UML基础与Rose建模实用教程

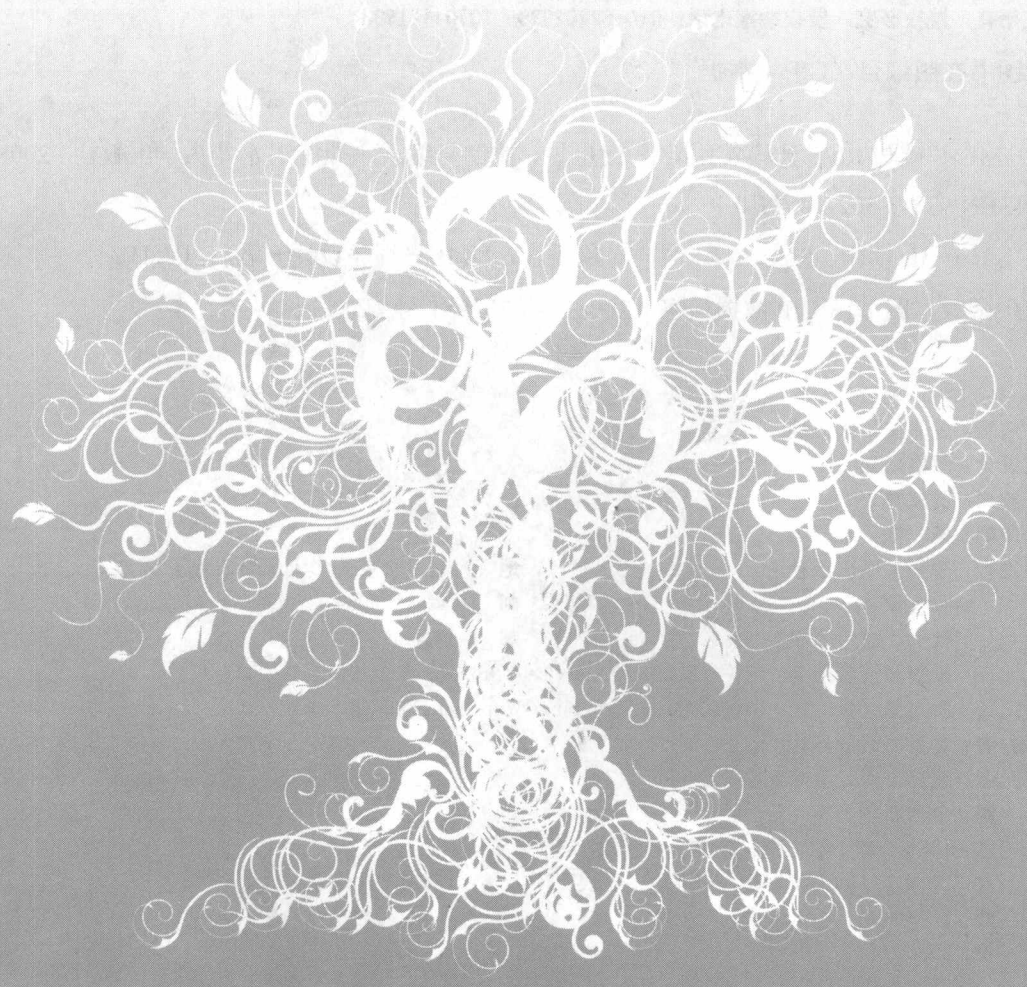
谢星星 沈懿卓 编著

- UML基础理论与Rose建模实践操作完美结合
- 完整、系统地学习UML图和图中模型元素的创建
- 两个完整的建模案例培养动手能力

清华大学出版社

UML基础与Rose建模实用教程

谢星星 沈懿卓 编著



清华大学出版社
北 京

内 容 简 介

本书系统地讲解了学习 UML 图和创建图中模型元素的基础理论，并通过两个完整的建模案例讲解 Rose 建模工具的使用。

全书内容分为 4 部分：入门基础（第 1～4 章）介绍了面向对象、UML、Rational Rose 和 Rational 统一过程的相关知识；图（第 5～12 章）针对 UML 的各种图，包括用例图、类图、对象图、序列图、协作图、状态图、活动图、包图、构件图和部署图进行介绍；案例（第 13～14 章），以图书管理系统和超市信息管理系统为例，介绍如何使用 UML 分析和设计一个实际的项目；附录（附录 A～附录 B）针对 Rational Rose 的安装和应用进行详细介绍，并给出章末练习答案。本书注重学习的渐进性和实践性，对 UML 每一种图的讲解均通过“图的基本概念→图的组成→图的创建概述→图的创建示例”方式进行，从而完整地把握每一种 UML 图。通过建模的具体案例，帮助读者达到学以致用的目的。此外，每章附有操作练习题，着重培养读者的动手能力，使其在练习过程中能快速提高实际应用水平。

本书适合软件设计与开发人员学习参考，更适合作为高等院校计算机软件工程相关专业的教材或教学参考书。

图书在版编目（CIP）数据

UML 基础与 Rose 建模实用教程/谢星星，沈懿卓编著.—北京：清华大学出版社，2008.10

ISBN 978-7-302-18539-0

I. U… II.①谢…②沈… III.面向对象语言，UML－程序设计－教材 IV.TP312

中国版本图书馆 CIP 数据核字（2008）第 138981 号

责任编辑：夏非彼　张　楠
装帧设计：图格新知
责任校对：贾淑媛
责任印制：杨　艳
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**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 刷 者：北京密云胶印厂
装 订 者：北京市密云县京文制本装订厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：24.5　**字　数**：596 千字
