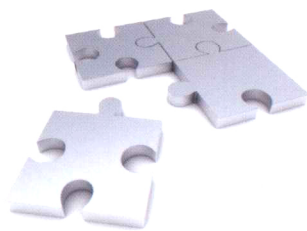
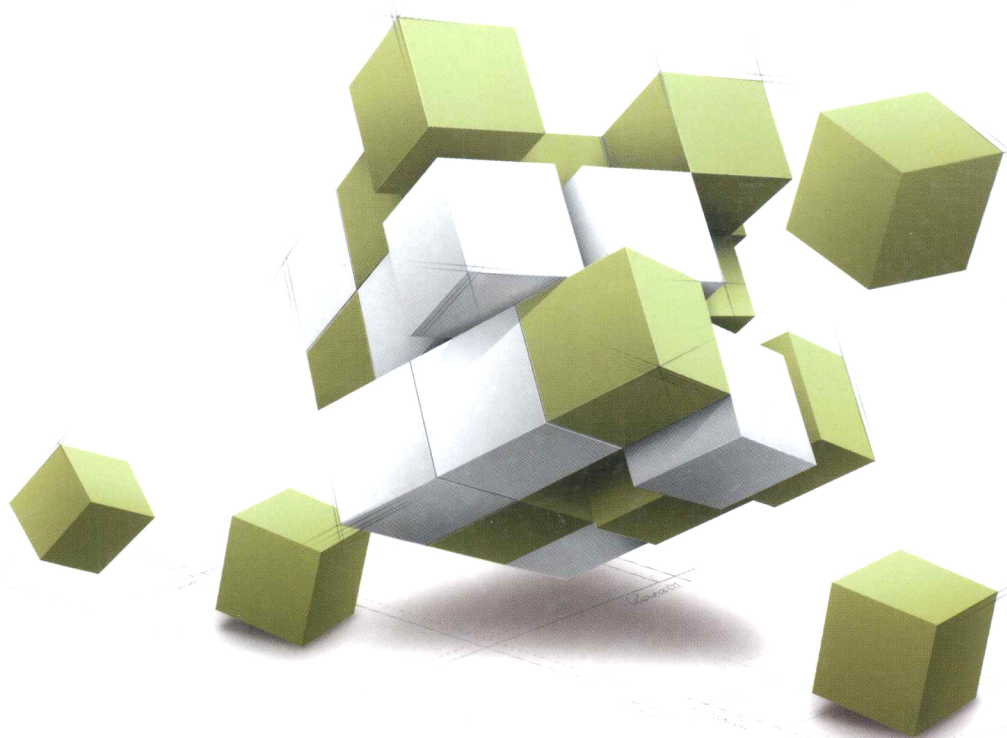


