



普通高等教育“十一五”国家级规划教材

普通高校本科计算机专业特色教材精选·算法与程序设计

Java语言程序设计 (第3版)

吕凤翥
马皓 编著

清华大学出版社



普通高等教育“十一五”国家级规划教材

普通高校本科计算机专业特色教材精选·算法与程序设计

Java语言程序设计 (第3版)

吕凤翥 马皓 编著

清华大学出版社
北京

内 容 简 介

本书全面而又系统地介绍了Java语言程序设计的基础知识,包括基本语法、编程方法、面向对象的特征,以及Java语言的异常处理、多线程、输入输出和文件操作等内容。在此基础上,本书阐述了Java语言在图形界面设计、集合操作和网络编程等方面的应用。

本书文字简练、概念准确、例题丰富,便于读者理解和掌握;每章均配有大量练习题和上机题,可帮助读者掌握书中的主要内容,检验学习效果。

本书既可作为高等院校相关专业的教材,也适合计算机爱好者、软件开发人员学习Java语言时使用。

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语言程序设计/吕凤翥,马皓编著. —3版. —北京:清华大学出版社,2016
(普通高校本科计算机专业特色教材精选·算法与程序设计)
ISBN 978-7-302-45118-1

I. ①J… II. ①吕… ②马… III. ①JAVA语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2016)第227218号

责任编辑:焦虹

封面设计:傅瑞学

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

印刷者:北京富博印刷有限公司

装订者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:28.25 字 数:646千字

版 次:2006年1月第1版 2016年11月第3版 印 次:2016年11月第1次印刷

印 数:1~2000

定 价:49.90元

产品编号:068793-01

出版说明

INTRODUCTION

在我国高等教育逐步实现大众化后，越来越多的高等学校将会面向国民经济发展的第一线，为行业、企业培养各级各类高级应用型专门人才。为此，教育部已经启动了“高等学校教学质量和教学改革工程”，强调要以信息技术为手段，深化教学改革和人才培养模式改革。如何根据社会的实际需要，根据各行各业的具体人才需求，培养具有特色显著的人才，是我们共同面临的重大问题。具体地，培养具有一定专业特色的和特定能力强的计算机专业应用型人才则是计算机教育要解决的问题。

为了适应 21 世纪人才培养的需要，培养具有特色的计算机人才，急需一批适合各种人才培养特点的计算机专业教材。目前，一些高校在计算机专业教学和教材改革方面已经做了大量工作，许多教师在计算机专业教学和科研方面已经积累了许多宝贵经验。将他们的教研成果转化为教材的形式，向全国其他学校推广，对于深化我国高等学校的教学改革是一件十分有意义的事。

清华大学出版社在经过大量调查研究的基础上，决定编写出版一套“普通高校本科计算机专业特色教材精选”。本套教材是针对当前高等教育改革的新形势，以社会对人才的需求为导向，主要以培养应用型计算机人才为目标，立足课程改革和教材创新，广泛吸纳全国各地的高等院校计算机优秀教师参与编写，从中精选出版确实反映计算机专业教学方向的特色教材，供普通高等院校计算机专业学生使用。

本套教材具有以下特点：

1. 编写目的明确

本套教材是深入研究各地各学校办学特色的基础上，面向普通高校的计算机专业学生编写的。学生通过本套教材，主要学习计算机科学与技术专业的基本理论和基本知识，接受利用计算机解决实际问题的基本训练，培养研究和开发计算机系统，特别是应用系统的基本能力。

2. 理论知识与实践训练相结合

根据计算学科的三个学科形态及其关系,本套教材力求突出学科理论与实践紧密结合的特征,结合实例讲解理论,使理论来源于实践,又进一步指导实践得到自然的体现,使学生通过实践深化对理论的理解,更重要的是使学生学会理论方法的实际运用。

3. 注意培养学生的动手能力

每种教材都增加了能力训练部分的内容,学生通过学习和练习,能比较熟练地应用计算机知识解决实际问题。既注意培养学生分析问题的能力,也注重培养学生解决问题的能力,以适应新经济时代对人才的需要,满足就业要求。