版　　次：2008 年 10 月第 1 版　　**印　　次**：2008 年 10 月第1 次印刷
印　　数：1～4000
定　　价：39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：(010)62770177 转 3103　　产品编号：030555－01

前言

Preface

软件是从20世纪50年代开始诞生，至今已经有近60年的历史。在20世纪70年代到80年代，面向对象技术开始有深入的研究并广泛予以应用，面向对象的建模也开始出现。经过近四十年的发展，面向对象技术已经成为软件开发中分析、设计、实现的主流方法和技术。在面向对象技术发展的同时，伴随着面向对象技术的各种软件设计工具、规范等也获得了较大发展。其中，最重要的一个成果就是统一建模语言（Unified Modeling Language，UML）的出现。

UML的创建首先开始于1994年10月，由Grady Booch、Jim Rumbaugh和Ivar Jacobson共同开发，并于1996年发布了UML版本。1997年11月17日，对象管理组织（OMG）开始采纳UML为其标准建模语言，并最终统一为大众所接受的标准建模语言。Rational Rose是由Rational软件开发公司设计、开发的一种重要的可视化建模工具。

本书分为入门基础、图、案例、附录共4个部分。

- 入门基础（第1~4章）：着重介绍UML和Rational Rose的预备知识，包括面向对象概述、UML概述、Rational Rose概述、Rational统一过程。
- 图（第5~12章）：着重介绍UML的各种图，包括用例图、类图、对象图、序列图、协作图、状态图、活动图、包图、构件图与部署图等，对UML图的介绍方式为：图的基本概念→图的组成→图的创建概述→图的创建示例。通过这种方式能够使读者完整而系统地去把握和了解每一种UML图。
- 案例（第13~14章）：着重介绍两个案例，即图书管理系统和超市信息管理系统。通过这两个案例全面而系统地对如何使用Rational Rose进行建模给予说明。
- 附录（附录A~附录B）：附录A介绍了Rational Rose的安装和应用，目的是帮助读者熟练使用Rational Rose进行建模，附录B是各章后练习的习题答案，方便读者参考。

不管您是从事面向对象软件开发的开发人员，还是希望通过学习UML帮助自己建模的人员，本书都能够帮助您全面了解UML的基本概念和建模方法，本书同样也适合作为高等院校计算机软件工程相关专业的教学用书或参考书。

