



Java

模拟实体与 网络应用 编程

■ John.Q.Wang 王黔平 编著

人民邮电出版社
www.pptph.com.cn



附光盘
CD-ROM



Java

Java 模拟实体与 网络应用 编程

■ John.Q.Wang 王黔平 编著

人民邮电出版社

图书在版编目 (CIP) 数据

Java模拟实体与网络应用编程 / (美) 王黔江 (John.Q.Wang), 王黔平编著.

—北京: 人民邮电出版社, 2001.11

ISBN 7-115-09812-3

I. J... II. ①王... ②王... III. JAVA语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第073963号

内 容 提 要

本书详细地介绍了使用Java语言进行窗口程序和网络程序设计的基本方法和技巧。全书共分为16章, 内容包括OOP基本概念、模拟实体编程基础、Java语言基础、如何制作软件开发说明、意外事件的处理、Java实用软件包、数据的传输与存储、Jar命令的使用、窗口应用软件基础、数据库的应用、在窗口应用软件中加注使用说明、串行口硬件设备控制、Thread使用、Servlet的使用、Java2D图形制作以及常用软件包简介等内容, 其中包括使用Sun公司推出的JHelp软件包制作应用软件辅助说明的完整过程。通过阅读本书以及书中提供的CD光盘, 除了计算机硬件外, 不需要任何其他工具, 就可以使用Java语言进行窗口程序和网络应用软件的编程了。

本书的特点是内容广而不深, 语言通俗易懂。书中提供了大量实例供读者上机练习和编程参考。本书以提出问题、解决问题的方式贯穿始终, 适合于有一定编程基础的读者学习, 可作为理工院校学生的选修教材, 也可供网络编程人员作为参考手册。

Java 模拟实体与网络应用编程

- ◆ 编 著 John.Q.Wang 王黔平
责任编辑 马 嘉
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@pptph.com.cn
网址 <http://www.pptph.com.cn>
读者热线: 010-67129212 010-67129211(传真)
北京汉魂图文设计有限公司制作
北京鸿佳印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 26.5
字数: 641千字 2001年11月第1版
印数: 1-5000册 2001年11月北京第1次印刷

ISBN 7-115-09812-3/TP·2561

定价: 44.00元(附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010)67129223

前言

Java是目前发展迅速的一门计算机语言。它继承了C++的很多优点，摒弃了C++的大多数缺点，对于有C++语言基础的人来说很容易学习。众所周知，21世纪是网络的世界，Java正是为网络编程而设计的，使用它可以很方便地编写网上应用软件，实现目前流行的网上交易、网上出版、网上信息交流，甚至可以实现在网上对设备进行遥控等功能。

本书的特点

无论哪一种语言，都有其自身的特点。根据笔者多年编写应用软件的经验，学习Java语言的读者完全可以从简单的操作入手，如学习怎样将数据写入文件，怎样在显示器上显示图形，怎样连接数据库等单一功能。把这些单一功能的程序块联系在一起，就能够构成有实际用途的应用软件。熟练掌握各种程序块的编写方法以后，就会对Java语言有越来越深入的了解，从而熟练地掌握Java程序设计方法。

本书的特点之一是内容广而不深。书中提供了大量实例，包括普通软件、窗口式应用软件、网上应用软件、数据库应用、商业应用软件、工程计算应用、网络连接应用、绘图软件和硬设备控制软件等。本书没有系统地讲述Java语言，希望系统学习Java的读者请参考其他书籍。本书从最简单的示例讲起，逐步深入讲述可完成较复杂完整的功能的示例。希望读者能通过学习本书在较短的时间内掌握Java语言。此外，读者可将本书作为手头参考书，随时查阅某一功能的编写方法。

