



面向“十二五”高职高专规划教材·计算机系列

Java程序设计与实验

■ 万年红 主编
■ 王雪蓉 田启明 副主编

清华大学出版社 · 北京交通大学出版社

面向“十二五”高职高专规划教材·计算机系列

Java 程序设计与实验

万年红 主 编
王雪蓉 田启明 副主编

清华大学出版社
北京交通大学出版社
· 北京 ·

内 容 简 介

本书以 J2SE 平台为基础, 主要介绍面向对象程序设计思想与方法, 主要包括 Java 语言基本概述, Java 类的实现、对象的产生, Java 程序运行环境的搭建安装、配置与使用, Java 程序的基本语法, 方法与变量, Java 包的定义与使用, 抽象性、封装性、继承性、多态性, 抽象类、抽象方法、接口及内部类的声明及使用, 常用的 Java API 实用类, 图形用户界面设计方法和步骤、常用控件、事件类、监听器、适配器及事件处理模型, 异常与异常处理机制, 多线程, Applet 与 WEB 开发应用、Applet 的多媒体程序开发, Java 数据库开发, 输入和输出流处理等内容。

本书非常适合作为高职高专计算机及相关专业学生的教材, 同时也可作为软件开发人员的参考书。

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 程序设计与实验/万年红主编. —北京: 清华大学出版社; 北京交通大学出版社, 2009. 11
(面向“十二五”高职高专规划教材·计算机系列)

ISBN 978-7-81123-905-8

I. ① J… II. ① 万… III. ① JAVA 语言-程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2009) 第 202966 号

责任编辑: 谭文芳

出版发行: 清华大学出版社

邮编: 100084

电话: 010-62776969

<http://www.tup.com.cn>

北京交通大学出版社

邮编: 100044

电话: 010-51686414

<http://press.bjtu.edu.cn>

印刷者: 北京东光印刷厂

经 销: 全国新华书店

开 本: 185 × 260 印张: 19.25 字数: 493 千字

版 次: 2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷

书 号: ISBN 978-7-81123-905-8/TP · 541

印 数: 1~4 000 册 定价: 29.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

当今的社会是信息的社会，以网络技术为核心的信息技术的迅猛发展，为计算机网络编程提供了前所未有的发展机遇。Java 是当前最为流行的纯面向对象的网络编程语言之一，它所具有的面向对象性、简单性、分布性、健壮性、安全可靠性和平台无关性、可移植性、解释性、多线程、动态性和高性能使得它特别适合开发网络应用程序。目前，用 Java 开发的手机程序、网络游戏比比皆是，在移动电话、家用电器等领域中无一没有 Java 技术的存在。Java 完全具备了面向对象的抽象性、封装性、继承性、多态性等特征，被 Bill Gates 确定为有史以来最伟大的面向对象程序设计语言。

本书主要内容如下。

第 1 章是概述部分，主要介绍程序设计与程序设计语言，面向对象基本概念及与面向过程的区别，面向对象技术基本要素、特征，Java 的发展历史及特点，Java 类的实现、对象的产生及构造方法，Java 程序运行环境的搭建安装、配置与使用，简单 Application 和 Applet 程序的编译与运行等内容。

第 2 章是 Java 程序设计基础部分，主要介绍 Java 程序结构及编写规范，注释、关键字、标识符，数据类型及其转换、变量和常量，运算符及其优先级、表达式、数组，程序的流程控制语句等内容。

第 3 章是方法与变量部分，主要介绍方法的声明与调用格式，参数传递方式，嵌套与递归，对象与变量的生命周期，变量初始化及赋值，static 方法和变量，main() 的语法等内容。

