




普通高等教育农业部“十二五”规划教材
全国高等农林院校“十二五”规划教材

Java

程序设计



肖 磊◎主编

 中国农业出版社

欢迎登录：中国农业出版
全国农业教育



封面设计 赵正刚

ISBN 978-7-109-16774-2



9 787109 167742 >

定价：24.00元

普通高等教育农业部“十二五”规划教材
全国高等农林院校“十二五”规划教材

J a v a

程 序 设 计

肖 磊 主 编



中国农业出版社

图书在版编目 (CIP) 数据

Java 程序设计/肖磊主编. —北京: 中国农业出版社, 2012. 8

普通高等教育农业部“十二五”规划教材 全国高等
农林院校“十二五”规划教材

ISBN 978-7-109-16774-2

I. ①J… II. ①肖… III. ①Java 语言—程序设计—
高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2012) 第 091778 号

中国农业出版社出版

(北京市朝阳区农展馆北路 2 号)

(邮政编码 100125)

策划编辑 朱 雷

文字编辑 李兴旺

人民教育出版社印刷厂印刷 新华书店北京发行所发行
2012 年 8 月第 1 版 2012 年 8 月北京第 1 次印刷

开本: 787mm×1092mm 1/16 印张: 13.5

字数: 315 千字

定价: 24.00 元

(凡本版图书出现印刷、装订错误, 请向出版社发行部调换)

内 容 简 介

Java 语言具有面向对象、平台无关、安全、稳定、多线程等特点，是目前软件开发的主流程序设计语言之一。本书循序渐进地介绍了 Java 语言的语法基础、开发工具、流程控制语句、数组、字符串、方法等程序设计的基本概念与方法；结合 Java 阐述面向对象程序设计的基本概念：类、对象、接口、抽象、封装、继承和多态等；讨论了 Java 的图形用户界面、异常处理、文件的输入与输出、Java Applet 程序等内容。

本书适合作为高等院校计算机及相关专业 Java 语言程序设计课程的教材，同时可供 Java 语言和程序设计初学者参考使用。

编写人员名单

主 编 肖 磊

参 编 毛宜军 朱 凯 彭红星

周运华 徐东风 史 广

前 言

在当今计算机应用飞速发展及广泛普及的时代，面向对象技术已经成为软件开发的主流技术。面向对象程序设计中的继承、封装、多态等特性非常接近人的思维方式，从而更容易理解和使用。与此同时，面向对象程序设计更加符合现代软件开发规模大、需求变化多的实际情况，有利于提高软件的重用性、可维护性和可扩展性。

Java 语言是主流的面向对象程序设计语言，具有简单、面向对象、分布式、安全、结构中立等特性，已经在软件开发行业广泛应用。从 Java 语言诞生到现在，Java 语言应用已经遍布软件开发的各个领域。在互联网飞速发展时期，Java 在开发 Web 应用程序、大型分布式企业应用中取得了巨大成功。在移动互联网、物联网突飞猛进的今天，Java 在移动应用、客户端及服务端软件开发中也占据了主流地位。

目前越来越多的高校将 Java 语言作为学习程序设计和面向对象技术的编程语言。传统的计算机学科，如计算机科学与技术、软件工程、网络工程等都加强了对 Java 开发能力的培养，与此同时，其他一些理、工、农等专业也采用 Java 培养具有较强软件应用能力的复合人才。

本书是以 Java 语言讲授程序设计的教材，主要让读者学习和理解基本的编程思维方式，包括语言基础、流程控制结构、数组和方法等；然后结合 Java 的特点阐述面向对象技术、图形用户界面、异常处理、流等重要概念与方法。学习本书之前不需要具备程序设计的知识，特别适合把 Java 作为第一门编程语言的读者。

全书共 11 章，阐述内容时注重基础知识与实践应用相结合。第 1 章对程序设计及 Java 语言进行了介绍。第 2 章介绍了 Java 语言的基本语法。第 3 章和第 4 章分别介绍选择结构和循环结构程序的特点及 Java 提供的流程控制语句。第 5 章讲述 Java 的数组和字符串的使用方法。第 6 章介绍了 Java 的方法。第 7 章和

