徐家福　吕建 著

软件语言及其实现

科学出版社
SCIENCE PRESS

软件语言及其实现

徐家福　吕建　著

科学出版社

内 容 简 介

本书是一部软件语言专著。本书以作者多年来在软件语言及其实现方面之工作为基础，但又不囿于此，撰写中力求系统化，并贯穿作者近年之学习心得。全书十八章，除第一章引言外，其余各章按语言级别归为四篇，第二章至第七章为需求级语言篇；第八、第九两章为功能级语言篇；第十章至第十二章为设计级语言篇；第十三章至第十八章为实现级语言篇。各章除阐明基本概念与发展概况外，着重讲述作者设计之语言及其实现系统，内容侧重语言兼及实现，对语言源流、设计背景、设计思想、设计原则、成分取舍、应用实况、利弊得失等均有详细讨论，俾读者能谙其本质、理顺关系、辨其优劣、冀收触类旁通之效。

本书可作为高等学校计算机、电子、通信等有关专业高年级学生及研究生课程的参考用书，同时对软件研究与开发人员也具有指导意义和参考价值。

图书在版编目(CIP)数据

软件语言及其实现/徐家福，吕建著．－北京：科学出版社，2000
ISBN 7-03-008485-3

Ⅰ．软… Ⅱ．①徐… ②吕… Ⅲ．程序语言 Ⅳ．TP312

中国版本图书馆 CIP 数据核字（2000）第 07778 号

JS4B/26

科学出版社 出版
北京东黄城根北街 16 号
邮政编码:100717
新蕾印刷厂印刷

科学出版社发行 各地新华书店经销
*
2000 年 7 月第 一 版 开本:787 × 1092 1/16
2000 年 7 月第一次印刷 印张:18 1/4
印数:1—4 000 字数:415 000

定价:28.00 元

(如有印装质量问题,我社负责调换〈杨中〉)

序

软件乃程序及其文档之总称。细言之，有个体含义、总体含义、学科含义之别。软件与硬件犹如阴与阳、天与地，两者互为依存，互促发展，缺一不可。两者结合，组成计算机系统，就其学科含义而论，二者之发展构成计算机科学技术之发展。

软件之作用有三，一为用户与硬件之接口界面，二为计算机系统之指挥机构，三为计算机体系设计之重要依据。

软件语言乃用以书写软件之语言。按级别可分为需求级、功能级、设计级、实现级。需求级语言用以书写需求定义，功能级语言用以书写功能规约，设计级语言用以书写设计规约，实现级语言则用以书写实现算法。

软件语言之作用有三，一为描述作用，二为交流作用，三为标志作用。无软件语言即无软件，其地位举足轻重，它是软件发展之重要标志。

20 世纪 50 年代，余在莫斯科大学修习程序设计，使用机器语言书写程序，深感其“苦”。归国后，决心探索程序设计自动化（后发展为软件自动化），40 年来对上述四级语言及其实现进行学习研究，开发出多种语言与相应软件自动化系统，今将部分成果整理成书，名曰“软件语言及其实现”，以飨读者。

本书特点如下：

(1) 系统性

全书除第一章引言着重讲述软件语言之含义、作用、级别、发展以及本书之内容与体式外，其余各章按语言级别分为四篇，即需求级语言篇、功能级语言篇、设计级语言篇、实现级语言篇。笔者工作自底向上，篇章安排则自顶向下，内容反映 40 年来对软件语言进行之系统研究，具系统性。

(2) 科学性

本书取材于笔者主持参与开发且已通过鉴定之多个软件系统及已在国内外计算机学术期刊上发表之论文，内容几经推敲修改，力求准确无误，具科学性。

(3) 实用性

本书取材之系统有些已臻实用，有些虽未实用，亦已试用有年。至于途径方案之决择、语言成分之取舍、系统结构之设计、理论探索之成果均有参考价值，具实用性。

本书为一集体研究成果，参与者逾百人，诸君功不可没，于此深致谢意。

软件开发昨日困难，今日困难，明日亦复困难。然而，困难之性质与程度乃随发展阶段而异。余深信，“柳暗花明又一村”之境地终必到来。

余年逾古稀，胆敢言老，不揣浅陋，与吕君合著此书，冀图嘉惠来学，疏漏、欠妥、谬误之处，尚请读者指正。

徐家福

1999 年 10 月于南京大学

目　录

需求级语言篇

功 能 级 语 言 篇

设 计 级 语 言 篇

实 现 级 语 言 篇

第一章 引 言

本章为引言。内容包括软件语言含义、软件语言作用、软件语言级别、软件语言发展以及本书内容与体式五节。

1.1 软件语言含义

1.1.1 语言

语言乃信息交流工具,其源可追溯至上古之"结绳记事",然本书所论语言乃现代意义下之语言。今援引如下几种定义。

(1) Webster 字典定义(下作 W 定义)

