....	108	6.1 GUI 组件概述 .....	173
4.4.2 接口的实现 .....	110	6.2 事件处理概述 .....	175
4.4.3 接口的继承 .....	118	6.2.1 事件分类 .....	175
4.5 包 .....	120	6.2.2 事件处理 .....	177
4.5.1 创建包 .....	120	6.3 标签 .....	178
4.5.2 引用包 .....	121	6.3.1 标签的构造方法 .....	178
4.5.3 Java 提供的包 .....	123	6.3.2 标签的常量和方法 .....	179
4.6 上机指导 .....	124	6.4 按钮与动作事件 .....	180
4.6.1 从原有类派生新类 .....	124	6.4.1 按钮的构造方法 .....	180
4.6.2 接口的创建和实现 .....	125	6.4.2 按钮的常用方法 .....	181
4.6.3 包的创建和引用 .....	126	6.4.3 按钮的动作事件 .....	181
4.7 习题 .....	127	6.4.4 动作事件的处理 .....	182
<b>第 5 章 系统常用类</b> .....	<b>130</b>	6.5 文本框与文本事件 .....	185
5.1 Object 类 .....	130	6.5.1 文本框的常用方法 .....	185
5.2 Applet 类与 Applet 小程序 .....	130	6.5.2 文本框的事件处理 .....	187
5.2.1 Applet 类的主要方法 .....	131	6.6 文本区与文本事件 .....	190
5.2.2 HTML 与 Applet 的 参数传递 .....	135	6.7 下拉列表与选择事件 .....	191
5.2.3 在命令行运行 Applet 程序 .....	136	6.7.1 下拉列表的常用方法 .....	191
5.3 字符串类 .....	137	6.7.2 下拉列表的事件处理 .....	192
5.3.1 String 类 .....	137	6.8 列表与选择事件 .....	194
5.3.2 StringBuffer 类 .....	143	6.8.1 列表的常用方法 .....	194
5.4 数学类 Math .....	145	6.8.2 列表的事件处理 .....	195
5.4.1 数学类提供的数学常量 .....	145	6.9 复选框与选择事件 .....	196
5.4.2 数学类提供的常用函数 .....	146	6.9.1 复选框的常用方法 .....	196
5.5 数据类型类 .....	150	6.9.2 复选框的事件处理 .....	197
5.5.1 数字类型的类 .....	150	6.10 单选按钮组与选择事件 .....	198
5.5.2 字符类 Character .....	154	6.10.1 创建单选按钮组 .....	199
5.5.3 布尔类 Boolean .....	155	6.10.2 单选按钮组的事件处理 .....	200
5.6 其他常用类 .....	156	6.11 滚动条与调整事件 .....	201
5.6.1 系统类 System .....	156	6.11.1 滚动条的常用方法 .....	201
5.6.2 日期类 Date .....	158	6.11.2 滚动条的事件处理 .....	202
5.6.3 随机数类 Random .....	161	6.12 鼠标事件与键盘事件 .....	203
5.6.4 向量类 Vector .....	163	6.12.1 鼠标事件 .....	204
5.7 上机指导 .....	166	6.12.2 键盘事件 .....	206
5.7.1 数学类 Math 的使用 .....	166	6.13 布局管理 .....	209
		6.13.1 FlowLayout 布局 .....	209