4. 注重教材的立体化配套

大多数教材都将陆续配套教师用课件、习题及其解答提示,学生上机实验指导等辅助教学资源,有些教材还提供能用于网上下载的文件,以方便教学。

由于各地区各学校的培养目标、教学要求和办学特色均有所不同,所以对特色教学的理解也不尽一致,我们恳切希望大家在使用教材的过程中,及时地给我们提出批评和改进意见,以便我们做好教材的修订改版工作,使其日趋完善。

我们相信经过大家的共同努力,这套教材一定能成为特色鲜明、质量上乘的优秀教材,同时,我们也希望通过本套教材的编写出版,为“高等学校教学质量和教学改革工程”作出贡献。

清华大学出版社

前言

PREFACE

诞生于 20 世纪 90 年代初的 Java 语言，伴随着互联网日新月异的发展，已广泛应用于网络编程、嵌入式和移动系统、游戏以及 Web 企业平台等方面。当前，越来越多的人期望学习和掌握 Java 语言，以便更好地了解和投身于互联网创新发展的大潮中。在此背景下，我们根据多年的教学经验和开发实践，于 2006 年推出了《Java 语言程序设计》第 1 版，2010 年又推出了第 2 版，广受读者欢迎与好评。此后，我们继续钻研、不断拓展，针对读者需求，引入 Java 语言的最新发展，对本书进行了进一步的修订，正式推出了第 3 版。

本书深入浅出、简明实用，在突出重点、详析难点、解答疑点的基础上，做到概念明晰、注重实用，通过大量例题，帮助读者理解相关概念和方法，检验学习效果，较快地掌握所学知识。

本书全面而又系统地介绍了 Java 语言的基本语法和重要应用，共分为 10 章，第 1、2 章介绍 Java 语言的特点、Java 程序的基本定义和实现方法以及语法基础；第 3 章着重讨论 Java 语言面向对象的特征；第 4 章介绍 Java Applet 及其应用；第 5 章详细描述作为 Java 语言的重要应用之一的图形用户界面的设计方法；第 6 章对异常处理进行了介绍；第 7 章详细解释线程创建、状态、同步和特性等基本知识；第 8 章介绍 Java 语言的主要特点之一的集合操作；第 9 章介绍 Java 语言的输入输出操作和文件操作；第 10 章介绍 Java 语言重要应用领域网络编程方面的一些典型应用。

学习计算机语言离不开上机实践。读者应在边学习边上机的过程中，深入理解、掌握本书的内容。本书尤其注重实际编程能力的培养与提高，备有上机练习题，读者应按照题目要求，认真上机练习。通过上机调试程序，发现自身概念理解上的模糊和方法运用上的错误，搞清概念和纠正错误对于深刻理解和熟练掌握 Java 语言十分重要。

限于编著者的水平，本书难免出现不足之处，恳请读者提出宝贵意见。

目 录

CONTENTS

第 1 章 Java 语言概述	1
1.1 Java 语言的历史	1
1.2 Java 语言的特点	2
1.3 Java 程序及其实现	4
1.3.1 Java Application 程序举例	5
1.3.2 Java Application 程序的结构特点	7
1.4 Java Application 程序的实现	8
1.4.1 编辑 Java 源程序	8
1.4.2 编译 Java 源程序	8
1.4.3 运行 Java 的字节码文件	9
1.5 Java Application 程序的不同界面输出	10
习题	13
第 2 章 Java 语言语法基础	19
2.1 词法规则	19
2.1.1 标识符	19
2.1.2 关键字	20
2.1.3 分隔符	20
2.1.4 注释符	21
2.2 数据类型	21
2.2.1 基本数据类型	21
2.2.2 复合数据类型	22
2.2.3 类型转换	22
2.3 常量和变量	24
2.3.1 常量	24
2.3.2 变量	27

2.4	运算符和表达式	32
2.4.1	运算符	32
2.4.2	表达式	37
2.5	语句	42
2.5.1	表达式语句和空语句	42
2.5.2	块语句	42
2.5.3	选择语句	43
2.5.4	循环语句	46
2.5.5	转向语句	49
2.5.6	其他语句	53
2.6	数组和字符串	53
2.6.1	数组	53
2.6.2	字符数组和字符串	60
2.6.3	Java Application 命令行参数	62
	习题	63
第 3 章 Java 语言面向对象的特征		77
3.1	面向对象的基本概念	77
3.1.1	对象和类	77
3.1.2	封装性	79
3.1.3	继承性	79
3.1.4	多态性	80
3.2	类的定义	80
3.2.1	类的定义格式	80
3.2.2	变量	81
3.2.3	方法	84
3.2.4	实例	90
3.2.5	静态变量和静态方法	92
3.2.6	抽象类和抽象方法	95
3.2.7	最终类、最终变量和最终方法	97
3.3	对象	99
3.3.1	对象的定义	99
3.3.2	对象的赋值	100
3.3.3	对象的使用	103
3.4	类的继承和多态	107
3.4.1	父类和子类	107
3.4.2	成员的继承和覆盖	113
3.5	接口和包	119

