

电子工业部“九五”规划图书

面向对象原理与应用

朱海滨 阳国贵 刘 仲 编著



Java

Smalltalk

国防科技大学出版社

TP311

428510

Z87-2

面向对象原理与应用

朱海滨 阳国贵 刘 仲 编著

国防科技大学出版社
· 长沙 ·

428840

图书在版编目(CIP)数据

面向对象原理与应用/朱海滨, 阳国贵, 刘仲. —长沙: ~~国防科技大学~~出版社,
1998.10

ISBN 7-81024-500-7

- I 面向对象原理与应用
- II 朱海滨 阳国贵 刘 仲
- III 面向对象
- IV TP311

国防科技大学出版社出版发行
电话: (0731)4555681 邮政编码: 410073
E-mail: gfkdcbs@public.cs.hn.cn
责任编辑: 何 晋 责任校对: 黄 煌
新华书店总店北京发行所经销
国防科技大学印刷厂印装

*

开本: 787×1092 1/16 印张: 20.75 字数: 479千
1998年10月第1版第1次印刷 印数: 1—3000册

*

定价: 25.00元

内 容 简 介

本书从原理、设计到应用由浅入深、循序渐进地展开对面向对象技术的讨论，是一部全面讨论面向对象程序设计技术和基本原理的文献。

本书分为两大部分，第一部分为一至五章，介绍面向对象技术的基本概念和基本原理，讲述面向对象程序设计的一般性概念和原则，并就基础性理论和方法做了一些探索；然后，对微软公司的OLE/COM标准给出了较为全面的介绍。第二部分为六至十章，比较详细地描述了Smalltalk语言、C++语言和Java语言的体系结构、面向对象机制和特点，并对三种语言进行了比较；讨论了面向对象数据库的原理、设计和实现；最后，介绍了面向对象技术的几个最新的典型应用，包括网上对象、面向对象操作系统、CORBA、DCOM和UML。

本书可作为大专院校计算机各专业高年级学生及研究生面向对象技术和面向对象程序设计课程的教材或参考书，也可供从事计算机相关领域研究、设计、应用和开发的工作者参考。

前 言

20世纪80年代以来,面向对象技术的研究不断深入,应用也日益广泛,从系统和语言设计、系统和语言的实现,到应用的各个领域都进行了深入的研究。面向对象已经被公认为一种非常有效的方法和技术,并应用到计算机领域的各个方面。

面向对象技术现已从原理讨论走向全方位应用。可以这样说,计算机及其相关专业的学生如果不懂得面向对象技术,将难立足于计算机研究和应用领域。所以,许多国外著名大学从20世纪80年代便开设了相应课程,我国各大学从80年代末开始也相继开设了类似课程,并出版了相关教材。但由于该领域的研究工作十分活跃,技术发展日新月异,教材也必须得到相应的更新,否则将无法适应技术的发展。1992年,我们编著出版了《面向对象技术——原理与设计》,受到广大读者的厚爱,1996年获电子工业部优秀教材二等奖。本书也被列入电子工业部“九五”规划教材。通过几年的教学和科研实践,我们感到原有教材在内容上需要大量更新,特别是应用方面更需要大量的补充和修改。本书就是本着不断更新知识的宗旨,参考国内外最新教材和国内外最新研究成果新编而成的。

本书假设读者已经具备计算机操作系统、编译原理和程序设计等基本知识。当然,如果没有上述这些知识,本书也给出了必要的知识铺垫。

本书分为两大部分。第一部分为一至五章,介绍面向对象技术的基本概念和基本原理;讨论了面向对象分析和设计的基本方法(Coad/Yourdon方法和Booch方法)及其工具支持;然后,对微软公司的OLE/COM标准给出了较为全面的介绍。第二篇为六至十章,较详细地描述了Smalltalk语言、C++语言和Java语言的体系结构、面向对象机制和特点,并对三种语言进行了比较;讨论了面向对象数据库的原理、设计和实现;最后,介绍了面向对象技术的几个最新的典型应用,包括网上对象、面向对象操作系统、CORBA、DCOM和UML。