UML



统一建模教程与实验指导



理论与实际相结合·教学与实验相结合

理论篇：深入浅出地介绍面向对象、UML语言的基本理论

绘图篇：如何使用Rational Rose建模工具创建各种视图和图

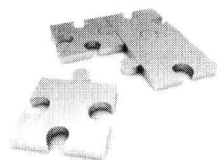
实战篇：通过综合案例详解用Rational Rose进行UML建模的全过程

谢星星 刘小松 王坚宁 编著



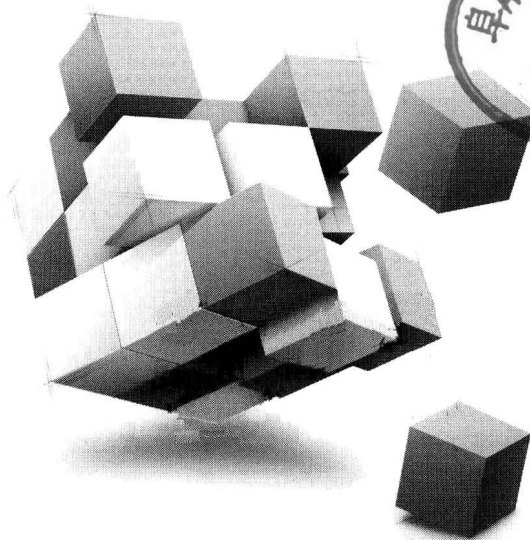
清华大学出版社

UML



统一建模教程与实验指导

谢星星 刘小松 王坚宁 编著



清华大学出版社
北京

内 容 简 介

本书是一本关于 UML 统一建模的实用教程,深入浅出、循序渐进地介绍了软件建模的概念、规范和方法。本书共有 3 大部分,第一部分是理论篇,着重于介绍面向对象、UML 建模语言的一些基本理论,详尽介绍了 UML 中类图、对象图、用例图、包图、序列图、协作图、活动图、状态图、构件图和部署图的概念;第二部分是绘图篇,着重于介绍如何使用 Rational Rose 建模工具来创建理论篇中的各种视图和图;第三部分是实战案例篇,通过一个综合实例对使用 Rational Rose 进行 UML 建模的全过程进行了详解的分析。此外,各章后配有适量的练习题和上机题,以加深读者的理解。

本书最大的特点是将理论和实际紧密结合起来,实例丰富、图文并茂、讲解详尽、实践性强。本书可以作为大专院校计算机软工程专业学生学习 UML 和面向对象的技术教材,也可作为广大软件开发人员和系统架构分析设计人员自学 UML 的参考和指导用书。

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 13701121933

图书在版编目(CIP)数据

UML 统一建模教程与实验指导/谢星星,刘小松,王坚宁编著. —北京:清华大学出版社,2013.1
ISBN 978-7-302-30621-4

I. ①U… II. ①谢… ②刘… ③王… III. ①面向对象语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字(2012)第 272649 号

责任编辑:夏非彼

封面设计:王 翔

责任校对:闫秀华

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:190mm×260mm

印 张:18.25

字 数:464 千字

版 次:2013 年 1 月第 1 版

印 次:2013 年 1 月第 1 次印刷

印 数:1~4000 册

定 价:32.00 元

产品编号:050342-01

前 言

Unified Modeling Language (简称 UML) 是一种定义良好、易于表达、功能强大且普遍适用的建模语言, 它融入了软件工程领域的新思想、新方法和新技术。UML 支持面向对象的分析与设计, 可以方便地与具体的软件开发过程相结合。Rational Rose 是目前业界最为瞩目的可视化软件开发工具, 通过它可以便捷高效地完成 UML 的建模工作。

本书以介绍面向对象的统一建模语言 UML 为主, 目的是使读者了解面向对象技术的基本概念, 掌握面向对象的分析和设计方法, 以及与面向对象技术相关的一些软件开发技术, 同时掌握在 Rational Rose 环境下使用 UML 进行分析和设计的技术。本书在讲述 UML 的元素时, 结合了大量实战的案例, 并且为了提高大家的学习效率, 在每个章节后面还提供了一定数量的习题, 完成这些习题可以使读者加深对 UML 的理解。

本书的内容

本书分为三个部分, 共 19 章。

第一部分是理论篇, 共有 10 章。内容着重于面向对象、UML 建模语言的一些基本理论, 详尽地介绍了 UML 中类图、对象图、用例图、包图、序列图、协作图、活动图、状态图、构件图和部署图的概念。

