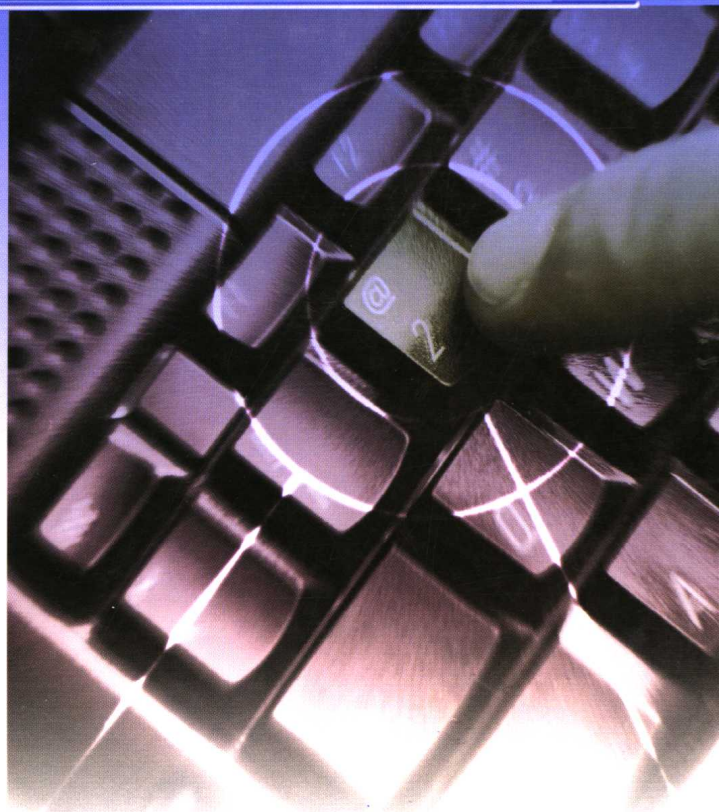


///

高等学校计算机基础教育教材精选



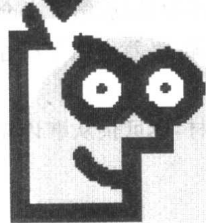
冯萍 主编

计算机软件技术 及应用基础



清华大学出版社

TP31
161



高等学校计算机基础教育教材精选

计算机软件技术

及应用基础

冯萍 主编

朱明 孔庆芸
赵光飞 姬亚利
编著

清华大学出版社
北京

北方工业大学图书馆



00543644

内 容 简 介

全书共13章。第1章简单介绍了程序和软件的基本概念,第2章介绍了VB程序设计的特点和Windows编程技术,第3章介绍了面向对象程序设计方法,第4章介绍了软件工程,第5章和第6章介绍了数据库系统、结构化查询语言SQL、VB数据库编程技术以及网络数据库编程基础,第7章介绍了管理信息系统,第8章和第9章介绍了计算机网络和网络编程技术,第10章至第12章介绍了线性和非线性数据结构,以及查找与排序方法,第13章介绍了多媒体技术。

本书语言通俗易懂,每章后附有习题,可作为高等学校非计算机专业的教材,也可供从事计算机软件工作的工程技术人员参考。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机软件技术及应用基础/冯萍主编. —北京:清华大学出版社,2004

(高等学校计算机基础教育教材精选)

ISBN 7-302-07905-6

I. 计… II. ①冯… III. 软件—高等学校—教材 IV. TP31

中国版本图书馆CIP数据核字(2003)第125780号

出 版 者: 清华大学出版社

<http://www.tup.com.cn>

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

组稿编辑: 焦 虹

文稿编辑: 张为民

印 装 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 23 字数: 528 千字

版 次: 2004年2月第1版 2004年2月第1次印刷

书 号: ISBN 7-302-07905-6/TP·5741

印 数: 1~5000

定 价: 29.00 元

本书如存在文字不清、漏印以及缺页、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)62770175-3103 或 (010)62795704