第 4 章介绍包和封装性，主要包括包的概念及分类，用户自定义包的声明及建立途径，import 语句的使用，类的封装、访问权限等内容。

第 5 章介绍继承性，主要包括继承性的概念，实现继承的语法格式，多级继承、单继承和多重继承，构造方法的调用次序，对象的类型转换等内容。

第 6 章介绍多态性，主要包括多态性概念，方法重载与方法覆盖，关键字 this、super 和 final 的使用情况。

第 7 章介绍抽象类、接口和内部类，主要包括抽象类和抽象方法，接口的概念，接口声明与使用，内部类和匿名内部类等内容。

第 8 章介绍 Java API 实用类，主要包括 Java API 分类，String 类和 StringBuffer 类，基本数据类型的对象包装类、Math 类、Vector 类等内容。

第 9 章介绍图形用户界面设计，主要包括 GUI 概念及设计 GUI 所需的程序包、一般要素与基本过程，AWT 基本组件类、容器类、布局管理器类，事件处理模型、要素、流程及常用方法，菜单系统和工具栏的设计，Swing 组件，组件的外观、颜色、字体控制等内容。

第 10 章介绍异常处理，主要包括异常及其发生的原因与异常类，异常处理机制，try...catch...finally 语句，throws 语句、throw 语句，重新抛出异常、自定义异常等内容。

第 11 章介绍多线程，主要包括程序、进程和线程的联系与区别，线程的创建途径，Thread 类和 Runnable 接口，线程的优先级、生命周期，线程控制，同步与锁定等内容。

第12章介绍 Applet 与 Web 应用开发, 主要包括 Applet 程序运行过程与生命周期, Applet 的多媒体程序开发, Applet 网络技术和应用开发, Web 开发应用等内容。

第13章介绍 JDBC 技术和数据库应用开发, 主要包括 JDBC 访问数据库的一般步骤, JDBC 访问数据库所需要相应接口、类及方法, 以及数据库编程的具体实现等内容。

第14章介绍输入输出流处理, 主要包括流的概念、特点和分类, 基本的输入输出流, 输入输出流类及接口的继承层次结构, InputStream 类、OutputStream 类、Reader 类、Writer 类, File 类, 标准输入输出流, FileInputStream 和 FileOutputStream 类, 过滤字节流等内容。

Java 程序设计是一门很难学的课程, 特别是对高职高专的学生来说, 更是无从入手。正因为难学, 编者通过多年从事 Java 课程教学的经验以及与同仁的交流沟通, 体会到 Java 教材的编写应该从高职高专的生源文化基础、学习接受能力、自学能力实际情况出发。因此, 本书总体编写思路如下。

(1) 本书在有限的计划课时内, 从高职高专计算机及相关专业学生实际接受能力出发, 集教师讲授与学生实验为一体, 实验部分以课后练习形式出现, 并提供例题代码、习题实验代码和重难点突出、内容较完整的课件。

(2) 本书突出章节主题及重点、难点, 整体层次结构及逻辑布局合理, 思路集中, 章节衔接, 通俗易懂, 知识连贯成系统, 代码简洁, 介绍前面的内容时基本不涉及后面未学的内容, 符合读者的思维, 避免知识点超前带来的问题, 使读者学习思路集中、快速进入学习状态, 提高读者开发 Java 项目的能力和以 Java 为先修课程的后续课程的学习兴趣, 在减少教师工作量的同时又能提高教学效果。

(3) 附录部分提供了一个实践教学案例系统, 从需求分析、可行性研究和基本信息模块、考勤考评信息模块、系统维护信息模块等方面入手, 介绍了后台数据库规划、前台界面设计、后台数据库的连接方法及前台界面操作事件, 逐渐培养学生正确运用面向对象的思维分析、解决问题的能力, 从而具备较强的软件开发能力。

综上所述, 本书非常适合作为高职高专计算机及相关专业的教材, 同时也可作为软件开发人员的参考书。

本书由浙江东方职业技术学院万年红老师主编并负责统稿。浙江东方职业技术学院王雪蓉老师, 温州职业技术学院田启明老师、张焰林老师、黄河老师作为副主编参与编写了部分内容。具体分工为: 第1章、第3章、第4章、第7章、第8章、第9章、第11章、第13章、附录部分由万年红老师编写; 第2章、第10章、第12章、第14章由王雪蓉老师编写; 第5章由田启明老师编写; 第6章由张焰林老师、黄河老师编写。