第 1 章, UML 与面向对象: 主要介绍了面向对象的基本知识、建立系统模型的方法以及 UML 统一建模语言的基本内容。

第 2 章, UML 建模基础知识: 对 UML 进行了系统的介绍, 包括常用的 UML 元素、UML 的通用机制、UML 的扩展机制等必须了解的知识点。

第 3 章, 用例图: 介绍了用例图的概念和作用, 讲解了用例图的重要组成元素。

第 4 章, 类图 and 对象图: 介绍了类图 and 对象图的基本概念以及在 UML 中的表示。

第 5 章, 包图: 对包图 and 包的基本概念以及组成元素逐一进行详细介绍。

第 6 章, 状态图: 介绍了构成状态图的元素 and 状态图在动态建模中的功能。

第 7 章, 活动图: 介绍了状态图的扩展——活动图的定义、作用及其各种标识符的表示方法。

第 8 章, 序列图 and 协作图: 讲解了 UML 动态建模中最为重要的两种交互图, 包括它们的概念、作用、区别 and 组成的元素。

第 9 章, 构件图 and 部署图: 对 UML 中描述系统物理实现 and 物理运行情况的构件图 and 部署图进行了详细的讲解。

第 10 章, Rational 统一过程: 介绍了统一过程的概念、结构、配置 and 实现 Rational 统一过程的方法。

第二部分是绘图篇, 共 8 章。内容着重于介绍如何使用 Rational Rose 建模工具来创建理

论篇中各种 UML 中的视图和图。

第 11 章, Rational Rose 入门: 对 UML 的主流开发工具——Rational Rose 进行介绍, 包括其安装、使用、四种视图模型和生成代码的方法。这一章为后面绘制 UML 图打好了基础。

第 12 章, 用例图的绘制: 演示了用例图、用例、参与者以及它们彼此关系的绘制步骤并通过实例讲解实现用例图的分析过程。

第 13 章, 类图和对象图的绘制: 演示了类图、对象图、类、类的属性、类的操作以及类之间四大关系的绘制步骤和分析实现的过程。

第 14 章, 包图的绘制: 演示了包图和包与包之间关系的绘制步骤和分析实现的过程。

第 15 章, 状态图的绘制: 读者在本章的学习后, 能够掌握状态图的画法。

第 16 章, 活动图的绘制: 详细地介绍了使用 Rational Rose 绘制活动图的完整步骤和过程。

第 17 章, 序列图和协作图的绘制: 详细地介绍了使用 Rational Rose 绘制序列图和协作图的完整步骤和分析过程。

第 18 章, 构件图和部署图的绘制: 介绍了构件图和部署图如何通过 Rational Rose 工具在 UML 中完成系统建模的过程和步骤。

第三部分是案例篇, 通过一个综合实例对使用 Rational Rose 进行 UML 建模的全过程进行了详细的分析。

第 19 章, BBS 网上论坛: 通过这个综合实例, 将前面各章介绍的 UML 的各种图形以及模型元素综合起来, 实现对一个完整系统的设计和建模过程。整个过程按照软件设计的一般流程进行, 让读者对实际项目的系统建模有一个直观的认识。

本书的特点

(1) 理论实际, 紧密结合

本书在介绍理论知识的同时, 在每一章给出了大量案例讲解, 力求让读者在理解基础知识后, 就能学以致用, 快速上手。在每章末尾都配有相应的课后习题和上机练习, 方便读者进行实践练习。

(2) 配套源码, 免费课件

为了方便读者自学, 本书不仅在每章附有精心设计的课后习题和上机练习, 还配有免费的课件、实例源码(有关资源, 请联系作者邮箱 myworks2001@263.net)等, 这样就能使读者牢固而扎实地掌握各个基本知识点。

(3) 图文并茂, 步骤详细

在具体介绍 Rational Rose 软件功能的时候, 本书提供了详细的图例, 详尽地说明了每一步功能的实现, 让读者一眼就能明了整个功能的使用方法和绘制步骤。每一个步骤都以通俗易懂的语言进行讲述, 读者只需要按照步骤操作, 就可以轻松完成软件的建模。

面向的读者

本书既可以作为大专院校计算机软工程专业学生学习 UML 和面向对象的技术教材, 也可作为广大软件开发人员和系统架构分析设计人员自学 UML 的参考和指导用书。

致谢

本书主要由谢星星、刘小松、王坚宁主持编写，其中江苏省财政厅信息管理中心工程师刘小松编写了第9章～第19章，并对全书进行了统稿。此外，胡书敏、张国华、李华、王林、李志国、冯慧、徐红、周建国、张建、陈浩、何武和许小荣等同志也参加了本书的编写，在整理材料方面给予了编者很大的帮助，在此，编者对他们表示衷心的感谢。