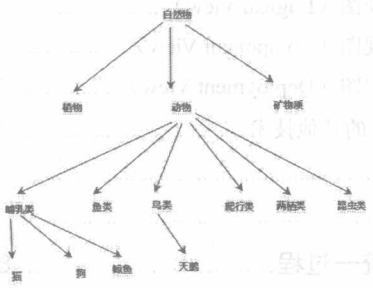
本书由谢星星和沈懿卓编写，参与本书编辑和修改的还有叶明、崔宁、卢宏、汪昔玉、卫平峰、程冬丁、王勤、张锐、汪小锋、李葵、叶浩、肖飞、宋海剑、林勇、朱衡等同志。在此，编者对以上人员致以诚挚的谢意！

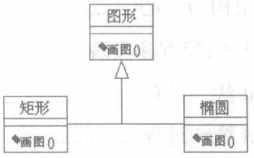
由于作者水平有限，书中难免存在疏漏和错误之处，恳请专家和广大读者批评指正。在学习过程中，遇到疑难问题，可以通过以下方式与我们联系：booksaga@126.com，也可以登录图格新知网站 http://www.booksaga.com 留言，我们将在第一时间给予答复！

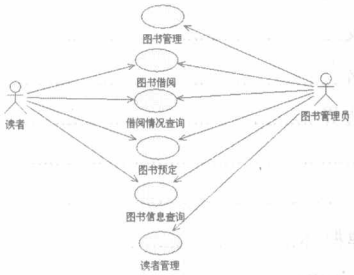
编　者

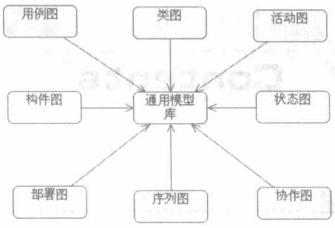
2008 年 6 月

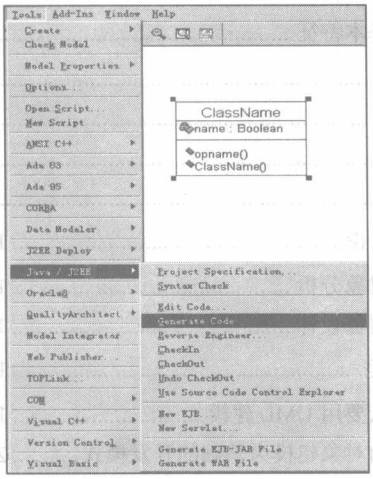
目录 Contents

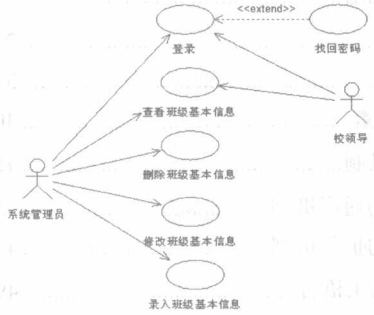


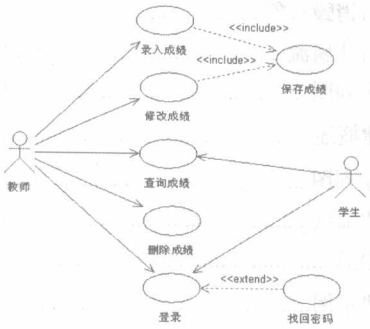


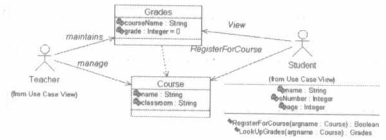


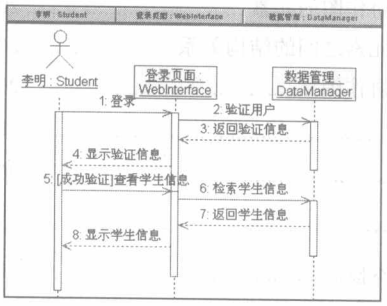


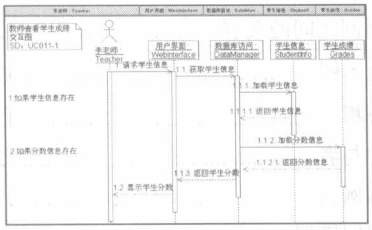


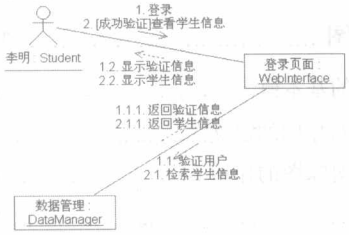


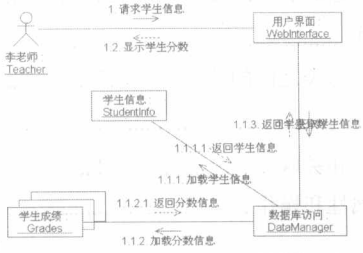


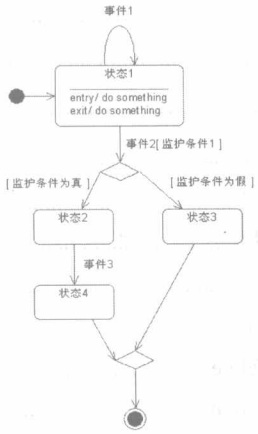


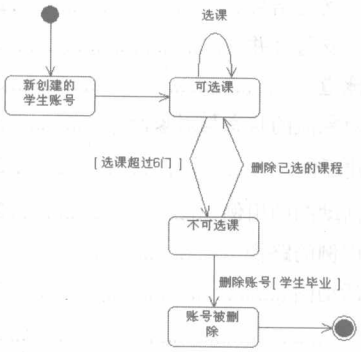


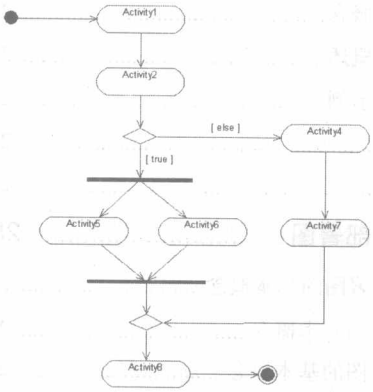


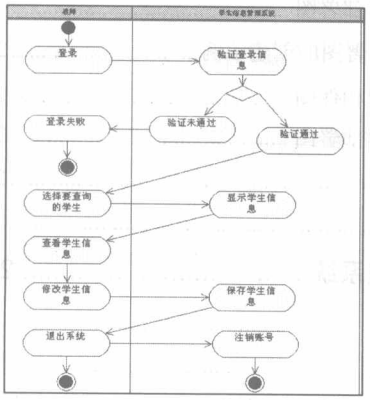