6.13.2 BorderLayout 布局 .....	210	7.8.1 通过 HTML 文件设置颜色 .....	262
6.13.3 CardLayout 布局 .....	211	7.8.2 显示图像 .....	263
6.13.4 GridLayout 布局 .....	213	7.8.3 实现动画 .....	263
6.14 容器 .....	215	7.9 习题 .....	264
6.14.1 容器类与容器事件 .....	215	<b>第 8 章 Java 高级编程</b> .....	268
6.14.2 面板与容器事件 .....	216	8.1 异常处理 .....	268
6.14.3 框架与窗口事件 .....	220	8.1.1 异常与异常类 .....	268
6.15 对话框的定义与使用 .....	222	8.1.2 异常类的层次结构 .....	269
6.15.1 用 JOptionPane 创建对话框 .....	223	8.1.3 异常的处理机制 .....	271
6.15.2 用 Dialog 或 JDialog 创建对话框 .....	224	8.1.4 抛出异常 .....	271
6.16 菜单的定义与使用 .....	227	8.1.5 捕捉异常 .....	274
6.16.1 设计菜单条式菜单 .....	228	8.1.6 自定义异常类 .....	278
6.16.2 设计弹出式菜单 .....	230	8.2 Java 多线程机制 .....	279
6.17 上机指导 .....	232	8.2.1 基本概念 .....	279
6.17.1 标签和按钮的使用 .....	232	8.2.2 实现多线程 .....	281
6.17.2 文本框和文本区的使用 .....	233	8.2.3 线程组 .....	288
6.17.3 其他组件的使用 .....	234	8.2.4 线程的同步控制 .....	290
6.17.4 键盘事件的处理 .....	235	8.3 流式输入输出与文件管理 .....	297
6.18 习题 .....	236	8.3.1 流与输入输出类库概述 .....	297
<b>第 7 章 Java 的多媒体技术</b> .....	239	8.3.2 基本输入输出流 .....	297
7.1 绘制图形 .....	239	8.3.3 基本输入输出流的子类 .....	300
7.1.1 绘图方法 .....	239	8.3.4 标准输入输出 .....	303
7.1.2 画布 Canvas .....	244	8.3.5 文件和目录管理 .....	304
7.2 设置文字显示效果 .....	245	8.4 网络通信 .....	306
7.3 设置颜色 .....	247	8.4.1 利用 URL 访问网上资源 .....	306
7.3.1 颜色类 .....	247	8.4.2 利用 URLConnection 与服务器交互 .....	309
7.3.2 设置颜色的方法 .....	249	8.4.3 基于连接的流式通信 .....	312
7.3.3 通过 HTML 文件设置颜色 .....	251	8.4.4 无连接的数据报通信 .....	319
7.4 绘图模式 .....	252	8.5 上机指导 .....	320
7.5 显示图像 .....	253	8.5.1 自定义异常类的 建立和使用 .....	320
7.6 实现动画 .....	255	8.5.2 多线程编程 .....	321
7.7 播放声音 .....	258	8.5.3 网络通信 .....	322
7.7.1 播放声音的方法 .....	258	8.6 习题 .....	323
7.7.2 编写播放声音的程序 .....	259	<b>附录 习题答案</b> .....	327
7.8 上机指导 .....	262		

# 第 1 章 概述

**教学提示:** 本章通过简单的 Java 程序实例介绍 Java 程序的分类和特点。详述 Java 程序的基本结构以及开发和运行 Java 程序的步骤及方法。

**教学目标:** 掌握 Java Application 程序和 Java Applet 程序的基本结构和特点。通过上机实践, 明确 Java 各类程序的编写和运行方式。

## 1.1 Java 程序简介

Java 语言诞生于 20 世纪 90 年代初, 是由 Sun 公司推出的一种面向对象的新兴网络编程语言, 其强大的网络功能和其他编程语言难以比拟的。Java 语言特别适合开发 Internet 应用程序, 已成为当前编写网络程序的首选工具之一。由于 Java 程序具有不依赖于机器结构、与平台无关、可靠性高、安全稳定和支持多线程等特点, 因此在当今计算机网络中占据关键性的地位。

按照程序的执行方式, 程序设计语言分为编译执行和解释执行两类(例如, C 语言是编译执行的程序设计语言, Java 语言是解释执行的程序设计语言)。

编译执行程序的方式是, 将编写好的程序源代码通过专门的工具软件(编译程序)编译成目标代码, 再转换成机器代码(即可执行程序), 然后由计算机直接运行机器代码, 实现程序的功能。编译执行的程序与机器之间存在一定的依赖性, 在一个系统上可以运行的程序到另一个系统上不一定能够运行。