由于作者水平有限，书中难免会有错误、纰漏之处，欢迎广大读者批评斧正。

编者

2012年9月

目 录

第 1 章 UML 与面向对象	1
1.1 面向对象开发	2
1.1.1 理解面向对象	2
1.1.2 面向对象的特征	5
1.1.3 面向对象的项目设计	8
1.1.4 面向对象开发的优点	11
1.2 UML 简介	11
1.2.1 对系统建模的必要性	11
1.2.2 系统建模的目标和原则	12
1.2.3 UML 的定义	13
1.2.4 UML 的优势	14
1.2.5 在何种场合需要 UML	14
1.2.6 多种 UML 建模类型	15
习题 1	16
第 2 章 UML 建模基础知识	17
2.1 UML 语言的构成	18
2.2 UML 的基本元素	18
2.2.1 结构 (Structure)	18
2.2.2 行为 (Behavior)	20
2.2.3 分组 (Grouping)	21
2.2.4 注释 (Annotation)	21
2.3 UML 中的关系	22
2.4 UML 中的视图和图	23
2.4.1 视图	23
2.4.2 图	27
2.5 UML 的公共机制	30
2.5.1 UML 的规格描述	30
2.5.2 UML 的修饰	31
2.5.3 UML 的通用划分	32
2.5.4 UML 的扩展机制	32
习题 2	35



第3章 用例图	36
3.1 用例图的构成	36
3.1.1 参与者	37
3.1.2 用例	38
3.1.3 关系	39
3.1.4 系统边界	43
3.2 确定参与者	43
3.3 确定用例	44
3.4 用例的粒度	45
3.5 用例规约	46
习题3	46
第4章 类图 and 对象图	48
4.1 类图的定义	49
4.2 类的构成	50
4.2.1 名称	51
4.2.2 操作	51
4.2.3 属性	53
4.2.4 注释	54
4.2.5 约束	54
4.2.6 职责	55
4.3 UML 中不同的类	55
4.4 构造型	56
4.5 类之间的关系	57
4.5.1 泛化关系	57
4.5.2 实现关系	58
4.5.3 依赖关系	59
4.5.4 关联关系	60
4.6 类的识别	61
4.6.1 名词识别	61
4.6.2 实体识别	62
4.6.3 用例识别	62
4.6.4 利用分解和抽象技术	62
4.7 对象图	63
4.7.1 对象的表示	64
4.7.2 链的表示	64
习题4	65

第 5 章 包图	66
5.1 包图的定义	66
5.1.1 模型的组织结构	67
5.1.2 包图和包	68
5.2 包的组成	69
5.2.1 名称	69
5.2.2 可见性	70
5.2.3 构造型	70
5.2.4 子系统	71
5.3 包图中的关系	72
5.3.1 依赖关系	72
5.3.2 泛化关系	74
5.4 包的嵌套	74
习题 5	75
第 6 章 状态图	76
6.1 状态图的定义	76
6.1.1 状态机	77
6.1.2 状态图	78
6.2 状态图中的标记符	79
6.2.1 状态	79
6.2.2 转换	80
6.2.3 判定	82
6.2.4 同步	83
6.3 状态图中的动作和事件	83
6.3.1 事件	84
6.3.2 动作	85
6.4 常用状态的种类	86
6.4.1 顺序状态	86
6.4.2 并发状态	86
6.4.3 历史状态	87
习题 6	87
第 7 章 活动图	89
7.1 活动图的定义	89
7.2 活动图中的标记符	91
7.2.1 动作	91
7.2.2 活动	92
7.2.3 转换	92



7.2.4 组合活动	92
7.3 其他标记符	93
7.3.1 分支与合并	93
7.3.2 分叉与汇合	93
7.3.3 泳道	94
7.3.4 对象流	95
习题 7	96
第 8 章 序列图和协作图	98
8.1 序列图的定义	98
8.2 序列图的组成	99
8.2.1 对象	99
8.2.2 生命线	100
8.2.3 激活	100
8.2.4 消息	101
8.3 对象行为	102
8.3.1 创建对象和销毁对象	102
8.3.2 分支	103
8.3.3 从属流	103
8.4 协作图的定义	104
8.5 协作图的构成	105
8.5.1 对象	105
8.5.2 链	105
8.5.3 消息	105
习题 8	106
第 9 章 构件图和部署图	108
9.1 构件和构件图	108
9.1.1 构件的定义	109
9.1.2 构件的种类	109
9.1.3 构件的表示	110
9.1.4 构件图的定义	112
9.2 部署图的定义	113
9.3 部署图的构成	113
9.3.1 节点	113
9.3.2 连接	115
习题 9	115
第 10 章 Rational 统一过程	117
10.1 Rational 统一过程的概念	118

