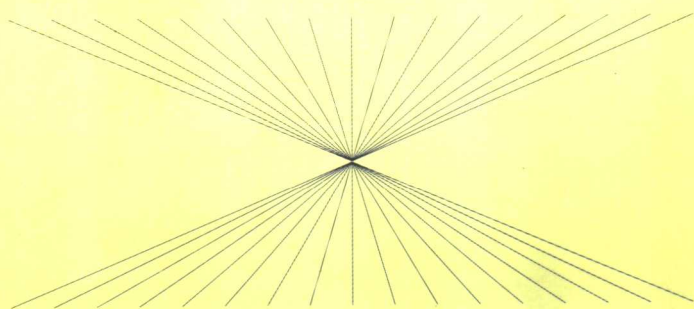


面向对象方法 与技术基础

黄永忠 陈新 陈海勇 编著
刘蓬晖 王磊 周蓓



国防工业出版社
National Defense Industry Press

面向对象方法与技术基础

黄永忠 陈 新 陈海勇
刘蓬晖 王 磊 周 蓓 编著

国防工业出版社

· 北京 ·

内 容 简 介

本书结合作者教学、科研和工程的实践,全面、系统地介绍面向对象方法与技术的基本原理。

全书共分8章,涵盖面向对象的基本概念、C++语言、Windows编程基础及MFC开发、面向对象软件工程、组件技术、设计模式、Java语言与技术等内容,并反映面向对象技术与方法的最新进展,内容翔实,论述深入浅出。

本书可以帮助读者理解和掌握面向对象技术,运用面向对象方法分析和解决软件系统设计的问题。可以作为计算机和其他相关专业的高年级本科生和研究生物面向对象技术和方法学课程的教材或参考书,也可供从事计算机软件领域研究、应用和开发的科技工作者参考。

图书在版编目(CIP)数据

面向对象方法与技术基础/黄永忠等编著. —北京:国防工业出版社,2006.5

ISBN 7-118-04473-3

I.面... II.黄... III.面向对象语言—程序设计
IV.TP312

中国版本图书馆CIP数据核字(2006)第023682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100044)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 22 字数 507千字

2006年5月第1版第1次印刷 印数 1—4000册 定价 36.00元

(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

面向对象技术是目前主流的软件构造技术，面向对象方法学是流行的软件方法学。在产业界，越来越多的公司从传统的领域开发技术转向面向对象技术；在学术界，面向对象方法和技术一直是最受关注的研究热点。新世纪，面向对象技术将进一步发展。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工作，使面向对象技术越来越体现出强大的生命力。计算机及相关专业的学生，如果不懂得面向对象方法和技术，将很难立足计算机及相关领域的软件开发。

目前各高校均开设了面向对象方法和技术方面的课程，但缺乏全面、系统地介绍面向对象理论、技术和方法的教材和专著。

本书结合作者教学、科研和工程的实践，参考大量的国内外有关著作和文献，全面、系统地介绍面向对象方法与技术的基本原理，包括面向对象的基本概念、C++语言、Windows 编程基础及 MFC 开发、面向对象软件工程、组件技术、设计模式、Java 技术等内容，并反映面向对象技术与方法的最新进展。

全书共分为 8 章。其中第 1 章引论，介绍面向对象技术的由来及发展、面向对象方法的基本思想以及面向对象技术的新进展；第 2 章面向对象方法与技术基础，详细介绍面向对象的基本概念以及形式化描述；第 3 章 C++程序设计基础，通过介绍 C++语言，让读者了解面向对象程序设计方法和思想在 C++中的具体体现；第 4 章 MFC 面向对象设计，介绍 Windows 环境下 WIN32 API 程序设计的基本概念和原理，以及 MFC 应用程序框架和面向对象设计思想；第 5 章面向对象软件工程，介绍如何将面向对象方法引入全软件生命周期，包括基本概念、分析与设计方法、经典面向对象方法等，最后介绍 UML 语言和统一过程；第 6 章分布组件技术，首先介绍软件复用技术，由于网络的发展使得分布组件技术的应用越来越广泛，因此本章对几种目前流行的分布组件模型，即 COM(以及建立在 COM 基础之上的 DCOM)组件模型、CORBA 组件模型、Web Service 和软件代理技术等进行了介绍；第 7 章 Java 语言与 Java 技术，首先给出 Java 的内涵，对 Java 和 Java 技术进行了简要的介绍，然后对 JDBC、远程方法调用 RMI、J2EE、JavaSpaces 等进行了介绍；第 8 章设计模式，首先介绍设计模式的概念、思想和原则，然后按照 GoF 设计模式的分类分别对创建模式、结构模式、行为模式进行了介绍，最后介绍了 J2EE 中常用的设计模式。

