



工业和信息化“十三五”
人才培养规划教材



Java

系统化项目|开发教程

Java Systematized Project Development Course

古凌岚 张婵 罗佳 © 编著



知识点结构的系统化，各个章节不仅安排了合适的知识点学习，还在章节的末尾做了知识点的梳理

大项目开发分解为各个子项目，做到**系统化讲解**

全书分“**先导项目**”和“**实战项目**”两个阶段进行，力求做到学习思维的**系统化**



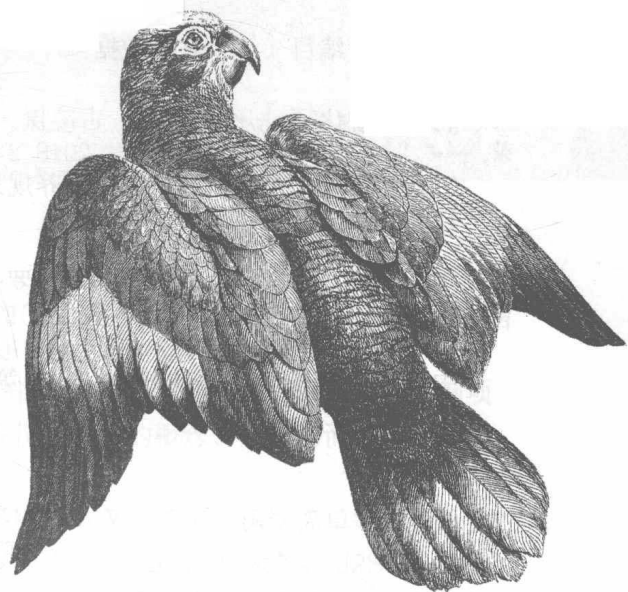
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



工业和信息化部“十三五”
人才培养规划教材



Java

系统化项目|开发教程

Java Systematized Project Development Course

古凌岚 张婵 罗佳 © 编著

人民邮电出版社

北京

图书在版编目 (C I P) 数据

Java系统化项目开发教程 / 古凌岚, 张婵, 罗佳编
著. — 北京: 人民邮电出版社, 2018. 2
工业和信息化“十三五”人才培养规划教材
ISBN 978-7-115-47670-8

I. ①J… II. ①古… ②张… ③罗… 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(2018)第000028号

内 容 提 要

本书以培养面向对象编程思维、Java 桌面应用开发技能为目标, 以提出问题、解读问题、知识探究和解决问题为基本思路, 采用控制台窗口呈现方式, 着重突出算法逻辑过程, 以“图形参数计算程序”为载体, 阐述面向对象的抽象概念及结合 Java 基本语法的编程应用; 并进一步利用对用户更为友好的图形界面方式, 以“闹钟工具软件”的设计实现为主线, 融入 Java 的主要编程机制, 如事件处理机制、输入/输出机制、异常处理机制等的应用, 适时地补充一些实际应用需要, 但常被忽视的知识点, 如 this 关键字、匿名类、集合 Collection 等, 达到学以致用用的效果。考虑到以案例设计实施为线索展开阐述, 可能会使得知识点较为松散, 每个项目单元结尾还对相关知识点进行系统化梳理, 便于学习者形成完整的知识链。

本书适合作为高等院校相关专业的 Java 课程教材使用, 也可供编程爱好者自学使用。

-
- ◆ 编 著 古凌岚 张 婵 罗 佳
责任编辑 范博涛
责任印制 马振武
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 19.5 2018 年 2 月第 1 版
字数: 488 千字 2018 年 2 月北京第 1 次印刷
-

定价: 49.80 元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

运用知识来解决实际问题学习的根本出发点和最终归宿。只有运用知识来解决问题，才能使所学知识成为学习者自身知识库的有机组成部分，进而逐步转化为专业能力。本书通过案例项目的开发过程，由浅入深地介绍 Java 知识，使之不再仅仅是抽象的概念、知识，而是切实可行的、解决实际问题的有力工具。

Java 语言自问世以来，就因其面向对象、支持多线程、与平台无关、语法简单等特点而独具魅力，很快得到了开发人员的青睐，尤其是在 Web 应用开发上。Java 技术包括 J2SE、J2EE 两个应用层面，前者是后者的基础，而后者则用于 Web 应用开发，本书主要介绍 J2SE 部分知识，同时注重后续 Web 应用开发的知识准备。

本书共 10 个项目，主要介绍了如何应用 Java 语言实现面向对象的编程。设计了一简一繁两个实际项目：利用简单项目，讲述面向对象思想、面向对象程序设计方法等抽象知识，让学习者更加容易接受；而复杂项目的实用性更强，涵盖了 J2SE 中的界面设计和编程机制，包括常用组件、事件处理的机制、异常处理的机制、读写文件/数据库表的操作、线程的实现及通信、网络应用程序开发等知识。