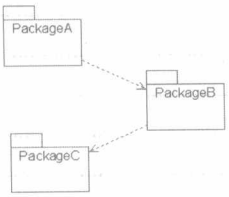


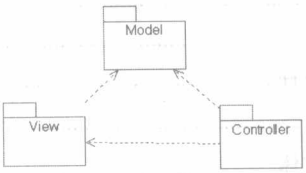


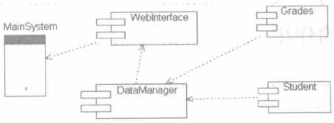


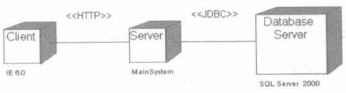


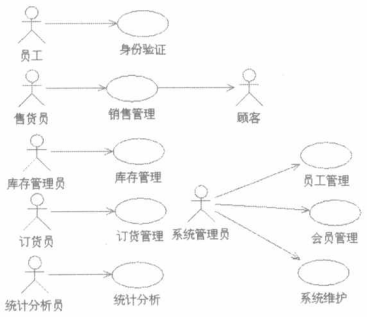




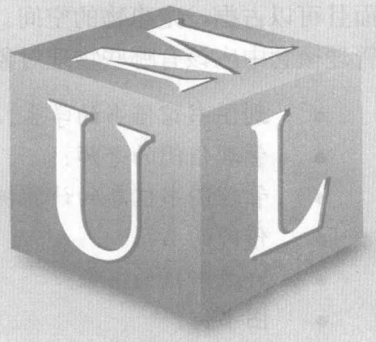








第 1 章



面向对象概述

目前，面向对象技术已经成为计算机软件开发中的一种主流技术了，随着其研究内容不断深化、应用领域不断扩大，特别是工业界对面向对象技术研究与产品化方面的支持，使得面向对象技术体现出越来越强大的生命力。面向对象技术作为一种全新的设计和构造软件的技术，使得计算机解决问题的方式更加符合人类的思维方式，更能直观地描述客观世界。面向对象的编程语言以及编程工具对代码可重用性、可扩充性以及自动化编程的深入支持，提高了编程效率、减少了软件的维护工作。面向对象技术越来越被众多的软件开发人员所接受。

本章首先介绍了面向对象的基本概念，并与面向过程进行对比，然后具体到面向对象的基本特征和实际的方法论技术，最后对为什么使用 UML 建模进行简要介绍。本章重点强调的是面向对象的基本特征和方法论。

1.1 面向对象的基本概念

面向对象是一种全新的软件技术，其概念来源于程序设计本身。从 20 世纪 60 年代提出面向对象的概念到现在，面向对象已经发展成为一种比较成熟的编程思想了，并且逐步成为软件开发领域的主流技术。面向对象的程序设计（Object-Oriented Programming，OOP）立意于创建软件重用代码，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的优胜者。它通过在程序中加入扩展语句把函数“封装”进编程所需要的“对象”中。面向对象的编程语言使得复杂的工作条理清晰、编写容易。

1.1.1 什么是对象

对象（Object）是面向对象（Object-Oriented，OO）系统的基本构造块，是一些相关的

