高等院校计算机教育系列教材

面向对象程序设计 (C#)

廖浩德 主编

杨 力 向海昀 杨 云 汪立欣 副主编

张豫新 符 晓 王世元 高 磊

扫描书中二维码
可观看讲课视频
（全书共77段微视频，时长520分钟）

赠送实例代码、电子课件
和课后习题答案

清华大学出版社

高等院校计算机教育系列教材

面向对象程序设计(C#)

廖浩德 主 编

杨 力 向海昀 杨 云 汪立欣
张豫新 符 晓 王世元 高 磊

副主编

清华大学出版社
北 京

内 容 简 介

面向对象程序设计范式具有封装、继承、多态等特点，能显著提高程序的可重用性和可扩展性，是现代开发大型应用软件的主要技术。掌握面向对象软件开发方法，可大幅度提高复杂软件系统的生产率和质量。本书用C#语言和.NET框架技术描述并介绍了面向对象程序设计的核心概念、基本原理、基本技术和方法，内容涉及变量、数据类型、运算符、程序流程控制等基础程序设计，类、对象、封装、继承、多态、接口等面向对象程序设计，数值、文字、集合、泛型、委托、事件、控件、图形、文件、数据库等实用化程序设计，重点培养读者用面向对象程序设计范式解决实际问题的能力。

全书共分9章。第1章介绍面向对象技术的由来、地位及其重要性。第2章从计算机的角度介绍程序设计基础，涉及变量机制和过程式程序设计思想。第3章从人的角度介绍高端程序设计，涉及分类机制和面向对象程序设计思想，重点解析抽象、封装、继承、多态、接口等概念及其实现机制。第4章对比分析过程式、面向对象、面向接口、组件化等程序设计范式的应用，体验利用面向对象思想进行程序设计所带来的好处。从第5章开始，按软件分层体系结构，介绍用户界面层、业务逻辑层、数据访问层的实现技术。其中，第5章涉及业务逻辑层技术，介绍科学计算、文字处理、时间、事件等常见数据结构类的使用。第6章涉及用户界面层技术，介绍各种控件类的使用。第7章涉及数据访问层技术，介绍文件和数据库类的使用。第8章涉及数据的可视化技术，介绍图形、图像、动画等多媒体类的使用。第9章基于企业信息化目标，用一个管理信息系统原型的实现过程介绍面向对象技术的综合运用。

本书思路新颖、图文并茂，适用于计算机类专业(包括但不限于计算机科学与技术、软件工程、网络工程、信息安全、物联网工程等)的面向对象程序设计、桌面应用软件开发等课程教学，也可供从事软件开发的科研人员使用。

图书在版编目(CIP)数据

面向对象程序设计(C#)/廖浩德主编. —北京：清华大学出版社，2018
(高等院校计算机教育系列教材)
ISBN 978-7-302-50798-7

Ⅰ. ①面…　Ⅱ. ①廖…　Ⅲ. ①C++语言—程序统计—高等学校—教材　Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2018)第179041号

责任编辑：姚　娜　刘秀青
装帧设计：李　坤
责任校对：吴春华
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
课件下载：http://www.tup.com.cn, 010-62791865
印 装 者：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**14.25　　**字　数：**346千字
版　　次：2018年9月第1版　　**印　次：**2018年9月第1次印刷
定　　价：39.80元

产品编号：079756-01

前　言

镰刀、锄头等第一代人力工具可把物质资源加工成材料，扩展了人的体质功能，孕育了农业时代的生产力，创建了农业文明。机车、机床等第二代动力工具可把能量资源转换成为动力，扩展了人的体力功能，形成了工业时代的生产力，建立了工业文明。

20 世纪后半叶，人类开始认识到信息也可以作为一种资源，甚至是更为重要的资源。综合利用物质材料、能源动力和信息知识，可制造新一代既有活力又有智能的生产工具。第三代生产工具用于扩展人类的智力功能，从而培育出信息时代的生产力，把工业文明进一步升华为更加辉煌的信息文明。

为迎接信息社会的来临，以信息化带动工业化，以工业化促进信息化，走出一条科技含量高、经济效益好、资源消耗低、环境污染少、人力资源优势得到充分发挥的新型工业化道路，是世界各国现代化的必然选择。