在知识点的引入及叙述方式上，本书以案例项目为载体，通过执行效果阐述学习目标，围绕问题，展开知识点的讲解和应用，并给出了完整的实施过程和源码。特别需要指出的是，在新概念的引入上，本书采用实际生活中大家所熟悉的例子来类比，从而使概念更加生动且人性化，更容易理解，进而对概念的运用也更加得心应手。在设计应用方面，先使用简单文本编辑工具，以便了解 Java 程序的编译、运行步骤，而后面的案例均使用主流开发工具软件 Eclipse。

本课程建议授课学时为 50，项目训练学时为 30。

本书的内容结构如下。

项目 1：通过面向过程到面向对象的演变，引入面向对象的基本概念，并介绍了 Java 的历史、特点以及应用领域。

项目 2：主要介绍 Java 程序分类、开发运行环境和工具，以及数据类型、变量、控制语句等基本语法。

项目 3：主要介绍 Java 面向对象的编程方法，并对类和对象的特性、接口等概念做进一步介绍。

项目 4：主要介绍 Java 的引用类型，主要 GUI 组件、布局管理的使用和 Java 2D 绘图机制。

项目 5：主要介绍事件及事件处理的机制。

项目 6：主要介绍 Java 的异常处理机制。

项目 7：主要介绍 Java 中的 I/O 机制，以及文件读写和数据库读写方法。

项目 8：主要介绍集合类、泛型，并重点阐述了 ArrayList、HashSet、HashMap、Properties 类的应用。

项目 9：主要介绍线程的概念、线程的创建，以及线程并发控制、线程通信机制和线程通信实现。

项目 10：主要介绍 TCP/IP、Socket 概念，以及利用 Socket 进行网络编程的方法。

为方便读者使用，书中全部实例的源代码及电子教案均免费赠送给读者，读者可登录人民邮电出版社教育社区（www.ryjiaoyu.com）下载。

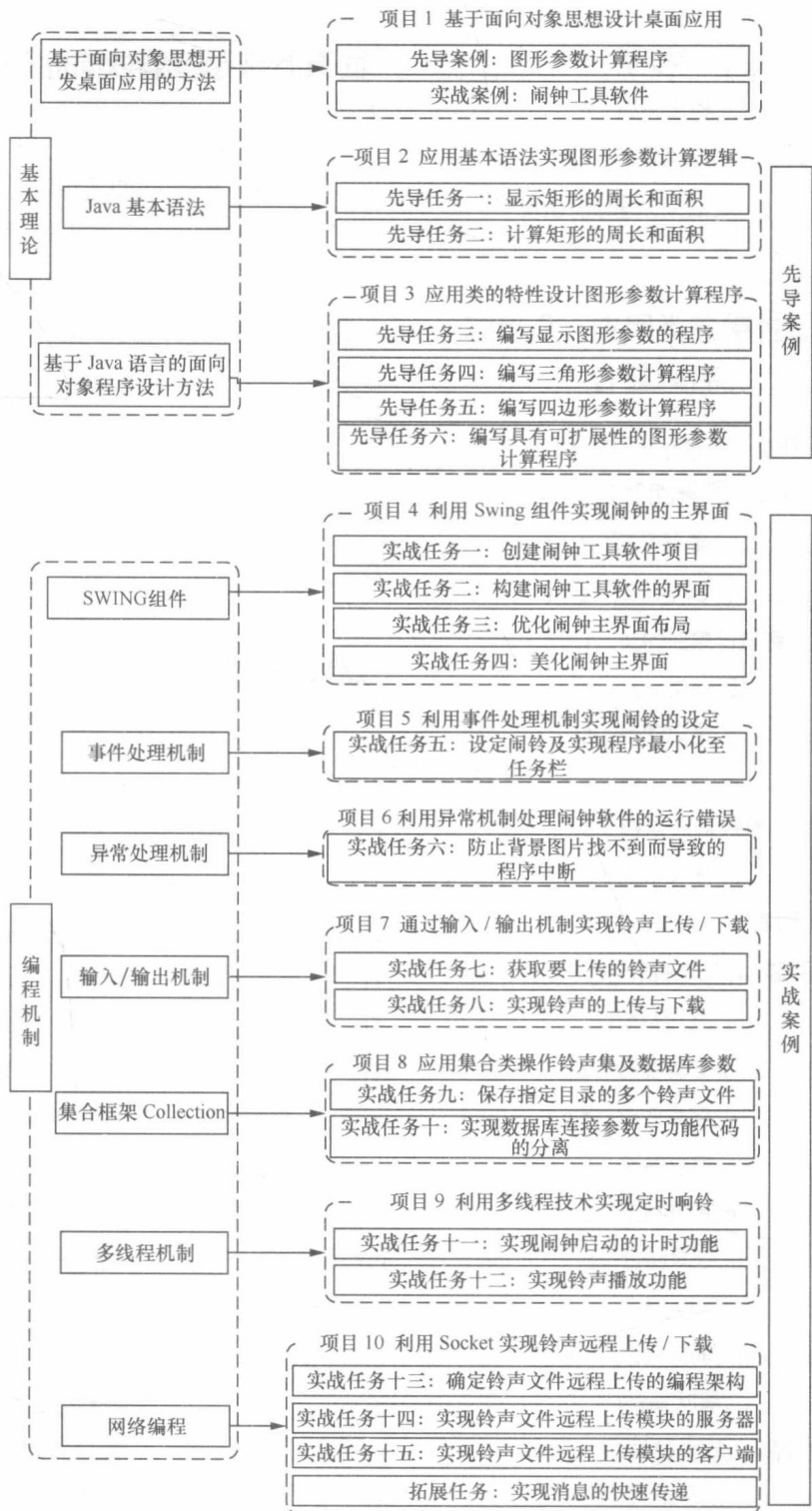
本书由古凌岚、张婵、罗佳编著，古凌岚审定。

由于编者水平有限，书中不妥之处在所难免，殷切希望广大读者批评指正。同时，恳请读者一旦发现问题，于百忙之中及时与编者联系，以便尽快更正，编者将不胜感激，E-mail：1999106010@gdltc.edu.cn。

编者

2017 年 11 月

知识组织结构图



项目 1 基于面向对象思想设计桌面应用..... 1

1.1 面向过程和面向对象..... 2	
1.1.1 两种分析问题的思维方式..... 2	
1.1.2 两种程序设计方法..... 2	
1.2 初识对象、类和实体概念..... 4	
1.3 了解 Java 语言..... 5	
1.4 项目案例及设计..... 7	
1.4.1 先导案例：图形参数计算程序..... 7	