出版说明

——高等学校计算机基础教育教材精选——

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验优秀的教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,面向各高校开设的计算机必修课、选修课以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本,出版一本,并保持不断更新)。坚持宁缺勿滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是出版质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是: jiaoh@tup.tsinghua.edu.cn。联系人: 焦虹。

清华大学出版社

2001年8月

前言

——计算机软件技术及应用基础——

为了贯彻教育部“面向 21 世纪教学内容与课程体系改革”的精神,适应计算机技术的迅速发展,工科计算机基础课程教学指导委员会确定了高等学校计算机基础教学的课程体系。本书为计算机的软件技术及应用基础类课程教材。

本书参考了国际上著名的计算机教材,力求使教材体系结构具有先进性、系统性、稳定性和发展性的特点。本书编写的指导思想是“宽编窄用”,以培养学生具有初步的应用系统软件应用开发能力为目标,以应用开发技术理论为主线,探索以学为主的教学模式,课堂教学覆盖不到之处让学生自学,重视学生创新能力和综合应用能力的培养。

根据目前软件技术的发展,以及近几年教学改革实践的成果,在教材中介绍了计算机发展的最新应用技术,例如,可视化编程技术、面向对象以及网络编程技术。我们认为综合应用能力的培养对于工科大学生来说,应该是其知识结构的一个重要部分,因此,对于这门课程的教学目标,定位在使学生掌握应用系统软件设计的基础知识,初步具有进行一个小型应用系统软件设计的基本能力上。

本书涵盖了软件工程、面向对象技术、数据库、计算机网络等方面的基础知识。在软件技术基础知识部分,主要介绍了程序、软件、面向对象、数据结构、计算机网络、关系数据库系统以及软件工程等基本知识;在软件设计方法部分,介绍了可视化、数据库以及网络编程技术。

本书共 13 章,第 1、8 章由冯萍执笔,第 2 章由姬亚利执笔,第 5、6、7、13 章由孔庆芸执笔,第 3、4、9 章由赵光飞执笔,第 10~12 章由朱明执笔,刘君瑞、孙蓬参与了编校工作,全书由冯萍统稿。冯博琴教授审阅了本书,对本书的教材体系结构进行了指导,在此表示感谢。由于时间仓促,水平有限,书中不妥和谬误之处在所难免,恳请读者批评指正。

编 者

2004 年 1 月

目录

—— 计算机软件技术及应用基础 ——

第 1 章 概论	1
1.1 程序设计语言	1
1.1.1 程序设计语言的分类	1
1.1.2 高级程序设计语言概述	1
1.1.3 程序设计语言的评价标准	5
1.1.4 程序设计语言的选择标准	5
1.2 程序的基本概念	6
1.3 软件的基本概念	6
1.4 应用系统软件的开发	11
习题 1	12
第 2 章 VB 程序设计基础	13
2.1 VB 程序设计特点	13
2.2 VB 编程基础	15
2.2.1 VB 的用户界面	15
2.2.2 VB 可视化开发和事件驱动	16
2.2.3 建立一个简单的 VB 应用程序	20
2.3 VB 语言基础	23
2.3.1 基本概念	23
2.3.2 变量、常量和数据类型	27
2.3.3 VB 语句与控制流	35
2.3.4 过程	41
2.4 Windows 编程技术	50
2.4.1 窗口、事件和消息	50
2.4.2 应用程序框架	51
2.4.3 过程编程	52
2.5 Windows API	54
2.5.1 Win32 API	54
2.5.2 在 VB 中使用 Windows API	54