变量和方法的软件集。对象经常用于建立对现实世界中的一些对象模型。对象是理解面向对象技术的关键。

在教室里的桌子、椅子、电脑等都可以认为是对象。根据《韦氏大词典》的解释，对象是某种可被人感知的事物，也是思维、感觉或动作所能作用的物质或精神体。

该解释的第一部分“某种可被人感知的事物”是指对象有可以看到和感知到的“东西”，而且可以占据一定事物的空间。以图书管理系统为例，想象一下图书管理系统中围绕图书管理概念中应该有哪些物理对象：

- 到图书馆借书的学生。
- 管理借阅的老师。
- 管理图书信息的计算机。
- 借阅的书籍。
- 存放书籍的书架。
- 图书馆。

解释的第二部分是“思维、感觉或动作所能作用的物质或精神体”，即“概念性对象”，以图书管理系统为例，可以列举出以下对象：

- 学生所在的院系。
- 学生的学号。
- 图书的编号。

对象是不能看到的、听到的，但是在描述抽象模型和物理对象时，仍然起着很重要的作用。

由上可知，软件对象可以这样定义：所谓软件对象，是一种将状态和行为有机结合起来形成的软件构造模型，它可以用来描述现实世界中的一个对象，也就是说软件对象实际上就是现实世界对象的模型，它有状态和行为。一个软件对象可以利用一个或者多个变量来标识它的状态。变量是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。方法是与对象相关联的函数（子程序）。

可以利用软件对象来代表现实世界中的对象，如用一个动画程序来代表现实世界中正在飞行的飞机，或者利用可以控制电子机械的程序来代表现实世界运行的机械车，也可以使用软件对象来描述抽象的概念，如按键事件就是一个用在 GUI 窗口系统的公共对象，它可以代表用户按下鼠标按钮或者键盘上的按键的反应。

1.1.2 面向对象与面向过程的区别

在面向对象程序设计（Object Oriented Programming，OOP）方法出现以前，结构化程序设计占据着主流。结构化程序设计是一种自上而下的设计方法，通常使用一个主函数来概括出整个程序需要做的事情，而主函数是由一系列子函数所组成。对于主函数中的每一个子函数，又都可以被分解为更小的函数。结构化程序设计思想就是把大的程序分解为具