解释执行程序的方式是, 当运行程序时, 由解释程序对源程序读入一句、解释一句, 并执行一句。解释执行程序并不需要将程序源代码转换成可执行程序。这样的语言与机器的相关性较小, 但运行速度较慢。Java 程序的执行方式是, 首先将 Java 源程序编译成字节码(Byte-Code)文件, 再由解释器对字节码文件进行解释执行。Java 程序的这种执行方式不仅提高了程序运行的速度, 而且使程序的运行不依赖于机器的结构, 从而使 Java 应用软件可以方便地在网络上的不同计算机中进行移植, 能够在任何环境的计算机系统上运行, 这正是 Java 成为网络编程语言的主要原因。

## 1.2 Java 程序举例

根据运行环境和程序结构的不同, 通常将 Java 程序分成两类, 一类是应用程序(Java Application), 另一类是小应用程序(Java Applet)。简单地说, Java Application 是一种独立完整的程序, 由 Java 解释器进行解释, 可以独立运行; Java Applet 是一种非独立程序, 需要置于由 HTML 语言建立的 WWW 网页文件中, 在用户访问嵌有 Java Applet 程序的 WWW 网页文件时, 由 WWW 浏览器内部包含的 Java 解释器对 Java Applet 程序进行解释, 使 WWW 网页中的 Java Applet 程序得以运行。

下面给出最简单的 Java 程序实例, 并通过这些例子详述 Java 程序的基本结构以及开发和运行 Java 程序的步骤及方法。

## 1.2.1 Java Application 程序举例

从编写 Java Application 程序到得到程序的运行结果, 要经过三个环节, 如图 1.1 所示。第一, 编辑 Java Application 源程序; 第二, 将 Java Application 源程序编译成字节码程序; 第三, 解释执行字节码程序, 得到程序的运行结果。下面分别进行说明。



图 1.1 Java 程序的上机步骤

### 1. 编辑源程序

可以用 Java 集成开发环境提供的源代码编辑器编写程序, 也可以使用其他文本编辑工具, 如 Windows 的记事本或 DOS 的 EDIT 软件等。Java 源程序是以 .java 为后缀(扩展名)的文本文件。

**注意:** 在用其他文本编辑工具编写 Java 程序时, 要将程序保存成后缀为 .java 的文件。

下面是一个 Java Application 程序。

#### 例 1.1 Java Application 程序:

```
//这是 Java Application 程序, 程序文件名为 ApplicationDemo.java
public class ApplicationDemo
{
    public static void main(String[] args)
    {
        System.out.println("这是一个最简单的 Java 应用程序");
        System.out.println("本程序练习使用输出语句");
    }
}
```

上述程序的功能是在屏幕(标准输出设备)上显示两行文字, 如图 1.2 所示。

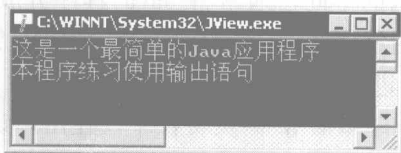



图 1.2 例 1.1 程序的输出结果

对上述程序说明如下:

- Java 程序由若干个类组成。上述程序只有一个类，类名是 ApplicationDemo。
- 上述程序的第 1 行是注释行。注释行是对程序的说明，它对程序的运行结果没有影响。注释行的标志是以“//”开头。这里用注释行说明程序文件名为 ApplicationDemo.java。Java 要求程序的文件名必须与类名相同。
- 程序的第 2 行用保留字 class 声明一个类，类名为 ApplicationDemo。类名的第一个字母通常为大写。Java 语言对字母大小写非常敏感，编写程序时要特别注意。修饰符 public 说明 ApplicationDemo 类的使用权限，表示它是公共类。public 类型的类可以被所有程序使用。ApplicationDemo 类的内容在 {} 中进行定义。一个 Java 程序中可以定义多个类，但是最多只能有一个 public 类，程序的文件名必须与 public 类的类名相同。
- 程序的第 4 行定义一个主方法 main()，并用 public、static 和 void 说明方法的类型。其中，public 表示访问权限，指明该方法是公用方法，即所有的类都可以引用这一方法；static 表示访问方式，指明该方法是一个静态方法，可以通过类名直接调用；void 指明 main() 方法没有任何返回值。
- 每个类中可以定义多个方法，对于任何一个 Java Application 程序，必须有且只能有一个 main() 方法，该方法是 Java Application 程序的入口，即程序从 main() 方法开始执行。main() 的方法头通常按照下面两种格式书写。
  - ◆ 格式一：public static void main(String[] args)
  - ◆ 格式二：public static void main(String args[])
- main() 方法中的 String[] args 或 String args[] 是传递给 main() 方法的参数，参数名为 args，是一个 String 类型的数组。main() 方法也可以有多个参数，参数之间用逗号分隔，如下面的写法：

```
public static void main(类型 参数1, 类型 参数2, ..., 类型 参数n)
```
- main() 方法的内容在 {} 中给出。本例只有两条输出语句，作用是把文字输出到显示器上。

 **注意：** Java 程序中的每个语句都要用分号“;”结束，这一点与 C 语言和 C++ 语言相同。

## 2. 将 Java 源程序编译成字节码程序

Java 源程序经过编译得到的目标代码称为字节码程序或字节码文件(Byte-Code)，字节码文件的后缀(扩展名)为 class。例如，例 1.1 的源程序文件名为 ApplicationDemo.java，字节码文件名为 ApplicationDemo.class。

由 Java 源程序生成字节码需要使用 Java 编译器。可以使用 Java 集成开发工具(如 JBuilder, Visual J++ 等)提供的编译器，也可以使用命令行开发工具 JDK 提供的编译器。详细介绍见 1.6 节的上机指导。

## 3. 解释执行 Java 的字节码程序

字节码程序文件是二进制文件，不能由操作系统直接执行，需要由 Java 解释器对字节码文件解释执行，才能得到程序的运行结果。通常将能够执行 Java 字节码文件的机器

称为 Java 虚拟机(Java Virtual Machine, JVM)。只要计算机上安装了可以执行 Java 程序的软件(Java 解释器),就构成了能够识别和执行 Java 程序的 Java 虚拟机。如浏览器(如 Internet Explorer)与 Java 的开发工具等都可看作 JVM。

## 1.2.2 Java Applet 程序举例

Java Applet 是另一种类型的 Java 程序。它与 Java Application 程序的区别是:Java Application 程序可以独立执行;而 Java Applet 程序不能独立执行,需要嵌入在用 HTML 语言编写的网页文件中,当打开网页文件,在 WWW 浏览器中显示网页内容时,嵌入在网页中的 Java Applet 程序就会执行。

从编写 Java Applet 程序到得到程序的运行结果,要经过四个环节:第一个环节是编辑 Java Applet 源程序,第二个环节是将 Java Applet 源程序编译成字节码程序,第三个环节是把字节码程序嵌入到 HTML 文件中,第四个环节是在网络上发布 Applet 字节码程序,由用户在本地计算机中下载 Applet 字节码程序,并在浏览器中解释执行 Applet 字节码程序,得到程序的结果。下面对这四个环节分别加以说明。

### 1. 编辑 Java Applet 源程序

可以用 Java 集成开发环境提供的源代码编辑器编写程序,也可以使用其他文本编辑工具编写程序。下面是一个简单的 Java Applet 小应用程序。其功能是在浏览器中显示两行文字。图 1.3 显示了该程序在 WWW 浏览器中的运行结果。图 1.4 显示了该程序在 Java 自带浏览器中的运行结果。