本书可以帮助读者理解和掌握面向对象技术，运用面向对象方法分析和解决软件系统设计的问题。本书可以作为计算机或其他相关专业的高年级本科生或研究生面向对象技术和方法学课程的教材或参考书，也可供从事计算机软件领域研究、应用和开发的科技工作者参考。

本书在正式出版之前作为解放军信息工程大学信息工程学院研究生自编教材已使用两届。本书由黄永忠主持编写，并执笔撰写了第1章和第2章；陈新、周蓓共同编写了第3章、第4章、第7章；陈海勇、王磊共同编写了第6章和第8章；刘蓬晖编写了第5章。在编写过程中，得到了学院训练部和计算机科学与技术系的领导和老师们的大力支持和帮助，实验室的研究生刘振林、李静、黄敬磊等学生对本书的文字、图表进行了部分录入和校对，张丹和任涛还整理了部分章节，在此一并表示衷心的感谢。

由于作者学识水平有限，疏漏和错误之处在所难免，敬请读者指教。联系方式：河南郑州1001信箱716号(450002)，电话0371—63531404、63531361、63530522。

作者

2006年3月于黄河之滨

目 录

第 1 章 绪论	1
1.1 面向对象方法与技术的由来及发展	1
1.1.1 传统软件方法的缺陷	1
1.1.2 面向对象方法与技术的产生和发展	4
1.2 面向对象的基本思想	9
1.2.1 面向对象方法与技术	9
1.2.2 面向对象的基本思想.....	10
1.2.3 行为共享与进化.....	11
1.2.4 面向对象方法和技术的主要优点.....	12
1.3 面向对象技术的新进展.....	13
1.3.1 将面向对象技术和数据库技术结合产生面向对象数据库.....	13
1.3.2 分布组件技术.....	13
1.3.3 网上对象(Objects on Web)	15
1.3.4 面向对象操作系统.....	16
1.3.5 面向对象软件工程.....	17
1.3.6 Java 技术.....	19
1.3.7 并行面向对象技术.....	19
1.4 小结.....	20
习题与思考题	22
第 2 章 面向对象方法与技术基础	23
2.1 面向对象设计范型.....	23
2.1.1 范型.....	23
2.1.2 过程性范型与面向对象范型的比较.....	24
2.1.3 语言对面向对象范型的支持.....	25
2.2 面向对象的基本概念.....	26
2.2.1 对象和类.....	26
2.2.2 消息和消息响应.....	28
2.2.3 继承和类层次.....	29
2.2.4 封装和接口.....	30
2.2.5 继承和多态(Polymorphism)	31
2.2.6 动态联编(Dynamic Binding)	31
2.2.7 重载(Overloading)	32

2.3 面向对象系统的形式化描述	33
2.3.1 对象	33
2.3.2 类	34
2.3.3 消息	35
2.3.4 对象间关系	37
2.4 面向对象系统的图模型	38
2.4.1 面向对象系统的基本图模型	38
2.4.2 面向对象 Petri 网(Object _ Oriented Petri Nets, OOPNs)	40
2.5 抽象数据类型(Abstract Data Type)	44
2.5.1 抽象	44
2.5.2 数据抽象	44
2.5.3 抽象数据类型	46
2.5.4 抽象数据类型的代数规范	49
2.6 面向对象涉及的其他概念	51
2.6.1 深拷贝与浅拷贝	51
2.6.2 作用域与语境	52
2.6.3 强类型与弱类型	53
2.6.4 对象的生命周期与持久对象	56
2.6.5 可扩充性	57
2.6.6 端对象与非端对象	59
2.6.7 主动对象	60
2.7 小结	61
习题与思考题	62
第3章 C++ 程序设计基础	64
3.1 类与对象	64
3.1.1 面向对象编程(COP)的基本特性	64
3.1.2 类的引入	67
3.1.3 类的定义	69
3.1.4 构造函数和析构函数	73
3.1.5 聚集的概念(对象成员)	78
3.2 继承	83
3.2.1 继承引入	83
3.2.2 继承表示语法	86
3.2.3 protected 访问指明符	86
3.2.4 派生类构造函数及析构函数的构造规则	87
3.2.5 继承指定	89
3.2.6 继承关系中的成员可见性	90
3.3 多态性与虚函数	90
3.3.1 多态性	90

