



普通高等院校电子信息类“十一五”规划教材

Java

面向对象程序设计

主编 赵付青 高峰
主审 邹建华



随书附赠
一张光盘



国防工业出版社

National Defense Industry Press

普通高等院校电子信息类“十一五”规划教材

Java 面向对象程序设计

主 编 赵付青 高峰

主 审 邹建华

副主编 郭玉坤 张聚礼 高玮军 马维俊

国防工业出版社

· 北京 ·

内 容 简 介

本书作为目前高校普遍使用的面向对象程序设计(Java)的教材,通过大量经典的程序实例,使学生在掌握 Java 知识的同时提高动手编程能力。全书共分 13 章,内容包括面向对象程序设计基础、Java 语言基础知识、类和对象、继承与派生、异常、多线程、图形用户界面设计、输入/输出流、applet、数据库编程、网络编程、J2EE 技术及常用工具类和包等。本书在介绍基本概念的同时,列举了大量典型而有意义的程序实例。读者通过阅读这些实例并结合上机操作,能较快地掌握 Java 语言的程序设计,并了解面向对象的设计方法。

本书是基于 Java2 的基础教程,以 SUN 公司的 JDK1.6 技术为背景,融合了面向对象的编程思想以及网络通信等相应的体系结构,是 Java 的普及性教材。

本书可作为高等院校计算机专业或其他专业程序设计的教材或教学参考书,也可作为计算机应用科技人员的自学或培训教材。

图书在版编目(CIP)数据

Java 面向对象程序设计/赵付青,高峰主编. —北京:国防工业出版社,2010.2

普通高等院校电子信息类“十一五”规划教材

ISBN 978-7-118-06693-7

I. ①J... II. ①赵...②高... 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(2010)第 016353 号

※

国防工业出版社 出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

腾飞印务有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 29½ 字数 682 千字

2010 年 2 月第 1 版第 1 次印刷 印数 1—4000 册 定价 48.00 元(含光盘)

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

前 言

随着计算机技术和网络技术的发展,Java 语言作为面向对象的、跨平台的编程语言,自 1996 年正式发布以来,迅速成为 IT 领域里的主流编程语言。面向对象的 Java 语言具备一次编程、任何地方均可运行的能力,使其成为软件服务提供商和系统集成商用以支持多种操作系统和硬件平台的首选解决方案。Java 作为软件开发的一种革命性的技术,其地位已被确定。如今,Java 技术已被列为当今世界信息技术的主流之一。

很多初学者都会提出一个共同的问题,即怎样才能学好 Java? 其实,没有什么捷径可走,唯有自己动手多编写程序。只有多编写程序和在上计算机上调试程序,才能把 Java 的语法规则、编程的基本方法和技巧、各种算法变成实实在在的知识。

目前市场上有关 Java 面向对象程序设计的书很多,但本书有其独特的方面,主要体现在以下几个方面:

(1) 突出面向对象。很多学生在学习 Java 之前,只学习过面向过程的程序设计语言——C 语言,如何顺利从面向过程过渡到面向对象是初学者学习时非常困难的问题。作者在编写本书时力图把面向对象和面向过程结合起来,在基本工具和算法部分引入面向过程的一些实例,从类和对象部分开始直接过渡到面向对象的思想。同时本书在第 1 章就引入了面向对象编程技术的概念和基本过程,使学生在开始学习 Java 之前就接触面向对象编程的概念,顺利实现从面向过程向面向对象的转变。

(2) 基本概念及知识点讲解精炼。本书用较少的篇幅讲解面向对象的概念,在基本知识点的讲解过程中尽量使用浅显易懂的语言加以描述,辅以大量具体的实例来说明基本知识点,可方便读者边看边练,加强基本知识点的理解。

(3) 指导性强。根据作者多年的教学经验,学生在课程上机时不知道干什么,或者遇到问题时无法解决。针对以上问题,我们专门编写了 Java 的上机工具的操作知识,可指导读者在很短的时间里,在掌握 Java 知识的同时提高上机操作能力。本书在每章后都配有练习题及上机实验题,以锻炼学生的能力,加深对知识的理解和掌握。

本书内容丰富,深入浅出,注重理论与实践的结合,突出对学生知识运用能力的培养。学生通过本书内容的学习和上机实验,可以较好地掌握面向对象设计技术、Java 核心内容及 Eclipse 编程的基础知识。