第8章是本书重点,阐述了Java的面向对象特性。第9章介绍的异常处理机制和使用流对文件进行读写操作的方法。第10章介绍了Java图形用户界面基本原理和应用Swing组件的方法。第11章简单描述了Applet的原理和编程方法。每章后面附有复习题和编程练习,同时编写电子版实验指导和教学课件,便于学习和教学。

本书由多名教学一线的专业教师共同编写完成,华南农业大学信息学院完成9章,其中徐东风编写第1章,毛宜军编写第4、5章,朱凯编写第6、11章,肖磊编写第7、8章,周运华编写第9章,彭红星编写第10章;山西农业大学资源环境学院史广编写第2、3章。全书由肖磊统稿。

由于时间仓促和编者水平有限,书中错误和不足之处在所难免,敬请批评指正。

编者

2012年4月

目 录

前言

第 1 章 Java 程序设计语言概述	1
1.1 程序与程序设计	1
1.1.1 程序设计的概念	1
1.1.2 程序设计语言	2
1.1.3 程序设计方法	2
1.2 Java 程序设计语言	5
1.3 Java 程序开发工具	6
1.3.1 JDK 开发工具	6
1.3.2 集成开发环境工具	9
1.4 Java 程序的基本结构	11
习题	13
第 2 章 基本数据类型与基本操作	14
2.1 标识符	14
2.2 基本数据类型	15
2.3 直接量与常量	16
2.3.1 直接量	16
2.3.2 常量	17
2.4 变量与赋值运算符	17
2.5 运算符与表达式	18
2.5.1 算术运算	18
2.5.2 关系运算	19
2.5.3 逻辑运算	20
2.5.4 位运算	20
2.5.5 赋值运算	21
2.5.6 条件运算	21
2.5.7 运算符与表达式的计算规则	21
2.6 数据类型的转换	22
2.6.1 数值型数据之间的转换	22
2.6.2 字符型数据与数值型数据之间的转换	23
2.6.3 字符串与基本数据类型之间的转换	23
2.7 使用 Math 类进行数学运算	24
2.8 基本输入输出	24
2.8.1 命令行输入输出	25



2.8.2 使用对话框进行输入输出	27
2.9 顺序结构程序举例	28
习题	30
第3章 选择结构程序设计	31
3.1 语句块	31
3.2 选择结构和条件判断	31
3.3 if 语句	32
3.3.1 单分支 if 语句	32
3.3.2 双分支 if 语句	34
3.3.3 if 语句的嵌套	36
3.4 switch 语句	39
3.5 选择结构程序举例	41
习题	45
第4章 循环结构程序设计	47
4.1 循环结构解决的问题	47
4.2 while 语句	48
4.3 do...while 语句	49
4.4 for 语句	50
4.5 循环语句的比较	51
4.6 循环嵌套	52
4.7 break 和 continue 语句	53
4.7.1 break 语句	53
4.7.2 continue 语句	55
4.8 循环结构程序举例	56
习题	58
第5章 数组与字符串	60
5.1 一维数组	60
5.1.1 声明数组变量	60
5.1.2 创建数组	60
5.1.3 访问数组元素	61
5.1.4 初始化数组	62
5.1.5 处理数组元素	62
5.1.6 for...each 循环	63
5.1.7 一维数组应用实例	64
5.2 二维数组	65
5.2.1 二维数组的定义与创建	65
5.2.2 二维数组的长度	66
5.2.3 二维数组应用实例	67