本书重点讨论了为什么应用面向对象技术、怎样进行面向对象程序设计以及如何使用面向对象语言,并讨论了面向对象相关标准和应用。通过对本书的学习,可以使学生掌握面向对象技术的基本原理和方法;熟悉目前应用最为广泛的三种面向对象语言的机制、特征和应用中值得注意的问题;比较深入地了解目前面向对象技术的重要应用领域。

本书可作为大专院校计算机各专业高年级学生及研究生面向对象技术和面向对象程序设计课程的教材或参考书,也可供从事计算机相关领域研究、设计、应用和开发的工作者参考。

本书第一、二、三、四、六、七、八、十章由朱海滨执笔,第九章由阳国贵执笔,第五章由刘仲执笔,全书由朱海滨统稿,刘仲还作了大量校阅工作。

本书出版之际,再次向带我入门面向对象领域的陈火旺院士表示最诚挚的感谢。感谢

硕士生宋辉、钱方、沈立强等同学提供的最新应用资料，同时感谢多年来选修《面向对象技术》课程的所有硕士生同学们，在每次给他们授课的同时，我都会得到许多新的感受和认识。还要感谢我的妻子张晶的支持与鼓励。

最后，敬请读者不吝指教，以便进一步改进。

朱海滨

1998年5月于长沙

目 录

第一章 导论	1
1.1 面向对象技术的发展	1
1.2 面向对象技术的特点	2
1.3 面向对象方法学	3
1.4 面向对象语言及系统	4
1.5 面向对象设计方法与其它设计方法的比较	5
1.6 面向对象与大型程序设计	7
1.7 面向对象方法学基础	9
1.7.1 程序设计方法学	9
1.7.2 面向对象语言与系统规范描述	10
1.7.3 数据抽象	11
1.7.4 抽象数据类型	13
1.7.5 面向对象设计与表示抽象	15
1.7.6 面向对象设计与基于状态的数据类型	17
习题	18
第二章 基本概念	19
2.1 对象	19
2.1.1 什么是对象	19
2.1.2 对象的生命周期	20
2.1.3 对象的特征	21
2.2 消息	22
2.2.1 消息和方法	22
2.2.2 消息序列	23
2.2.3 消息分类	24
2.3 封装和协议	25
2.4 类和实例	26
2.4.1 类及其描述	26
2.4.2 实例	28
2.5 抽象数据类型与类	30
2.6 继承	33
2.6.1 继承及分类	33
2.6.2 多重继承	34
2.6.3 继承与其它概念的比较	35
2.6.4 类的层次	37
2.6.5 泛化和特化	39

2.6.6 推迟功能实现	39
2.6.7 继承机制的应用	39
2.7 小结	41
习题	41
第三章 对象关系及其它相关概念	43
3.1 对象间的关系及关系图示	43
3.1.1 类的继承关系	44
3.1.2 对象间的创建关系	44
3.1.3 对象之间的聚合关系	45
3.1.4 对象通讯(消息)关系	45
3.1.5 实例化关系	46
3.2 面向对象系统的形式化描述	46
3.2.1 面向对象系统的定义	47
3.2.2 面向对象系统的有向图表示	48
3.3 面向对象语言的相关问题	49
3.3.1 多态与功能重载	49
3.3.2 作用域和语境	51
3.3.3 强类型与弱类型	53
3.3.4 端对象与非端对象	56
3.3.5 可扩充性	57
3.3.6 深拷贝与浅拷贝	59
3.4 小结	60
习题	61
第四章 面向对象分析与设计方法	62
4.1 面向对象分析与设计	62
4.2 COAD/YOURDON 方法	63
4.2.1 基本步骤	63
4.2.2 基本图符及意义	68
4.2.3 举例	69
4.3 BOOCH 方法	72
4.3.1 基本步骤	72
4.3.2 基本图符及意义	74
4.3.3 Coad/Yourdon 方法和 Booch 方法的比较	76
4.4 面向对象分析与设计环境	77
4.4.1 面向对象分析/设计环境的总体结构	77
4.4.2 功能界定	78
4.4.3 用户界面	78