1.4.2 实战案例：闹钟工具小软件..... 7	
知识梳理..... 8	

项目 2 应用基本语法实现图形参数计算逻辑..... 9

2.1 先导任务一：显示矩形的周长和面积..... 10	
2.1.1 什么是 Java 程序..... 10	
2.1.2 Java 开发环境及工具..... 11	
1. Java 开发包..... 11	
2. Java 开发工具..... 12	
3. JDK 的安装..... 12	
2.1.3 计算机处理 Java 程序的过程..... 13	
2.1.4 任务实施..... 15	
2.2 先导任务二：计算矩形的周长和面积..... 15	
2.2.1 任务解读..... 15	
2.2.2 知识学习..... 15	
1. 基本数据类型..... 15	
2. 变量和常量..... 17	
3. 数据类型转换..... 18	

4. 操作符..... 19	
5. 控制语句..... 24	
2.2.3 任务实施..... 32	
知识梳理..... 33	

项目 3 应用类的特性设计图形参数计算程序..... 34

3.1 先导任务三：编写显示图形参数的程序..... 35	
3.1.1 任务解读..... 35	
3.1.2 知识学习..... 35	
1. 类的抽象与封装..... 35	
2. Java 类的定义..... 35	
3. 类的成员..... 38	
4. 包..... 41	
5. 编写 Java 程序..... 42	
3.1.3 任务实施..... 43	
3.1.4 知识延伸：代码编写规范..... 44	
3.2 先导任务四：编写三角形参数计算程序..... 45	
3.2.1 任务解读..... 45	
3.2.2 知识学习..... 45	
1. 类的继承..... 45	
2. 访问控制符..... 46	
3. 包的划分..... 49	
3.2.3 任务实施..... 49	
3.2.4 知识延伸：this、super 关键字和 Class 对象..... 51	
3.3 先导任务五：编写四边形参数计算程序..... 55	

3.3.1 任务解读.....	55
3.3.2 知识学习.....	55
类的多态.....	55
3.3.3 任务实施.....	57
3.4 先导任务六：编写具有可扩展性的图形参数计算程序.....	58
3.4.1 任务解读.....	58
3.4.2 知识学习.....	59
1. 抽象类和接口.....	59
2. 非访问限制符 abstract、static 和 final.....	62
3.4.3 任务实施.....	64
知识梳理.....	68