#### 例 1.2 Java Applet 程序:

```
//这是 Java Applet 程序,程序文件名为 AppletDemo.java
import java.awt.*;
import java.applet.*;
public class AppletDemo extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("这是一个最简单的 Applet 程序", 20, 20);
        g.drawString("Applet 程序要嵌入到网页中执行", 20, 40);
    }
}
```

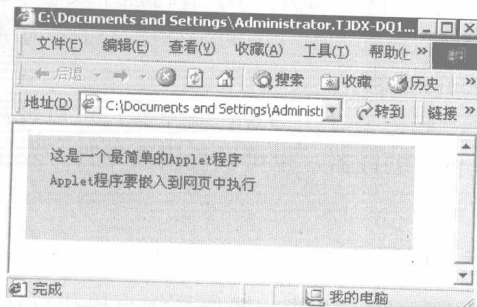


图 1.3 例 1.2 程序在 WWW 浏览器中的运行结果

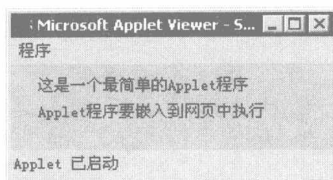


图 1.4 例 1.2 程序在 Java 自带浏览器中的运行结果

对上述程序说明如下：

- 该程序由 AppletDemo 类构成。程序的第 1 行是注释行，说明该程序的文件名为 AppletDemo.java。
- 程序的第 2、3 行用 import 语句将 java.awt 包(类库)和 java.applet 包引入到程序中，这样，程序就可以使用这些包中提供的类。Import 语句类似于 C 语言中的 #include 语句。
- 程序的第 4 行声明一个公共类 AppletDemo，用 extends 指明它是 Applet 类的子类，即 Applet 类是 AppletDemo 类的父类。AppletDemo 类的内容在 {} 中定义。
- 在 AppletDemo 类中编写了 paint() 方法，参数 g 的类型为 Graphics 图形类。在 paint() 方法中可以填写向屏幕输出内容的语句。本例通过 g.drawString 语句向屏幕输出内容：第一个 g.drawString 语句的作用是从坐标(20, 20)开始输出字符串“Applet 程序”；第二个 g.drawString 语句的作用是从坐标(20, 40)开始输出字符串“嵌入网页执行”。这里的坐标单位是像素点。

## 2. 将 Applet 源程序编译成字节码程序

同 Java Application 程序一样，也需要将 Java Applet 源程序编译成字节码程序。可以使用 Java 集成开发工具，也可以使用 JDK 命令行开发工具提供的编译器进行编译。

## 3. 把字节码文件嵌入到 HTML 文件中

从例 1.2 的程序中可以看出，Applet 程序没有 main() 方法，这是 Applet 程序与 Application 程序的明显区别。由于没有 main() 方法作为程序的执行入口，因此 Applet 程序不能独立运行，需要用超文本标记语言 HTML(Hyper Text Markup Language)编写网页文件，并把字节码文件嵌入到 HTML 网页文件中。例如，AppletDemo.class 是 AppletDemo.java 的字节码文件，应将 AppletDemo.class 置于网页文件中。

下面的例 1.3 是为例 1.2 程序编写的网页文件，网页文件的后缀是 htm，本例将网页文件命名为 Page1.htm。要查看例 1.2 程序的运行结果，只要在 WWW 浏览器或 Java 自带的浏览器中打开网页文件 Page1.htm，便会看到嵌入在其中的 Applet 程序的运行结果，如图 1.3 和图 1.4 所示。

**例 1.3** 编写网页文件 Page1.htm，将 AppletDemo.class 文件嵌入其中：

```
<HTML>
<TITLE></TITLE>
<HEAD></HEAD>
<BODY>
<APPLET code=AppletDemo.class height=100 width=200>
</APPLET>
```

```
</BODY>
</HTML>
```