5.3 数组的操作	69
5.3.1 数组的复制	69
5.3.2 数组的排序	70
5.3.3 数组的查找	72
5.3.4 使用 Arrays 类对数组进行操作	73
5.4 字符串	74
5.4.1 使用 String 类处理字符串	74
5.4.2 使用 StringBuffer/StringBuilder 类处理字符串	77
5.4.3 字符串应用实例	78
习题	79
第 6 章 方法	81
6.1 定义方法	81
6.2 调用方法	82
6.3 参数的值传递	84
6.4 方法重载	85
6.5 方法与数组	86
6.6 局部变量	89
6.7 方法的递归调用	90
习题	93
第 7 章 对象和类	95
7.1 对象和类的概念	95
7.2 定义类	96
7.3 创建对象	97
7.4 访问对象	98
7.4.1 引用类型和引用变量	98
7.4.2 访问对象的成员	98
7.4.3 数据域的初始值	99
7.4.4 类与对象应用实例	99
7.5 类成员的可访问性	100
7.5.1 包	100
7.5.2 访问修饰符	102
7.5.3 类的封装	103
7.6 类的静态成员	104
7.7 类和对象应用	107
7.7.1 变量的作用域	107
7.7.2 关键字 this	108
7.7.3 对象作为方法的参数与返回值	109
7.7.4 对象数组	110
7.8 实例分析：二维坐标系的圆	112



习题	115
第 8 章 继承和多态	118
8.1 继承	118
8.1.1 使用继承	118
8.1.2 创建子类对象	120
8.1.3 方法覆盖	122
8.1.4 关键字 final	123
8.1.5 对象类型转换与 instanceof 运算符	123
8.1.6 对象类 Object	125
8.2 多态	127
8.3 抽象类	129
8.4 接口	133
8.5 内部类	136
8.5.1 成员内部类	136
8.5.2 匿名内部类	138
8.6 基本数据类型与包装类	138
习题	139
第 9 章 异常处理与输入输出	140
9.1 异常处理	140
9.1.1 理解异常和异常处理	140
9.1.2 抛出异常	141
9.1.3 声明异常	142
9.1.4 捕获异常	143
9.1.5 重新抛出异常	145
9.1.6 自定义异常类	146
9.2 文件类 File	146
9.3 文件的输入输出	148
9.3.1 Java 的流	149
9.3.2 FileReader 和 FileWriter 类	150
9.3.3 PrintWriter 和 Scanner 类	151
9.3.4 FileInputStream 和 FileOutputStream 类	152
9.3.5 DataInputStream 和 DataOutputStream 类	153
9.4 对象的输入输出	155
9.4.1 ObjectInputStream 和 ObjectOutputStream 类	155
9.4.2 可序列化接口 Serializable	156
9.4.3 数组的序列化	157
习题	158
第 10 章 图形用户界面	161
10.1 Java 图形用户界面概述	161

10.2 Swing 容器类	162
10.2.1 顶层容器	162
10.2.2 面板容器	164
10.2.3 向容器中添加组件	164
10.3 布局管理器	166
10.3.1 流式布局 FlowLayout	166
10.3.2 网格布局 GridLayout	167
10.3.3 边框布局 BorderLayout	167
10.3.4 布局管理器应用	167
10.4 Swing 组件	169
10.4.1 Swing 组件的通用特征与辅助类	169
10.4.2 标签组件	172
10.4.3 文本输入组件	172
10.4.4 按钮组件	175
10.5 事件驱动程序设计	178
10.5.1 事件与事件源	178
10.5.2 事件监听器与事件驱动模型	179
10.5.3 动作事件	181
10.5.4 窗口事件	183
10.6 绘图	185
10.6.1 选择绘图组件	186
10.6.2 Graphics 对象的绘图方法	186
10.6.3 绘图与鼠标事件示例	188
10.6.4 绘图与键盘事件示例	190
10.7 对话框	191
10.7.1 JOptionPane 对话框	191
10.7.2 文件对话框	192
习题	195
第 11 章 Java Applet 编程	196
11.1 Applet 基础知识	196
11.2 Applet 程序结构	196
11.2.1 基于 Applet 类的小程序	196
11.2.2 基于 JApplet 类的小程序	197
11.2.3 运行 Applet 小程序	198
11.3 Applet 程序示例	198
11.4 Applet 的安全机制	200
习题	201
主要参考文献	202

第 1 章 //



Java 程序设计语言概述

我们生活的现实世界中充满了各种各样的信息，人们通过运行在计算机中的各种应用软件来高效地处理这些信息。软件是使用程序设计语言开发出来的，程序设计语言有很多种，为什么选择 Java 语言呢？原因是计算技术的未来属于 Internet，而 Java 是非常适合 Internet 的程序设计语言。