本书编委会成员由西安交通大学电信学院邹建华教授、博导,兰州理工大学赵付青、张聚礼、高玮军、马维俊,河西学院高峰、郭玉坤等老师组成。

全书由赵付青主编,第 1 章~第 3 章由赵付青撰写,第 4 章由高玮军撰写,第 5 章和第 6 章由高峰撰写,第 7 章和第 8 章由张聚礼撰写,第 9 章及第 13 章由郭玉坤撰写,第 10 章~第 12 章由马维俊撰写。本书由西安交通大学电信学院博士生导师邹建华教授主审。

兰州理工大学计算机与通信学院 09 级研究生唐建新、张自如、王吉哲和赵玲在书稿的文字录入、排版及程序的验证方面做了大量工作。本书在编写过程中得到了兰州理工大学教务处及计算机与通信学院领导和同事的大力支持和帮助。本书还得到了国防工业出版社丁福志老师、西安交通大学电信学院的领导和老师、河西学院计算机学院领导和老师的鼎力支持和帮助,在此一并表示衷心的感谢。

由于时间仓促,加之作者水平有限,不当之处在所难免,敬请读者批评指正。

编者

2009 年 10 月

目 录

第 1 章 面向对象软件开发概述	1
1.1 对象、类与实体,对象的属性与相互关系	1
1.1.1 对象	2
1.1.2 类	2
1.1.3 对象的属性与相互关系	3
1.2 面向对象的软件开发过程	5
1.2.1 有关的术语	5
1.2.2 面向对象的分析	6
1.2.3 面向对象的设计	7
1.2.4 面向对象的实现	7
1.3 面向对象的开发方法	7
1.4 面向对象程序设计方法的特点	8
1.5 面向对象的编程技术(OOP)	10
1.5.1 OOP 的概念和术语	11
1.5.2 OOP 特性	12
1.5.3 OOP 中的抽象	15
1.6 小结	15
习题	16
第 2 章 Java 简介	17
2.1 Java 的诞生	17
2.1.1 需求催生了语言	17
2.1.2 Java 的发展史	18
2.1.3 Java 的现状	19
2.1.4 Java 的深远影响	19
2.2 Java 语言的应用前景	20
2.3 Java 语言的特点	20
2.4 Java 语言的特色	24
2.5 与其他语言的比较	24
2.6 Java 编程环境	28
2.6.1 JDK 软件包的下载	28
2.6.2 JDK 软件包的安装	30

2.6.3	JDK 环境变量的配置	31
2.7	Java 的运行原理	33
2.8	基本工具	34
2.9	第一个 Java 程序	34
2.9.1	Java Application 程序	35
2.9.2	Java Applet 程序	36
2.9.3	Applet 的运行	37
2.10	JDK1.6 的新特性	38
2.11	开发工具软件	41
2.12	Eclipse 集成开发环境	42
2.12.1	Eclipse 简介	42
2.12.2	Eclipse 的安装与启动	43
2.12.3	Eclipse 编写 Java 的流程	45
2.13	小结	49
	习题	49
第 3 章	Java 基本程序设计	50
3.1	基本知识	50
3.1.1	字符集	50
3.1.2	标识符	50
3.1.3	关键字	51
3.1.4	注释	51
3.2	基本数据类型	51
3.3	基本数据类型转换	57
3.3.1	优先关系	57
3.3.2	自动类型转换	57
3.3.3	强制类型转换	58
3.4	Java 类库中对简单类型数据的类包装	59
3.5	表达式	60
3.5.1	赋值运算符	61
3.5.2	算术运算符	61
3.5.3	关系运算符	63
3.5.4	布尔逻辑运算符	63
3.5.5	位运算符	64
3.5.6	条件运算符?:	64
3.5.7	其他运算符	65
3.5.8	运算符的优先级与结合性	65
3.6	语句	66
3.7	Java 的控制语句	67
3.7.1	条件语句	67

3.7.2 循环语句	71
3.7.3 转移语句	74
3.8 package 语句	76
3.9 import 语句	76
3.10 数组	78
3.10.1 声明数组	78
3.10.2 创建数组	78
3.10.3 数组元素的使用	79
3.10.4 数组的初始化	79
3.10.5 length 的使用	79
3.10.6 数组的引用	80
3.10.7 二维数组	82
3.11 字符串 String 类	88
3.11.1 声明	88
3.11.2 创建字符串	88
3.11.3 字符串常用操作	89
3.12 StringBuffer 类	92
3.13 小结	93
习题	93