使用通俗易懂的语言来讲述Java语言是本书的另一特点。笔者在美国学习和工作期间，对于使用什么样的语言讲述现代科学有着十分深刻的亲身体会。中国从古老的文言文到现代的白话文，已经取得了可喜的进步，但在中国仍然有书面用语和口头用语的严格区分。记得作者上大学时接触到物理学中“干涉”一词，当时花了很多时间去弄懂干涉的含义。有一次无意中看到3个美国孩子在玩玻璃球的游戏，一个孩子不小心碰到了球，另外的孩子马上大叫“Interference! Interference! Doesn't count!”如果译成中文就很怪了，似乎孩子们都在说：“干涉！干涉！不算！”。中国的孩子们绝不会在这种情形下使用“干涉”这个词（是否会说“搅和”或“捣乱”，这还与不同地方的方言和习惯有关）。而英文在最深奥的物理学中讲述光的干涉现象时，使用的竟是孩子们在玩游戏时常用的语言！由此例可看出，对于讲英文的孩子们来说，很容易理解“干涉”一词的含义及其演变的过程，讲中文的孩子则要从深奥的物理学中了解干涉现象，进而掌握“干涉”一词的应用。因此，书面用语的进一步通俗化势在必行。

本书的结构

全书共分16章。

第1章讲述了OOP基本概念。与通常“面向对象”的译法不同，首次将OOP译成模拟实体编程法，使初学者更容易理解OOP的本质。并将实体这个概念及模拟实体编程的思想贯穿全

书的始终，形成一个初步完整的OOP的中文翻译体系。

第2章结合示意图来讲解有形世界的物，将其与软件世界抽象的实体类联系起来。明确了模拟实体编程的4大要素，即抽象性、继承性、封闭性和多样性。完善了模拟实体编程的中文译法。

第3章从零开始详细地介绍了如何下载安装Java开发软件包以及如何使用PFE编写、运行Java源程序。概括地介绍了Java的语言结构。

第4章主要介绍如何使用javadoc.exe来制作软件开发说明以及在源程序中加注文字说明。

第5章介绍什么是意外事件，如何处理意外事件，怎样编写处理意外事件的程序块。

第6章着重介绍了如何使用常用的Vector、Hashtable、Properties、Enumeration、Calendar、StringTokenizer等实体类，特别介绍了使用TreeSet对实体个体排序的具体方法。

第7章详细介绍了文件的存储方法，着重举例说明网上实体的传输。

第8章介绍了Jar文件的制作方法。

第9章详细介绍了使用Swing图形开发软件包制作窗口应用软件的具体方法。这样的窗口应用软件可在各种操作系统（如Windows NT、Windows 2000、UNIX、Linux等）下运行。

第10章详细介绍了实体与数据库连接的具体方法。

第11章详细介绍了使用JHelp软件包制作软件使用说明书的方法，简单介绍了XML的应用。

第12章介绍了Java程序调用其他程序（如C语言编写的驻机程序）的基本方法。

第13章介绍了使用Java语言编写、运行多功能（multi-thread）程序的方法。

由于网络技术的广泛应用，本书在第14章加重了分量，详细介绍了利用Servlet小服务器制作网络应用的每一个步骤，包括前台显示、商业问题处理以及数据库管理相互之间的关系，并附有实例。可以这样说，如果读者掌握了第14章，制作网上动态的应用软件就不是太难的事了。

第15章详细介绍了如何应用Java语言进行图形处理的具体办法。

第16章简单介绍了JavaBean的概念、使用打印机输出文字及图形的方法以及Java服务网页JSP语言的编写过程。

所有书中的示例都将存放在随书附带的光盘中，其中还包括Java编译软件。可以说有了这本书，除了计算机硬件外，不需要任何其他工具，就可以进行Java语言学习了。作者认为，上机运行Java程序是学习Java语言的一条捷径。

杨晓卓博士、郭治教授对本书的编写提出一些很好的意见和建议，为本书的出版做出了较大的贡献，在这里我们深表谢意！同时，对所有关心和帮助我们的女士们和先生们表示衷心的感谢。