4.4.4 分析/设计工具	79
4.4.5 维护工具	80
4.6 小结	83
习题	83
第五章 OLE/COM 技术	84
5.1 引言	84
5.2 COM: OLE 的对象总线	85
5.2.1 部件对象模型(COM)	85
5.2.2 COM 对象服务	89
5.3 自动化、脚本和类型库	93
5.3.1 OLE 自动化	93
5.3.2 OLE 自动化服务器	96
5.3.3 创建和管理类型信息	98
5.4 一致数据传输和 OLE 拖放	102
5.4.1 OLE 的数据传输模型	102
5.4.2 剪贴板传输	103
5.4.3 拖放数据传输	105
5.4.4 链接数据传输	106
5.5 结构化存储和标记	108
5.5.1 OLE 的结构化存储	108
5.5.2 永久对象	111
5.5.3 标记	113
5.6 复合文档和 OCX	114
5.6.1 OLE 复合文档模型	114
5.6.2 最小容器/服务器	115
5.6.3 最大容器/服务器	117
5.6.4 OLE 定制控件(OCX)	120
5.7 小结	124
习题	124
第六章 SMALLTALK	125
6.1 概述	125
6.2 SMALLTALK 基本语法	126
6.2.1 文字表达式	126
6.2.2 变量名表达式	126
6.2.3 消息表达式	127
6.2.4 块表达式	127
6.3 类	130

6.3.1 类的描述	130
6.3.2 变量说明	130
6.3.3 方法	133
6.4 子类	135
6.4.1 应用继承机制的类描述举例	136
6.4.2 方法的查找与匹配	137
6.4.3 抽象类	139
6.4.4 子类的框架消息	143
6.5 元类	143
6.6 类的组织与管理	148
6.6.1 类的组织结构	148
6.6.2 与类管理有关的类的功能协议	149
6.7 SMALLTALK 系统结构	153
6.8 SMALLTALK 虚拟机	156
6.8.1 对象存储器	156
6.8.2 原语	159
6.8.3 解释器	160
6.9 小结	166
习题	167
第七章 C++	168
7.1 概述	168
7.1 类与封装	168
7.1.1 类的定义	168
7.1.2 成员函数及其定义	171
7.1.3 成员变量的访问和消息表达式	172
7.1.4 静态成员	173
7.1.5 关键字 this	174
7.2 对象的创建与删除	175
7.2.1 创建和删除方式	175
7.2.2 构造函数和析构函数	176
7.3 继承	179
7.3.1 继承的定义及继承成员的访问特性	179
7.3.2 多重继承	181
7.3.3 重复继承	186
7.4 重载与多态	189
7.4.1 函数的重载	189
7.4.2 多态与虚函数	190
7.5 C++中的其它概念和机制	194

7.5.1 友元	194
7.5.2 类模板	195
7.6 评述与小结	196
习题	197
第八章 JAVA	198
8.1 引言	198
8.2 JAVA 及其基本特征	198
8.2.1 Java 的特征	198
8.2.2 Java 语言中的面向对象概念	200
8.3 类——JAVA 的抽象数据类型	200
8.3.1 类说明	201
8.3.2 变量说明	201
8.3.3 类的方法定义和实现	203
8.3.4 类的继承	205
8.4 对象	206
8.4.1 对象的生成	206
8.4.2 对象的使用	206
8.4.3 对象的撤消	207
8.5 接口	207
8.6 包(PACKAGE)	208
8.7 标准 JAVA 包	209
8.7.1 java.lang 包	210
8.7.2 java.util 包	210
8.7.3 java.io 包	211
8.7.4 java.awt 包	212
8.7.5 java.net 包	212
8.7.6 java.applet 包	213
8.8 JAVA 的体系结构	213
8.8.1 体系结构特征	213
8.8.2 Java 程序开发过程	214
8.9 JAVA 虚拟机	216
8.9.1 Java 源文件的编译、装载、解释和执行	216
8.9.2 JVM 规格描述	218
8.9.3 Java 对象的存储模型	221
8.10 JAVA、SMALLTALK、C++的比较	223
8.11 小结	225
习题	225