本章将介绍有关程序与程序设计的基本概念、Java 语言的基本情况，并说明开发 Java 应用程序的环境和方法。

1.1 程序与程序设计

1.1.1 程序设计的概念

为了让计算机能够按照人的意图处理事务，人们必须事先设计好完成各种任务的程序，并预先将它们存储到计算机系统的存储器中。

从普遍意义上来说，程序是用以解决问题、按一定顺序安排的操作序列，它主要用于描述完成某项功能所涉及的对象和动作规则。在计算机领域，程序是指计算机为完成某项任务所必须执行的指令集合。通过在计算机上运行程序，向计算机发出一系列指令，就可以使计算机按照人们的要求解决特定的问题。

人们可以借助计算机能够理解的语言编写程序，用以告诉计算机应该处理哪些数据以及如何来处理数据，称为程序设计。严格来说，程序设计是指设计、编制和调试程序的方法和过程。瑞士著名的计算机科学家 Niklaus Wirth 提出了一个有名的公式：

$$\text{程序} = \text{算法} + \text{数据结构}$$

从这个公式可以了解到，要编写程序必须要研究如何组织数据，如何对这些数据进行操作。数据是程序操作的对象，因此首先应该确定要解决的问题中需要处理哪些数据，然后设计出对数据进行操作的具体步骤，这些步骤称为算法。对于不同的数据结构，算法是不同的。例如，对 10 个数进行排序，采用数组组织数据和使用十个简单变量组织数据，算法的差别是非常大的。

上面所提到的公式是针对较小规模的程序提出的，对于大型程序，还需要研究如何组织程序的结构，例如可以将一个大型任务分解为若干较小的任务，然后分别对每一个小任务进行程序设计，从而降低程序设计的复杂度。这就要求研究程序设计方法，目的是使程序设计工作规范化，保证程序的质量。因此，进行程序设计要掌握程序设计方法，也就是说采用合理的方法编写程序。

1.1.2 程序设计语言

程序设计语言 (programming language) 是用于编写计算机程序的语言, 使用它把人们解决问题的方法用计算机可以识别的形式表示出来。

在程序设计发展的几十年中, 先后出现的程序设计语言有几千种, 但是真正被广泛使用的不过有十几种。程序设计语言按照发展过程大致可以分成三类。

1. 机器语言 机器语言是指由二进制代码按照一定规则组成的命令集合, 也称为机器指令集合。机器语言是唯一能被计算机直接理解和执行的程序设计语言。机器语言的每一条语句实际上是一条机器指令, 它以二进制形式表示, 书写时为了方便经常使用十六进制形式。机器语言的优点是编写的程序计算机可以直接识别, 执行效率高; 其缺点是指令难记忆, 编程非常困难, 可读性差, 而且由于每种计算机的指令集是不同的, 所以机器语言编写的程序是无法移植的。

2. 汇编语言 为了解决机器语言难记忆、可读性差的缺点, 人们把机器语言中的指令用英文助记符和符号地址来表示, 这种助记符语言称为汇编语言。汇编语言的可读性、可记忆性比机器语言有了很大改善, 而且保持了其执行效率高的优点。由于助记符与机器指令一一对应, 依赖于具体的硬件, 而且汇编语言语句简单, 导致程序编写比较复杂, 可移植性极差。汇编语言编写的程序, 只有先翻译成计算机可以识别的机器指令后才能在计算机上执行。

3. 高级语言 高级语言是更接近人类的自然语言和数学语言的表示方法。其特点是在一定程度上与具体计算机的结构无关, 具有更强的表达能力。使用高级语言可以方便地表示数据的运算和程序的控制结构, 使用户更容易掌握。

高级语言使用方便、易学, 使用范围覆盖了计算机应用的各个方面。现在比较流行的高级语言从编程方法上可以分为面向过程的语言, 如 BASIC、PASCAL、C 等; 面向对象的语言, 如 C++、Java、C# 等。