习题 2	60
第 3 章 面向对象的程序设计方法	62
3.1 面向对象的程序设计	62
3.2 面向对象方法学导论	63
3.2.1 什么是对象	63
3.2.2 面向对象的概念	65
3.2.3 面向对象建模	68
3.2.4 对象模型	68
3.2.5 面向对象的分析	71
3.2.6 面向对象的设计	72
3.2.7 面向对象的实现	73
3.2.8 面向对象的程序实例	74
3.3 用 C++ 语言编程	77
3.3.1 定义类	77
3.3.2 说明数据成员	78
3.3.3 说明和定义成员函数	79
3.3.4 完整的 C++ 程序	80
习题 3	84
第 4 章 软件工程	86
4.1 软件工程概述	86
4.2 软件工程模式	89
4.3 软件生存周期	89
4.3.1 软件生存周期各个阶段的主要任务	90
4.3.2 软件开发模型	91
4.4 面向对象的设计	93
4.4.1 面向对象的分析	93
4.4.2 面向对象的设计方法	94
4.4.3 基于对象的设计步骤	96
4.4.4 面向对象的实现	96
4.4.5 面向对象的测试	97
4.5 软件维护	97
习题 4	98
第 5 章 数据库基础	99
5.1 数据库理论	99
5.1.1 数据模型	99



5.1.2	数据库的体系结构	101
5.1.3	关系模型	103
5.1.4	关系数据库理论	106
5.1.5	关系数据库的完整性和安全性	110
5.1.6	数据库分类及发展	113
5.2	结构化查询语言(SQL)	118
5.2.1	SQL 概述	118
5.2.2	SQL 的数据定义	120
5.2.3	SQL 的数据操纵	125
5.2.4	SQL 的数据控制	131
习题 5		132
第 6 章	VB 数据库编程技术	134
6.1	数据库设计基础	134
6.2	本地数据库设计	136
6.3	数据控件	140
6.3.1	数据控件属性	140
6.3.2	数据控件的事件和常用方法	143
6.3.3	记录集的属性与方法	145
6.3.4	数据库记录的增加、删除和修改	148
6.4	ODBC	152
6.5	ADO 数据控件	154
6.6	VB 中 SQL 的实现	159
6.7	在 VB 中使用 ODBC	164
6.8	网络数据库设计	168
6.8.1	客户机/服务器数据库系统	168
6.8.2	VB 中的网络数据库设计	170
习题 6		172
第 7 章	管理信息系统	174
7.1	概述	174
7.2	管理信息系统的开发	175
7.2.1	管理信息系统的开发环境	175
7.2.2	数据库技术和网络技术	176
7.2.3	MIS 系统发展趋势——Intranet	179
7.3	管理信息系统应用举例	180
习题 7		181

第 8 章 计算机网络基础	182
8.1 概述	182
8.2 计算机网络体系结构	184
8.2.1 层次结构	184
8.2.2 ISO/OSI 网络体系结构	185
8.3 局域网技术	192
8.3.1 以太网	193
8.3.2 IEEE 802 标准	194
8.4 TCP/IP 协议	198
8.4.1 TCP /IP 分层	198
8.4.2 IP 协议	200
8.4.3 TCP 协议	202
8.5 网络互联技术	205
8.6 WWW 技术	206
8.6.1 浏览器/服务器	207
8.6.2 超文本传输协议	208
8.6.3 Internet 信息服务器	209
习题 8	211
 第 9 章 网络编程技术	 213
9.1 HTML 基础	213
9.1.1 HTML 语言基础	213
9.1.2 DHTML 语言基础	224
9.2 ASP 编程技术	230
9.3 Java 语言的网络程序设计	235
9.3.1 Java 的特征	235
9.3.2 Java 的语法机制	237
9.3.3 Java 与 Internet 环境下的软件开发	242
9.4 JSP 编程技术	252
习题 9	259
 第 10 章 线性数据结构	 260
10.1 数据结构概述	260
10.1.1 什么是数据结构	260
10.1.2 数据类型与抽象数据类型	261
10.1.3 算法特性和算法效率	262
10.2 线性表结构及相关算法	264
10.2.1 线性表的逻辑结构及运算	264