3.3.2	虚函数	92
3.3.3	抽象类	97
3.3.4	多继承与命名冲突	98
3.3.5	重复继承(共享继承)	100
3.4	C++ 其他成分	101
3.4.1	友元	101
3.4.2	常类型	103
3.4.3	引用	105
3.4.4	异常处理	105
3.4.5	函数重载与运算符重载	107
3.4.6	模板	110
3.5	小结	115
	习题与思考题	117
第4章	MFC 面向对象设计	119
4.1	WIN32 API 概述	119
4.2	WIN32 API 基本概念	122
4.2.1	句柄	122
4.2.2	常用数据类型	122
4.2.3	消息机制与事件驱动	123
4.3	WIN32 API 应用程序框架	124
4.3.1	应用程序入口	124
4.3.2	消息循环	124
4.3.3	窗口函数	126
4.3.4	窗口类	126
4.3.5	窗口的创建与显示	127
4.3.6	WIN32 API 程序执行流程	127
4.4	WIN32 API 程序示例	128
4.5	MFC 应用程序框架	130
4.5.1	MFC 概述	130
4.5.2	MFC 类库	131
4.5.3	MFC 基础类	134
4.5.4	MFC 程序示例	142
4.6	MFC 对 WIN32 API 的封装	143
4.6.1	MFC 对消息循环机制的封装	143
4.6.2	MFC 对窗口创建流程的封装	146
4.6.3	MFC 对消息处理的封装	147
4.7	文档/视图	151
4.7.1	文档/视图概述	151
4.7.2	文档/视图架构分析	152

4.8 小结	155
习题与思考题	156
第5章 面向对象软件工程	157
5.1 概述	157
5.1.1 传统软件工程的缺陷	157
5.1.2 面向对象软件工程	160
5.2 面向对象软件的开发过程	161
5.2.1 面向对象的应用生存期模型	161
5.2.2 面向对象分析的基本任务与过程	164
5.2.3 面向对象设计的任务	170
5.3 经典面向对象方法	172
5.3.1 Booch 方法	172
5.3.2 Rumbaugh 的 OMT 方法	176
5.3.3 Coad/Yourdon 方法	179
5.3.4 Jacobson 方法	182
5.4 标准的面向对象建模语言——UML	183
5.4.1 UML 简史	183
5.4.2 UML 语义	185
5.4.3 UML 表示法	189
5.4.4 UML 静态建模	190
5.4.5 UML 动态建模	193
5.4 Rational 统一开发过程——RUP	196
5.4.1 概述	196
5.4.2 RUP 的基本思想	197
5.4.3 迭代式开发	199
5.4.4 以构架为中心的过程	199
5.4.5 用例驱动的过程	200
5.6 小结	201
习题与思考题	202
第6章 分布组件技术	203
6.1 基于组件的软件复用是解决软件质量问题的有效手段	203
6.1.1 软件复用技术	203
6.1.2 组件技术	205
6.1.3 流行的组件技术	207
6.2 分布组件技术初探	207
6.2.1 网络 Socket 编程	207
6.2.2 分布组件技术的概念	209
6.3 COM/DCOM 模型	214
6.3.1 ActiveX、OLE 和 COM	214

6.3.2	基本概念	215
6.3.3	实例分析	215
6.3.4	DCOM	222
6.3.5	接口	223
6.3.6	IDL 语言	224
6.4	CORBA 技术	226
6.4.1	CORBA 的产生与发展	226
6.4.2	CORBA 体系结构概述	227
6.4.3	CORBA 开发	232
6.5	Web Service 技术	238
6.6	代理(Agent)技术	246
6.6.1	概述	246
6.6.2	一个基于 Java 的流动代理模型	248
6.7	小结	251
	习题与思考题	252
第 7 章	Java 语言与 Java 技术	253
7.1	Java 简介	253
7.1.1	Java 的内涵	253
7.1.2	Java 技术的组成	253
7.1.3	Java 语言的特点	256
7.1.4	Java 语言与 C/C++	257
7.1.5	Java 开发环境	259
7.1.6	Java 语言的跨平台	259
7.1.7	Java 语言面向对象特性的语法体现	260
7.1.8	Java 本机方法	268
7.2	JDBC	273
7.2.1	JDBC 的出现	273
7.2.2	JDBC 的组成	274
7.2.3	JDBC 编程的基本概念	275
7.2.4	JDBC 的常用接口和类	276
7.2.5	使用 JDBC 进行数据库编程	277
7.3	远程方法调用 RMI	278
7.3.1	RMI 的体系结构	278
7.3.2	RMI 工作机制	280
7.3.3	RMI 应用程序的开发	280
7.4	J2EE	284
7.4.1	Java Servlet	286
7.4.2	JSP	287
7.4.3	EJB	289