1.1.3 程序设计方法

程序设计方法是指导程序设计的思想方法, 包括程序设计的原理和所应该遵循的基本原则, 帮助人们从不同角度去描述问题域。在程序设计过程中, 选择一种良好的程序设计方法将有助于提高程序设计的效率, 保证程序的可靠性, 增强程序的可扩充性, 改进程序的可维护性。

程序设计发展到今天, 用于指导程序设计的方法已经有很多种, 它们均有各自的特点和优势, 其中结构化和面向对象是发展最为成熟、应用最为广泛的两种程序设计方法。

1. 结构化程序设计方法 结构化程序设计方法 (structured programming) 是 20 世纪 70~80 年代非常流行的程序设计方法。其所强调的结构化主要体现在以下三个方面。

(1) 自顶向下, 逐步求精。结构化程序设计方法把程序设计看成一个逐步演化的过程。自顶向下是指将分析问题的过程划分为若干个层次, 每一个新的层次都是上一个层次的细化, 即步步深入、逐层细分。

(2) 模块化。整个系统经过层层分解后形成若干模块, 每个模块实现特定的功能, 最终整个系统将由这些模块组装而成。模块之间通过接口传递信息, 力求模块具有良好的独立



性。模块实际上是对自顶向下、逐步求精产生的各个组成部分的实现，即每个模块实现一个子系统的功能。如果某子系统继续划分为更加具体的，那么模块之间形成上下层关系，上层模块通过调用下层模块来实现自己的功能。

(3) 流程控制结构化。首先支持结构化程序设计方法的语言必须提供函数或过程等语言要素实现模块的概念；其次，结构化程序设计要求在程序中只允许出现顺序、分支和循环三种控制结构，并且每一种结构必须符合单入口/单出口原则。

2. 面向对象程序设计方法 结构化程序设计方法在程序设计发展过程中发挥了重要作用，但是进入 20 世纪 80 年代以后它的弱点逐步显露了出来，这要求人们继续研究新的程序设计方法来代替它，这就是面向对象的程序设计方法 (object oriented programming)。

面向对象的程序设计方法是指用面向对象的方法指导程序设计的整个过程。所谓面向对象是指以对象为中心，分析、设计和构造程序的机制。与结构化方法相比，面向对象方法实现了从问题域到求解域的直接映射，从而实现了计算机系统对现实世界的真正模拟。

Alan Kay 在 1993 年提出了面向对象程序设计的基本特征：

- 一切都是对象 (object)。
- 计算通过对象间相互通信，请求其他对象执行动作来实现。对象通信通过发送和接收消息 (message) 实现。
- 每个对象都有自己的存储空间，其中可能包含了其他对象。
- 每个对象都是每个类 (class) 的实例 (instance)。
- 类是具有共同属性和行为的对象进行抽象的结果。
- 类被组织成单个根节点的树状结构，也称继承的层次结构。

因此，面向对象程序设计方法可以用下面的公式形象地描述出来：

面向对象 = 类 + 对象 + 继承 + 消息 + 通信

面向对象的程序设计方法包括一些基本概念，准确地理解这些概念对于深入理解面向对象的程序方法非常重要。

(1) 抽象 (abstract)。抽象是人类认识世界的本能方式。所谓抽象是指从许多事物中，舍弃个别的、非本质的属性，抽取出共同的、本质属性的过程，它是形成概念的必要手段。例如，在学生成绩管理程序中，研究学生对象时，要抽象出其与分析问题相关的信息，班级、课程、成绩等，而忽略身高、体重等信息。

抽象包括过程抽象和数据抽象两个部分。所谓过程抽象是指功能抽象，即抽取共同的、本质的行为而舍弃个别的行为。数据抽象是更高级别的抽象，它将现实世界中的客体作为抽象单元，其抽象内容既包括客体的属性特征，也包括行为特征。数据抽象是面向对象程序设计的核心方法。