#### 例 1.4 网页文件 Page1.htm 的另一种写法:

```
<HTML>
<TITLE>Applet 程序</TITLE>
<HEAD>这是 Applet 程序<BR></HEAD>
<BODY>
<APPLET code=AppletDemo.class height=100 width=200>
</APPLET>
</BODY>
</HTML>
```

在上面的网页文件中增加了 TITLE 标记对和 HEAD 标记对。对应的运行结果如图 1.5 所示。通过 TITLE 标记对, 设置浏览器的标题为“Applet 程序”。通过 HEAD 标记对, 设置网页标题为“这是 Applet 程序”。符号<BR>表示将光标移到下一行, 即换行。

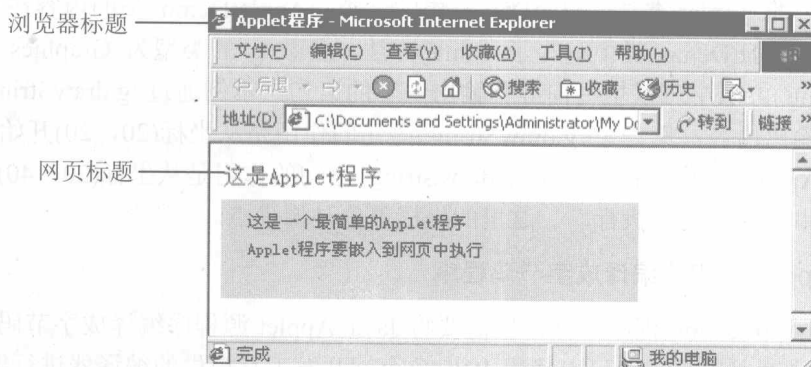


图 1.5 设置浏览器和网页标题

#### 4. Applet 程序的网络发布和下载运行

Applet 程序的运行机制是程序存储在 Web 服务器上, 下载后在用户计算机上运行。如图 1.6 所示。

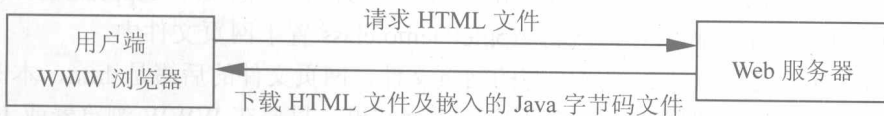


图 1.6 Java Applet 的下载执行过程

Java 程序的运行过程就是网络应用程序的发布过程。所有提供给用户的 Java Applet 字节码文件都要保存在 Web 服务器上。要在用户本地机上运行 Applet 字节码文件, 首先需要把它下载到本地机上, 然后再在本地机上运行。当需要修改或维护 Applet 程序时, 只需改动服务器一处的程序即可。

将 Applet 程序在网上发布, 供用户下载执行的步骤如下。

(1) 编写 HTML 文件, 并通过<APPLET>和</APPLET>标记对嵌入编译好的 Java

Applet 字节码文件。将 HTML 文件和 Java Applet 字节码文件保存在 Web 服务器的合适路径下。

(2) 当用户在用户机上请求 HTML 服务时, 由 WWW 浏览器下载 HTML 文件到用户机器, 并显示对应的网页, 同时自动下载 HTML 文件中指定的 Java Applet 字节码文件, 然后调用浏览器中的 Java 解释器, 解释执行下载到本机的字节码文件, 这样, 字节码文件的运行结果就会在网页中反映出来。

### 1.2.3 超文本标记语言 HTML 简介

由于 Java Applet 程序的字节码文件需要嵌入到 HTML 文件中才能执行, 因此需要编写 HTML 文件。编写 HTML 文件之前, 需要了解超文本标记语言 HTML。

HTML 语言是一种与平台无关的描述性语言, 由浏览器对它的源代码解释执行。可用任何一种文本编辑器编写 HTML 语言代码。在 HTML 中不区分大小写字母。HTML 语言用各种标记对(“<>”和“</>”)来描述和编排网页的内容。下面介绍常用的标记对。