本教材编写过程中得到了浙江东方职业技术学院及工程技术系吴建平老师, 温州职业技术学院田启明老师, 北京交通大学出版社及谭文芳编辑的大力支持和重要指导, 温州职业技术学院叶展翔老师参与了书稿第5章内容的校对工作, 在此一并表示诚挚的感谢。

本书所涉及的一些参考内容, 均在本书最后以参考文献形式列出, 对其作者或提供者表示感谢。

限于编者水平有限和时间匆忙, 书中有些错误在所难免, 敬请广大读者批评指正。

万年红

2009年9月于温州

目 录

第1章 概述	1
1.1 程序设计	1
1.2 程序设计语言及其类型	1
1.3 面向过程程序设计与面向对象程序设计	2
1.3.1 面向过程程序设计	2
1.3.2 面向对象程序设计	2
1.4 类的实现	4
1.4.1 Java 类的定义	5
1.4.2 对象的产生与使用	5
1.4.3 构造方法	6
1.5 面向对象的基本特性	8
1.5.1 封装性	8
1.5.2 继承性	9
1.5.3 多态性	9
1.6 Java 语言概述	9
1.6.1 Java 语言发展历史	9
1.6.2 Java 语言特点	9
1.6.3 Java 语言与 C、C++ 的区别	10
1.7 Java 开发环境的搭建	11
1.7.1 Java 技术的架构	11
1.7.2 Java 程序与 C 程序的编译、执行过程比较	12
1.7.3 JDK 安装与配置	13
1.7.4 Java 类库	16
1.7.5 Java 程序类型	16
1.8 Java 集成开发环境	18
1.9 简单 Java 程序编写、编译与运行	18
1.9.1 Application 程序编译与运行	19
1.9.2 Applet 程序编译与运行	21
1.9.3 Application 和 Applet 运行过程的比较	22
小结	22
习题	22
第2章 Java 程序设计基础	25
2.1 Java 程序结构及编写规范	25

2.1.1	Java 程序基本规范	25
2.1.2	关键字	26
2.1.3	标识符	27
2.2	数据类型、常量与变量	27
2.2.1	数据类型	27
2.2.2	常量	28
2.2.3	变量	29
2.3	数据类型转换	31
2.3.1	自动类型转换	31
2.3.2	强制类型转换	32
2.4	Java 中的运算符与表达式	32
2.4.1	运算符	32
2.4.2	表达式及运算符优先级、结合性	36
2.4.3	语句和语句块	37
2.5	程序的流程控制	38
2.5.1	顺序结构	38
2.5.2	分支语句	39
2.5.3	循环语句	44
2.5.4	跳转语句	48
2.6	数组	50
2.6.1	数组概念	50
2.6.2	数组的声明与使用	50
2.6.3	数组的分类	51
2.6.4	一些与数组操作有关的方法	54
小结		56
习题		56
第3章	方法与变量	60
3.1	方法	60
3.1.1	方法的声明	60
3.1.2	方法的调用	61
3.1.3	方法嵌套与递归	62
3.1.4	方法参数传递方式	64
3.2	变量的作用域和生命周期	66
3.2.1	变量的作用域和生命周期	66
3.2.2	对象的作用域和生命周期	67
3.3	变量初始化	68
3.3.1	成员变量初始化	68
3.3.2	局部变量初始化	69
3.4	static 方法和变量	69