3.5.1	接口	119
3.5.2	包	124
3.6	常用的工具类简介	126
3.6.1	Java 语言的类库	126
3.6.2	Java 语言基础类库	128
	习题	142
第 4 章	Java Applet 及其应用	155
4.1	Applet 概述	155
4.1.1	一个 Applet 的例子	155
4.1.2	HTML 文件	156
4.1.3	Applet 的特点和工作原理	157
4.2	Applet 与 Applet 类	160
4.2.1	Applet 与 Applet 类的关系	160
4.2.2	Applet 类的主要方法	161
4.3	Applet 程序和 HTML 文件	163
4.3.1	HTML 文件	163
4.3.2	向 Applet 传递参数	164
4.4	Applet 的应用	166
4.4.1	绘制图形	166
4.4.2	演示图像	181
4.4.3	播放声音	184
	习题	186
第 5 章	图形用户界面设计	193
5.1	图形用户界面概述	193
5.1.1	图形用户界面的构成	193
5.1.2	图形用户界面的实现	194
5.1.3	界面元素的类结构	197
5.2	事件和事件处理	198
5.2.1	事件处理方法	198
5.2.2	事件类和监听者接口	201
5.3	基本控制组件的使用	208
5.3.1	按钮和标签	209
5.3.2	文本框和文本区	211
5.3.3	单复选框和列表	216
5.3.4	下拉列表和滚动条	221
5.3.5	画布	226

5.4	布局设计	228
5.4.1	顺序布局	229
5.4.2	边界布局	231
5.4.3	卡片布局	232
5.4.4	网格布局	235
5.4.5	复杂的网格布局	237
5.5	常用容器组件的使用	241
5.5.1	容器概述	241
5.5.2	容器 Panel	242
5.5.3	窗口和菜单	245
5.5.4	对话框	257
	习题	263
第 6 章	异常和异常处理	275
6.1	异常和异常类	275
6.1.1	异常	275
6.1.2	异常类	276
6.2	异常处理	278
6.2.1	程序运行时异常	279
6.2.2	捕获异常	280
6.2.3	抛出异常	283
6.3	创建异常	285
6.3.1	用户创建自己的异常	285
6.3.2	举例分析	287
	习题	293
第 7 章	线程	301
7.1	概念	301
7.2	线程的创建	302
7.2.1	继承 Thread 类	302
7.2.2	实现 Runnable 接口	304
7.2.3	两种方式的比较	306
7.3	线程的状态	306
7.3.1	概述	306
7.3.2	启动	307
7.3.3	终止	307
7.3.4	暂停	308
7.3.5	状态判断	310

7.4	线程的同步	311
7.4.1	非同步实例	311
7.4.2	关键词 synchronized 和锁	313
7.4.3	等待 wait()和唤醒 notify()	315
7.4.4	完整的同步实例	316
7.5	线程的其他特性	318
7.5.1	主线程	318
7.5.2	守护线程	319
7.5.3	线程组	321
7.5.4	线程优先级	323
7.5.5	死锁	323
7.5.6	线程池	325
7.5.7	不再使用的方法	327
	习题	328
第 8 章	Java 语言的集合操作	333
8.1	概念	333
8.2	数组 (Arrays) 类	333
8.3	基本的集合接口	335
8.4	集合 (Set)	336
8.5	列表 (List)	340
8.6	映射 (Map)	342
8.7	Collections 类	346
8.7.1	排序	347
8.7.2	同步	350
8.7.3	只读	350
8.7.4	特殊集合	351
8.8	枚举和迭代	352
8.9	历史集合类	354
8.10	使用原则	355
8.11	Java 泛型初探	356
	习题	360
第 9 章	Java 输入/输出	365
9.1	流的概念	365
9.2	输入/输出类	366
9.2.1	基本类	366
9.2.2	类别和功能	367