第4章 Java 语言的面向对象基础

4.1 类与对象	104
4.1.1 Java 语言中类的声明	104
4.1.2 成员变量和局部变量	105
4.1.3 成员方法的声明	107
4.1.4 方法的重载	109
4.1.5 构造方法	110
4.1.6 对象	112
4.1.7 关键词 this 的用法	114
4.2 继承与子类	114
4.2.1 子类的创建	115
4.2.2 方法的重写和变量隐藏	117
4.3 抽象方法	119
4.4 抽象类	120
4.5 内部类	124
4.5.1 内部类的分类	124
4.5.2 内部类的使用	124
4.6 Java 的接口	128
4.6.1 什么是接口	128
4.6.2 接口的声明	129

4.6.3 接口的实现	130
4.7 接口的继承	133
4.8 抽象类与接口的区别	134
4.8.1 理解抽象类	134
4.8.2 抽象类和接口的语法定义	135
4.8.3 抽象类和接口的设计理念	135
4.9 Java 语言的包	138
4.9.1 包的概念	138
4.9.2 创建包	138
4.9.3 声明包	138
4.9.4 包的生成	139
4.10 包中类的使用	139
4.11 Jar 命令的使用	142
4.11.1 Jar 命令的用法	143
4.11.2 打包自己的类	143
4.12 Java 的系统包	144
4.13 修饰符	144
习题	145

第 5 章 异常处理 147

5.1 异常的概念	148
5.1.1 什么是异常?	148
5.1.2 异常处理机制的结构	152
5.2 异常类的继承关系	153
5.3 捕获异常	158
5.4 正确地使用异常	173
5.5 小结	173
习题	174

第 6 章 多线程机制 176

6.1 程序、进程与线程	176
6.2 多线程机制	177
6.2.1 什么是多线程机制	177
6.2.2 多线程实现机制	177
6.3 Thread 类	184
6.3.1 Thread 类成员变量及方法	185
6.3.2 线程状态	186
6.3.3 Thread 优先级	188
6.3.4 线程调度	190

6.3.5 线程同步	190
6.3.6 wait()、notify()和 notifyall()方法	196
6.4 线程组	198
6.5 线程通信	199
6.5.1 主存读/写通信	199
6.5.2 管道流通信	200
6.6 多线程的互斥与同步	201
6.7 小结	206
习题	207
第7章 图形用户界面技术	210
7.1 概述	210
7.1.1 AWT 体系结构	210
7.1.2 java. awt 包	211
7.1.3 Swing 体系结构	211
7.1.4 javax. swing 包	212
7.2 AWT	213
7.2.1 容器	213
7.2.2 布局管理器	214
7.2.3 AWT 事件模型	227
7.2.4 AWT 组件	234
7.3 Swing 技术	265
7.3.1 构造 Swing GUI	270
7.3.2 模型—视图—控制器	272
7.3.3 Swing GUI 实例	274
7.3.4 JComponent	277
7.3.5 Swing 组件	277
7.4 小结	277
习题	278
第8章 输入和输出	279
8.1 I/O 流基础	279
8.1.1 InputStream 和 OutputStream	280
8.1.2 Reader 和 Writer	284
8.1.3 流的结构	285
8.1.4 过滤器流	286
8.2 典型的流	288
8.2.1 字节流	288
8.2.2 字符流	289
8.2.3 缓冲流	291

8.2.4	扫描器和格式化	292
8.2.5	命令行 I/O	296
8.2.6	数据流	298
8.2.7	对象流	301
8.2.8	管道流	304
8.2.9	文件流	309
8.3	文件	310
8.4	随机存取文件	311
8.5	小结	316
	习题	316

第9章 Java applet 317

9.1	概述	317
9.1.1	applet 的工作流程	318
9.1.2	applet 的安全机制	318
9.1.3	java. applet 包	320
9.1.4	applet 的生命周期	320
9.2	编写 applet	321
9.2.1	编辑 applet 源文件	322
9.2.2	编译 applet	323
9.2.3	建立 HTML 文件	323
9.2.4	运行 applet	325
9.2.5	生命周期方法	326
9.2.6	图形显示和更新方法	327
9.2.7	事件处理	328
9.2.8	其他方法	330
9.3	绘图	330
9.4	加载数据	335
9.4.1	加载图像	337
9.4.2	加载声音	338
9.5	applet 高级功能	339
9.5.1	显示状态信息	339
9.5.2	从 HTML 获取信息	340
9.5.3	显示文档	341
9.5.4	动画	346
9.6	编写 applet 和应用程序都支持的代码	349
9.7	小结	351
	习题	351