第九章 面向对象数据库	227
9.1 概述	227
9.2 面向对象数据库系统的基本特征	229
9.2.1 必备特征	229
9.2.2 可选特征	234
9.2.3 开放的可选特征	234
9.3 ODMG	235
9.3.1 对象模型	236
9.3.2 OQL	244
9.4 面向对象数据库系统	248
9.4.1 ONTOS	248
9.4.2 GemStone 系统	262
9.4.3 其他	267
9.5 实现技术	267
9.5.1 对象的存储结构及索引组织	267
9.5.2 OODB 中的事务管理	272
9.5.3 模式演化与版本组织和管理	275
9.5.4 其他	278
9.6 对象关系数据库	278
9.6.1 对象关系数据库技术	279
9.6.2 ORDB 系统与实现途径	281
9.7 未来发展	282
习题	284
第十章 面向对象应用新进展	285
10.1 自治对象	285
10.1.1 自治对象的特点	285
10.1.2 自治对象与现有系统的比较	286
10.1.3 Java 的 Applet	287
10.1.4 自治对象的实现	287
10.1.5 小结	290
10.2 网上对象(OBJECTS ON WEB)	290
10.2.1 概述	290
10.2.2 W3 Objects	291
10.2.3 W3 Object 体系结构	292
10.2.4 举例	293
10.3 面向对象操作系统	295
10.3.1 基本问题	295

10.3.2 面向对象操作系统原型 Choices	296
10.4 CORBA 标准	300
10.4.1 简介	300
10.4.2 对象管理结构(OMA)	301
10.4.3 CORBA 基本要点	302
10.4.4 对象模式	304
10.4.5 CORBA 技术的应用	305
10.5 DCOM 标准	306
10.5.1 简介	306
10.5.2 DCOM 的体系结构	307
10.5.3 DCOM 的无关性	308
10.5.4 DCOM 的连接与通讯管理	308
10.5.5 DCOM 的可扩展性	309
10.6 统一建模语言 UML	309
10.6.1 概述	310
10.6.2 UML 的基本内容	312
10.6.3 UML 未涉及的领域	314
10.6.4 UML 与其他建模方法的比较	315
10.6.5 UML 的新特点	315
参 考 文 献	316

第一章 导论

1.1 面向对象技术的发展

“对象”(object)一词可以说是源远流长,从亚里士多德的哲学论著,到笛卡尔的《哲学原理》,多次用过“对象”一词,但是“面向对象”一词的使用却只有十几年的历史。20世纪80年代以来“面向对象”一词的广泛使用,反映了计算机科学发展的需要。计算机科学的发展与社会发展是相互促进的,计算机科学技术的发展推动了社会的现代化,计算机应用的广泛深入,也反过来对计算机科学提出了新的要求。人们越来越希望能尽可能直接地与计算机进行交互,而不再需要专门的训练。这给计算机特别是软件领域的技术发展提出了更高的要求。

60年代末70年代初,软件设计人员普遍感到负担越来越重。软件系统越来越庞大,复杂度越来越高,但仍然不能满足用户的需要,低效的软件生产率已越来越不适应软件的发展,因此,出现了软件工程、程序设计方法学等研究领域。

为了控制大型软件系统的复杂程度,必须解决以下两个问题:

第一,更新语言中的类型定义机制,使类型的内部表示细节尽可能对外不可见,使用户程序不依赖于数据的某一具体表示;