9.3	标准输入/输出	369
9.4	文件读/写	372
9.4.1	字节流方式	372
9.4.2	字符流方式	375
9.4.3	文件缓冲类	377
9.5	流的包装	381
9.5.1	概念	381
9.5.2	连接多个输入流	383
9.5.3	基本数据流	385
9.5.4	管道操作	388
9.6	文件类	391
9.7	随机访问文件	397
	习题	399
第 10 章	Java 网络编程	403
10.1	概述	403
10.2	URL 访问	405
10.3	InetAddress 类	408
10.4	NetworkInterface 类	409
10.5	Socket 应用	411
10.6	安全套接字应用	417
10.7	UDP 数据报	424
10.8	组播	427
10.9	对 IPv6 的支持	430
	习题	433
	参考文献	439

Java 语言概述

Java 语言是由美国 Sun Microsystems 公司开发的一种面向对象的程序设计语言。Java 是从 C++ 程序语言发展而来的,但比 C++ 语言简单。该语言具有简单、高效、健壮、安全、与平台无关、可移植性好和多线程等特点,是当前网络编程中的首选语言。

1.1 Java 语言的历史

Java 语言是在 C++ 语言已广泛使用,Internet(因特网)发展迅速的历史背景下出现的。因此,Java 语言源于 C++ 语言,又与 Internet 同步发展。

1991 年,美国 Sun Microsystems 公司成立了一个开发面向家电(电视机、烤面包箱等)的编程软件小组,项目名称为 Green,软件名称为 Oak(一种橡树名)。当时 C++ 语言已经比较成熟,自然成为他们开发软件的首选语言。后来,人们发现 C++ 语言比较烦琐,并且安全性较差,不能满足要求,于是决定开发一种新的语言,当时取名为 Oak。

Oak 语言出世后几经磨难,是 WWW 挽救了它。Sun 公司的一位创始人认为 Oak 语言的安全性和分布式系统非常适于 Internet,于是采用了在 Internet 上免费使用的策略。后来经过继续发展和完善,1995 年,Oak 语言改名为 Java 语言。

Java 语言发布后,发展顺利。Sun 公司很快公布了完整的 Java 技术规范,在 WWW 厂商支持下,用 Java 编写的 Applet 小程序在浏览器内部嵌入也获得了成功。不久,又推出一套全新的基于 Java 语言的开放式多媒体格式和应用程序编程接口(Application Program Interface, API),从而为 Java 语言的进一步推广使用开创了美好前景。

由于 Java 语言具有较好的安全性和可移植性及与平台无关等特性,因此,它为 Internet 开辟了一个新时代;而 Internet 的发展,又促进了 Java 语言的开发利用。

1.2 Java 语言的特点

Java 语言是一种卓越的程序设计语言,它具有下述特点。

1. 面向对象

Java 语言是一种面向对象的程序设计语言。面向对象是当前软件开发的先进技术和重要方法。面向对象的方法是基于信息隐藏和数据抽象类型的概念,利用类和对象的机制将数据和方法封装在一起,通过统一的接口与外界交互;通过类的继承机制实现代码重用。Java 语言只支持单继承,但它却支持多接口。Java 语言还支持方法重载和动态调用。总之,Java 语言支持面向对象方法中的 3 个基本特性:封装性、继承性和多态性。面向对象方法反映了客观世界中现实的实体在程序中的独立性和继承性。这种方法有利于提高程序的可维护性和可重用性,还有利于提高软件开发效率和程序的可管理性。虽然,C++语言是面向对象的,但是它为了兼容 C 语言,不得不保留一些面向过程的成分;而 Java 语言去掉了 C++语言中非面向对象的成分,因此,它是一个完全的面向对象的程序设计语言。

2. 简单性

与面向对象程序设计的 C++语言相比较,Java 语言更加简单、高效,生成的可执行目标代码也更加短小。Java 语言中取消了 C++语言中很少使用的、难以理解的、容易混淆的和复杂麻烦的特性和内容。例如:

(1) 在 C++语言中,结构、联合和类的概念重合使用,而 Java 语言只保留了类的概念。

(2) Java 语言取消了 goto 语句,优化了结构;Java 语言支持带标号的 break 语句和 continue 语句。

(3) 运算符重载是 C++语言的特点之一,由于它实际使用较少,作用也不大,因此 Java 语言不支持运算符重载。

(4) C++语言支持单继承和多继承;Java 语言只支持单继承,不支持多继承,但它支持多接口。因为多继承给编程者带来了复杂性,在实际使用中发现多继承难以使用,容易混淆,又浪费资源。因此,取消多继承,使得 Java 语言更简单。

(5) 指针是 C 语言的特点。在 C++语言中指针比在 C 语言中的使用要少,这是因为指针使用困难,容易混淆。因此,Java 语言中取消了指针。

此外,Java 语言还取消了 C++语言中的预处理功能和内存空间的动态申请和释放功能,增添了内存空间自动回收功能。这些措施都使 Java 语言更简单、更方便。

总之,Java 语言是消除累赘和难以理解的部分之后的 C++语言,它是更纯粹的面向对象的程序设计语言。

3. 可移植性

与其他语言相比,用 Java 语言编写的程序可移植性比较高。Java 语言为了保证可移植性采用了下述机制。

(1) Java 语言规定同一种数据类型在各种不同的实现中,必须占据相同的内存空间。例如,short 类型为 16 位,int 型为 32 位,long 类型为 64 位,它们与硬件平台无关。而 C++ 语言不同,数据类型的长度与硬件环境或操作系统有关。例如,int 型数据在 Windows 3.1 中占 16 位,而在 Windows 2000 中占 32 位。由于 Java 语言在数据类型的空间大小方面采用了统一标准,因此保证了其程序的平台独立性。

(2) Java 程序的最终实现需要经过编译和解释两个步骤。Java 语言的编译器生成的可执行代码称为字节码。这种代码是由一种称为 Java 虚拟机(Java Virtual Machine, JVM)生成的,它与任何硬件平台无关。由于 Sun 公司规定的 JVM 规范没有涉及任何硬件平台,因此只要根据 JVM 规范创建的平台便可以实现 Java 程序。另外,Java 程序生成字节码后,使用 Java 解释器来解释执行字节码,这意味着只要装有 Java 解释器的机器 Java 程序都可以运行。Java 语言采用的这种先编译后解释的方法是以牺牲执行速度来换取与平台无关的特性,从而提高了可移植性。

4. 稳定性和安全性

在网络上,应用程序要求更高度的稳定性和安全性。Java 语言在稳定性和安全性方面采取了一些措施。

(1) Java 语言取消了指针概念。C++ 语言在稳定性和安全性方面最大的隐患是指针的使用和缺乏内存的自动管理。Java 语言使用下标表示数组,并加强了对下标的检查;使用类和类的方法访问变量,这样就解决了指针在数组和结构方面的应用问题。由于取消了指针的不安全性,因此增强了系统的健壮性。

(2) Java 语言稳定性的另一个原因是自动的内存管理。在 C++ 语言编程中,要经常申请资源和释放资源,稍不小心就会造成系统故障或存储空间的浪费。而在 Java 语言中,Java 虚拟机增加了一个称为自动垃圾收集器的后台程序,它可用来对内存扫描,自动释放掉不再使用的内存碎片,进而实现了内存的自动管理。

(3) Java 语言的运行环境中提供了安全机制的保障。首先,使用 Java 编译器生成的字节码在进入 Java 解释器时,必须通过字节码校验器的检查,防止字节码在编译或解释期间有意或无意地被破坏,进而增加了安全性。其次,Java 解释器决定了程序中类的内存布局,使得有意破坏者无法预先知道类的内存布局结构,也就无法利用该信息来破坏系统。最后,用户可以限制从网络上装载的类只能访问某些被允许的系统,如文件和硬盘。另外,在信息传输过程中使用密码算法,增加了传输中的安全性。总之,通过上述种种措施,使得 Java 语言具有较好的稳定性和安全性,在一定程度上可以防止病毒和黑客的攻击。

5. 高性能

Java 语言采取了一些软件开发中的先进技术,从而表现出了它的高性能。

(1) 结构中立,与平台无关。在设计 Java 语言时,考虑到该语言能在不同的平台上都具有生命力,因而采取了一种中性结构方式。主要表现在字节码的中介方式,它的生成与平台无关。另外还表现在类的定义上比较严格。对于类的数据成员在内存中占有空间大小是固定不变的,与系统无关。例如,int 型数据占有 4 字节。定义的变量都具有默认值(初值),并且类中定义的变量可以直接初始化(C++语言是不允许的)。