3.5	main()方法	71
3.6	垃圾收集	72
	小结	73
	习题	73
第4章	包和封装性	76
4.1	包的创建和访问	76
4.1.1	包的概念及分类	76
4.1.2	用户自定义包的创建	79
4.1.3	包创建的过程实例	81
4.2	import 语句与包的使用	82
4.3	类的封装与作用域	83
4.3.1	封装的含义及目的	83
4.3.2	访问权限	84
4.3.3	使用 set 和 get 方法	86
	小结	87
	习题	87
第5章	继承性	90
5.1	继承性的概念	90
5.2	继承的实现	91
5.3	多级继承和单继承	92
5.4	继承关系下构造方法的调用次序	93
5.5	对象的类型转换	95
	小结	96
	习题	96
第6章	多态性	99
6.1	多态性概念	99
6.1.1	方法重载	99
6.1.2	方法覆盖	101
6.2	this 和 super 的使用	103
6.2.1	this 的使用	103
6.2.2	super 的使用	104
6.3	final 关键字	105
	小结	106
	习题	106
第7章	抽象类、接口和内部类	109
7.1	抽象类和抽象方法	109
7.1.1	抽象类	109
7.1.2	抽象方法	110
7.2	接口	113

7.2.1	接口的概念	113
7.2.2	接口的声明	113
7.2.3	接口的继承	114
7.2.4	接口的实现	115
7.3	内部类	117
7.3.1	内部类概念及声明	117
7.3.2	内部类功能特点	118
7.3.3	匿名内部类	119
	小结	121
	习题	121
第8章	Java API 实用类	124
8.1	Java API 分类	124
8.2	String 类和 StringBuffer 类	124
8.2.1	String 类	125
8.2.2	StringBuffer 类	130
8.3	其他类	131
8.3.1	Math 类	131
8.3.2	基本数据类型及其包装类对应关系	133
8.3.3	Vector 类	135
	小结	136
	习题	137
第9章	图形用户界面设计	140
9.1	图形用户界面	140
9.1.1	设计 GUI 所需的程序包	140
9.1.2	AWT 基本知识	141
9.2	AWT 组件类	144
9.2.1	LayoutManager 类	144
9.2.2	Container 类	150
9.2.3	非容器基本组件类	152
9.3	事件处理模型	156
9.3.1	基本要素与流程	156
9.3.2	事件处理常用方法	158
9.4	菜单系统的设计	165
9.4.1	与菜单系统相关的 AWT 类	165
9.4.2	菜单系统的实现方法与步骤	166
9.5	组件的外观控制	168
9.5.1	颜色控制	168
9.5.2	文字显示效果控制	169
9.6	Swing 组件	171

9.6.1 与 AWT 组件的比较	171
9.6.2 几种常用的 Swing 组件	173
小结	176
习题	176
第 10 章 异常处理	181
10.1 异常	181
10.1.1 了解异常	181
10.1.2 异常发生原因及异常类	182
10.2 异常处理	184
10.2.1 try...catch...finally 语句	185
10.2.2 重新抛出异常	187
10.2.3 自定义异常类	189
小结	190
习题	190
第 11 章 多线程	193
11.1 理解与创建线程	193
11.1.1 线程概念	193
11.1.2 建立多线程	194
11.1.3 线程的优先级	201
11.1.4 线程的生命周期	203
11.2 线程控制	203
11.3 线程的同步与锁定	205
小结	206
习题	206
第 12 章 Applet 与 Web 应用开发	209
12.1 Applet 类	209
12.1.1 如何运行 Applet	209
12.1.2 Applet 生命周期及其方法	210
12.2 Applet 多媒体程序开发	210
12.2.1 Graphics 类	211
12.2.2 图形的绘制	211
12.2.3 图像操作	214
12.2.4 音频处理	215
12.2.5 视频程序开发	218
12.3 Applet 网络技术和应用开发	220
12.3.1 Java 网络技术概述	220
12.3.2 Applet 与 URL	220
12.3.3 基于流套接字的通信	223
12.3.4 基于数据报套接字的通信	225