7.5	Tspaces,JavaSpaces,Linda	298
7.5.1	TupleSpace	298
7.5.2	Tspaces	300
7.5.3	JavaSpaces	302
7.6	小结	303
	习题与思考题.....	304
第8章	设计模式	306
8.1	概述	306
8.2	设计模式的概念	307
8.3	设计模式的原则	309
8.4	GoF 设计模式的分类	311
8.4.1	创建模式	311
8.4.2	结构模式	312
8.4.3	行为模式	313
8.5	创建模式	314
8.5.1	工厂模式	314
8.5.2	生成器模式	322
8.5.3	原型模式	325
8.5.4	单例模式	327
8.6	结构模式	328
8.6.1	适配器模式	328
8.6.2	外观模式	330
8.7	行为模式	332
8.7.1	状态模式	332
8.7.2	观察者模式	334
8.8	其他设计模式	336
8.8.1	J2EE 设计模式	336
8.8.2	J2EE 设计模式的特点	337
8.8.3	J2EE 设计模式的分类	337
8.9	小结	340
	习题与思考题.....	340
	参考文献	341

第1章 绪论

多年以来，计算机领域的科学家、工程技术人员一直在不断探索能够控制软件复杂性的开发机制，以降低开发成本、提高软件生产效率、增加软件的可重用性以及改善软件可维护性等，并推出了许多解决软件复杂性问题的开发规范。

面向对象技术是当前计算机领域的主流技术。面向对象方法作为一种独具优越性的新方法引起了全世界越来越广泛的关注和高度的重视，它被誉为“研究高技术的好方法”，更是当前计算机界关心的重点。在学术界，面向对象方法和技术已成为最受关注的研究热点，越来越多的学术会议和期刊把面向对象作为主要议题之一，并且每年都有面向对象方面的专著出版；在产业界，越来越多的公司从传统的开发技术转向面向对象技术，并以此作为提高公司形象和产品信誉的标志。特别是在一些发达国家，几乎所有的新软件开发，都全面或部分地采用了面向对象技术。

本章就面向对象的产生背景、面向对象方法学的基本思想以及面向对象技术的新发展做一简要介绍。

1.1 面向对象方法与技术的由来及发展

1.1.1 传统软件方法的缺陷

计算机软件是能完成某种功能的计算机程序系统。其设计目的就是模拟和仿真现实世界中的某一个系统来完成某个具体的功能。从这个角度看，可以认为软件系统是现实世界中某个系统的模型或模拟。软件是逻辑实体，而非物理实体，软件的这种本质特征决定了软件的开发既困难艰巨、耗时费力，又不易控制开发过程和产品质量。

从第一台计算机问世至今已有半个多世纪了，人们设计了许多计算机软件。能够借助计算机做很多工作：从早期的数值计算、事务处理到人工智能及知识处理，计算机已经日益渗透到生活的每一个角落。如图 1-1 所示，事实上，软件的发展就是不断挑战现实世界复杂问题求解的过程。

现在，软件开发的需求表现在以下三个方面。

(1) 提高生产能力。早期的计算机处理能力有限，软件主要应用于科学处理，输入、输出量不大，计算量大，编制程序工作复杂、繁琐、费时、易出错，衡量程序的标准是功效，设计与编制程序采用个体工作方式。随着计算机应用的日益普及、应用范围的逐步扩大，软件的规模也越来越大，比如航天飞机的计算机系统有 4000 万行代码，空间站计算机系统有 10 亿行代码，大家比较熟悉的 Windows 系统也有 4500~6000 万行代码。星球大战计划导弹防御系统设计时估计有 4000 万行代码，最后被取消的原因之一就是软件太大。据统计，软件需求量每年增加 12%，而生产率每年只增加 4%。个体化