项目 4 利用 Swing 组件实现闹钟的主界面 70

4.1 实战任务一：创建闹钟工具软件项目.....	71
4.1.1 Eclipse 安装与配置.....	71
4.1.2 任务实施.....	73
4.2 实战任务二：构建闹钟工具软件的界面.....	79
4.2.1 任务解读.....	79
4.2.2 知识学习.....	80
1. 引用类型.....	80
2. 图形用户界面的常用组件.....	86
3. 用户界面设计的基本原则.....	102
4.2.3 任务实施.....	103
4.3 实战任务三：优化闹钟主界面布局.....	108
4.3.1 任务解读.....	108
4.3.2 知识学习.....	109
布局管理器.....	109
4.3.3 任务实施.....	117
4.4 实战任务四：美化闹钟主界面...	122

4.4.1 任务解读.....	122
4.4.2 知识学习.....	123
1. Java2D 绘图机制.....	123
2. 颜色设置.....	124
3. 字体设置.....	124
4. 图像加载.....	125
5. 图形绘制.....	128
4.4.3 任务实施.....	129
知识梳理.....	135

项目 5 利用事件处理机制实现闹铃的设定.....137

5.1 实战任务五：设定闹铃及实现程序最小化至任务栏.....	138
5.1.1 任务解读.....	138
5.1.2 知识学习.....	138
1. 事件模型.....	138
2. 事件处理机制.....	139
3. 内部类.....	142
4. 事件处理的应用.....	145
5. 事件适配器类.....	153
5.1.3 任务实施.....	155
知识梳理.....	170

项目 6 利用异常机制处理闹钟软件的运行错误.....172

6.1 实战任务六：防止背景图片找不到而导致的程序中断	173
6.1.1 任务解读.....	173
6.1.2 知识学习.....	174
1. 异常.....	174
2. 异常处理机制.....	174
3. 异常的捕获和处理.....	175
4. 自定义异常.....	182

10.1 实战任务十三：确定铃声文件
远程上传的编程架构 279

10.1.1 任务解读.....	279
10.1.2 知识学习.....	279
1. 初识网络编程.....	279
2. 网络协议.....	280
3. 套接字.....	281
4. Java 网络类.....	282
10.1.3 任务实施.....	288
10.2 实战任务十四：实现铃声文件 远程上传模块的服务器	288
10.2.1 任务解读.....	288
10.2.2 任务实施.....	289
10.3 实战任务十五：实现铃声文件 远程上传模块的客户端	293

10.3.1 任务解读.....	293
10.3.2 任务实施.....	294
10.4 拓展任务：实现消息的快速 传递.....	296
10.4.1 任务解读.....	296
10.4.2 知识学习.....	296
数据报 Socket 编程.....	296
10.4.3 任务实施.....	298
知识梳理.....	300

附录 知识图谱	301
---------------	-----

Java

Chapter 1

项目 1

基于面向对象思想设计桌面应用

【知识要点】

- 面向过程和面向对象
- 类、对象和实体
- Java 语言特性



引子：程序设计思想与程序设计语言有什么关系？

当你使用软件时，你会发现同类软件的响应速度、易用性等方面可能有很大的差异。影响软件性能的主要因素是软件本身，而软件是由程序构成的。什么样的程序设计能让程序在实现功能的同时能够高效运行？我们先来看一个式子：程序=程序设计方法+算法+数据结构+语言工具及环境。程序设计思想是指程序设计方法和问题的分析模式；程序设计语言则是一种具体的表达方式。当你面对一个问题，经过分析思考有了清晰的解决思路，就可以用计算机能够接受的描述方式（某种编程语言），在计算机上实现对问题的处理。简言之，程序设计思想就是使用程序设计方法去描述现实世界，程序设计语言则是在计算机世界中，对程序设计思想的具体表达。

“思想”比“语言”更重要，这话不无道理。一方面，我们要学习前人经验的结晶，如面向过程、面向对象思想，将其应用到程序设计中；另一方面，我们还可以通过语言的学习，体会这些编程思想，不断运用、总结、领悟，从而形成自己的思想。

1.1 面向过程和面向对象

1.1.1 两种分析问题的思维方式

设计程序是为了解决问题，而分析问题所采用的思维方式不同，将会导致编码方式的不同。面向过程和面向对象是目前主要的两种思维方式。我们先通过一个例子来初步了解一下。

例子：张三使用洗衣机洗衣服。

1) 从面向过程的思想来看，张三先打开洗衣机的电源，打开洗衣机盖子，放衣服进去，洗好后关闭电源。

2) 从面向对象的思想来看，张三告诉洗衣机自己要洗衣服，“洗衣机.打开电源”，“洗衣机.打开盖子”，“洗衣机.放衣服进去”，“洗衣机.关闭电源”。

对上面洗衣的过程做一个比较如下（见图 1-1）。

面向过程：张三是主体，需要完成一系列操作洗衣机的动作，如接电源、打开盖子……；

面向对象：洗衣机是主体，它就像是一个写好代码的机器人，张三只需要提出自己的要求，洗衣机会主动地打开电源、打开盖子……。

由上面的例子可知，面向过程和面向对象本质上都是可以解决问题的，相对而言，面向过程主要关注的是解决问题的每个步骤，面向对象会显得更为方便些，可直接使用某个对象（如洗衣机）提供的功能，当然，前提条件是对象事先已具备了这些功能。