An artificially constructed primarily formal system of signs and symbols (as symbolic logic) including rules for the formation of admissible expressions and for their transformation [基于一组记号与符号由人工构作之(基本上说)形式系统(如符号逻辑),包括合法表达式之形成规则与转换规则。][见 Webster's Third New International Dictionary, G&C Merriam Co, 1976, pp. 1270, c:]

(2) Longman 字典定义(下作 L 定义)

Any system of signs, movements, etc., used to express meanings or feelings(任何表情达意之记号系统)。[见 Longman Contemporary English-Chinese Dictionary, 现代出版社, 1988 年 11 月,第 789 页。]

(3) 英汉双解计算机辞典定义(下作"双计"定义)

A set of characters, conventions and rules, that is used for conveying information. The three aspects of language are pragmatics, semantics and syntax[一种用于传递信息之字符、约定和规则的集合。语言的三个方面是语用学(笔者按,此处应为语用)、语义学(笔者按,此处应为语义)和语法。][见"英汉双解计算机辞典",清华大学出版社,1996 年 2 月第 2 版,第 553 页。]

(4) 中国大百科全书,电子学与计算机卷定义(下称大百科定义)

语言的基础是一组记号和一组规则,根据规则由记号构成之记号串的总体就是语言。[见"中国大百科全书,电子学与计算机"卷 1("程序设计语言"词条),第 86 页,中国大百科全书出版社,1986 年 9 月。]

笔者认为,前述四种定义各有千秋,其中 W 定义基本可行,惜文字表达欠佳,且严谨不够;L 定义立论过泛,且欠严格;双计定义内容欠严谨;大百科定义相对较为合适,今以之为基础,将"语言"定义如下:

语言是基于一组记号与一组规则、根据规则由记号构成之记号串的总体。任何语言均包括语法、语义和语用三方面。

1.1.2 软件

计算机软件(简称软件)是计算机系统中的程序及其有关文档。程序是计算任务之处理对象与处理规则的描述,文档是为了便于理解程序所需之资料说明,程序必须装入到机器内部才能工作,文档一般是给人看的,不一定装入机器。

细言之,软件具有如下三层含义:

第一,个体含义,软件是指计算机系统中各别程序及其有关文档,如特定语言之编译程序及其有关文档。

第二,整体含义,软件是指计算机系统中个体含义下软件之总体。

第三,学科含义,软件是指开发与维护前述含义下之软件所涉及的理论、原则、方法与技术。在这种含义下,宜称为软件学,一般亦称软件。

1.1.3 软件语言

软件语言是用以描述软件的语言。申言之,软件语言用以描述软件、软件开发过程中间结果,以及软件开发过程本身(当然,后二者亦可视为软件)。软件语言质量关系到软件质量(如易读性、易维性、功效性、安全性等)、软件开发功效以及软件、软件技术、软件科学之发展。它一直是计算机系统之重要组成部分,一直是计算机科学与技术之重要组成部分。

1.2 软件语言作用

1.2.1 描述作用

软件语言是软件描述工具,程序及其文档赖于语言得以表述,未经描述的程序及其文档只能存在于开发人员脑中,无法为人知晓,从而也难以臻于实用。"内容决定形式,形式影响内容"。一方面,相对软件内容说来,语言是描述工具,描述是形式,但另一方面,语言本身也有内容与形式两面。语言成分是内容,语言描述(表示)是形式,因此,语言之优劣也会影响到软件内容,例如,语言结构决定了程序结构,程序结构又影响到程序之易读性、易维性、安全性等等。文章之好坏,意境固然重要,但文笔之通畅优美亦不容忽视。它往往影响到读者情绪,影响到作品传世之久远。举凡传世之作,莫不意境高超、文笔优美。程序设计语言 ALGOL 68 就其内容而论,确有不少创新之处,惜其"语言报告"文笔晦涩,不便阅读,难于流传。反之,PASCAL"语言报告"却文笔流畅,清晰易懂,二者成一鲜明对比,类似事例比比皆是,不胜枚举,描述之重要性可见一斑。