第10章 数据库编程 353

10.1	关系型数据库的基本概念	353
------	-------------------	-----

10.2	数据库的标准查询语言	356
10.3	使用 JDBC 连接数据库	362
10.3.1	JDBC 总体结构	362
10.3.2	JDBC 驱动程序	366
10.3.3	用 JDBC 连接数据库	367
10.4	数据库编程实例	369
10.4.1	用 MySQL 建立数据库及表	369
10.4.2	用 JDBC 进行数据库操作简单实例	370
	习题	372
第 11 章	网络编程	373
11.1	网络编程基础	373
11.1.1	计算机网络的定义	373
11.1.2	网络体系结构	374
11.1.3	TCP/IP 参考模型	376
11.1.4	URI 和 URL	377
11.1.5	端口和端口号	378
11.2	基于 Socket 的低层次 Java 网络编程	381
11.2.1	网络地址 InetAddress 类	382
11.2.2	Socket 通信	383
11.3	基于 URL 的高层次 Java 网络编程	388
11.3.1	URL 类	388
11.3.2	URLConnection 类	390
11.3.3	URL 与 Socket 通信的区别	393
11.4	一个简单的 Socket 通信程序	393
11.4.1	服务器端主程序	393
11.4.2	客户端主程序	410
	习题	422
第 12 章	J2EE 概述	423
12.1	J2EE 的概念	423
12.2	为何使用 J2EE 技术规范	425
12.3	J2EE 核心技术简介	427
12.3.1	常用概念	427
12.3.2	J2EE 核心 API	428
12.4	J2EE 规范的轻量级实现	430
	习题	433
第 13 章	常用工具包和类	434
13.1	Java 常用工具包	434

13.1.1	核心 Java API 包	434
13.1.2	关于 java.lang	435
13.2	简单类型包装器	436
13.2.1	Number 类	436
13.2.2	Double 类和 Float 类	436
13.2.3	Byte、Short、Integer 和 Long	439
13.2.4	Character 类	441
13.2.5	Boolean 类	443
13.2.6	关于 Vector	443
13.2.7	Void 和 Process	444
13.3	Object 类	444
13.3.1	Object 类的方法	445
13.3.2	使用 clone() 和 Cloneable 接口	445
13.4	Class 类	448
13.5	Package 类	450
13.6	Runtime 类	451
13.6.1	内存管理	452
13.6.2	执行其他的程序	453
13.7	System 类	454
13.7.1	使用 currentTimeMillis() 方法	455
13.7.2	使用 arraycopy() 方法	455
13.7.3	环境属性	456
13.8	Math 类	457
13.8.1	超越函数	457
13.8.2	指数函数	457
13.8.3	舍入函数	457
13.8.4	其他数学方法	458
习题		459
参考文献		460

第1章 面向对象软件开发概述

传统结构化软件开发技术其本质是功能分解，是围绕实现处理功能的过程来构造系统的。结构化方法强调过程抽象化和模块化，是以过程（或操作）为中心来构造系统和设计程序的。然而用户需求的变化大部分是针对加工的，因此这种变化对基于过程的设计来说是难以想象的。传统软件开发方法存在的问题可概括为如下几点：

- (1) 软件生产率低。
- (2) 软件重用程度低。
- (3) 软件维护艰辛。
- (4) 软件非用户所需。

面向对象作为一种思想及编程语言，为软件开发的整个过程——从分析设计到实现——提供了一个完整解决方案。面向对象堪称是软件发展取得的里程碑式的伟大成就。

面向对象的软件开发和相应的面向对象的问题求解是当今计算机技术发展的重要成果和趋势之一。面向对象是一种思想，面向对象思想为软件设计与开发赋予了哲学的意义。在哲学的世界里，小至沙粒微尘，大至日月星辰乃至宇宙，均可视为单独的个体对象而存在。

面向对象的程序设计和问题求解力求符合人们日常自然的思维习惯，降低、分解问题的难度和复杂性，提高整个求解过程的可控制性、可监测性和可维护性，从而达到以较小的代价和较高的效率获得较满意效果的目的。面向对象程序设计以人类习惯的解决问题的方法进行软件开发，从而使软件开发过程和人类求解问题过程一致。