(2) 封装 (encapsulation)。封装是将现实世界中某个个体的属性和行为组合一个逻辑单元 (内部) 的机制。利用封装可以将属性和行为隐藏起来，外部只能通过特定的接口方法访问隐藏的属性和方法。封装有两层含义：一是可以将数据和代码结合成一个完整的对象；二是对象可以通过封装拥有私有成员，将对象内部实现细节隐藏起来的能力。

封装是构造类的基本方法，保证了类具有较好的独立性，防止外部程序修改类的内部数据，提高了程序的可维护性。当需要修改程序时，把修改范围限制在类的内部，将程序修改带来的影响降到最低。



(3) **对象 (object)**。对象是面向对象程序设计的核心概念。在现实世界中，对象可以泛指任何事物，可以是具体实体，如一个人、一辆车等；也可以是抽象的概念，如一项计划、一个行动等。任何对象都具有自己的状态，如汽车有停止、行驶状态；同时对象具有特定的行为，如汽车有刹车、加油等行为。

在面向对象的程序设计中，对象概念是对现实世界对象的模型化，它是代码和数据的组合，同样具有自己的状态和行为。对象的状态用数据的值表示，称为属性 (property)；对象的行为用代码的集合表示，称为对象的方法 (method)。为了区别同一程序的不同对象，每个对象都有一个唯一的标识。

(4) **类 (class)**。类是一组具有相同属性和行为的对象的抽象描述，具体地说，类是面向对象程序的基本构成单位，是抽象数据类型的具体实现，是创建对象的模板。类与对象关系非常密切，类是对象的抽象结果，对象是类具体化，是类的实例。

类与类之间存在三种主要的关系：关联、聚合和泛化。关联是指两个或多个类之间的特定关系，最常见的是两个类之间的关系，称为二元关联。例如，学生类与课程类之间存在“学生选修课程”的关联。聚合也称为“整体—部分”关系，它描述的是整体类由部分类组成的关系。例如，计算机类是由主机类、显示器类、键盘类等组成。泛化描述了两个类之间的“一般—特殊”的关系。例如，形状类与矩形类之间存在泛化关系。在设计类时，不仅要考虑类内部的属性和方法，还要明确类之间的各种关系。

(5) **消息 (message)**。面向对象程序运行过程中，对象之间通过消息传递进行通信，从而推动程序的运行。消息是一个对象对另一个对象的请求，它包括消息的接收者、要求接收者完成的方法及必要的参数。一个对象只有在获得了另一个对象的标识后才能向其发送消息，消息传递一般通过方法调用实现。

(6) **继承 (inheritance)**。继承是泛化关系的实现，其中一般概念称为“父类”或“基类”，特殊概念称为“子类”或“派生类”。继承为类与类之间共享数据和方法提供了一种良好的机制。通过继承，子类可以得到父类的可继承成员，包括属性和方法，在此基础上，子类可以扩充或覆盖父类的成员。

继承具有可传递性。如果类 C 继承类 B，而类 B 继承类 A，那么类 C 也获得了类 A 的可继承成员，此时称类 A 是类 C 的祖先类，类 C 是类 A 的后代类。如果一个类可以有多个直接父类，称为多继承；反之，如果一个类只能有一个直接父类，称为单继承。并不是所有的面向对象程序设计语言都支持多继承，如 Java、C# 等只支持单继承，而 C++ 支持多继承。

(7) **多态 (polymorphism)**。对象在收到消息时要做出响应，不同对象在收到同一消息时可以产生完全不同的响应结果，这种现象称为多态。利用多态机制，可以在程序中发送一个通用消息，而收到消息后完成工作的细节由接收消息的对象自行实现，这样同一个消息会导致调用不同的方法。例如，在绘图程序中，不同的图形对象都有自己的 draw 方法，程序中只需发出一个调用 draw 方法的消息，不同的图形对象收到后会在屏幕上绘制出不同的图形。

上面阐述的几个基本的概念构成了面向对象的程序设计方法的四大特性：抽象、封装、继承和多态。深入理解这四个特性是掌握面向对象程序设计方法的关键。

面向对象的程序设计具有很多优点：能够实现对现实世界的自然描述；可控制程序的复