#### 1. <HTML>和</HTML>标记对

所有 HTML 文件都以<HTML>开始, 以</HTML>结束。在<HTML>和</HTML>标记对之间描述和编排网页内容, 即<HTML>和</HTML>之间的内容是网页内容。

#### 2. <TITLE>和</TITLE>标记对

该标记对是可选项, 可有可无。用于设置浏览器标题栏的标题。例如, 可以通过“<TITLE> Applet 程序</TITLE>”设置浏览器的标题为“Applet 程序”。

#### 3. <HEAD></HEAD>头标记对

该标记对是可选项, 可有可无。用于描述 HTML 文档的属性、网页标题等。例如, 可以通过“<HEAD>这是 Applet 程序</HEAD>”设置网页标题为“这是 Applet 程序”。

#### 4. <BODY>和</BODY>主体标记对

在该标记对之间编排的内容是网页的可视内容。

#### 5. <APPLET>和</APPLET>标记对

用于嵌入 Java Applet 小程序。此标记对必须包含三个参数, 分别是 CODE、WIDTH 和 HEIGHT。其中 CODE 用于指明嵌入的 Java 字节码文件名。WIDTH 和 HEIGHT 分别用于指明 Applet 程序在网页中占用的宽度和高度

## 1.3 Java 程序结构

从前面的例子中可以看出, Java 程序由类构成, 一个 Java 程序可以包含一个或多个类。每个 Java 程序必须有一个主类。主类是程序的入口, 程序从这里开始执行。在 Java Application 程序中, 主类是包含 main()方法的类。在 Java Applet 程序中, 主类是 Applet 类的子类。



Java 程序中的类由数据成员(变量或对象)和方法(函数)成员组成。类的每个成员也称为类的属性。其中, 类的数据成员称为类的静态属性, 类的方法成员称为类的动态属性。类的全部内容也称为类体。类的数据成员是定义的变量或对象, 用于存储数据。变量的数据类型可以是任何基本数据类型, 对象的数据类型可以是任何引用数据类型(类)。一个类可以包括多个数据成员。类的方法类似于函数, 是类完成的功能或操作。一个类可以包括多个方法。每个程序的文件名必须与主类的类名相同, 后缀为 java。下面仍以 Applet 程序为例说明 Java 程序的构成。

### 例 1.5 Java 程序构成说明:

```
public class Applet1 extends Applet ← 主类类头
{
    String s;
    Label k;
    TextArea t;
    public void init()
    {
        s="欢迎登录本网站";
        k=new Label("请留言");
        t=new TextArea(2, 50);
        add(k);
        add(t);
    }
    public void paint(Graphics g)
    {
        g.drawString(s, 25, 100);
    }
}
```

变量或对象

方法 1

方法 2

类体

例 1.5 程序的运行结果如图 1.7 所示。说明如下:

- 上述程序由类 Applet1 组成。类的内容写在 {} 中。类中包括三个变量(s、k、t)和两个方法(init、paint)。
- 方法的具体内容称为方法体, 写在 {} 中。方法体由若干语句构成, 每条语句用分号“;”结束。init()方法包括 5 条语句。paint()方法包括 1 条语句。
- 方法名前的修饰符 public 说明方法的使用权限。public 类型的方法可以被任何类引用。修饰符 void 说明方法只完成操作, 没有返回值。

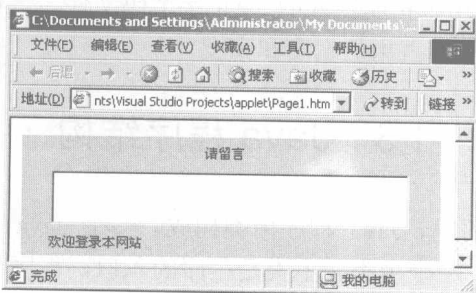


图 1.7 例 1.5 程序的运行结果