作坊式的软件开发手段已远不能完成其使命，传统结构化软件开发方法和技术也面临更多的挑战。

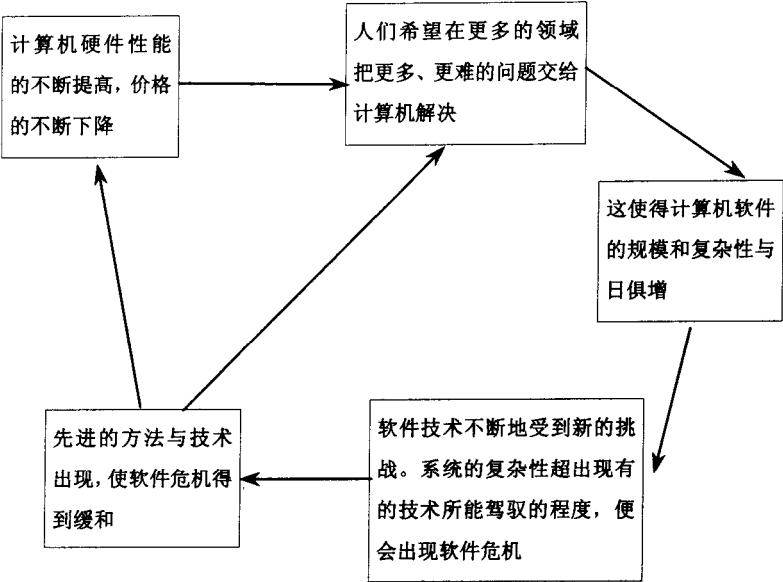


图 1-1 软件技术的发展

(2) 扩大处理范围。软件已广泛应用于社会的各个领域、各个层次，以及人类生活的各个方面。当世界步入信息时代，作为信息产业核心的软件及信息服务产业，已成为新世纪全球的第一大产业。1999 年 2 月美国总统信息技术顾问委员会给总统的报告中列举了大量的事实论证 IT(信息技术)将引起社会和国家以及人民生活方式的重大改变，包括通讯、信息使用方式、学习、医疗保健、商业贸易、工作、设计/建造、科学研究、对环境理解、政府工作性质和方式等 10 个方面。计算机辅助设计、计算机辅助制造、计算机集成制造系统等先进制造技术，正在彻底地改变着传统工业的面貌；管理信息系统、电子数据交换、企业信息系统集成、电子商务等商贸自动化技术，成为提高企业竞争力的必要手段；遥感技术、地理信息系统和全球定位系统等信息获取与处理技术，为人类活动提供了精确的、数字化的地理信息；信息技术进入消费电子领域，形成巨大的消费市场，加速了社会信息化的进程，也改变着人们的生活方式；因特网的兴起，更是空前地加速了信息技术的应用和渗透，掀起了全球的信息化热潮。信息技术的核心是软件，相应地，计算机软件也将适应这些转变，从而扩大信息处理范围。

(3) 面向新的环境。因特网爆炸性地发展，是 20 世纪末信息技术最伟大的成就。随着因特网的飞速发展，软件生产与应用方式正在发生显著的转变。我们个人将来的工作、学习、娱乐就像现在离不开电灯、电话、电视一样将离不开因特网。社会上的各行各业的运行也将最终离不开因特网。全球的经济一体化进程将随因特网的发展而加快，地球也将变得愈来愈小。这种网络化的趋势对计算机软件技术产生了巨大影响。软件的运行环境(平台)正在从封闭、静态、稳定的单机环境转变到开放、动态、多变的网络环境，出

现了 Web 操作系统、中间件、网上搜索引擎、信息挖掘、Agent、CSCW、网格计算等新的软件技术方向。因特网快速发展使得软件已深入、广泛地渗透到人类社会生活的各个方面，软件系统的规模、复杂性都有了大幅度提高。当前的软件生产能力难以满足社会对软件的巨大需求。软件产业迫切需要更优良的技术与支撑环境，以提高能力，适应社会需要；同时对软件质量提出了更高的要求。

随着软件系统越来越大，越来越复杂，开发人员和产品管理人员都能够深切地感受到传统软件开发方法和技术的不力与束缚。原因在于人们的思想长久地局限于冯·诺依曼体系结构的思想之内。