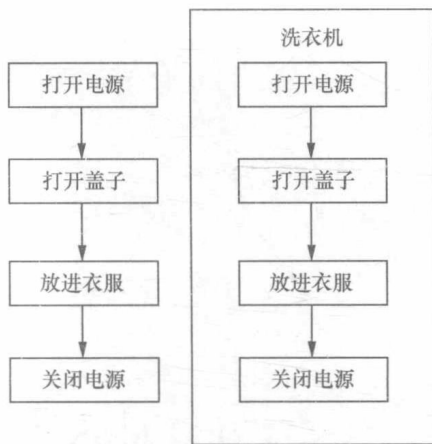


图1-1 面向过程和面向对象分析问题的比较

1.1.2 两种程序设计方法

面向过程的程序设计方法，是针对解决问题的一系列步骤，编写相应的一系列函数来完成，且每个函数又通过基本程序结构（顺序、选择和循环结构），来描述对数据的操作。因此，这种程序设计方法，是以功能为中心来设计函数，但函数和对其操作的数据是分离开来的。

面向对象的程序设计方法，是先提炼数据，再编写一系列操作这些数据的函数，并将函数与其操作的数据有机地组装在一起，作为一个整体来处理。所以，这种方法是以数据为中心来描述系统的，相对而言，数据较功能更具稳定性，程序也就更加易于维护。

由于面向过程的程序设计结构与问题的解决流程相对应，所以，整个程序的执行控制流程也是由预先确定的顺序来决定的；而面向对象程序在运行时，是根据用户的实际需要，去触发相应的函数执行，如洗衣机打开电源后，可以多次或多人去洗衣服（即多次放衣服进去），最后再关闭电源，因此，这种方法更符合实际需要。

下面再通过使用上述两个方法设计“五子棋”游戏（见图 1-2），进一步了解各自的特点。

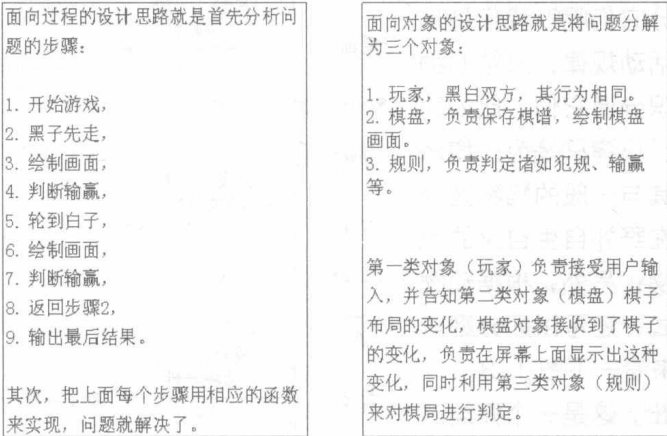


图 1-2 面向过程和面向对象分析五子棋游戏的比较

有人形象地将面向过程和面向对象程序方法分别比喻为蛋炒饭和碟头饭（盖浇饭），蛋炒饭是所有材料味道融为一体，你要么接受这种混合味道，要么放弃，无法改变部分；而碟头饭是由饭、菜拼放在一起，各自独立，你可以只要部分，还可以再加些什么。显然，面向对象中“各司其职”的特点，使得其可扩展性强，比如，“五子棋”要加入悔棋功能，面向过程的设计中的 2~7 步都必须修改，甚至调用顺序也要调整；而面向对象的设计中，只要修改棋盘对象，根据棋谱回溯一下即可，其他不变。

表 1-1 对面向过程和面向对象进行了比较，也让我们有了更全面的认识。

表 1-1 面向过程和面向对象比较

对比项	面向过程（PO）	面向对象（OO）
设计思想	分析问题，划分为多个步骤，用函数实现	分析问题，划分为多个功能，并用多个对象实现
构成公式	程序=算法+数据	程序=对象+消息
特点	基于算法，过程驱动	基于对象，事件驱动
优缺点	运行效率相对高，可重用性差，可扩展性差	运行效率相对低，可重用性好，可扩展性好
编程语言	C	Java、C#
适用范围	数据少操作多，如设备驱动程序、算法实现	数据多操作单一，如信息管理系统、网站