为了方便读者，附上一个读者信箱，地址是：john-wang@houston.rr.com，请读者与我们联系。

由于作者水平有限，书中难免存在疏漏之处，敬请广大读者批评指正。

作者

本书术语诠释

本书力图使读者从字面本身去理解文字的实际含义，使词义更贴切且通俗易懂，对许多书中关于“OOP”、class和object等译法进行了修改和充实。

本书中最大的不同是将OOP译为“模拟实体编程方法”（在许多书中，OOP被译成“面向对象的编程语言”）。由此引出了对其他专业词汇的不同译法。比如将class译成了“实体类型”，将object译为“实体”或“个体”或“实体类个体”（在许多书中，“object”被译为“对象”），将instance译为“实体类个体”，而将package译为“实体类型集合”。所有这些译法，其本身自成体系，而且贯穿一个主题，那就是“实体”。既然我们讨论的是模拟实体编程方法，所用的词汇都围绕着“实体”这个主题，可谓是丝丝入扣。而且换了一个与众不同的说法，也体现了本书出版的意义。

那么将OOP译为“模拟实体编程方法”是否合适呢？让我们先把这3个英文单词拆开，然后再组合起来，看看哪种译法对讲中文的读者更容易理解。

- object一词，在王运主编的《新编英汉科技大辞典》中是这样注释的：① 物（体），实物，事物；② 对象；③ 目标；④ 项目，科目。可见将object译为“实体”乃是首选。
- oriented一词的注释为：① 定向的，有向的；② 面向...的，适于的。可见将oriented译为“面向”就是字面直译。
- programing一词的注释为：① 规[计]划（的），大纲（的）；② 程序设计[编制，控制]，程序；③ 广播节目。

让我们再来看一下简单组合的结果：

一般将“OOP”译为“面向对象的编程”，但对象指的是什么？编程序为什么要面向某些对象呢？单从字面看很费解。需要更进一步的解释。

如果所译的新词，能让人们顾名思义，直观地了解其英文原文的内涵，这样的译法是最可取的。比如我们将“OOP”译为“模拟实体编程方法”，人们很容易从字面本身得知，这是一种模拟实际事物的编程方法，而这正是OOP的真实内涵，无需更多的解释。

有争议的是，在字典中oriented一词，根本没有“模拟”的词意，那么“模拟”一词是从哪里来的呢？其实，仔细揣摩，oriented一词有“朝...方向的”、“向...靠进的”、“与...接近的”等意思，如果我们将其译为“与实物接近的编程方法”，它岂不就是“模拟实体编程方法”吗？

综上所述，OOP的译法是本书的根。有了对OOP译法的认同，其他词汇的译法就是水到渠成、顺理成章的事了。接下来谈谈其他几个相关词汇的翻译及说明。

- class：实体类型，简称为“类”（在其他有关计算机书中，都把class译成“类”）。

在模拟实体编程中，我们将具有相同属性的事物根据其共同特征抽象成“类”，而用这个“类”来制作实体个体，最终目的是用这些个体来完成我们要计算机完成的工作。我们认为

“类”是“实体类型”的简化说法。比如说，Person类的实体具有“人”的特征，Account类的实体具有“账目”的特征等。但当人们还没有建立实体类型的概念时，在书中使用较为完整的说法比较合适。

我们将class译为实体类型的另外一个原因，是类又可以细分为3种，分别是：❶ 真实实体类；❷ 抽象实体类；❸ 功能标识类。它们有很多相同之处，也有各自独特的功能。

➤ object, instance: 实体、个体、实体类个体、具体的实体。

具体的个体和描述个体共同特征的“类”有本质的不同。具体的个体用来完成具体的任务，“类”只是用来制作个体，其本身不能完成工作，除非该“类”中的功能块是属于“类”的功能块。有关具体的说明，请参考本书的有关章节。每个个体都有与其他个体不同的描述，而使其区别于其他个体。在计算机中，不同个体保存在不同的内存区域内。如张三属于Person类，他的“name”是“张三”，而李四也属于Person类，他的“name”是“李四”。如果我要求张三告知他的电话号码，这是没有问题的。但如果我要“人”这个“类”告诉我电话号码，就无法办到了。因为张三的电话号码与李四的不同，向Person这个“类”要号码实在是无的放矢，没有具体的对象。注意到在这里用了“对象”一词，这大概就是“实体个体”和“对象”的内在联系吧。上面讲的例子是自然界的，还有很多事物如银行，是人们创造的，“中国人民银行”属于“Bank”类，是“Bank”类的具体个体，“农业银行”也属于“Bank”类，却是完全不同的另一个“Bank”个体。如果我们在这里将“object”译为“目标”实属牵强，翻成“物体”也不恰当。我们列出几个简单的句子加以比较：