有层次结构的若干个模块，每个模块再分解为下一层模块，如此自顶向下、逐步细分，从而把复杂的大模块分解为许多功能单一的小模块。结构化程序设计的特征就是以函数为中心，也就是以功能为中心来描述系统，用函数来作为划分程序的基本单位，数据在过程式设计中往往处于从属的位置。结构化程序设计的优点是易于理解和掌握，这种模块化、结构化、自顶向下与逐步求精的设计原则，与大多数人的思维和解决问题的方式比较接近。

然而，对于比较复杂的问题或在开发中需求变化比较多的情况下，结构化程序设计往往显得力不从心。这是因为结构化程序设计是自上而下的，这要求设计者在一开始就要对解决的问题有一定的了解。在问题比较复杂时，要做到这一点会比较困难，而当开发中的需求变化时，以前对问题的理解也许会变得不再适用。事实上，开发一个系统的过程往往也是一个对系统不断了解和学习的过程，而结构化程序设计方法忽略了这一点。结构化程序设计的方法把密切相关、相互依赖的数据和对数据的操作相互分离，这种实质上的依赖与形式上的分离使得大型程序的编写比较困难，并难于调试和修改。在多人进行协同开发的项目组中，程序员之间很难读懂对方的代码，代码的重用变得十分困难。由于现代应用程序的规模越来越大，对代码的可重用性和易维护性的要求也越来越高，面向对象技术对以上问题提供了很好地支持。

面向对象技术是一种以对象为基础、以事件或消息来驱动对象执行处理的程序设计技术。它是一种自下而上的程序设计方法，它不像面向过程程序设计那样一开始就需要使用一个主函数来概括出整个程序，面向对象程序设计往往从问题的一部分着手，一点一点地构建出整个程序。面向对象设计是以数据为中心，使用类作为表现数据的工具，类是划分程序的基本单位。而函数在面向对象设计中成了类的接口，以数据为中心而不是以功能为中心来描述系统，相对来讲，更能使程序具有稳定性。它将数据和对数据的操作封装到一起，这种作为一个整体进行处理并且采用数据抽象和信息隐藏技术最终被抽象成一种新的数据类型——类。类与类之间的联系以及类的重用使得类出现了继承、多态等特性。类的集成度越高，越适合大型应用程序的开发。另外，面向对象程序的控制流程运行时是由事件进行驱动的，而不再由预定的顺序进行执行。事件驱动程序的执行围绕消息的产生与处理，靠消息的循环机制来实现。更加重要的是，可以利用不断成熟的各种框架，如.NET的.NET Framework 等，在实际的编程过程中迅速地将程序构建起来。面向对象的程序设计方法还能够使程序的结构清晰简单，从而大大提高代码的重用性、有效地减少程序的维护量、提高软件的开发效率。

在结构上，面向对象程序设计和结构化程序设计也有很大不同。结构化程序设计首先应该确定的是程序的流程怎么走、函数间的调用关系怎么样、函数间的依赖关系是什么。一个主函数依赖于其子函数，子函数又依赖于更小的子函数，而在程序中，越小的函数处理的往往是细节实现，具体的实现又常常变化。这种变化的结果就是程序的核心逻辑依赖于外延的细节，程序中本来应该是比较稳定的核心逻辑，也因为依赖于易变化的部分而变得不稳定起来，一个细节上的小改动也有可能在依赖关系上引发一系列变动。可以说这种依赖关系也是过程式设计不能很好处理变化的原因之一，而一个合理的依赖关系应该由细节实现依赖于核心逻辑。面向对象程序设计由类的定义和类的使用两部分组成，主程序中

定义对象并规定它们之间消息传递的方式，程序中的一切操作都是通过面向对象的发送消息机制来实现的。对象接收到消息后，启动消息处理函数完成相应的操作。

仍然以图书管理系统为例，使用结构化程序设计的方法时，首先需要在主函数中确定图书管理系统要做哪些事情，并使用函数将这些事情进行表示、使用一个分支选择程序进行选择，然后将这些函数进行细化实现，并确定调用的流程等。使用面向对象技术来实现图书管理系统时，以学生为例，要了解图书管理系统中学生的主要属性（如学号、院系等）、学生要进行什么操作（如借书、还书）等，并且把这些当成一个整体进行对待，形成一个类，即学生类。使用这个类可以创建不同的学生实例，即创建许多具体的学生模型，每个学生拥有不同的学号，都可以在图书馆借书和还书。学生类中的数据和操作都是可共享的，可以在学生类的基础上派生出大学生类、大专生类、研究生类等，从而实现代码的重用。