冯·诺依曼结构认为计算机系统由处理器(CPU)、内存(M)、输入/输出(I/O)三大部件组成。内存是一个存储单元的线性空间，程序和它加工的数据共同存放于内存之中；程序是某种指令的序列，描述了既定执行过程，处理器通过逐条执行程序中的指令来完成某项工作。

围绕冯·诺依曼结构，伴随计算机系统的发展，程序设计经过了三代语言的发展历程。

第一代(1GL)：机器语言。机器语言是最早的程序设计语言，它是由机器指令代码组成的语言，其抽象程度最低。机器语言是与机器相关的，对于不同的机器就有相应的一套机器语言。用这种语言编写的程序，都是二进制代码的形式，且所有的地址分配都是以绝对地址的形式处理。存储空间安排，寄存器、变址的使用都由程序员自己计划。因此使用机器语言编写的程序很不直观，在计算机内的运行效率很高，但编写出的机器语言程序出错率也高，并且开发效率极低。

第二代(2GL)：汇编语言。汇编语言是采用助记符的符号化二进制语言，比机器语言直观，使得编程人员从晦涩、难记的二进制代码中解脱出来。它的每条符号指令与相应的机器指令有对应关系，同时又增加了一些诸如宏、符号地址等功能。存储空间安排可由机器安排。但汇编语言仍然是与机器相关的，不同指令集的处理器系统就有自己相应的汇编语言，其抽象层次还是很低。

第三代(3GL)：高级语言。高级语言采用类似于人类自然语言进行编程，在较高的层次上进行抽象，使得软件开发人员从机器指令中解脱出来，提高了软件开发效率。高级语言的发展经历了非结构化到结构化的过程。早在 20 世纪六七十年代，大型软件开发引起的软件危机，导致了 Yourdon 和 De Marco 的结构化分析与结构化设计的软件工程方法的盛行。其中结构化程序设计语言基于自顶向下、逐步求精的思想，采用模块化、抽象、信息隐蔽等手段，提供了高效的开发工具，在软件开发的历史中起到巨大的作用。而源于结构化程序设计语言的结构化软件开发方法则主宰了几十个春秋。所谓结构化方法，其本质就是为了保证软件开发的质量、提高软件的灵活性和软件的生产效率，通过工程化方法，建立系统的软件开发过程，使开发的软件具有好的结构，即所谓可拼装、可裁剪的模块化结构。

从三代语言的发展过程来讲，进步确实很大，然而三个阶段均没有摆脱“过程化”描述，即使是高级语言，元件只是操作单位更大的表达方式和控制语句，相当于引进“宏指令”，软件的结构仍然是“语句”的串行序列，语句和数据仍然混在一起。

现实世界中，万物在运行、运动，其组织的机理十分丰富多样，当然“串行化”也

是其中之一，但只是一种最简单的基础性机理。在自然界有并行机制、联想机制、消息响应机制、抽象机制、功能数据集约化机制等。到目前为止，自然界依然比人造出来的“计算机”神奇得多，其中一个原因就是人们的软件思想太长久地束缚在冯·诺依曼结构之中。按照这种体系结构，非得将丰富多彩的世界用一种单一的“串行过程”来描述，使得计算机软件系统是一个现实世界的扭曲变形的描述。这种软件既难于设计开发，又难于维护，所以从 20 世纪 60 年代起软件界人士就开始呼吁“软件危机”。人们在硬件设计上和软件设计上都在力求打破这种结构束缚，而面向对象设计就是这种努力的结果，而且证明是为人们广为欢迎的、有效的、有价值的技术。

1.1.2 面向对象方法与技术的产生和发展

今天，面向对象技术不仅是一种思想方法，同时也是一种现实可用的技术，可以借助面向对象工具来进行软件设计和开发。

影响面向对象(Object-Oriented, OO)产生的因素很多。影响 OO 产生的诸多因素中，最重要的可能要算是编程语言与方法的进步了，当然哲学和认知科学的层次和分类理论也促进了 OO 的产生和发展。语言不仅是我們描述系统和过程的工具，而且语言本身也表征我们思维的水平 and 特征。在过去的几十年中，编程语言中对抽象机制的支持已经发展到了一个较高的水平，图 1-2 给出了高级语言的演变过程。这种抽象的进化从地址(机器语言)到名字(汇编语言)，到表达式(第一代高级语言，如 Fortran)，到控制(第二代高级语言，如 COBOL)，到过程和函数(第二代和早期第三代高级语言，如 Pascal)，到模块和数据(晚期第三代高级语言，如 Modula)，最后到对象(基于对象和面向对象的语言)。Smalltalk 和其他面向对象语言的发展使得新的分析和设计技术的实现成为可能。