“中国人民银行”是属于“银行”类的“目标”；

“中国人民银行”是属于“银行”类的“物体”；

“中国人民银行”是属于“银行”类的“对象”；

“中国人民银行”是属于“银行”类的“具体个体”。

由上述4个例子的比较，也很容易看出把object译为“个体”是较好的翻法。这说明，好的翻译要为进一步的应用铺平道路。

还要说明的是，使用object一词不仅给讲中文的人带来困惑，即使是讲英文的人也会产生类似的误解。这就是为什么在很多英文书中使用instance，而不使用object的原因。

instance一词，在王运主编的《新编英汉科技大辞典》中是这样注释的：❶ 样品，实例；❷ 情况，场合；❸ 请[要]求，提议；❹ 举...例，引证，用例子说明。

其中也没有“个体”一说，但“个体”与“样品”和“实例”的意思已经非常接近了。样品是做实验的，实例是做说明的，都有“具体的”这一含意，但似乎都不是用来完成具体任务的，与模拟实体编程的内涵不符。不过从object和instance这两个词的组合意义“具体的事物”来看，“实体个体”的译法是贴切的。

我们认为“实体类”和“实体类个体”的译法准确地表述了“模拟实体编程方法”中这两个相关词语的内涵，是不可多得的好译法。把它们简化为“类”和“个体”也是可取的。

➤ package: 实体类型集合，简称“类集”或“软件包”（其他有关计算机书中，都把package译成软件包）。

与“软件包”相比，“实体类型集合”更为准确，简化为“类集”也是可取的。这不仅由于它使得我们所有词汇的译法看起来更为统一，而且实际意义更为准确。“软件包”包含某种完整的意思，是指能够独立完成许多工作的计算机应用程序的组合。而在“模拟实体编程方

法”中的package，是指在某种程度上有相关联系的实体类的集合。多数情况下，它是不完整的，不能构成一个完整的计算机应用软件。可以说，软件包包括一个或多个应用软件，而应用软件又包含多个package。从某种意义上讲，软件包与实体类集合的差别并非十分严格，因此我们有时也把package译为软件包。

以上是我们与其他书在一些名词术语上的不同译法的说明和补充，在此加以强调是为引起读者的重视。还有一些我们会在书中陆续说明。另外，所有计算机术语与英文的对照将列在附录1中供读者查阅。

作者简介

John.Q.Wang

1982年毕业于山西大学物理系。毕业后留校任教。在校期间，系统地学习了Z80汇编语言，并应用它编写X光衍射仪的自动控制和数据采集的计算机应用软件，从而奠定了个人的计算机软硬件的坚实基础。1986年赴美国攻读博士，获物理博士学位。在美国期间，系统地学习了Fortran、C++等计算机语言，并使用这些语言完成了大量的科学计算，以及X光四圆衍射仪的自动控制，掌握了在美国仪器设备上进行计算机自动控制的大量经验。而后又在NASA美国国家宇航中心担任软件开发和系统分析员。在此期间，积累了丰富的UNIX操作系统、X Window、X Mofit以及G2的编程经验。随后又在一家网络应用软件公司Net Explorer工作，制作了很多使用Java、HTML和XML编写的商业应用软件。目前在美国一家较大的软件公司BMC Software就职，从事网络主机服务程序运行状态的监控程序的制作。