12.4 Web 开发应用	226
12.4.1 什么是 Servlet	226
12.4.2 Servlet 编程有关的类和接口	227
12.4.3 JSP 技术	227
12.4.4 JavaBean 在 JSP 中的使用	229
小结	231
习题	232
第 13 章 JDBC 技术和数据库应用开发	236
13.1 Java 数据库程序开发原理	236
13.1.1 JDBC 技术简介	236
13.1.2 JDBC 访问数据库的一般步骤	236
13.2 JDBC 提供的主要类和接口	238
13.2.1 DriverManager 类	238
13.2.2 Connection 接口	239
13.2.3 Statement 接口	240
13.2.4 ResultSet 接口	241
13.3 数据源的建立	243
13.3.1 建立 Access 数据源	243
13.3.2 建立 SQL Server 数据源	245
13.4 Java 数据库编程实现	255
小结	259
习题	259
第 14 章 输入输出流处理	262
14.1 基本的输入输出流	262
14.1.1 概述	262
14.1.2 InputStream 类	265
14.1.3 OutputStream 类	266
14.1.4 Reader 类	267
14.1.5 Writer 类	267
14.2 File 类	268
14.2.1 建立 File 对象	268
14.2.2 File 对象的属性和操作	269
14.3 文件输入输出流类	270
14.3.1 FileInputStream 类	270
14.3.2 FileOutputStream 类	271
14.4 过滤字节流类	273
14.4.1 DataInputStream 和 DataOutputStream 类	273
14.4.2 BufferedInputStream 和 BufferedOutputStream 类	275
14.5 InputStreamReader 和 OutputStreamWriter	275

14.5.1	BufferedReader 和 BufferedWriter 类	275
14.5.2	FileReader 和 FileWriter 类	277
	小结	278
	习题	279
附录 A	人事管理系统开发案例	282
A.1	系统需求分析及可行性研究	282
A.2	功能模块	282
A.3	后台数据库规划	282
A.4	前台界面设计	284
A.5	后台数据库的连接	287
A.6	前台界面操作事件	287
A.7	程序完整的代码	291
	参考文献	296

第 1 章 概 述

本章要点:

- ☒ 程序设计与程序设计语言
 - ☒ 面向对象基本概念及与面向过程的区别
 - ☒ 面向对象技术基本要素、特征
 - ☒ Java 的发展历史及特点
 - ☒ Java 类的实现、对象的产生及构造方法
 - ☒ Java 程序运行环境的搭建安装、配置与使用
 - ☒ 简单 Application 和 Applet 程序的编译与运行
 - ☒ Java 集成开发环境
-

1.1 程序设计

程序设计实际上就是编程,程序设计的两个基本要素是数据结构和算法。数据结构是计算机的操作对象;算法是利用操作对象对特定任务求解过程的一种描述,是指令的有序序列。实际上,程序设计就像造汽车,数据结构就像发动机、轮胎等原部件,算法则是汽车设计图纸。这些部件就是物质基础和原材料,但是这些部件不会自动组装起来,必须按照汽车设计图纸一个部件一个部件地去组装,否则,汽车是不可能生产出来的,二者相辅相成。因此,进行程序设计时,数据结构和算法是相互依存、缺一不可的。没有主题思想的程序不能叫作程序,只是一堆杂乱无章的符号而已。

另外一个典型的例子是机器人程序设计,假如数据结构是机器人的各种部件,算法则是机器人的思想,如果机器人从一个位置(初始位置)移到另一个位置(目标位置),我们可以用算法去支配机器人的各种部件随意运动,使它可以跳过去、走过去、跑过去、爬过去,但是无论如何,机器人的部件没有变化,而从初始位置到目标位置方式却是多样的,这就是算法表现出来的弹性。因此可以说,数据结构的内容是有限的,是死的,但是算法却不同,算法是鲜活的,它可以让数据以用户想要的方式排列。