总之，面向过程是分析得到解决问题的步骤，然后用函数把这些步骤逐个实现，使用的时候逐个依次调用。面向对象是把构成问题的事务分解成各个对象，建立对象的目的是为了完成一个步骤，而是为了描叙某个事物在整个解决问题步骤中的行为。它们都遵循将问题分解再解决的基本原则，但对问题的思考、处理方式以及编码实现却迥然不同，不能简单地评判孰优孰劣，而是要从实际出发，考虑系统规模、应用领域、扩展需要、执行效率等方面，选择适用的方法，或

是结合起来运用，以达到最佳开发效益的最终目标。

1.2 初识对象、类和实体概念

对象、类与实体是面向对象中的重要概念。对于初学者来说，理解起来比较困难。面向对象的方法类似于人类认识现实世界的思维过程，所以，先从身边事说起，想想我们生活的每一天，都会接触到许多事物，比如：乘公车去上学，为客户提供服务，看到飞着的鸟，小区里跑着的小狗小猫，去银行开个账户……，其中划线词都是事物，如何区分它们呢？我们是从这些事物的外观形态活动规律，总结归纳其共同点，逐渐地认识分清它们。比如动物，一般家养的小猫、小狗只是为了增添生活情趣，我们会将其与一般的猫狗区分开，称为宠物；而将在野外自生自灭的那些动物，称为野生动物；另外，根据动物的外貌特征、出行方式，还可以区分哪些是飞禽、走兽。我们来看一下图 1-3。

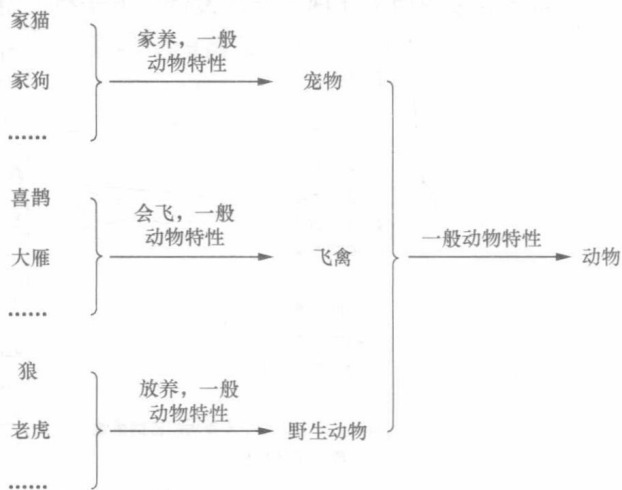


图1-3 动物的归类过程

从图 1-3 可以看出，这是一个归类过程，即将具有相似特征的事物归为一个类（Class）。为了更好地认识类，我们再画一个分类图（假设有只猫的名字叫小黑），如图 1-4 所示。

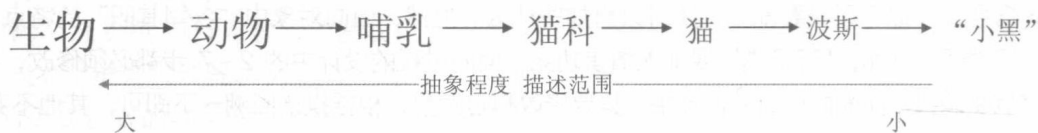


图1-4 归类过程

看到猫的归类过程，我们会发现从波斯到生物类，抽象度越高的类，概括性越强，所指范围也越大，且抽象度高包含抽象度低的类，如哺乳类包括猫科、犬科……；猫科包括猫、虎……；猫类包括波斯、索马里……。进一步，以波斯类为例，我们知道，波斯猫具有头圆且大、长毛、短尾、毛色多为银灰或白、举止文雅、捕鼠高手等特点，其中“头圆且大、长毛、短尾、毛色为银灰或白”是描述猫的状态，而“举止文雅、捕鼠高手”是描述猫的行为。也就是说，类概括了一组相似事物的状态和行为，称状态为属性（Property），行为为方法（Method）。读者可以分析图中其他类，来进一步加深理解。接着，我们来看“小黑”，它是一只只能看见摸得着的猫，毛长 5 厘米、银灰色、喜爱吃巧克力。那么，“小黑”是独一无二的，且有具体的属性（毛长、毛色）和行为（喜爱吃巧克力），我们将这样存在于现实世界中的具体事物称之为对象（Object）。要说明的一点是，这里所说的对象不一定是看得见的，像上面提到的账户，它只是银行处理系统中的一些数据，但它也是对象。

另外，我们还可以发现类与对象间的联系，类描述的一些属性，如波斯猫毛的长度、颜色只是一个定性或范围，当这些属性定量时，如毛长为 5 厘米、毛色为银灰的那只叫“小黑”的猫，就是指一个对象了。