王黔平

1982年毕业于河北矿冶学院（现更名为河北理工学院）化工系。毕业后留校任教。1986~1987年，在武汉工业大学进修研究生主要课程。在从事教学和科研工作的近20年间，进行了大量无机非金属材料的研究，并在《中国生物医学工程学报》、《中国陶瓷》、《硅酸盐通报》、《陶瓷》、《情报杂志》、《中国建材》等全国性科技期刊及国家核心期刊上发表了30余篇论文。1998~2000年，作为访问学者到美国学习，主要学习Java语言和计算机操作系统在工业控制等方面的研究。2001年，被河北省教委聘为河北理工学院教授。

目 录

第1章 OOP 基本概念	1
1.1 什么是OOP和OOD	1
1.2 为什么要使用OOP	3
1.3 什么是class	3
1.4 什么是method	4
1.5 什么是interface	5
1.6 什么是abstract class	6
1.7 实用实体类、抽象实体类及功能标识类概述	7
1.8 什么是classpath	8
1.9 什么是package	8
第2章 模拟实体编程基础	11
2.1 如何用图形来表示实体类型及相互关系	11
2.2 什么是数值传递和地址传递	17
2.3 什么是功能块的同名异形和同形异能	19
2.4 什么是向上强制改型和向下强制改型	21
2.5 功能标识类与功能填充	23
第3章 Java 语言基础	27
3.1 Java 起步	27
3.2 Java语言中的专用词	29
3.3 Java语言中常用的运算操作算符	31
3.4 使用PFE编写源程序	32
3.5 什么是Java源程序的基本结构	34
3.6 何谓constructor	35
3.7 什么是功能块的标识	35
3.8 功能块main()有什么主要用途	35
3.9 数组 (Array) 的使用	37
3.9.1 数组的确定	38
3.9.2 数组变量的产生	39
3.9.3 数组变量的初始化	41
3.10 数据的类型转换	41
3.11 什么是静态变量? 如何使用静态功能块	42

第4章 如何制作软件开发说明	45
4.1 如何安装Java语言使用说明	45
4.2 如何使用Java语言使用说明	45
4.3 如何应用javadoc.exe来制作软件开发说明	46
4.4 如何在源程序中加注文字说明	52
4.4.1 变量、功能块、实体类型的说明	52
4.4.2 特殊的标签	53
4.4.3 整体说明	55
4.4.4 标签的使用位置	55
4.4.5 借助文件简化javadoc命令	56
4.5 如何使用Java的自学工具	56
第5章 意外事件的处理	59
5.1 什么是意外事件	59
5.2 如何处理意外事件	60
5.3 如何编写自己的意外事件处理程序块	62
第6章 Java 实用软件包	65
6.1 如何使用Vector	65
6.2 如何使用Hashtable	66
6.3 如何使用Properties	69
6.4 如何使用Enumeration	73
6.5 如何使用Calendar	74
6.6 如何使用StringTokenizer	75
6.7 排序-TreeSet 的使用	76
第7章 数据的传输与存储	81
7.1 数据流	81
7.2 使用数据流存取一个字节	82
7.3 数据流的族谱	83
7.4 怎样读入文字文件	84
7.5 怎样将数据写入计算机文件	85
7.6 网上数据流传输实例	87
7.7 文件的随机存取	90
7.8 文件中直接存取实体	92
7.9 网上实体传输	95
7.10 File实体类的应用举例	102
第8章 Jar 命令的使用	105
8.1 什么是Jar文件	105
8.2 如何编辑Jar文件	105