在引领时代的软件行业，软件工程师始终是最为紧俏的科研人才。当今软件开发人才的培养速度难以企及软件行业的发展，主要在于对程序设计的片面理解和传统的教育模式。随着软件技术的发展，企业对软件人才的需求不再呈现金字塔式的结构。现在，许多初级程序设计工作更多的是使用自动化工具完成，程序设计的门槛已经降低。在人才培养上，过多地强调程序设计语言的语法式教学或过细地解析 API 的列表式培训已经不合时宜，难以有效地培养合格的软件工程师。

众所周知，作为第三代智能工具的典型代表，计算机的主要功能是实现计算的自动化，涉及计算的对象(数据)和计算的过程(算法)。数据和算法用程序来描述，计算自动化的核心任务就是程序设计。除了数据和算法，程序设计还涉及程序设计语言、计算环境、程序设计范式等多个方面。程序设计类的教材，有的突出程序设计语言，有的偏重程序设计工具，难以将程序设计所涉及的方方面面有效结合起来。本书以面向对象范式为主线，将程序设计语言、工具库和方法学等有机“串接”起来，注重文化传承，中西结合，以及现实世界与机器世界的关联，旨在培养深刻理解程序设计核心概念、基本原理，掌握实用程序设计技术和方法，具备自主学习和终身学习的意识，具有不断学习、适应发展、能解决实际应用问题的能力的实用型软件工程师。

面向对象程序设计范式具有封装、继承、多态等优点，能显著提高程序的可重用性和可扩展性，是现代开发大型应用软件的主要技术。支持面向对象程序设计范式的程序设计语言有很多，如 C++、Java、C#等。20 世纪 80 年代以来，C/C++一直是使用最为广泛的商业化程序设计语言。高校计算机相关专业普遍开设有面向对象程序设计类课程，使用的教材一般是用 C++进行描述的。但是，C++过度的功能扩张破坏了面向对象的设计理念，而且学习周期长，开发效率低，软件行业迫切需要一种能在控制力和生产率之间达到良好平衡的全新程序设计语言。因此，C++已经难以适应行业和高校的教学要求。C#是一种简单、现代、通用、完全面向对象的程序设计语言。它从 C/C++发展而来，汲取了 C/C++、Delphi、Java 等多种语言的精华，具有语法简洁、与 Internet 结合紧密、安全高效、灵活兼容等优点。

C#语言简洁易懂，更适合高校和培训机构传授面向对象设计理念和技术。从C#入手，可以更容易体验和感悟现代化程序设计方法和技术，掌握可重用面向对象软件的开发方法，大幅度提高复杂软件系统的生产率和质量。本书是我校“面向对象程序设计”精品资源共享课程教改研究的结晶，用C#语言描述和介绍面向对象程序设计范式，思路新颖、图文并茂，不仅适用于本科院校的学生，也可作为各类培训班学生面向对象程序设计或桌面应用开发类课程的首选学习用书。

本书作者是具有软件开发和项目管理经验的大学教师。作为国家注册的高级程序员，在软件企业长期从事软件开发、程序设计、技术培训等工作，开发过多项软件系统。从教后，主讲计算机科学基础、面向对象程序设计、软件工程、程序设计范式、软件设计模式、软件项目管理等多门课程，对软件工程、程序设计、技术培训、专业教育等有着深刻的理解和丰富的实践经验。本书是作者教学和培训经验的积累，具有如下特色。

(1) 概念探源：计算机科学知识源于欧美国家，从源头梳理概念可以帮助读者把握知识发展脉络，为跟踪学习先进技术指引方向，培养技术研究能力和终身学习意识。本书的大部分核心概念都从 Wikipedia 指出出处，对一些容易引起混淆的概念，都针对原文进行了详细解析。我国计算机相关术语来自英文资料，在引进时可能会遇到翻译障碍。例如，C 语言的“函数”由 function 翻译而来，而“函数”术语本身是由清朝数学家李善兰翻译而来。但从程序设计角度，function 译为“功能模块”或“过程模块”也许更便于理解。本书的概念探源试图引导学生从概念入手逐步加深对程序设计语言实现机理的理解，进而掌握程序设计技术和方法。

(2) 注重思想：一种程序设计语言可以体现多种范式，如C#语言既支持过程式，也支持面向对象、组件化等思想；一种范式也可以在多种程序设计语言中体现，如C++、Java、C#等语言都支持面向对象程序设计范式。每门语言都有各自的特点及难点。针对不同的任务，应该用不同的语言实现。同一个任务，用同一种语言实现，不同的方法会有不同的效率。本书解析了用不同思想解决同一问题的优缺点，以加深对面向对象程序设计范式的理解。书中还适当点缀中国文化思想，在增强趣味性的同时，对于中西方文化的结合和传承也有一定的启示意义。