简而言之,程序是指按时间顺序依次安排的工作步骤,而程序设计则是有机地集成一堆指令实现某种功能,就是对这些步骤的编排和优化。

1.2 程序设计语言及其类型

在计算机中,实现程序设计的语言工具很多,如 Visual Basic、Java、C、C#、C++、Power-

Builder、Delphi 等,我们把这些语言统称为程序设计语言,它们是用于编写计算机程序的语言。那么程序设计语言的基础是什么呢?就像人的语言由一些符号和语法组成一样,程序设计语言也是由一组符号和一组规则组成的,符号串和规则的有机集成就是程序设计。程序设计语言的基本构成涉及语法、语义和语用这三个方面。从字面理解,语法表示构成程序的各个符号之间的组合规则,就像学习英语时用单词造句一样,不指明这些单词的含义及其使用者;语义其实就是语法中的表示各个符号的特定含义,但不指明是谁在使用;语用则表示程序与用户之间的关系。

从语言级别的角度,程序设计语言可以分为低级语言和高级语言。低级语言包括机器语言和汇编语言。机器语言是由 0 和 1 组成的原始二进制编码,是表示成编码形式的基本指令集,是最早出现的、最基础的计算机编程语言。计算机只能识别机器语言,由其他语言编写的程序最终都要转换成机器语言才能被计算机识别。汇编语言比机器语言更灵活一些,它是把原始二进制地址部分符号化的结果,它的每条语句都对应一条机器语言。高级语言是利用人类语言中的词和语法的程序设计语言,最接近于人类语言的智能思维和解决问题的方法。高级程序设计语言种类众多,例如 FORTRAN、COBOL、PASCAL、ALGOL69、Modula、Ada 等早期程序设计语言和目前流行的 Visual Basic、Java、C、C#、C++、PowerBuilder、Delphi 等程序设计语言。

从用户要求的角度,程序设计语言可以分为面向过程程序设计语言和面向对象程序设计语言。面向过程程序设计语言的主要特征是以过程和功能为中心,即先干什么,接着干什么,最后干什么。常见的面向过程程序设计语言是 FORTRAN、COBOL、PASCAL、C 等,不过已经逐渐在淘汰;面向对象程序设计语言以客观现实中的对象为基础,无论对象是怎么来的,我们只需知道它可以干什么,怎么去干,而不必知道它的执行过程。当前程序设计语言的主流就是面向对象,常见的面向对象程序设计语言是 C++、C#、Java 等。但实际上 C++、C#既有面向对象的特征,又有面向过程的特征,而 Java 是纯面向对象的。

1.3 面向过程程序设计 with 面向对象程序设计

1.3.1 面向过程程序设计

在面向过程的程序设计中,一个任务由许多可以被单独设计、编码、测试的模块(子任务)

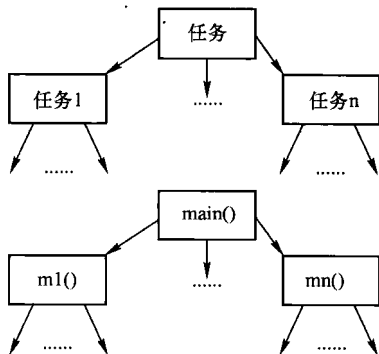


图 1-1 面向过程的程序设计流程

组成,然后这些模块被组装成一个完整的程序,这些模块就是过程或函数,一个过程或函数就是一个语句序列所表示的功能,各模块之间可以相互调用。即面向过程程序设计的总体编程思想是以功能和过程为中心,执行步骤是先做什么,再做什么,最后做什么。面向过程的程序设计流程(如 C 语言)如图 1-1 所示。

1.3.2 面向对象程序设计

1. 对象及其要素

对象(object)是一件事、一个实体、一个名词,可以获

得的、可以想象有自己的标记的任何东西,它不仅能表示客观世界中的任何具体的实体事物,还能表示抽象的规则或计划。比如一辆汽车、一个人、一间房子、一张桌子、一株植物、一张支票、一件雨衣、一只鸟、一个计划书等都是一个对象。概括起来说就是:万物皆对象。

对象是属性(又称为状态或域)和行为(又称为操作)的封装体。属性用来描述对象的状态;行为用来描述对象的操作,用于改变对象的状态。对象实现了属性和操作的结合,使属性和操作封装在对象实体中,即对象 = 属性 + 操作。例如“学生 a”这个对象,具有学号、姓名、班级、年龄、性别等属性和在讲话、在看书等行为。当然,“学生 b”这个对象也有学号、姓名、班级、年龄、性别等属性和在讲话、在看书等行为。即同类对象具有相同属性和行为,但属性值和行为实现方式可以不同,见表 1-1。