第二,寻求连接模块的新方法,尽可能缩小模块间的界面。

为解决上述两个问题,必须更新模块概念,寻求访问类型数据的新方法。因此,程序设计语言不断得到发展。到了20世纪70年代,随着结构化程序设计方法的推广,开始出现面向对象的思想方法,例如SIMULA语言中出现了对象和类的概念。进入80年代后,面向对象方法已经被广大程序设计人员所接受,Smalltalk语言和C++语言便是面向对象程序设计语言的典型代表。

1992年,北美CIO(Computer Information Organization)的Deloitte和Touche所作年度总结报告中称,该年度中有30%的应用开发采用了面向对象技术,到1994年底,面向对象技术的利用率增加到70%。著名大公司与机构纷纷将面向对象技术作为其组织中的主导技术,如Apple、IBM和HP三大公司联合投资一亿美元开发面向对象的软件环境,微软公司也斥巨资开发继Windows NT之后的新一代面向对象的操作系统Cairo。例如,一个美国公司,其主要业务是国防监控系统,在一个项目(为五艘战舰建立监控系统,包括命令、控制和通讯等功能)中采用了OOA和OOD技术,每艘战舰上安装的系统约含600多个程序,大约150万行代码,该系统的软件生产率比该公司未采用面向对象技术以前提高了一倍多。这一提高主要得益于OOD的软件重用性,系统中有近一半代码可重用。这使该公司一次性节省开发费用达400万~780万美元。还有一个美国的系统集成商,开发了一个基于大型机上MVS和CICS的财务管理软件包,采用OOA和OOD技术,系统包含约200万

行代码，该集成商获得了每工程小时 3.7 行的生产率，比原来 1.8 行的生产率提高一倍多，节省开发费用约 240 万美元，并大大缩短了开发时间。

目前，面向对象技术已不仅仅局限于程序设计领域，而是逐步渗透到了软件开发、系统模拟和仿真、CAD/CIMS、图形处理、数据库组织与管理、人工智能和体系结构等计算机领域的各个方向，许多专家学者已开始从认识方法论等基础理论的角度研究这一方法。目前，面向对象已经出现了系统、技术、语言、工具/环境和方法学等研究分支。

1.2 面向对象技术的特点

面向对象方法追求的是现实问题空间与软件系统解空间的直接模拟。它希望用户用最小的气力，最大程度地利用软件系统来求解问题。可以这样看，现实世界包含两大部分，也就是哲学的基本问题——物质和意识。映射到面向对象系统的解空间就是具体事物（对象）和抽象概念（类）。如图 1-1 所示。

面向对象并非一种全新的技术方法，面向对象技术中的主要思想和机制于 20 世纪 60 年代末便开始出现在“主流”语言（如 C 语言和 Pascal 语言）中，在 SIMULA67 中就更完整一些。

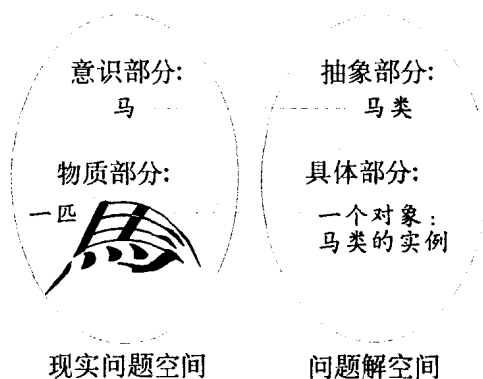


图 1-1 面向对象解空间与现实问题空间的对应关系

面向对象的设计方法基于 Parnas 的信息隐蔽和 Cuttag 的抽象数据类型概

念。它把系统中所有资源，如数据、模块以及系统都看成对象，每个对象把一组数据和一组过程封装在一起，使得仅这一组过程可以对这组数据进行处理，并在定义对象时可以规定外界的请求权限。使用这一方法，设计人员可以依照自己的意图创建自己的对象，并将问题映射到该对象上。这一方法直接、自然，使设计人员把主要精力放在系统一级上，而对细节问题可以较少地关心。