(3) 分层递进：从基础级的变量与过程到对象级的封装、继承与多态，从模式级的委托与事件到实用级的集合与泛型，从应用级的图形处理、文件存储、数据库访问到企业级的复杂软件项目开发，逐层递进，分类学习。本书前半部分(第 1～4 章)以概念及 C#语言实现机理为主，强调计算机与现实之间的关系；后半部分(第 5～9 章)以应用.NET 框架类为主，强调程序设计的实用性。

(4) 案例驱动：本书所涉及的主要概念都以完整的案例加以说明，与现实紧密结合，避免了技术的枯燥性，增强了实用性和趣味性。第 6～8 章用一个完整的案例串接起来形成一个有机的整体，为实现多层应用程序打下基础。第 9 章以企业信息化为目标，实现了一个基于分层软件体系结构的管理信息系统的原型。以此案例作为软件开发能力构建的目标，可有的放矢地驱动学习的进程。

另外，本书还为重要的知识点配备了全程板书式授课的教学微视频，可用于 MOOC 模式的教学或自学。

在本书的编写过程中，参考了很多国内外同行的有关资料，西南石油大学计算机科学

学院的廖浩德、杨力、杨云、高磊、王世元，现代教育中心的向海昀、汪立欣，教务处的符晓等教师参加了写作思路的研讨、收集资料、编写和程序调试等工作。张豫新全程负责教材的编写和出版事宜，包括案例设计、文字录入、图形绘制、内容合成和编辑审校等。西南石油大学教务处、教材科、计算机科学学院和理学院等部门的领导、工作人员和教师多年来对作者始终给予了热情的支持和鼓励。清华大学出版社对本书的出版十分重视并做了周到的安排，使本书得以在短时间内顺利出版。在此向他们表示诚挚的谢意。

由于作者水平有限，疏漏之处在所难免，敬请广大读者批评指正。

编　者

目　录

第1章 概 述

1.1 面向对象探源

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.

——摘自 https://en.wikipedia.org/wiki/Object-oriented_programming

开宗明义，概念先行。这段摘自 Wikipedia 的英文原文介绍的 Object-oriented programming 在我国大陆译为“面向对象程序设计”。这个术语一般用其缩写 OOP 简称，由来已久，早已响彻业界。后续章节将以这个定义为主线，围绕相关概念展开学习和讨论。本节介绍与此相关的行业大背景，追根溯源，了解面向对象概念、.NET 平台，以及 C#语言的来龙去脉，为深入学习相关理论和技术做好准备。

1.1.1 关于计算

说到计算，人们并不陌生。形容一个人“精于计算”，一般是指其数学功底深厚。这里的计算(computing)，特指与计算机相关的目标导向活动，包括计算机软硬件系统的设计和建造、信息的采集和处理、通信和娱乐媒体的创建与使用，以及用计算机进行科学研究等。简而言之，这里的计算是计算机设计和使用的研究，包括理论、实验和工程等。计算机的主要功能是实现计算的自动化，涉及计算的对象(data，即数据)和计算的过程(algorithm，即算法)。数据和算法用程序(program)来描述，计算自动化的核心任务就是程序设计(programming)。

随着计算理论的日渐成熟和计算系统的飞速发展，计算科学已划分成许多理论和实践领域。从工程角度看，计算机硬件制造和软件开发各自发展，形成了计算机工程和软件工程两大独立的学科。计算机硬件和软件产品集成起来，可应用于不同的领域，形成各种各样的信息系统(相关技术统称为信息技术)。当然，不管怎么演化和划分，程序设计都是最为基本的活动，是各计算学科都不可或缺的内容。计算机科学的发展如图 1-1 所示。

程序由描述计算对象的数据结构和描述计算过程的算法构成，程序设计还涉及表示程序的语言(即程序设计语言)和运行程序的平台(即计算环境)。就程序运行平台来说，从早期基于单机的主机计算到后来基于 Internet 的网络分布计算，计算环境发生着深刻的变革。只有真正了解计算环境的这种翻天覆地的变化，才有可能理解新一代程序设计语言所提供的机制和特性，并快速掌握这些更为先进的程序设计技术和方法。