表 1-1 两个学生的属性和行为列表

对象	属性	属性值	行为	行为实现方式	表 达 式
学生 a	姓名	‘张三’	讲话	大声讲话	学生 a. 姓名 = ‘张三’
	年龄	20	看书	精读	学生 a. 年龄 = 20
学生 b	姓名	‘李四’	讲话	小声讲话	学生 b. 姓名 = ‘李四’
	年龄	21	看书	泛读	学生 b. 年龄 = 21

2. 面向对象程序设计总体思想

面向对象程序设计把一个任务分解成若干个对象,描述的是对象在解决某任务时的属性和行为,表现为对象与对象之间的相互关系及作用。面向对象程序设计的总体编程思想是:以客观现实中的对象为基础,不管对象怎么实现,只需知道它可以干什么,怎么让它去干。面向对象的程序设计流程如图 1-2 所示。

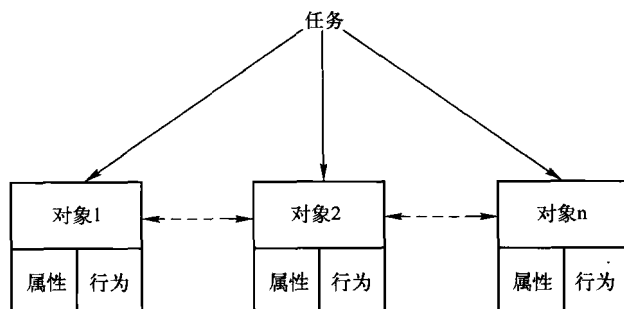


图 1-2 面向对象的程序设计流程

因此,实现设计面向对象程序,仅有一个对象是不够的,需要若干个对象共同作用,这就涉及类的概念。编写面向对象程序的过程实际上是定义类及创建对象的过程。

3. 类

具有相同或相似性质的对象的抽象就是类。也就是说,将一组对象抽象出共同属性和行为,就形成了类,即类是一组具有相同属性和行为的对象的抽象。因此,对象的抽象是类,类具有属性,它是对象状态的总体抽象,用数据结构来描述类的属性。类具有操作,它是对象行为的总体抽象,用操作名来描述。

类的具体化就是对象,类的实例(Instance)是对象,即对象是类的实例化结果。

类与对象的关系是抽象与具体的关系,如同一个模具与用该模具铸出的铸件之间的关系。类给出了属于该类的全部对象的抽象定义,是一组对象的模板;对象则是符合这种定义的一个实例,即把一个类实例化后即可生成该类的一个对象。同一个类可按同种方法创造出多个对象,它们刚开始的状态都是一样的,例如,按照“本田 Exi”型设计图纸生产出来的汽车刚开始

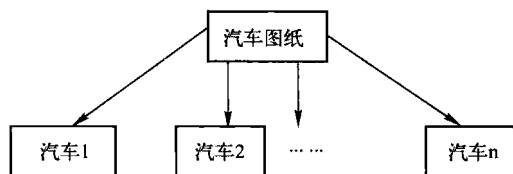


图 1-3 汽车图纸和汽车实例的关系

都是一样的,其中一辆“本田 Exi”汽车被改装后,是不会影响到同型号的其他“本田 Exi”汽车的。但如果修改了“本田 Exi”型汽车的设计图纸,就会影响到以后所有出厂的“本田 Exi”汽车。汽车图纸和汽车实例的关系如图 1-3 所示。

再比如,给定一组学生 1、2、3、……从中抽象出它们共同的属性(班级、姓名、年龄、性别、成绩等)和行为(查询、修改、打印、看书、写字等),就形成了一个名为“学生”的类。再由学生类实例化若干个对象:学生 a、b、c、d、……抽象与实例化过程如图 1-4 所示。