面向对象方法有以下优点：

①模块性：对象是一个功能和数据独立的单元，相互之间只能通过对象认可的途径进行通信。互相没有预料不到的影响，也可以较为自由地为各个不同的软件系统所重用；

②封装功能：为信息隐蔽提供具体的实现手段，用户不必清楚对象的内部细节，只要了解其功能描述就可以使用；

③代码共享：继承性提供了一种代码共享的手段，可以避免重复的代码设计，使得面向对象的方法确实有效；

④灵活性：对象的功能执行是在接收到消息时确定的，使得对象可以根据自身的特点进行功能实现，提高了程序设计的灵活性；

⑤易维护性：对象实现了抽象和封装，使其中可能出现的错误限制在自身，不会向外传播，易于检错和修改；

⑥增量型设计：面向对象系统可以通过继承机制不断扩充功能，而不影响原有软件运行；

⑦局部存储与分布处理：每个对象通过数据抽象和数据隐蔽将其内容和状态置于自身独立的存储结构中。对象的处理也是自治的，由对象构成的整个系统的运行和处理是分布式的。

1.3 面向对象方法学

面向对象技术是 20 世纪 90 年代计算机研究领域中广泛应用的技术，它既是一种软件开发方法，也可以作为一种建立系统的基础结构。

从广义来讲，面向对象的原则就是：

- ①一切事物皆为对象；
- ②任何系统（也是对象）均由对象构成；
- ③系统的发展和进化均由系统的内外部对象的相互作用完成。

从应用上看，Smalltalk 语言是坚持上述三原则的最杰出代表，而 C++ 和 Java 则在第一条原则上进行了许多修正。

首先它是一种从一般到特殊的演绎方法，也是一种分类方法。一个按照面向对象方法设计的系统的构造恰好符合于这一演绎过程。

另外，面向对象系统又为人们对问题领域采用从特殊到一般的归纳方法提供了帮助。由一大批相同或相似的对象构造出类的过程恰是这样一个归纳过程。

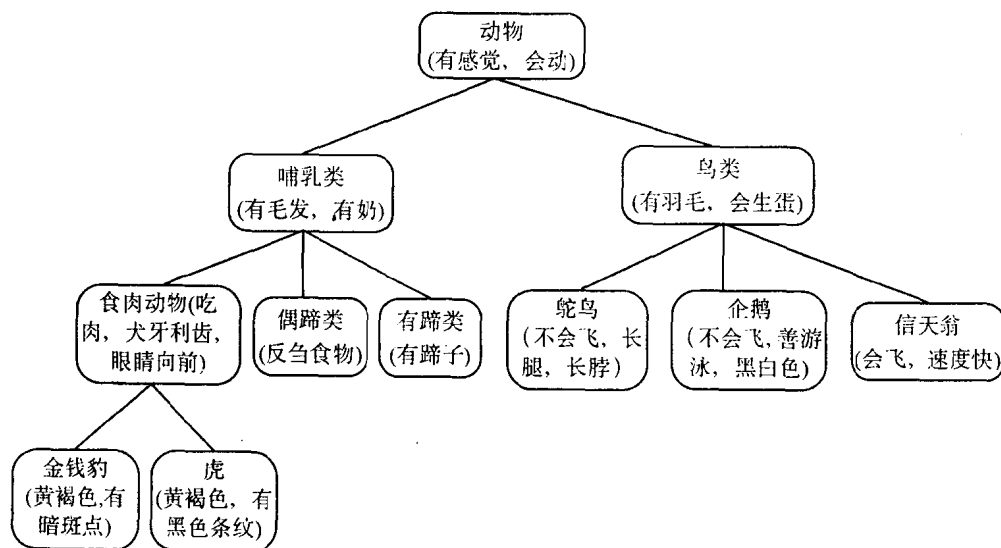


图 1-2 动物界的演绎分解

例如，动物世界可以由分类图示(图 1-2)表征，这一图示基本上与面向对象方法的表示是一样的，这是一种演绎方法。其中，每个结点表示一类动物，括号内表示该类对象的属性。