思考题：分析自行车的状态、行为，并描述某辆自行车的具体属性和行为。

刚才所讲到的事物如猫、账户、客户、“小黑”……，都是现实世界客观存在的，我们称为实体。但它们所描述的内容不同，如果所描述的内容是一个类，如猫、客户，称为实体类，若所描述的内容是现实世界中的一个对象，如“小黑”、一个真实的账户，则称为实体实例。

对于类、对象和实体有了一些感性认识后，再来看一下面向对象中的专业定义。

1) 对象：将一组数据和作用于其的一组操作封装而形成的实体。

这里的数据是用于描述对象的状态，操作是指对象自身或外界施加的行为，通过操作将改变对象的状态。举个例子，运动员听到哨音，起跑。对象（运动员）通过自身的行为（起跑），改变了对象（运动员）的状态（静止→跑动）。

2) 类：对具有相同或类似性质的一组对象的共同描述。

3) 实体：客观存在并可相互区别的事物称之为实体。

我们研究上述概念的目的，实际是为了传递消息，所以，这里不得不提到消息（Message）这个概念。

4) 消息：对象之间进行通信的结构。

举个例子，张三（对象）打电话给李四（对象），请他吃饭。消息包括：李四（消息接收对象）和应邀吃饭（接收对象要执行的操作信息）。所以，一条消息将包含消息的接收者和要求接收者完成某项操作。

思考题：你骑自行车，识别其中的对象，分析自行车状态的改变，以及你和自行车之间的消息。

1.3 了解 Java 语言

Java 是一种面向对象的程序设计语言，它诞生于 1995 年，在不长的发展历史中，很快受到业界和许多开发人员的推崇。它具有以下特性。

（1）简单性

Java 语言是一种面向对象的语言，它通过提供最基本的方法来完成指定的任务，只需理解一些基本的概念，就可以用它编写出适合于各种情况的应用程序。Java 略去了运算符重载、多重继承和数据类型自动转换等模糊的概念，并且通过实现自动垃圾收集大大简化了程序设计者的内存管理工作，有助于减少软件出现的错误。另外，Java 也适合于在小型机上运行，它的基本解释器及类的支持只有 40 KB 左右，加上标准类库和线程的支持也只有 215 KB 左右。

Java 语言的简单性是以增加运行时系统的复杂性为代价的。以内存管理为例，自动内存垃圾处理减轻了面向对象编程的负担。对开发人员而言，Java 的简单性可以使我们的学习曲线更趋于合理化，加快了我们的开发进度，减少了程序出错的可能性。

（2）面向对象

Java 语言的设计集中于对象及其接口，它提供了简单的类机制以及动态的接口模型。对象中封装了它的状态变量以及相应的方法，实现了模块化和信息隐藏；而类则提供了一类对象的原型，并且通过继承机制，子类可以使用父类所提供的方法，实现了代码的复用。

（3）可移植性

程序的可移植性（平台无关性）是指 Java 程序不经修改可以方便地被移植到网络上的不同机器上运行，包括在不同硬件和软件平台上运行。同时，Java 的类库中也实现了与不同平台的接口，使

这些类库可以移植。另外,Java 编译器是由 Java 语言实现的,Java 运行时系统由标准 C 实现,这使得 Java 系统本身也具有可移植性。可移植性在一定程度上决定了程序的可应用性。

可移植性分为两个层次:源代码级可移植性和二进制代码可移植性。C 和 C++ 只具有一定程度的源代码级可移植性,其源程序要想在不同平台上运行,必须重新编译,而 Java 不仅源代码级是可移植的,甚至源代码经过编译之后形成的字节码,同样也是可移植的。

(4) 安全性和稳定性

网络分布式计算环境要求软件具有良好的稳定性和安全性。为此,Java 首先摒弃了指针数据类型,这样,程序员就不能凭着指针在内存空间中任意“遨游”;其次,Java 提供了数组下标越界检查机制,从而使网络“黑客”无法构造出类似 C 和 C++ 语言所支持的那种指针;第三,Java 提供了自动内存管理机制,可以利用系统的空闲时间来执行诸如垃圾清除等操作。例如,Java 不支持指针,一切对内存的访问都必须通过对象的实例变量来实现,这样就防止了网络“黑客”使用“特洛伊木马”等欺骗手段访问对象的私有成员,大大提高了网络的安全性和稳定性,同时也避免了指针操作中容易产生的错误。

(5) 高性能

在一般情况下,可移植性、稳定性和安全性几乎都是以牺牲性能为代价的,解释型语言(如 Python/JavaScript/Perl/Shell 等语言)的执行效率一般也要低于直接执行源码的速度,而 Java 字节码的设计和多线程的支持,很好地弥补了这些性能的不足,从而得到较高的性能。