1.1 对象、类与实体，对象的属性与相互关系

面向对象的开发方法把软件系统看成各种对象的集合，对象就是最小的子系统，一组相关的对象能够组合成一个更复杂的子系统。这种开发方法有以下几点优点：

- (1) 把软件系统看成是各种对象的集合，这更接近于人类的自然思维方式。
- (2) 软件需求的变动往往是功能的变动，而功能的执行者——对象，一般不会有大的变化。这使得按对象设计出来的系统结构比较稳定。
- (3) 对象包括属性(数据)和行为(方法)，对象把数据以方法的具体实现方式一起封装起来，这使得方法和与之相关的数据不再分离，提高了每个系统的相对独立性，从而提高了软件的可维护性。
- (4) 支持封装、抽象、继承和多态，提高了软件的可重用性、可维护性和可扩展性。

我们一般的软件开发是从需求分析开始到绘制类图的结束。先从“问题领域的事物”中抽取与问题领域相关的事物的属性和行为，抽象出来为对象。再从“对象模型中的对象”中把具有相同属性和行为的对象抽象为类(class)。再从“对象模型中的类”中把具有相同属性和

行为的类中抽象出来为“对象模型中的父类”(abstract class)。

从上面的分析可以看出，在面向对象的开发过程中主要运用了抽象思维。

1.1.1 对象

在面向对象的系统中，对象是基本的运行时的实体，它包括数据属性，也包括作用于数据的数据的操作(行为)。所以一个对象把属性和行为密封成一个整体。

现实世界中的对象具有以下特性：

(1) 每一个对象必须有一个名字来与其他对象区分开来，如在一个班级中，每一个学生对象是通过姓名来与其他学生进行区分的。

(2) 用状态或属性来描述它的具体特征。如学生由学号、专业以及身高、体重等属性来刻画具体某一个学生对象。

(3) 有一组操作或方法，每一组操作或方法决定对象的一种行为。如学生具有学习、娱乐、社交等操作，学习操作决定学生学习的行为，社交操作决定学生与其他同学及外界的交流活动等。

(4) 对象的操作可以分为两类，即自身所承受的操作和施加于其他对象的操作。如学生学习是自己施加到自身的操作，用于提高自己的知识水平，修改自身的知识储备；而学生社交是学生对象向公众提供的接口，用于该学生对象与外界进行交流。

软件的对象概念由现实世界对象抽象而来。现实世界对象的状态保存在软件对象的变量中；现实世界对象的行为通过软件对象的方法(函数)来实现。

在面向对象程序设计中，对象是描述其属性的数据以及对这些数据施加的一组操作封在一起构成的统一体。对象可以认为是“数据+操作”构成的最小操作单元。对象所能完成的操作表示它的动态行为，通常也把操作称为方法。

在面向对象的程序中使用一个对象时，只能通过对象与外界的接口来操作它。对象与外界接口也就是该对象向公众开放的操作。使用对象向公众开放的操作就好像操作电视机对象，只需要使用电视机提供给我们的操作键就可以了，通过操作的名称如更换频道就可以实现更换频道的功能了，不需要知道具体实现这些操作的方法。

事实上，实现对象操作的代码和数据是隐藏在对象内部的，一个对象就像是一个黑盒子，表示它内部状态的数据和实现各个操作的代码都被封装在这个黑盒子内部，在外面是看不见的，更不能从外面去访问或修改这些数据或代码。

属性值规定了对象所有可能的状态。对象的操作是指该对象可以展现的外部服务。例如，我们周围的每一个人可视为对象，它具有姓名、性别、身高、体重等属性，该对象具有吃饭、睡觉、思考、社交等操作，这些操作与属性同属于一个具体人的状态信息。按照人们习惯的方式，上述属性与操作本身就属于一个整体，人所具有的状态可通过其自身的方法与外界进行交流。

1.1.2 类

类是某些对象的共同特征(属性和操作)的表示，对象是类的实例。

例如，人类包含姓名、性别、身高、体重等属性，具有走路、思考、社交等操作，如程序 1-1 所示。是否建立了一个丰富的类库是衡量一个面向对象程序设计语言成熟与否的重要标志。