10.2 Rational 统一过程的构成.....	119
10.2.1 Rational 统一过程的核心工作流.....	119
10.2.2 Rational 统一过程的迭代开发模式.....	121
10.2.3 Rational 统一过程的最佳实践.....	122
10.3 Rational 统一过程的开发模型.....	124
10.3.1 统一过程的动态开发.....	125
10.3.2 统一过程的静态开发.....	127
10.3.3 面向架构的过程.....	128
10.4 Rational 统一过程的配置和实现.....	130
10.4.1 配置 Rational 统一过程.....	130
10.4.2 实现 Rational 统一过程.....	131
习题 10.....	132
第 11 章 Rational Rose 入门.....	133
11.1 Rational Rose 工具.....	134
11.1.1 Rational Rose 简介.....	134
11.1.2 Rational Rose 的功能.....	134
11.2 Rational Rose 的安装.....	136
11.3 Rational Rose 操作指南.....	138
11.3.1 Rational Rose 主界面.....	140
11.3.2 Rational Rose 的常用操作.....	145
11.4 Rational Rose 的 4 种视图模型.....	154
11.4.1 用例视图 (Use Case View).....	154
11.4.2 逻辑视图 (Logical View).....	159
11.4.3 构件视图 (Component View).....	162
11.4.4 部署视图 (Deployment View).....	164
11.5 Rational Rose 代码生成.....	166
11.5.1 代码生成的方法.....	166
11.5.2 逆向工程.....	170
11.6 上机题.....	170
第 12 章 用例图的绘制.....	171
12.1 创建用例图.....	171
12.2 创建参与者.....	174
12.3 创建用例.....	175
12.4 创建用例和参与者之间的关联.....	176
12.5 创建用例之间的关系.....	177
12.6 创建用例图实例分析.....	178



12.6.1	需求分析	178
12.6.2	确定参与者	178
12.6.3	确定用例	179
12.6.4	构建用例模型	179
12.7	上机题	180
第 13 章	类图和对象图的绘制	182
13.1	类图的创建	182
13.2	类的创建	184
13.2.1	类属性的创建	186
13.2.2	类操作的创建	187
13.2.3	构造型的设置	187
13.2.4	设置类的分类	188
13.3	绘制类之间的关系	189
13.4	绘制对象图的方法	190
13.5	类图创建实例	191
13.5.1	确定类和关联	192
13.5.2	确定属性和操作	193
13.5.3	根据类图创建对象	193
13.6	上机题	194
第 14 章	包图的绘制	196
14.1	包的创建	196
14.1.1	在包图中添加包	197
14.1.2	包中类的添加	198
14.1.3	包之间关系的绘制	199
14.2	包图创建实例	199
14.2.1	确定包的分类	200
14.2.2	创建包和包之间的关系	200
14.3	上机题	201
第 15 章	状态图的绘制	203
15.1	状态图的创建	203
15.1.1	绘制初始和终止状态	204
15.1.2	绘制状态	205
15.1.3	绘制转换	206
15.1.4	设置事件	206
15.1.5	设置动作	207
15.1.6	设置监护条件	207
15.2	状态图创建实例	208