8.3 如何修改Jar文件中的manifest.mf文件	107
第9章 窗口应用软件基础	111
9.1 如何制作一个简单的HTML网页	111
9.1.1 HTML的基本结构	111
9.1.2 图像的显示	111
9.1.3 链接	112
9.1.4 一些常见的表示	112
9.1.5 简单表格	113
9.1.6 举例说明HTML的用法	114
9.1.7 什么是Applet	116
9.2 如何编写和运行简单的Applet	116
9.3 什么是本地机窗口应用软件	118
9.4 如何编写本地机窗口应用软件	118
9.5 如何使用自动排版管理	123
9.5.1 BorderLayout	123
9.5.2 FlowLayout	126
9.5.3 GridLayout	127
9.5.4 GridBagLayout	128
9.6 如何使用内部实体类型定义	130
9.6.1 内部实体类型 (inner class)	130
9.6.2 匿名内部实体类型 (anonymous inner class)	132
9.7 如何使用JfileChooser	134
9.8 如何使用JColorChooser	136
9.9 如何使用对话框窗口	138
9.10 如何使用JTable实体类型	146
9.11 如何写一个简单的工业应用软件	148
第10章 数据库的应用	173
10.1 Java如何与数据库连接	173
10.2 实用有效的数据库连接方法	175
10.3 如何由数据库建立实体的双向连接	187
10.4 双向连接中应注意的问题	198
第11章 在窗口应用软件中加注使用说明	205
11.1 如何下载和安装制作窗口应用程序使用说明软件包	205
11.2 什么是XML	205
11.3 help软件包中的制作实例	207
11.4 按分类目录制作的help	208
11.5 按给定关键字制作的help	215

11.6 全部文件通查方式	216
11.7 小结	217
第12章 串行口硬件设备控制	219
12.1 如何安装Java串行口信息传递软件包	220
12.2 如何将ASCII 码的命令送出串行口	220
12.3 如何将ASCII 码的数据由串行口读出	222
12.4 串行口命令传输中的注意事项	224
12.5 JNI 简介	224
12.5.1 Java 程序调用本地机C程序	225
12.5.2 在Java和C/C++程序之间传递字符串变量	229
12.5.3 原始类数组的传递	235
12.5.4 实体类个体的传递	238
第13章 Thread使用	245
13.1 什么是Thread	245
13.2 一个简单的Thread程序	245
13.3 Thread程序的基本结构	247
13.4 为什么要使用synchronize	248
13.5 Thread中功能块的正确应用	249
13.6 什么是Deadlock	250
第14章 Servlet 的使用	251
14.1 什么是Servlet	251
14.2 Jetty 引擎的安装与运行	251
14.3 Jetty 文件结构和编程习惯	254
14.4 如何编译HelloWorldServlet.java 这个最简单的Servlet程序	255
14.5 什么是PathMap	257
14.6 一个简单实用的先尝后买的Java综合开发软件-KAWA	258
14.7 按Jetty的方式制作HelloWorld服务软件	261
14.8 对HelloWorld服务软件的具体分析	266
14.9 小服务器的基本结构和编写方法	267
14.10 如何使用Template解决网页输出问题	269
14.11 Jetty的其他管理简介	275
14.12 一个检查用户使用资格的通用程序段	279
14.13 一个较为完整的网上销售软件的设计制作	279
14.14 Java服务网页JSP简介	292
14.15 Jetty新版本的系统设置及其运行	298
14.16 网上文件的上传与下载	304

第15章 Java2D 二维图形软件包	311
15.1 图片 (image) 的显示及处理	312
15.1.1 图片显示	312
15.1.2 图像处理	314
15.2 文字的显示	317
15.3 二维图形的绘制	322
15.3.1 用笔方式	323
15.3.2 填充方式	325
15.3.3 重叠组合方式	327
15.3.4 区域间的结合	329
15.3.5 图形的截取显示	331
15.3.6 图形 (Shape)	335
15.3.7 图形变换	341
15.4 数据曲线图	352
15.4.1 显示曲线的简单程序	352
15.4.2 圆形百分图	354
15.4.3 直方图	356
15.4.4 坐标的绘制	358
15.5 模拟仪表	363
第16章 常用软件包简介	369
16.1 将文字和图形送往打印机	369
16.1.1 文字打印	369
16.1.2 图形打印	372
16.2 JavaBean简介	374
16.2.1 JavaBean开发系统的安装	375
16.2.2 绘制分子结构的Bean	377
16.2.3 简单的JavaBean的制作	380
16.3 JavaMail简介	386
16.4 TCP/IP的使用	396
16.4.1 什么是Socket	396
16.4.2 如何编写提出服务申请的用户一方 (client) 的程序	396
16.4.3 如何编写提供服务的服务 (server) 一方的程序	397
附录1 本书中英文对照表	399
附录2 光盘说明	400
附录3 本书附带软件的版权声明	403
附录4 软件版权协议原文拷贝件	404