类与对象是面向对象程序设计中最基本和最重要的概念，也是创建和使用 UML 图的基础，有必要仔细理解和掌握，并且在学习中不断深化。

1.1.3 对象与类的确定

面向对象技术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，不同对象之间的相互联系和作用构成了各种系统。在面向对象程序设计中，系统被描绘成由一系列完全自治、封装的对象组成，对象和对象之间是通过对象暴露在外的接口进行调用的。对象是组成系统的基本单元，是一个组织形式的含有信息的实体，如一个叫张三的人，“人”作为实体对象，包含姓名为“张三”的信息。而类是创建对象的模板，在整体上可以代表一组对象，如创建“人”此类，它就代表人这个概念，可以使用此类来表达张三、李四等。设计类而不是设计对象，这样可以避免重复编码，类只需要编码一次就可以被实例化这个类的对象。

对象（Object）是由状态（Attribute）和行为（Behavior）构成的。事实上，状态（Attribute）、属性（Property）、状态（State）、数据（Data）这些在各种书中提到的概念，意思都相似，是用于描述一个对象的数据元素，这些概念具体到各种语言便有不同的叫法。对象的状态（Attribute）值用来定义对象的状态，如当判断学生是否能够借书时，可以利用学生的学号（称为第一个状态）和学生目前的借书数量（称为第二个状态）进行判断。行为（Behavior）、操作（Operation）以及方法（Method）这些在各种书中提到的概念，是用于描述访问对象的数据或修改（维护）数据值的方法，如“告诉图书管理员你的学号”和“选择需要借阅的图书”等。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，行为用于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用“对象=数据+数据的操作”来进行表达。

类（Class）是具有相同属性和操作的一组对象的组合，即抽象模型中的“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，如学生类被用于描述能够到图书馆借阅图书的借阅的学生对象。

类的定义需要包含以下要素：

- 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，即类的对象要被调用执行哪些操作以及进行这些操作时对象还要执行哪些操作，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——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。类用来在内存中开辟一个数据区，用于存储新对象的属性或把一系列行为和对象关联起来。

一个对象又被称作类的一个实例，即实体化（Instantiation）。实体化（Instantiation）是指对象在类声明的基础上创建的过程，如声明了一个学生类，可以在这个基础上创建一个姓名叫张三的学生对象。

类的确定和划分没有一个统一的标准和方法，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。确定一个类的步骤通常包含以下方面：

- 确定系统的范围，如图书管理系统需要确定和图书管理相关的内容。
- 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，如图书管理系统中，计算机系有一个名叫张三的学生，英语系名叫李四的人和张三一样都是学生。
- 将对象抽象成为一个类，按照上面的类的定义确定类的数据和操作。

在面向对象的程序设计中，类和对象的确定非常重要，是软件开发的第一步，软件开发中类和对象的确定将直接影响软件的质量。如果划分得当，对于软件的维护、扩充以及软件的重用性方面都有很大帮助。

1.1.4 消息和事件

当使用某一个系统时，单击一个按钮后通常会显示相应的信息，以图书管理系统为例，单击图书管理系统界面的某一个按钮时会显示出当前的图书信息。那么当前的程序是如何运行的呢？

01 图书管理系统界面的某一个按钮发送鼠标单击事件的消息给相应的对象。
02 对象接收到消息后有所反应，它提供了图书的相关信息给界面。
03 界面将图书的相关信息显示出来，从而完成任务。

在这个过程中首先要触发一个事件，然后发送消息，那么消息是什么呢？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作用的方式。一个消息主要由5部分组成：消息的发送对象、消息的接收对象、消息的传递方式、消息的内容（参数）、消息的返回。传入消息内容的目的有两个，一个是让接受请求的对象获取执行任务的相关信息，另一个是行为指令。