另一种认识方法是从特殊到一般的归纳方法。我们今天看到一条猎狗，它是一个对象，明天又看到一条警犬，它也是一个对象，这两个对象除了作用不同外，其它狗的特征完全一样。这样，我们便可以构造一个类——“狗”，其中描述了狗的所有共同特征，如会叫、有犬齿、嗅觉灵敏、有颜色、忠实等等，而猎狗和警犬都是这个类的实例。因此，面向对象很适合这种认识方式的组织（如图 1-3）。

按图自上而下生成树，便是一种演绎方法，反之，自底向上观察树又是一种归纳方法。

面向对象既提供了从一般到特殊的演绎手段(如继承等)，又提供了从特殊到一般的归纳形式(如类等)，从而说明它是一个很好的认知方法。只不过目前以面向对象为基础的归纳方法的研究尚未展开并引起足够的重视。

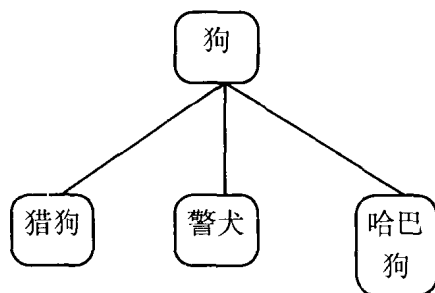


图 1-3 狗类的归纳总结

狭义上看，OOSD（面向对象系统开发）= OOA

（面向对象分析）+ OOD（面向对象设计）+ OOP（面向对象程序设计）。其中，面向对象分析是系统分析师对于应用系统的需求定义和分析，得到对象类及相互间关系的抽象描述；面向对象设计是指软件设计人员将对象的抽象描述及对象类间的关系抽象描述转化为程序设计语言中的具体描述；面向对象程序设计则是程序设计人员利用面向对象程序设计语言，根据设计得到的对象类，生成对象实例，建立对象间的各种联系，从而建立实际可运行的系统。

1.4 面向对象语言及系统

自 20 世纪 70 年代末 80 年代初以来，多种面向对象的语言和系统相继问世，虽然对于面向对象程序设计机制的取舍各不相同，但它们对于促进面向对象技术的发展都起到了巨大的推动作用。

Smalltalk 是第一个面向对象的语言，它的风格起源于一个模拟语言 Simula。它最初是由 Alan Kay 于 1972 年在施乐公司的 PARC 实验室工作时设计的，这个名字取自“少说话 (talk small)”，意思是你通过很少的话语就可以完成很多工作。从那时起，许多人为 Smalltalk 的发展做出了很大贡献，经过 Smalltalk-72、Smalltalk-74 和 Smalltalk-78 的不断改进完善，最后形成最完整的 Smalltalk-80 系统。Smalltalk-72 是继设计人之后推出的第一个正式 Smalltalk 解释器版本；Smalltalk-74 第一次加上了多窗口界面，这是 Smalltalk 系统的一大特点；Smalltalk-78 第一次采用了中间代码的设计和实现，并用微指令实现了中间代码，大大提高了系统速度，使该系统第一次有了可用性。之后，施乐公司制订了商用 Smalltalk 计划，并成立了以 Adele Goldberg 为首的 ParcPlace System 公司，最后推出 Smalltalk-80 系统。它在系统的设计中强调了对象概念的统一，并引入完善了类、方法、实例等概念和术语，应用了单重继承机制和动态连接，其中许多概念为其它语言所采用。Smalltalk-80 是程序设计及语言领域中一个重要的里程碑。

C++ 是由 AT&T 公司的 Bjarne Stroustrup 对 C 语言进行改进扩充，并增加了类似于