第 1 章 OOP 基本概念

学习任何一种新的语言首先都需要建立一些基本概念。在这一章里要介绍一些OOP编程的概念。如果读者急于动手上机实习，可以先跳过这一章，等到需要的时候再返回来学习这些基本概念。

1.1 什么是OOP和OOD

OOP英文为Object Oriented Programming，在本书中将其译为模拟实体编程法。OOD英文为Object Oriented Design，在本书中译为模拟实体设计法。所谓实体就是任何客观存在的物体或人们想象中的个体，如人、书、用户、银行账户、产品、经理、雇员以及商店等等。简单地说，模拟实体编程方法就是用一段固定格式的程序块模拟实体的全部应用特征，而模拟实体设计就是用上述方法设计计算机应用软件。

模拟实体编程，首先是将生活中各种各样的事物分类，抽象出它们共同的特征和行为，然后用计算机变量及功能程序模块模拟这些特征和行为，给出这类事物的类型定义。接着用这些类型定义在应用程序中制造模拟实体，由这些模拟的实体最终完成具体的工作。这很像由实践到理论，再由理论指导实践这样一个完整的过程。模拟实体编程示意图如图1-1所示。

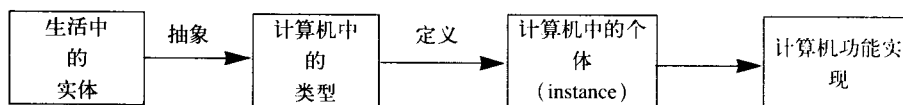


图1-1 模拟实体编程过程示意图

一旦确定了计算机中的实体类型，就可以用这个类型创建出众多的、具有相同特征的模拟实体，并用这些实体来完成工作。在英文中，对这样单一的实体有一个专门的名词，称为该实体类型的一个instance。在本书中，我们把这个词译为“个体”。比如学生（Student）是一个生活中的实体，可以用学生号码、姓名、课程和平均分等变量来描述所有学生的共同特性，而用“计算平均成绩”来描述学生的行为。它们共同给出了学生这个实体类型的定义。如图1-2所示。

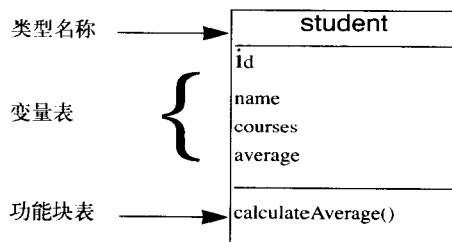


图1-2 用图形表示实体类

有了Student类型，在实际编程过程中就可以用它制作出一系列的学生实体类个体。如：

```
Student zhao = new Student("Zhao");
Student qian = new Student("Qian");
Student sun = new Student("Sun");
Student li = new Student("Li");
```

其中zhao、qian、sun、li都是Student类的个体。如果我们要计算学生zhao的平均成绩，就调用

```
zhao.calculateAverage();
```

同样，要计算学生li的平均成绩，就调用

```
li.calculateAverage();
```

需要注意的是，这个抽象定义的学生实体类型，对真实世界的实体是一个极为不完整的描述。比如，它根本不能回答：今天学生li穿的是什么衣服，学生qian吃过早饭了吗？等一系列的问题。其实，在计算机模拟实体中，根本不可能，也完全没有必要去完整地描述真实世界中的任何实体。对于给定的计算机应用软件，只要部分地、抽象地描述真实实体的某些特征和行为就足够了。从这里也可以看出，对于同一个真实世界中的实体，在不同的应用软件中，所给出的类型定义可能是不同的，因为它们侧重点不同，抽象的部分也不同。