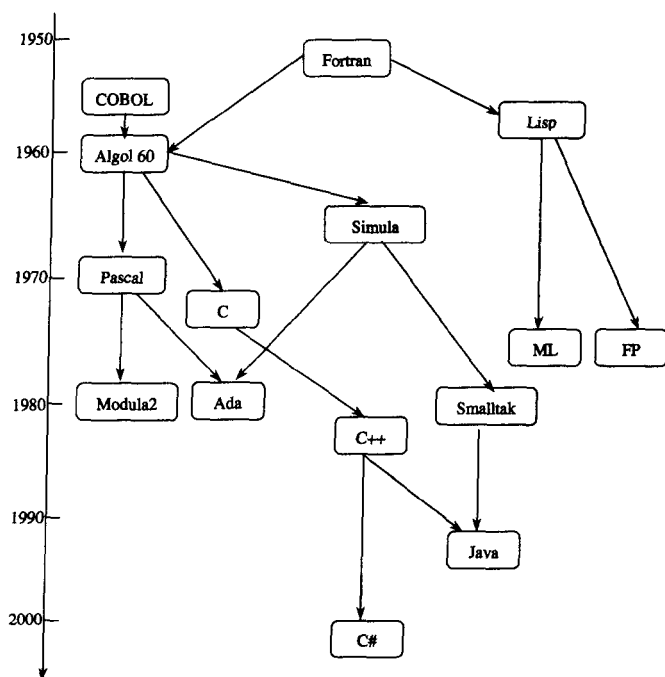


图 1-2 高级语言谱系及其演变

影响 OO 发展的因素也有很多。计算机硬件体系结构的进步、性能价格比的提高和硬件设计中对象概念的引入都对 OO 的发展产生了一定的影响。OO 的程序通常要更加频繁地访问内存,需要更高的处理速度,需要并且也正在利用强大的计算机硬件功能。计算机系统不断增长的规模、复杂度和分布性都对 OO 技术起了或多或少的推动作用。

1. 第一阶段——起源阶段

面向对象方法起源于面向对象的编程语言(简称面向对象程序设计语言)。面向对象程序设计语言是能够支持面向对象程序设计的计算机语言。

20 世纪 50 年代后期,人们在用 Fortran 语言编写大型程序时,常出现变量名在程序不同部分发生冲突的问题。鉴于此,Algol 语言的设计者在 Algol60 中采用了以“Begin…End”为标识的程序块,使块内变量名是局部的,以避免它们与程序块外的同名变量相冲突。这是编程语言中首次提供封装(保护)的尝试。此后程序块结构广泛用于高级语言如 Pascal、Ada、C 之中。

面向对象编程语言发展的一般说法是追溯到 1960 年出现的 Simula67 语言,这是挪威研制的一种计算机语言,当时研究者的目的是想开发一种模拟交叉关联过程的方法,当时 Algol 和 Fortran 语言均不适合。Algol 是分程序结构(嵌套),凡在某个分程序中创造的变量和分程序,一旦出了这个分程序的作用域它们将自然死亡,即不能有更长久的生命周期,不能独立生存。而自然界则有许多这样的过程,一个对象被创造出来,应有与创造者相对独立的生命力(生命周期)。20 世纪 60 年代中后期,Simula 语言在 Algol 的基础上研制开发,它将 Algol 的块结构概念向前发展一步,提出了对象的概念,并使用了类,也支持类继承。Simula 是第一个引进“类”(Class)概念的语言,其中许多思想是当前面向对象程序设计语言的前驱,如下面的一些概念。

(1) 类是数据和方法的包装体,这样,类就不单是数据类型,数据类型可以定义变量(数据),但它是“被动的”;而类可以用来生成类个体(现在相当于对象),它可以响应“请求”执行某个动作,因此是“积极的”、“能动性”的。

(2) 引入了关联变量,相当于现在的对象引用:

```
ri=new vehicle;
```

(3) 类个体具有独立的生命期,可以响应别的成分对其发出的请求,提供“服务”。

(4) 虽然 Simula 没有明确提出继承,但它提出了“类并置”,本质上是想解决继承问题。

Simula 在当时提出这些概念非常了不起,但很可惜这个语言当时影响不大,不过后来对 Ada、Smalltalk 等语言有很大的参考作用。这些思想与今天讲的面向对象的基本概念是十分相符合的,当然今天的面向对象技术更精良和丰富,所以人们一般认为 Simula 是面向对象程序设计语言的鼻祖语言。