计算机科学的发展.mp4

图 1-1　计算机科学的发展

1.1.2　主机计算

现代计算机遵循的是匈牙利数学家约翰·冯·诺依曼(John von Neumann)于 1945 提出的体系结构，如图 1-2 所示。这种体系结构由中央处理器(Central Processing Unit，CPU)、存储器(Memory Unit)、输入设备(Input Device)和输出设备(Input Device)构成。其中，CPU 又由算术/逻辑运算器(Arithmetic/Logic Unit)、处理器寄存器、含有指令寄存器和程序计数器的控制器(Control Unit)构成。由于存储器用于存储数据和指令，这种体系结构的计算机又称为存储程序(stored-program)计算机。

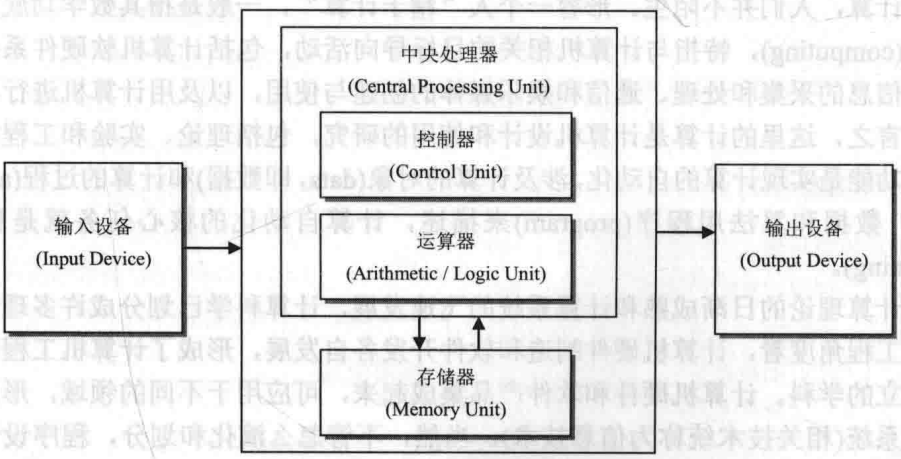

图 1-2　冯·诺依曼体系结构

基于冯·诺依曼体系结构的计算系统(computing system)由硬件(hardware)和软件(software)组成，如图 1-3 所示。硬件是构成计算系统的物理部件的集合，都是有形的物体，包括主机和外部设备两大部分。主机的核心是一块集成电路主板，用于连接计算机的其他部分，包括 CPU、存储器、磁盘驱动器(如 CD、DVD、硬盘等)。主机可以通过主板上的端口或扩展槽连接外部设备，如显示器、键盘、打印机、音箱、麦克风等。软件是能够被

硬件存储和执行的指令的集合，一般分为系统软件和应用软件两个部分。系统软件包括设备驱动程序、操作系统和一些用于计算机维护的实用工具软件；应用软件则泛指系统软件之外的所有软件。