1.2.2 交流作用

软件语言又是软件交流工具。经软件语言描述之软件方可在社会上交流。如前所述,未经语言描述之软件只存在于开发人员脑中,无法进行交流。软件开发之目的是为了实用,为了在社会上得到广泛应用,以促进物质生产、科技文化、人民生活之发展与提高。但是,就特定软件而言,经语言描述后,原则上均可交流,但交流范围之广狭,生存期之长短,除了取决于软件本身质量外,其描述之优劣亦颇重要,历史上有关机构曾竭力推广

PL1 语言，然终难如愿，原因乃在于语言本身。ADA 语言虽经美国国防部硬性推广，且内容不断更新，能否经久而不衰，殊难逆料。由于计算机网络之盛行，由于社会信息化之发展，JAVA 语言一经推出，顿成热点，原因仍在语言本身，如能对之改进润色，更上一层楼，则用户幸甚。

1.2.3 标志作用

软件语言既是描述工具，又是交流工具，无软件语言即无软件，其地位可谓举足轻重。环顾软件发展历史，代表性语言莫不成为软件发展之重要标志。50 年代中期 FORTRAN 语言出现，显著减轻了程序人员的手工劳动，提高了程序设计功效，促进了计算机的推广应用。ALGOL 60 及其相应语言族之发展，推动了科学计算领域及其有关科技领域之发展。COBOL 语言出现大大推动了数据处理领域之发展。LISP 语言推动了表处理之发展。PASCAL 语言推动了程序设计语言教学以及科学计算领域之研究。SIMULA 67 语言开创了对象式(面向对象)语言之先河。随后，ADA, SMALLTALK, C, C + +, EIFFEL 以及新近开发之 JAVA 等语言均为对象式(包括基于对象)语言之发展标志。另一方面，各种函数式语言与逻辑式语言促进了计算机体系结构研究以及逻辑问题研究与发展，此外，设计级、功能级、需求级语言之发展促进了软件开发、软件开发过程、计算机辅助软件工程以及软件自动化等领域之研究与发展。凡此种种，莫不表明软件语言水平是软件科学技术水平之重要标志，同时也是计算机科学技术水平之重要标志。

1.3 软件语言级别

软件语言可区分为需求级语言、功能级语言、设计级语言以及实现级语言。

1.3.1 需求级语言

需求级语言意指用于软件需求阶段之语言，特别是用于书写软件需求定义(或规约)之软件需求定义语言。软件需求定义是软件需求之完整描述。软件需求包括功能需求和非功能需求。功能需求从用户角度明确软件必须具备之功能；非功能需求可包括功能限制、设计限制、环境描述、数据与通信规程和项目管理等。软件需求定义主要面向用户，采用基于客观外界之描述模型，以便于用户理解。

在计算机发展早期，待解问题规模较小，一般采用自然语言书写软件需求。随着待解问题规模日益增大，其复杂度不断提高，自然语言之非形式性导致需求定义出错，纠正并非易事。70 年代起，以软件方法学为基础，着手研究需求级语言形式化，先后提出基于自顶向下途径之“结构化分析”(SA)，基于自底向上途径之“问题陈述语言”(PSL)，基于对象式思想之“需求建模语言”(RML)，以及以对象及其相互关系为核心、以图形化表示机制为刻画手段、基于对象式需求模型之各种需求级语言。

按照形式化程度，需求级语言可分为非形式化语言、半形式化语言以及形式化语言三类。

需求级语言研究涉及基本模型、语言结构以及语用分析等。

需求级语言是需求工程核心内容之一，其研究已取得较大进展，现有各种语言已逐步

用于软件工程实践,效果良好,今后发展将更为迅速。

1.3.2 功能级语言

功能级语言意指用于书写软件功能规约之语言。软件功能规约是软件功能精确而完整之描述。它描述软件"要做什么",以及"只做什么"。功能规约是需求定义之功能抽象,它对需求定义中用户所需功能重组,使之尽可能表述为数学语言,以作为设计与实现依据。功能级语言又称功能规约语言。

在软件发展早期,软件功能规约主要用自然语言书写,用户易学易用。但自然语言之歧义性、模糊性和不完备性致使用以书写的功能规约难以成为设计与实现之依据,同时也难以开展软件形式化与自动化研究。为了提高软件生产率与软件产品质量,出现了形式化功能级语言(又称形式化功能规约语言)。这类语言理论严谨,便于研究规约之性质,如一致性、完备性、等价性等等。不仅避免了自然语言之歧义性、模糊性和不完备性,而且奠定了 WP 和 VDM 等形式化方法之基础,使软件自动化之实现成为可能。