模拟实体编程有4大特性，即抽象性、继承性、封闭性和多样性。

模拟实体编程的抽象性是指先将实体所具有的共同特性，如状态、职能、实体之间的相互关系抽象为程序中的变量和程序块，然后再以此分别定义该实体的状态、职能的这样一种性质。

继承性是指一个实体可以从另外一个实体继承其全部特征，再加上自身的特征构成新的实体的这样一种性质。

封闭性是指描述实体的程序块中的变量或功能块，不能随意让其他实体使用和修改的这样一种性质。

多样性简单地说是由一个实体发出一个问题，由不同的实体给出不同的答案。发出问题的实体为父辈实体，而回答问题的为子辈实体。子辈实体是由父辈实体继承发展而来的。这样解释多样性实在是过于简单，而多样性的完成也是模拟实体编程中较为难懂的部分。可以这样说，如果弄懂了多样性，就算弄懂了模拟实体编程方法。

为了使读者能对模拟实体编程方法有一个感性的认识，下面用一个简单的例子来加以说明。

人，是一个实体，可以用姓名、年龄、职务、身份证号码、家庭住址等变量来描述人的状态，而用“设定姓名”、“提供姓名”、“设定身份证号码”、“提供身份证号码”等功能来确立“人”这类实体所能完成的工作。这就是抽象性。同样一个实体，所抽象出来的变量和功能不一定相同，如何确定实体的状态变量和职能是根据软件的实际应用而决定的。

经理，是一个实体，它具有所有人的特征，因此可以把“人”作为父辈实体，把“经理”作为子辈实体继承“人”的特征。经理有其自身的状态和职能，如“开除雇员”、“招收雇员”等。

雇员，是一个实体，它具有“人”的全部特征，我们把“雇员”作为子辈继承“人”的

特征。雇员不同于经理，有其自身的特征，如具有“销售”、“汇报工作”等职能。这就是继承性。由“人”这个实体发出“你的职务是什么？”的问题，“经理”和“雇员”两个实体将给出不同的答案，这就是多样性。我们会在今后的实际编程例子里，学习Java是如何完成这些编程过程的。

Java是全方位的模拟实体编程语言。

1.2 为什么要使用OOP

旧的编程方法为顺序编程。所编的程序由头至尾，按顺序执行。对于小的应用程序，这种方法并没有太大的弊病。一旦应用程序变大，就会带来很大的麻烦。如牵一发而动全身，一个小小的改动，将使整个程序受到影响。而模拟实体编程方法将大的应用程序分成若干个小块实体，而且尽量减少实体之间的程序联系（coupling）。这样一来，一个实体的变化、修改，对其他实体影响不大，使程序的应变能力大大加强，易于纠错、维护、便于修改。更由于使用固定格式的程序块来模拟实体，使程序读起来更接近实际，程序的可读性大大提高。人们在大量的编程实践中，积累了很多经验，已经为模拟实体编程制订了很多规范，对于不同的应用软件，有很多现成的格式。只要遵循这些格式制作软件，就可以避免这样或那样的错误。模拟实体编程有很多技巧，很多问题当没有用到时不会感觉到，所以只有不断地吸取成功经验，才会收到事半功倍的效果。

1.3 什么是class

class在本书中译为实体类型。Java正是使用了class使得OOP成为可能。Java中用class来定义具有共同特性的实体，它由3部分组成：一是实体类型的名字，二是描述该类型实体的变量，三是对这些变量进行处理的程序功能块。如人可定义为一个实体类型People，可以把姓名（name）、身份证号码（id）、家庭住址（address）作为它的变量，而把给出姓名（getName）、给出身份证号码（getID）、给出家庭住址（getAddress）作为功能块。程序清单1-1列出了源程序。

程序清单1-1 People.java(OOP示意举例)

```
1 public class People
2 {
3     String _name;
4     String _id;
5     String _address;
6
7     public People(String name, String id, String address)
8     {
9         _name = name;
10        _id = id;
11        _address = address;
12    }
13
14    public String getName()
15    {
```