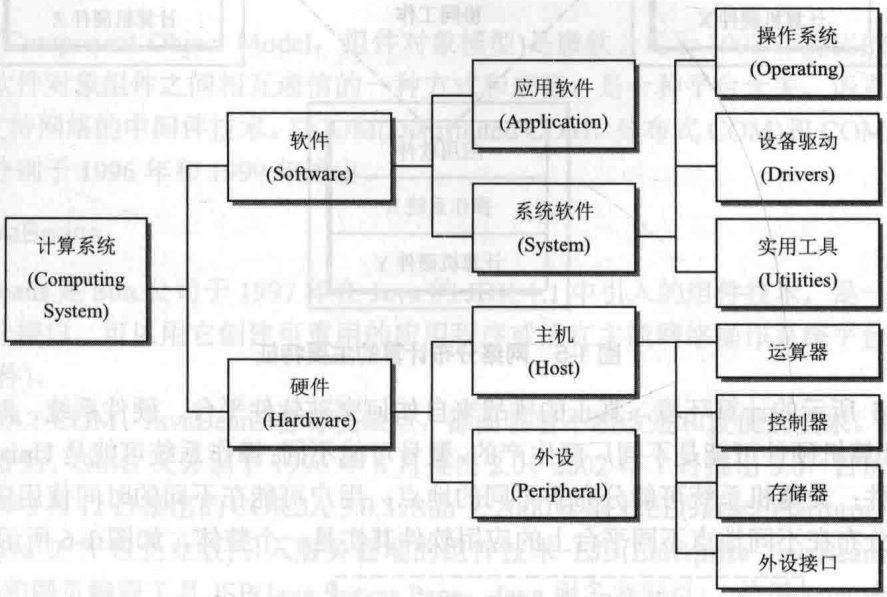

图 1-3　计算系统组成

计算环境是指运行应用程序的平台，包括硬件平台和软件平台，如图 1-4 所示。

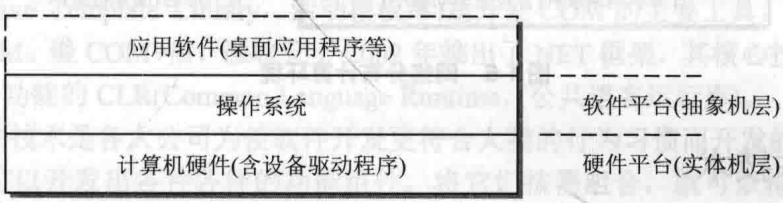

图 1-4　主机计算环境

在图 1-4 中，底层是计算机实体层，可以是各种品牌和类型的计算机。硬件之上是操作系统，用于管理计算机的软硬件资源并提供公共服务，就像一台扩展了硬件功能的虚拟机，一般视为抽象机层。抽象机也可以有多层，如运行 Java 程序的 JVM(Java Virtual Machine)、运行 C#程序的 CLR(Common Language Runtime)等都属于虚拟机。这里的主机计算是指基于单台计算机的程序运行环境，对应的程序设计相对比较简单。

1.1.3　网络分布计算

网络分布计算.mp4

Internet 出现后，随着网络应用需求的飞速增长，网络分布计算逐渐成为新一代计算和应用的主流。这时的计算涉及主机之间的资源共享和协同工作，如图 1-5 所示。

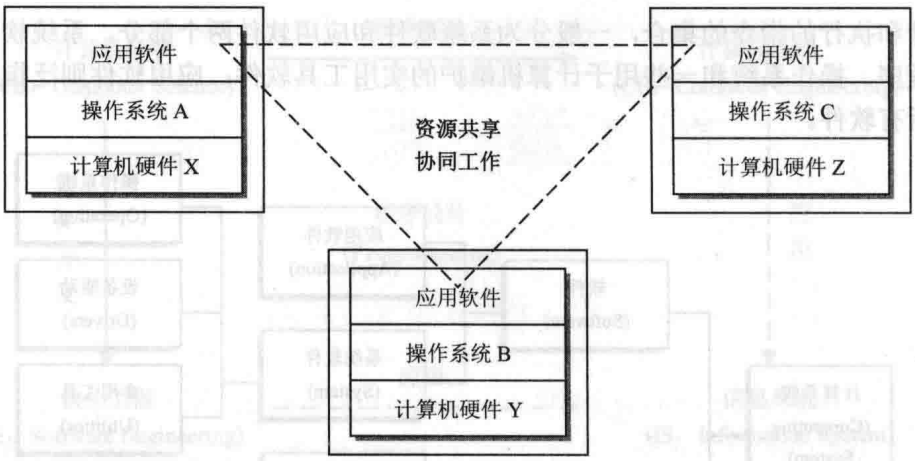

图 1-5　网络分布计算的主要特征

图 1-5 所示的计算环境，真正的挑战来自如何突破软件平台、硬件系统、时间、地点的限制。计算机硬件可能是不同厂商生产的，型号可能不同；操作系统可能是 Unix、Windows 等不同软件；计算机系统可能分布于不同的地点；用户可能在不同的时间使用应用软件。图中看似分布在不同地点不同平台上的应用软件其实是一个整体，如图 1-6 所示。

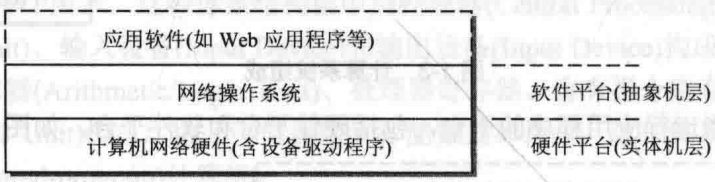

图 1-6　网络分布计算环境

1.1.4　组件技术

组件技术.mp4

在从主机计算向网络分布计算过渡的过程中，软件系统的规模和复杂度呈几何级数增加，程序设计语言和方法都面临着前所未有的挑战。

面对规模越来越大的软件，为了降低复杂度，提高开发效率，人们提出了组件式程序设计方法。组件式方法是“搭积木”思想在程序设计领域的开拓性应用。因为组件(积木)具有可重用性和互操作性，可以通过组件集成来高效地构建复杂的软件系统。