程序 1-1

//Java 的类实例

```
class People {  
    String name; //姓名属性  
    String gender; //性别属性  
    int height; //身高属性  
    int weight; //体重属性  
  
    public void walk(){ //走路操作  
        System.out.println( "people walk..." );  
    }  
    public void think(){ //思考操作  
        System.out.println( "people think..." );  
    }  
    public void communication(){//社交操作  
        System.out.println( "people communication..." );  
    }  
}
```

在软件中，类就是一个模板，它定义了通用于一个特定种类的所有对象的状态(变量)和行为(方法)。

类是创建对象的模板，对象是类的实例。

从程序设计的角度看，类是面向对象程序中最基本的程序单元。类实质上定义的是一种数据类型，这种数据类型就是对象类型。所以我们可以使用类名称来声明对象变量。

声明对象变量之后，还不能使用对象。必须用运算符 **new** 创建对象实体之后，才能使用对象。

1.1.3 对象的属性与相互关系

状态和行为是对象的主要属性。

状态又称为对象的静态属性，主要指对象内部所包含的各种信息，也就是变量，这些变量的值标明了对象所处的状态。当对象经过某种操作和行为而发生状态改变时，具体地就体现为它的属性变量的内容的改变。通过检查对象属性变量的内容，就可以了解这个对象当前所处的状态，如程序 1-2 所示的学生对象的姓名属性的检查及修改。

程序 1-2

//Java 的对象属性实例

```
class student {  
    String studentID; //学号属性  
    String name; //姓名属性  
    String gender; //性别属性  
    int height; //身高属性  
    int weight; //体重属性
```



```

public void setName( String name){    //设置姓名操作
    this.name=name;
}

public String getName(){    //检查姓名操作
    return name;
}

public void modifyName(String modifyName){    //修改姓名操作
    this.name= modifyName;
}

}

```

行为又称为对象的动态属性，即对象的操作或方法。它一方面把对象的内部变量包裹、封装、保护起来，使得只有对象自己的方法才能操作这些内部变量；另一方面是对象与外部环境和其他对象交互、通信的接口，对象的环境和其他对象可以通过这个接口来调用对象的方法、操纵对象的行为、改变对象的状态。如程序 1-2 所示，对象只有通过 setName、getName 及 modifyName 方法对静态属性 name 进行操作。

在面向对象方法学中，“对象”是现实世界的实体或概念在计算机逻辑中的抽象表示。对象是具有唯一对象名和固定对外接口的一组属性和操作的集合，是用来模拟组成或影响现实世界问题的一个或一组因素。其中对象名是区别于其他对象的标志；对外接口是对象在约定好的运行框架和消息传递机制中与外界通信的通道；对象的静态属性表示了它所处的状态；而对象的操作则用来改变对象的状态以达到特定的功能。对象的最主要特点是以数据为中心，它是一个集成了数据和其操作的独立、自恰的逻辑单位。

面向对象的问题求解就是力图从实际问题中抽象出这些封装了数据和操作的对象，通过定义属性变量和操作来表述它们的特征和功能，通过定义接口来描述它们的地位及与其他对象的关系，最终形成一个广泛联系的可理解、可扩充、可维护、更接近于问题本来面目的动态对象模型系统。

对象从形式上看是程序员或用户所定义的抽象数据类型的变量，当用户定义了一个对象，就创建出了具有丰富内涵的新的数据类型。它是构成和支撑整个面向对象程序设计(Object Oriented Programming)最重要的细胞与基石，它具有以下特性：

(1) 封装性 对象将私有元素和实现操作的内部细节隐藏起来，并能管理自己的内部状态。外部只能从对象所表示的具体概念、对象提供的服务和对象提供的外部接口来认识对象。通过向对象发送消息来激活对象自身的动作，达到施加特定操作的目的。

(2) 模块独立性 一个对象是一个独立存在的模块，具有独立的操作、计算能力。从外部看模块，只需了解它具有哪些功能，至于它如何实现这些功能和使用哪些局部数据来完成它们，则都隐藏在模块内部，其自身的变化不受外界的干扰，也不会涉及到其他的模块；模块间的依赖性极小或几乎没有；各模块可以可作为独立的单元成为软件系统的组建而不必担心波及或破坏其他模块。

(3) 动态连接性 对象的动态连接性是指对象通过消息传递机制将对象动态地连接在一起的特性。对象能接收其他对象发送的消息，也能向其他对象发送消息，通过这种联系，若