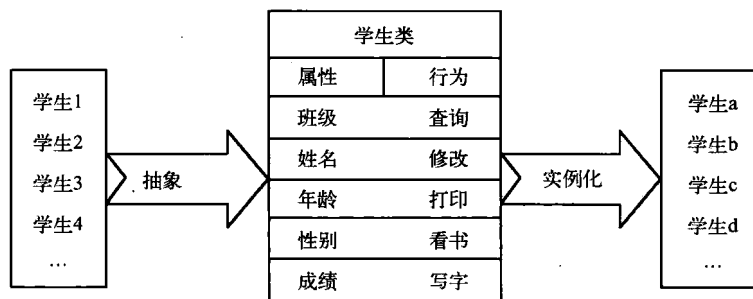


图 1-4 抽象与实例化过程

4. 类的结构

面向对象的程序实际设计过程中,所用到的某些类之间具有一定的结构关系,即“一般与具体”结构关系、“整体与部分”结构关系。

“一般与具体”结构关系称为分类结构,表现为“或”关系或是“is a”关系。

“整体与部分”结构关系称为组装结构,表现为“与”关系或是“has a”关系。

5. 消息和方法

面向对象的程序中,对象间必须进行交互来实现复杂的行为。一个对象是通过它的外部接口和某种操作与其他对象进行交互的。对象间进行交互的单元叫作消息。一个对象向另外一个对象发送一条消息时,消息至少应该包含接收者和发送者的名称和操作名(即对象名、方法名)。

1.4 类的实现

类是一种重要的复合数据类型,封装了一组对象的状态和行为,是这一类对象的原型。为了对类有进一步的认识,本节先简单了解一下用 Java 创建图 1-4 中的学生类及其对象的过程。

我们先做如下约定:用成员变量表示属性,用成员方法(成员函数)表示行为。创建 Java

类时要用到关键字 `class`。

1.4.1 Java 类的定义

1. 完整定义格式

一个完整的 Java 类的定义包含类头和类体两部分,定义格式如下:

```
[修饰符] class 类名 [extends 父类]    //类头
{
    [修饰符] 类型 成员变量                //属性、状态、域
    [修饰符] 类型 成员方法() { 行为实现方式 } //行为、操作
    [修饰符] 构造方法() { ..... }
}
```

成员变量和成员方法统称类的成员。方法用来实现对象的行为,一般对类的属性进行操作,在多数情况下要完成一个具体、独立的任务,方法编写好后,可被重复使用。一个 Java 程序可以由若干个不同的类组成,但其中最多只能有一个类被 `public` 修饰符修饰,被 `public` 修饰的类称为主类,主类名就是源程序的名称,其他类称为功能类。若一个程序中没有一个类被 `public` 修饰,则包含 `main()` 方法的类就是主类,它的名称就是源程序的名称。

2. 简化定义格式

为了方便、快速地定义一个 Java 类,可以对完整的 Java 类的定义格式进行简化。

简化定义格式如下:

```
class 类名
{
    类型 成员变量;
    类型 成员方法() { ... }
}
```

下面用 Java 来实现图 1-4 中的学生类。

【例 1-1】 学生类的定义。

```
class StudentClass //类名
{
    String Class; //定义变量(班级属性)
    String Name; //定义变量(姓名属性)
    int age; //定义变量(年龄属性)
    String Sex; //定义变量(性别属性)
    int score; //定义变量(成绩属性)
    void print() //定义行为,方法头
    {
        System.out.print("打印"); //方法体,即行为实现方式
    }
}
```

该程序定义了一个学生类 `StudentClass` (若要编译,请在 `class` 前面加上 `public`)。接下来可以利用 `StudentClass` 类创建图 1-4 中的学生对象。

1.4.2 对象的产生与使用

1. 对象的产生

在例 1-1 中实现了图 1-4 所示的学生类,那么如何从 `StudentClass` 类中具体化学生对象