20 世纪 90 年代以来，出现了三种典型的组件技术，即 CORBA、COM、JavaBeans。

1. CORBA

CORBA (Common Object Request Broker Architecture，公共对象请求代理体系结构)是 OMG(Object Management Group，对象管理组织)于 1991 推出的组件技术。OMG 于 1989 年由 3COM、Apple、美国航空、佳能、DG、HP、IBM、Philips、Unisys 和 Sun 等多家公司联合创建，是一个开放型非营利组织，负责制定和维护协同企业应用的计算机工业规范。发展至今，已有八百多家公司、大学和国际组织参与其中。OMG 制定的其他标准还有

UML(Unified Modeling Language，统一建模语言)和 IDL(Interface Definition Language，接口定义语言)等。

2. COM/DCOM/COM+

COM(Component Object Model，组件对象模型)是微软公司于 1993 年提出的一种组件技术，是软件对象组件之间相互通信的一种方式和规范，是一种平台无关、语言中立、位置透明、支持网络的中间件技术。DCOM(Distributed COM，分布式 COM)和 COM+是 COM 的发展，分别于 1996 年和 1999 年推出。

3. JavaBeans

JavaBeans 是 Sun 公司于 1997 年在 Java 的 JDK 1.1 中引入的组件技术，是一个面向对象程序设计接口，可以用它创建可重用的应用程序或能在主流网络操作系统平台配置的程序模块(组件)。

CORBA、COM、JavaBeans 各有优缺点，都面临着不断改进和发展的要求。例如， 继 CORBA 1.0 后，OMG 又分别于 1996 年 8 月推出 2.0、2002 年 7 月推出 3.0，目前的最新标准是 2004 年 3 月 12 日推出的 CORBA 3.0.3。Sun 于 2000 年随 J2EE(Java 2 Platform, Enterprise Edition，Java 2 平台企业版)引入服务器端的组件技术 EJB(Enterprise JavaBeans，企业级 JavaBeans)和网页编程工具 JSP(Java Server Page，Java 服务器网页)，使得 Java 成为一种功能完备的分布式计算环境。COM 源自 OLE(Object Linking and Embedding，对象链接和嵌入)，OLE 源自 DLL(Dynamic Link Libraries，动态链接库)，ActiveX 控件是 COM 的具体应用，ATL(Active Template Library，活动模板库)是开发 COM 的主要工具，也可以用 MFC 直接开发 COM。继 COM+后，微软又于 2002 年推出了.NET 框架，其核心技术就是用来代替 COM 组件功能的 CLR(Common Language Runtime，公共语言运行库)。

这些组件技术是各大公司为使软件开发更符合人类的行为习惯而开发的新技术。利用这些技术，可以开发出各种各样的功能组件，将它们按需组合，就可以构成复杂的应用系统。

这样做不仅能提高软件定制的效率和软件产品的质量，也使得软件系统易于升级和维护。例如可以“现场”替换软件系统中的组件、可以在多个软件系统中重用同一个组件、可以方便地将组件部署到分布式网络环境等。

CORBA、COM、JavaBeans 等组件技术都与面向对象技术密切相关。要掌握组件式程序设计方法，面向对象技术是关键。

1.1.5 面向对象技术

面向对象核心概念.mp4

现在回到本章开篇主题“面向对象程序设计”，中英对照如下：

Object-oriented programming (OOP) is a programming paradigm
面向对象程序设计(OOP)是一种程序设计范式，
based on the concept of “objects”,
它基于“对象”概念，

which may contain
对象可以包含：
data, in the form of fields, often known as attributes;
数据，以字段的形式体现，常称为属性；
and code, in the form of procedures, often known as methods.
代码，以过程的形式体现，常称为方法。


类的世界.mp4

“对象”，也许是学习程序设计技术的路上最易明白的概念了。我们看到的每个从身边走过的人，就是一个个具体的对象。对象有其自身的特性，如年龄、身高等，也有其自身的行为，如走路、微笑等。对应到程序设计，由于派别或翻译等原因，很容易把这些原本简单的概念弄混淆了。这涉及三个“世界”的术语转换问题，如图 1-7 所示。