① 高效的字节码

Java 字节码格式的设计充分考虑了性能因素,其字节码的格式非常简单,能很容易地直接转换成对应于特定 CPU 的机器码。

② 多线程

多线程机制使应用程序能够并行执行,而且同步机制保证了对共享数据的正确操作。通过使用多线程,开发人员可以分别用不同的线程完成特定的行为,而不需要采用全局的事件循环机制,这样就很容易地实现网络上的实时交互行为。Java 提供了完全意义上的多线程支持。

③ 即时编译和嵌入 C 代码

Java 的运行环境还提供了另外两种可选的性能提高措施:即时编译和嵌入 C 代码。即时编译的另一个作用是,在运行时把字节码编译成机器码,这意味着代码仍然是可移植的,但在开始时会有一个编译字节码的延迟过程;而嵌入 C 代码在运行速度方面效果当然是最理想的,但会给开发人员带来一些负担,同时还会降低代码的可移植性。

④ 动态性

Java 程序的基本组成单元为类,在类库中可以自由地加入新的方法和实例变量,类库升级后不会影响用户程序的执行,使 Java 程序适应于一个不断发展变化的环境。它允许程序动态地装入运行过程中所需要的类(在 C++ 中,类的变化必将要重新编译),Java 在运行时才确定引用类,而类的编译早在编译阶段已完成,正在编译到运行间的延迟使得 Java 可以引用最新的类。Java 的动态性使用户能够真正拥有“即插即用”的软件模块功能。

⑤ 分布性

分布式包括数据分布和操作分布。数据分布是指数据可以分散在网络的不同主机上,操作分布是指把一个计算分散在不同主机上处理。Java 支持浏览器/服务器(B/S)和客户端/服务器(C/S)两种分布式计算模式。对于前者,Java 提供了一个叫作 URL 的对象,利用这个对象,可以打开并访

问指定 URL 地址上的对象, 访问方式与访问本地文件系统相同。对于后者, Java 提供了 Socket 编程机制, 以及一整套网络类库, 开发人员可以利用类库进行网络程序设计, 实现 Java 的分布式特性。

1.4 项目案例及设计

本书内容的展开, 是以图形参数计算程序(项目 1~项目 3)和闹钟工具小软件(项目 4~项目 10)的实现为主要线索的, 因此, 我们有必要先了解下这两个案例的背景和设计思路。

1.4.1 先导案例: 图形参数计算程序

(1) 背景

Java 是一个纯面向对象语言, 面向对象思想的理解和掌握是学习 Java 编程的基础, 前三个单元就是主要介绍面向对象的基本概念, 以及 Java 的基本语法。由于面向对象概念较为抽象, 不易理解, 因而, 采用控制台窗口方式, 突出算法逻辑过程, 选取大家熟知的图形参数计算为载体, 以便着力于面向对象编程方法的学习和应用。

(2) 功能描述

实现常见图形名称的显示, 以及周长和面积计算的功能。

(3) 设计思路

常用图形包括三角形、四边形、梯形等多种, 每种图形均有名称、周长和面积等参数, 周长与面积的计算公式视图形形状的不同而不同。采用面向过程的编程方法, 在设计图形参数计算程序时, 根据每种图形形状不同, 需要设计相应的计算逻辑, 过程重复, 可复用性差。

根据面向对象的编程思想, 设计一个图形类, 图形的参数作为该类的属性, 参数计算则可通过定义该类的方法来实现。对于不同形状的图形, 则可通过类的继承、多态特性来实现不同的计算逻辑, 以适应多种图形特点的周长和面积计算需要, 同时具有较好的可扩展性。

1.4.2 实战案例: 闹钟工具小软件

(1) 背景

在了解了 Java 面向对象编程方法基础上, 后续单元则着重阐述如何应用 Java 基础类库和编程机制, 来进行软件开发。由于篇幅有限, 以知识点覆盖率高、业务流程完整、项目小而精为原则, 选取了闹钟工具软件作为载体。

(2) 功能描述

这是一个可视化的工具软件, 主要功能如下。

- ① 设置闹钟(可以是一个或多个)。
- ② 选择铃声和试听铃声。
- ③ 实现主界面的图标最小化及还原。
- ④ 实现两种方式上传/下载铃声文件: 一是利用服务器上传和下载; 二是利用数据库实现上传和下载。

闹钟工具小软件运行效果如图 1-5 所示, 其中图 1-5a 为主界面, 图 1-5b 为版本说明界面。

(3) 设计思路

- ① 为了提供更好的用户体验, 采用 Java 提供的图形界面方式, 进行用户界面设计, 包括背