2. 第二阶段——Smalltalk 语言问世

面向对象源出于 Simula,但真正意义上的面向对象程序设计(Object-Oriented Programming, OOP)由 Smalltalk 奠基。Smalltalk 现在被认为是最纯的面向对象程序设计语言。它由 Xerox 公司于 1972 年开发,名字“Small talk”,意思是少说话。经过 72、74、78、80 版本的发展,最后推出 Smalltalk80。正是通过 Smalltalk80 的研制与推广应用,使人们注意到面向对象方法所具有模块化、信息封装与隐蔽、抽象性、继承性、多样性

等独特之处，这些优异特性为研制大型软件、提高软件可靠性、可重用性、可扩充性和可维护性提供了有效的手段和途径。

在 Simula67 问世后不久，犹他大学的博士生 Alan Kay 在个人计算机上设计了一个实验性的语言 Flex。该语言从 Simula67 中借鉴了许多概念，如类、对象、继承等。此后 Kay 加入 PARC 继续从事 Smalltalk 的研究，Smalltalk 是 Xerox 公司 Palo Alto 研究中心在 1970 年研制开发的一种程序设计语言，它取 Simula 的类为核心概念，它的很多内容借鉴于 Lisp 语言，其研究目的是想研制一种人们更为适用的程序设计语言。其有两个研究目的：其一，语言更为简单，与人们的思维更相近一些；其二，是提供更直观的人机界面，基于图形用户接口(GUI)，开发者企图让小孩也能使用。1972 年 PARC 发布了 Smalltalk72，其中正式使用了“面向对象”这个术语。在 Smalltalk 中任何东西都是对象，即类的实例。对象具有封装性，因此是一个较为独立的程序单位。所有的对象都是类的实例；类有超类、子类之分，子类继承超类的所有信息。作为面向对象语言主要特征的继承与封装，在 Smalltalk 中都具备了。因此，Smalltalk 的问世标志着面向对象程序设计方法的正式形成。当然这个时期的 Smalltalk 语言还不够完善，还处在一种实验性的阶段。此后，PARC 一直坚持对 Smalltalk 的研究与改进，经过对 Smalltalk72、76、78 等版本进行持续 10 年不断的研究和改进之后，于 1981 年推出最完善的语言版本 Smalltalk80，它在系统设计中强调对象概念的统一，引入对象、对象类、方法、实例等概念和术语，采用动态联编和单继承机制。它是第一个完善的、能够实际应用的面向对象语言。它的发布使越来越多的人认识并接受了面向对象的思想，形成了一种崭新的程序设计风格，引发了计算机软件领域一场意义深远的变革。此外，Smalltalk80 不仅是一个编程语言，而且是一个具有类库支持和交互式图形用户界面的编程环境，这对于它的迅速流传也起到了很好的作用。

Smalltalk 语言第一次系统地引进了面向对象的一整套完整概念，它的问世被认为是面向对象语言发展史上最重要的里程碑，可以说这是现代面向对象技术的真正开始。

Smalltalk 语言具有以下特征。

(1) 纯面向对象性质。Smalltalk 语言完全支持信息隐蔽、数据抽象、动态绑定和继承(Inheritance)4 个特性，并具有下列一些特性：

- ① 对象作为唯一的实体概念。
- ② 对象标志作为唯一的对象引用方式。
- ③ 类和子类作为唯一的分类模式。
- ④ 消息作为唯一的一种计算、通信和控制机制。
- ⑤ 使用一种消息发送机制。
- ⑥ 所有对象处于同一地位。
- ⑦ 一种语义解释(方法)。
- ⑧ 一种参数传递机制(传值)。

(2) 表达式语言。Smalltalk 的计算对应着向对象发送消息，消息表达式用于描述向对象发送的消息。消息的发送必须说明接收对象(Receiver)和消息模式(Message Pattern)。消息模式由消息选择符加上任意个数的参数组成。例如

```
ages at: 'Brett Jorgensen' put: 3
```

其中，ages 是消息接收者；“at:”是消息选择符；Brett Jorgensen 和 3 是 2 个参数。