15.2.1 确定状态图的实体	208
15.2.2 确定实体的状态	208
15.2.3 确认相关事件	209
15.3 上机题	209
第 16 章 活动图的绘制	211
16.1 活动图的创建	211
16.1.1 绘制起始和终止状态	213
16.1.2 绘制动作	213
16.1.3 绘制活动	214
16.1.4 绘制泳道	216
16.1.5 绘制转换	217
16.1.6 绘制分支	218
16.2 活动图创建实例	219
16.2.1 确定需求用例	219
16.2.2 确定用例路径	220
16.2.3 创建完整的活动图	220
16.3 上机题	221
第 17 章 序列图和协作图的绘制	223
17.1 序列图的创建	223
17.1.1 对象的创建	225
17.1.2 消息的绘制	227
17.1.3 消息的设置	228
17.1.4 生命线的设置	229
17.1.5 对象的销毁	230
17.2 协作图的创建	230
17.2.1 对象的创建	232
17.2.2 链和消息的绘制	233
17.2.3 序列图和协作图的转换	233
17.3 序列图创建实例	234
17.3.1 需求分析	234
17.3.2 确定序列对象	235
17.3.3 创建的序列图	235
17.4 协作图创建实例	236
17.4.1 协作元素的确认	236
17.4.2 确定元素之间的关系	236
17.4.3 创建协作图	237
17.5 上机题	237



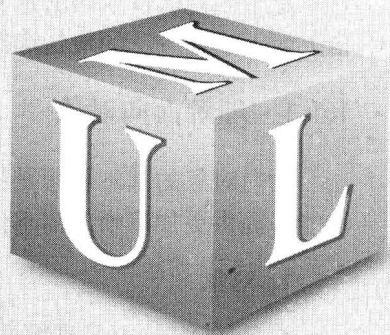


第 18 章 构件图和部署图的绘制	239
18.1 构件图的创建	239
18.1.1 构件的绘制	241
18.1.2 构件间关系的绘制	242
18.2 部署图的创建	243
18.2.1 节点的绘制	243
18.2.2 节点的设置	244
18.2.3 连接的绘制	245
18.3 构件图和部署图创建实例	247
18.3.1 创建构件图	247
18.3.2 创建部署图	248
18.4 上机题	249
第 19 章 BBS 网上论坛	251
19.1 系统需求分析	251
19.2 系统建模	254
19.2.1 创建系统用例模型	254
19.2.2 创建系统静态模型	257
19.2.3 创建系统动态模型	260
19.2.4 创建系统部署模型	269
习题参考答案	272



第 1 章

UML 与面向对象



本章要点

- 理解面向对象的含义
- UML 的概念和作用



本章导读

- 基础内容：类和对象的关系
- 重点掌握：面向对象的四个特征
- 一般了解：面向对象项目的设计



课堂讲解

面向对象开发是一种先进的设计和构造软件的技术，使计算机以更符合人的思维方式去解决一系列的编程问题，极大地提高了代码复用程度、代码可扩展性和编程效率，同时减少了软件维护的代价。在面向对象开发的发展过程中，统一建模语言 UML 应运而生，统一了过去相互独立的数十种面向对象建模语言共同存在的局面。



1.1 面向对象开发

面向对象的程序设计（Object-Oriented Programming，简称为 OOP）是一种软件设计方法，它立足于创建软件代码的重复使用，具备更好地模拟现实世界环境的能力，这使它被公认为是自上而下编程方法的最佳选择。

1.1.1 理解面向对象

面向对象的基本思想是尽可能按照人类认识世界的方法和思维来分析和解决问题，这种方法能够提供更加清晰的需求分析和设计，是指导软件开发活动的系统方法。

1. 对象

对象（Object）是理解面向对象的关键，它经常用于建立现实世界中一些对象的模型。对于个人而言，所认识的东西都是对象，例如，房间里面的桌子、椅子、电脑等，我们都可以认为是对象。一旦认识某样东西，就能说出其特点，并将它与别的对象进行比较。

了解了事物的特征和行为，就表示对该事物有了认识和概念，尽管有些东西并不存在，但只要对其有概念，就是对象了。例如，龙、凤凰都是我们熟悉的对象，这些对象是我们看不到的、听不到的，但是在描述抽象模型和物理对象时，仍然起着非常重要的作用。

软件工程设计中的对象和上面所描述的对象的意义有所不同，软件工程中的对象是指一种将状态和行为有机结合起来形成的软件构造模型，它可以用来描述或代表现实世界中的一个对象。也可以这样说，软件对象其实就是现实世界中对象的一种模型，它有自己的状态和行为。