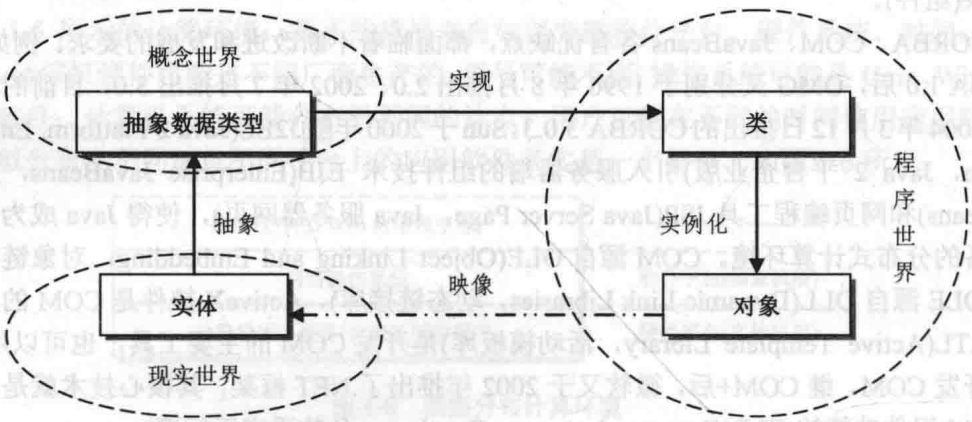

图 1-7　三个世界的变换

首先是从现实世界向概念世界过渡。例如，现实世界中的一个人事部门，有许多实实在在的员工“实体”。要对这些员工进行有效的管理，需要对他们进行了解。人事部门经理分析这些员工，在大脑(概念世界)中对员工信息进行抽象，形成了自己的看法(关注点)，建立了信息模型(即抽象数据类型)，如图 1-8 所示。

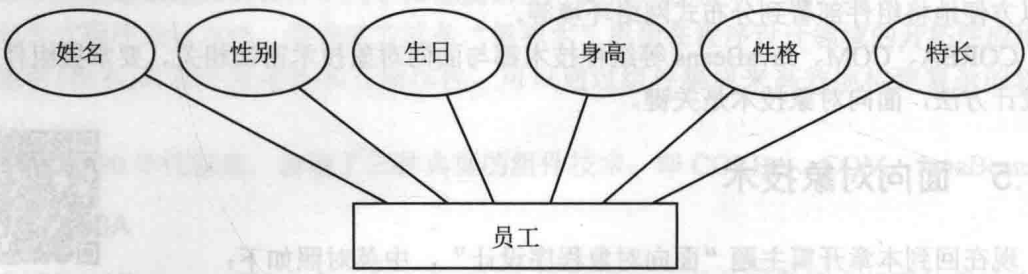

图 1-8　员工信息

当然也要关注他们的行为表现，如签到、写作、说话等。

作为软件工程师，要为这个部门经理开发一套人事管理软件，就得把部门经理的概念模型转换成程序世界的“类”，如图 1-9 所示。

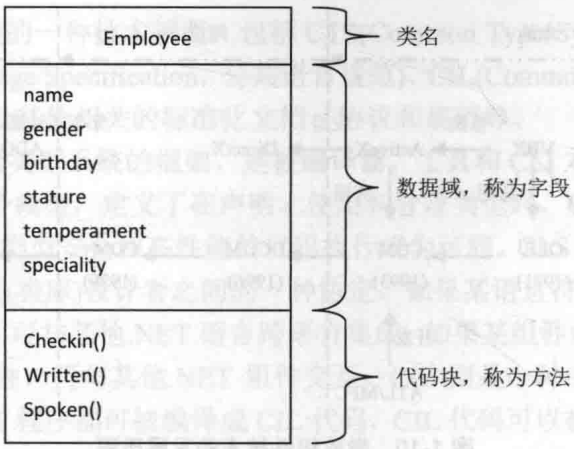

图 1-9　员工类

程序世界的类相当于一个模板，利用这个模板可以创建具体的“对象”。例如：

```
Employee emp = new Employee();   //用 Employee 类实例化一个 emp 对象
emp.name = "张三";               //emp 对象映射了现实世界中张三这个员工
```