(2) 高级语言通常分为两种:一种是为面向机器的编译型语言,这种语言执行效率高,但移植性差;另一种是解释型语言,这种语言编程效率高,但执行速度慢。Java 语言程序是先编译后解释,介于二者之间。Java 语言编译后生成的字节码与机器语言很接近,从而获得了较高的性能。

(3) 操作系统中两个重要的概念是进程和线程。进程(process)的特点是在执行过程中拥有自己独立的内存空间和系统资源,不同进程的内存数据和状态是彼此独立的。线程(thread)在执行过程中共享一块内存空间和一组系统资源,线程之间可以直接进行数据交换。因此,线程比进程的开销要小。多个线程并行执行,类似于多个 CPU 在同时运行。例如,有两个线程同时工作,一个线程在执行某种复杂的运算,而另一个线程可与用户实现交互。可见多线程可提高程序的运行效率。

Java 语言真正支持多线程的。它通过同步关键字(synchronized)来保证多个线程不会同时访问一个被说明为同步的方法,从而使得某些关键操作不被打断。但是,这种做法有时会降低运行效率。

6. 分布性

Java 语言是一种面向网络的高级语言。Java 语言的应用程序编程接口具有支持超文本传输协议(Hypertext Transfer Protocol,HTTP)和文件传输协议(File Transfer Protocol,FTP)等 TCP/IP 的类库。Java 应用程序可以通过统一资源定位器(Uniform Resource Locator,URL)地址直接访问 Internet 上的各种对象,就如同访问本地对象一样。Java 的小应用程序(Applet)是跨平台的网络应用程序,它被嵌入超文本标记语言(Hypertext Markup Language,HTML)文件中,通过主页发布到 Internet。用户访问到服务器的 Applet 程序时,它们将从 Internet 上传输,然后在支持 Java 的浏览器中运行。

1.3 Java 程序及其实现

Java 语言程序可分为两大类:一类是控制台应用程序 Java Application,它是一种独立完整的程序;另一类是基于浏览器运行的程序 Java Applet,又称小程序,它不是一种独立完整的程序,需要在 WWW 浏览器这种特定环境下运行。这里,仅介绍 Java Application 程序的结构、特点及其实现,而 Java Applet 程序在本书后面的章节中介绍。

1.3.1 Java Application 程序举例

这里先举两个 Application 程序的简单例子。通过这两个例子,读者可初步了解 Application 程序的结构和书写方式。

【例 1-1】 第一个简单的 Java Application 程序。该程序的文件名为 Exam1_1.java。

(1) 程序 Exam1_1.java:

```
import java.io.*;
public class Exam1_1
{
    public static void main(String args[])
    {
        System.out.println("This is a Java Application Program. ");
    }
}
```

该程序的功能是在显示器屏幕的当前光标处输出显示字符串: This is a Java Application Program.

(2) 程序分析:

将该程序中出现的各条语句按其顺序说明如下。

```
import java.io.*;
```

该语句是把 Java 类库中 java.io 包中的所有程序引入到该程序中来。Java 类库中提供了很多个包,java.io 是其中一个有关输入输出的包。每个包中包含若干个模块,每个模块是一个类的定义,它是经过编译的字节码形式的程序。语句 import 是用来将系统提供的具有特定功能的模块引入到用户编写的程序中,用户便可在该程序使用引入模块的各种功能。上述语句是在该程序中引入了 java.io 包中的所有模块,程序中后边出现的输出语句“System.out.println(“...”);”中包含的对象和方法都被包含在 java.io 包中的模块中。

Java 程序是由若干个类组成的。该程序中只有一个类,其名为 Exam1_1,类的定义格式如下:

```
<修饰符> class <类名>
{
    <类体>
}
```

其中,class 是定义类的唯一的关键词。〈类体〉是用花括号括起来的语句组。

该程序中,Exam1_1 类体内只定义一个主方法 main()。该方法无返回值,有一个形参,该参数是一个字符串数组 args[]。主方法 main()是静态的(static),可以通过类名直接调用,其访问权限是公有的(public),该程序中所有类都可使用该方法。

在 main()方法的实现(方法体)中,只有一条语句:

```
System.out.println("This is a Java Application Program. ");
```