从形式化程度看,功能级语言可分为非形式化语言与形式化语言。

功能级语言主要涉及规约对象、规约方法以及规约性质等。

功能级语言,特别是,形式化功能规约语言研究已有较大进展,其理论基础日臻完善,方法日趋成熟,并已逐步用于软件工程实践,进一步工作包括应用于大型软件开发以及基于形式化功能规约语言之软件形式化与自动化研究。

1.3.3 设计级语言

设计级语言意指用于书写软件设计规约之语言。软件设计规约包括概要设计规约与详细设计规约,其描述对象是软件系统组织或其组成部分之内部结构。设计级语言一方面不同于刻画软件及其组成部分界面特性之功能级语言,另一方面,它并不刻画重在执行高效之软件细节,从而也有别于实现级语言(即程序设计语言)。

在计算机发展早期,软件设计人员使用图形化或半图形化设计级语言,其中夹以自然语言正文描述,以书写概要设计规约或详细设计规约。稍后,相继出现表格形式之设计级语言和设计性程序语言。目前,形式化设计级语言颇受重视,在详细设计中应用尤多。

设计级语言主要涉及规约结构、规约对象、规约方法等。

按照传统观点,设计级语言应是可扩充语言。一方面,通过扩充,可以包容数据结构与控制结构新概念,如多任务、并行处理、进程间通信、声象界面等;另一方面,通过扩充,可支持不同应用领域之特定结构。然而,近来却倾向语言具有固定结构,这是由于计算机辅助软件工程(CASE)技术促进了设计过程自动化。因此,语言中宜包括表达能力颇强之成分,以适应不断发展之应用要求,例如,用户自定义抽象数据类型、类属模块、多继承机制、并行循环、非确定等待、进程之条件激活机制等,而领域特定结构则可通过领域专家库提供。

1.3.4 实现级语言

实现级语言意指用于书写处理算法的语言,即程序设计语言(又称编程语言),相对其它级语言而言,实现级语言发展最早、相对成熟,其优劣不仅影响程序人员使用是否方便,

程序人员所写程序质量之高低，而且对实际处理功效也影响至大。今就其基本成分、语言分类，以及主要语言简述如下：

语言种类千差万别。但是，一般说来，其基本成分不外四种，即用以描述程序中所含数据之数据成分；用以描述程序中所含运算之运算成分；用以描述程序中控制构造之控制成分；以及用以刻画程序中数据传输之传输成分。

按照语言级别，有低级语言和高级语言之分。低级语言包括字位码、机器语言和汇编语言。其特点是与特定机器有关，功效高，但使用复杂、繁琐、费时、易出差错。其中字位码是计算机唯一可直接理解之语言，但由于字位码程序为一串字位，复杂、繁琐、冗长，几乎无人直接使用。机器语言是表示成数码形式之机器基本指令集，或者是操作码经符号化后之基本指令集。汇编语言是机器语言基本指令中不仅操作码已符号化，而且地址部分也符号化后之结果，或者进一步还包括宏构造。高级语言表示方法要比低级语言更接近于待解问题之表示方法，其特点是在一定程度上与具体机器无关，易学、易用、易维护。当高级语言程序翻译成相应低级语言程序时，一个高级语言程序单位一般要对应多条低级语言指令，相应编译程序所产生之目标程序往往功效较低。

按照用户要求，有过程式语言和非过程式语言之分。过程式语言主要特征是，用户可以显式指明一列可顺序执行之运算，以表述相应计算过程。例如，FORTRAN, COBOL, ALGOL60 等都是过程式语言。非过程式语言的含义是相对的，凡是用户无法显式指明表述计算过程之一列可顺序执行之运算的语言都是非过程式语言。PROLOG 语言即其一例。

按照应用范围，有通用语言和专用语言之分。目标非单一的语言为通用语言，FORTRAN, COBOL, ALGOL60 等均是。目标单一的语言为专用语言，如 APT 等。

按照使用方式，有交互式语言和非交互式语言之分。具有反映人－机交互作用成分之语言为交互式语言，BASIC 即是。语言成分不反映人－机交互作用之语言为非交互式语言，前述 FORTRAN, COBOL, ALGOL60 以及 PASCAL 等均是。