本书第 2 章、第 3 章将具体介绍数据字段的表示、代码过程的实现。

1.2　.NET 框架

微软公司于 2000 年 6 月推出了用来代替 COM 的.NET。这是微软面向第三代 Internet 的计算计划，是微软继用 Windows 取代 DOS 之后的又一项战略性举措。.NET 是一个分布式计算环境，提供了一个安全、一致、标准的模型和环境，简化了分布式应用程序开发的难度，能大幅度地提高软件系统的生产率和质量。它面向异构硬件平台、操作系统和网络，为软件提供最大限度的可重用性、互操作性和可扩展性，以实现软件系统之间的智能交互和协同工作，提高整个网络的利用率和效率，特别是企业级的系统集成和资源优化，给开放性企业的生产力水平带来质的飞跃。目前，.NET 已成为 Windows 应用和 Web 应用的主流开发模型。

1.2.1　微软技术的发展

.NET 技术.mp4

微软技术的发展路线如图 1-10 所示。

20 世纪 90 年代末，使用 Microsoft 平台的 Windows 程序设计演化出了许多分支：大多数程序员使用的是 Visual Basic、C 或 C++，使用 C 和 C++的程序员中，有的使用 Win32 API(Application Programming Interface，应用程序设计接口)，有的使用 MFC(Microsoft Foundation Classes，微软基础类库)，有的程序员已经转向 COM。

这些技术都有自身的问题。例如，Win32 API 不是面向对象的，使用它比使用 MFC 需要更多的工作量；MFC 是面向对象的，但缺乏一致性；COM 概念简单，但实际编码很复杂且代码也较难阅读。况且，这些程序设计技术主要针对的是桌面应用开发，对 Internet 则显得力不从心。

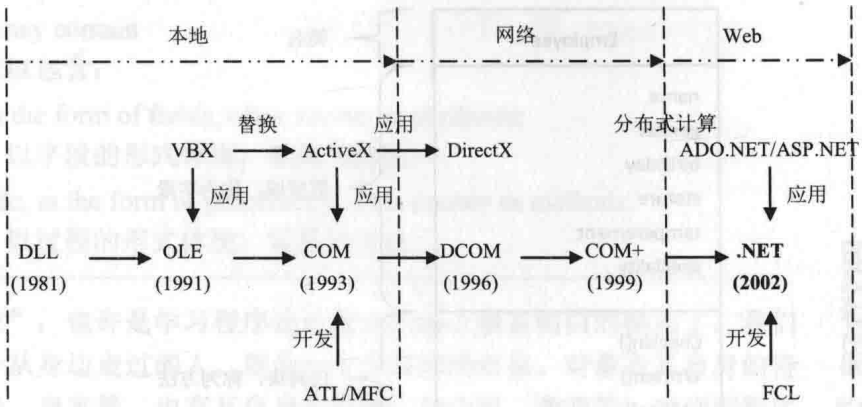

图 1-10　微软组件技术的发展历程

早期的程序代码短小精悍。随着问题规模的越来越大，程序代码也越来越复杂。难以阅读的程序代码必然会给开发和维护带来困难。于是，程序员开始重温那“激情燃烧的岁月”，希望用一种集成的、面向对象的开发框架把一致性和优雅性带回到程序中，回归到代码简洁的时代。为此，他们对下一代计算平台提出了新的要求，希望达到以下目标。

1. 运行环境(Execution Environment)

(1) 安全(Security)；
(2) 多平台(Multiple Platforms)；
(3) 性能(Performance)。

2. 开发环境(Development Environment)

(1) 面向对象的开发环境(Object-Oriented Development Environment)；
(2) 一致的程序设计体验(Consistent Programming Experience)；
(3) 用行业标准进行沟通(Communication Using Industry Standards)；
(4) 简化开发(Simplified Development)；
(5) 语言独立(Language Independence)；
(6) 互操作(Interoperability)。

为了满足这些需求，微软公司开始开发一个能满足这些目标的代码运行环境和应用开发环境，这就是.NET。

.NET 将 Internet 作为构建新一代操作系统的基础，在理念中包含了对操作系统和网络设计思想的延伸。微软计划用.NET 彻底改变软件的开发、发行和使用方式，构建第三代 Internet 平台，解决各种协同合作的问题，实现信息的高效沟通和分享，让整个 Internet 为人们提供全方位的服务。

1.2.2　.NET 规范及其实现

.NET 规范.mp4

微软为.NET 技术制定了一套完整的规范 CLI(Common Language Infrastructure，公共语言基础结构)。CLI 是针对可执行代码格式，以及能执