可以利用一个或者多个变量来标识软件对象的状态。变量是指由用户标识符来命名的数据项，软件对象可以利用它的方法来执行它的行为，而方法则是与对象相关联的函数（子程序）。

面向对象认为客观世界是由各种各样的对象组成的，每个对象都有自己的数据和操作，对象和对象之间是通过对象暴露在外部的接口进行调用的，不同对象之间的相互联系和作用构成了各种系统。

对象是组成系统的基本单元，是一个有组织的含有信息的实体，它是由状态（Attribute）和行为（Behavior）构成的。实际上，状态（Attribute）、属性（Property）还有数据（Data）等这些在相关书本中提到的概念，都是用于描述一个对象的数据元素，这些概念具体到不同语言便有不同的叫法。

对象只有在具有状态和行为的情况下才有意义，状态用来描述对象的静态特征，行为描述对象的动态特征。对象是包含客观事物特征的抽象实体，封装了状态和行为，在程序设计领域，可以表示为“对象=数据+数据的操作”。

2. 类

类（Class）是创建对象的模板，在整体上可以代表一组对象。它是具有相同属性和操作的一组对象的组合，也就是说，抽象模型中的类描述了一组相似对象的共同特征，为属于该类的全部对象提供了统一的抽象描述。

类的定义包含以下要素：

- 定义该类对象的数据结构（属性的名称和类型）。
- 对象所要执行的操作，也就是类的对象要被调用执行哪些操作，以及执行这些操作时对象要执行哪些操作，如数据库操作等。

类是对象集合的再抽象，类与对象的关系如同一个模具和使用这个模具浇注出来的铸件一样，类是创建软件对象的模板，即一种模型。类给出了属于该类的全部对象的抽象定义，而对象是符合这种定义的一个实体。类的作用是：

- 在内存中开辟一个数据区，存储新对象的属性。
- 把一系列行为和对象关联起来。

一个对象又被称为类的一个实例，也称为实例化（Instantiation）。它是指对象在类声明的基础上创建的过程。比如，我们创建“人”这个类，它就代表“人”这个概念。现在有一个名字叫“小明”的人，就表示小明是“人”这个类的一个实体对象，并且包含了“叫李平”这样一个信息。我们可以使用这个类来表达更多其他具体对象，如“小张”、“小王”等。这样来看，设计类而不是设计对象就可以避免重复编码，类只需要编码一次，就可以实例化属于这个类的无数个对象。比如说，我们声明了一个“学生”类，就可以在这个基础上创建一个姓名叫“小明”的学生对象。

类的确定和划分没有统一的标准和方法，基本上依赖于设计人员的经验、技巧以及对实际项目中问题的把握。通常的标准是“寻求共性、抓住特性”，即在一个大的系统环境中，寻求事物的共性，将具有共性的事物用一个类进行表述，在具体的程序中，具体到某一个对象，要抓住对象的特性。确定一个类的步骤通常包含以下方面：

- 确定系统的范围。
- 在系统范围内寻找对象，该对象通常具有一个和多个类似的事物。
- 将对象抽象成为一个类，按照上面类的定义，确定类的数据和操作。

在面向对象程序设计中，类和对象的确定是软件开发非常重要的一步，类和对象的确定直接影响软件设计的优劣。如果划分得当，对于软件的维护与扩充以及软件的重用性，都有好处。

3. 消息和事件

当使用某一个系统的时候，我们单击一个按钮，这时通常会显示相应的信息。以“网上书店系统”为例，我们单击界面上某一个按钮的时候，会显示出当前查询书籍的信息。那么当前的程序是如何运行的呢？

- “网上书店系统”界面的某一个按钮发送鼠标点击事件的消息给相应的对象。
- 对象接收到消息后有所反应，它提供书籍的相关信息给界面。
- 界面将书籍的相关信息显示出来，完成任务。

在这个过程中，我们首先要触发一个事件，然后发送消息，那么什么是消息呢？所谓消息（Message）是指描述事件发生的信息，是对象间相互联系和相互作用的方式。一个消息主要由