按照成分性质，有顺序语言、并发(并行)语言和分布语言之分。只含顺序成分的语言为顺序语言，FORTRAN, COBOL 等均是。含有并发(并行)成分的语言为并发(并行)语言，并发 PASCAL, MODULA, ADA 等均是。考虑到分布计算要求的语言为分布语言，如 MODULA * 即是。

传统程序设计语言大都以冯·诺依曼式计算机为其设计背景，因而，又称为冯·诺依曼式语言，J. Backus 于 1977 年提出的函数式语言 FP 则以非冯·诺依曼式计算机为其设计背景，因而，又称为非冯·诺依曼式语言。

主要语言举例如下：

1) 自动数控程序语言 APT(Automatically Programmed Tools)：第一个专用语言，用于数控机床加工，1956。

2) 公式翻译程序设计语言 FORTRAN(FORmula TRANslation)：第一个广泛使用之高级语言，为广大科学和工程技术人员使用计算机创造了条件，1956。

3) 面向商业通用语言 COBOL(COmmon Business Oriented Language)：广泛使用之商用语言，1960。

4) 算法语言 60 ALGOL60(ALGOrithmic Language 60)：程序设计语言由技艺转向科

学之重要标志,其特点是局部性、动态性、递归性和严谨性,1960。

5) 表处理语言 LISP(LISt Processing):引进函数式程序设计概念和表处理设施,在人工智能领域内广泛使用,1960。

6) APL(A Programming Language):一种提供很多高级运算符之语言,它可使程序人员所写程序甚为紧凑,特别是涉及到矩阵计算之程序,直到 1967 年才定义出其实现文本,1962。

7) 模拟语言 SIMULA(SIMUlation LAnguage):一种主要用于模拟的语言,它是ALGOL60之扩充,1966。SIMULA67 是 1967 年 SIMULA 之改进,其中引进了"对象"及"类"等概念,它是第一个对象式语言。

8) 菲利浦自动顺序计算机语言 PASCAL(Philips Automatic Sequence CALculator):它是在 ALGOL60 基础上发展起来的重要语言,其最大特点是简明性与结构性,1971。

9) 逻辑程序设计语言 PROLOG(PROgramming in LOGic):一种处理逻辑问题之语言,它已广泛用于关系数据库、数理逻辑、抽象问题求解、自然语言理解等多种领域,1971。

10) SMALLTALK:一种对象式程序设计语言。自 1971 年出现后,曾有多种不同文本,其中使用最广的是 SMALLTALK80,其显著特点是利用"对象",以使对象式程序设计得到广泛应用,1971。

11) C:一种使用颇为广泛的程序设计语言。它原先为辅助开发 UNIX 操作系统而设计,后来广泛用于研究、开发与教学,主要用于系统程序设计,同时也用于其它领域,1974。

12) FP:一种无副作用、具有组合子风格的函数式程序设计语言。其特点是引用透明、便于表示递归函数、具有潜在并行性以及具有良好代数性质。由 John Backus 提出,1977。

13) ML:一种严格的函数式程序设计语言,但非纯函数式,非引用透明,且其 I/O 系统具有副作用。由 Robin Milner 提出,1975～1976。

14) ADA:一种现代模块化语言,它属于 ALGOL-PASCAL 语言族,但有较大变动。其主要特征是强类型化和模块化,便于实现分别编译,提供类属设施与异常处理,适于嵌入式应用,1979。

15) MIRANDA:一种非严格的纯函数式程序设计语言,具有惰性计值和多态强类型特征,应用较为广泛,已有商业产品。由 David Turner 提出,1985～1986。

16) C++:一种使用广泛之对象式通用程序设计语言,它具有 SIMULA 语言之程序组织设施以及 C 语言之灵活性与功效性,1986。

17) EIFFEL:一种对象式通用程序设计语言,该语言严格区分静态类和动态对象,类是唯一程序构造单位,并支持多继承,1987。

18) JAVA:一种适于网络的程序设计语言。其特点是简明性、对象式、分布性、健壮性、安全性、多平台运作、易移植性、多线程等,1994。

除了上面所举语言外,还有一些通用语言,特别是 BASIC, PL/1, SNOBOL, ALGOL68等。BASIC 虽然简单、易学,使用广泛,但其中并无新概念,而且并非第一个交互式语言。PL/1 之设计思想源于 JOVIAL,其功能源于 FORTRAN, COBOL, ALGOL60,具有中断和表处理等设施。SNOBOL 是一种好语言,对 COMIT 中若干概念做了明显改进。ALGOL68 在语言成分和描述方法方面虽有所创新,但应用尚不广泛。