10.2.2	线性表的顺序存储结构	265
10.2.3	线性表的非顺序存储结构——链表	269
10.3	堆栈和队列	276
10.3.1	堆栈	276
10.3.2	队列	280
10.4	串和数组	285
10.4.1	串	285
10.4.2	数组	287
10.5	算法示例	290
习题 10		294
第 11 章	非线性数据结构	296
11.1	树	296
11.1.1	树的定义及基本概念	296
11.1.2	二叉树	299
11.1.3	二叉树的存储结构	301
11.1.4	二叉树的遍历	302
11.1.5	二叉树遍历的序列	304
11.1.6	树、森林与二叉树的转换	306
11.2	图	307
11.2.1	图的基本概念和术语	307
11.2.2	图的存储结构	309
11.2.3	图的遍历	311
习题 11		313
第 12 章	查找与排序	315
12.1	查找的基本概念	315
12.2	基于线性结构的查找算法	315
12.2.1	顺序查找算法	315
12.2.2	折半查找算法	317
12.2.3	分块查找算法	318
12.3	基于二叉排序树结构的查找算法	319
12.4	哈希表及其查找算法	321
12.4.1	哈希表的基本概念	321
12.4.2	哈希函数的构造方法	322
12.4.3	冲突解决的方法	323
12.4.4	哈希查找	325
12.5	排序	325

12.5.1	基本概念	325
12.5.2	直接插入排序	326
12.5.3	直接选择排序	327
12.5.4	冒泡排序	327
12.5.5	快速排序	328
习题 12		330
第 13 章	多媒体技术	331
13.1	多媒体定义与多媒体计算机	331
13.2	图形与图像处理及多媒体图像压缩与编码技术	333
13.2.1	图形与图像处理	333
13.2.2	多媒体图像压缩与编码技术	335
13.3	VB 多媒体编程技术	338
13.3.1	VB 多媒体编程技术概述	338
13.3.2	媒体控制接口	339
13.3.3	使用 API 函数设计多媒体程序	345
13.3.4	使用 OLE 控件进行多媒体程序设计	350
习题 13		353
参考文献		354



1.1 程序设计语言

1.1.1 程序设计语言的分类

程序设计语言可以从不同的角度进行分类。

从发展过程来分,计算机程序设计语言可分为第一代语言(机器语言)、第二代语言(汇编语言)、第三代语言(高级语言)、第四代语言与第五代语言。

从应用范围来分,计算机程序设计语言可分为通用语言与专用语言;再细分又可分为系统程序设计语言、科学计算语言、事务处理语言、实时控制语言、教学用语言以及用于解决非确定性问题的语言等。

从程序设计方法上来分,计算机程序设计语言可分为结构化语言、模块化语言与面向对象语言。

1.1.2 高级程序设计语言概述

计算机能读懂的语言是机器码,但对人来说,由1和0组成的二进制程序既难编写又难读懂。于是出现了用英文字母代表操作码的汇编语言,汇编语言是机器语言的符号化,汇编语言是面向机器的,使用汇编语言编程需要直接安排存储,规定寄存器和运算器的动作次序,汇编语言与计算机紧密相关,不同的计算机在指令长度、寻址方式、寄存器数目和指令表示等方面都不一样。由于汇编语言不便于进行数学描述,而且不可移植,于是出现了高级语言。高级语言是面向计算过程的和面向问题的语言,只与解题的步骤有关,而将高级程序设计语言“翻译”成机器语言的工作则是由编译程序来完成的。程序员的工作是把要计算的问题化成高级程序设计语言的表达式、语句、过程/函数和对象,而不是机器指令序列。

把高级语言程序翻译成机器语言程序有两种做法:编译和解释。相应的翻译工具分别叫做编译器和解释器。以下分别讲述。

1. 编译器工作原理

编译器逐行扫描高级语言源程序,编译的过程如下。

① 词法分析(Lexical Analysis)。识别关键字、字面量、标识符(变量名、数据名)、运算符、注释行以及特殊符号(续行、语句结束和数组)等6类符号,分别归类等待处理。

② 语法分析(Syntax Analysis)。把每个语句看作一串记号(Token)流,由语法分析器进行处理,按照语言的文法检查判定是否是合乎语法的句子。如果是合法的句子,就以内部格式保存,否则报错,直至检查完整个程序。

③ 语义分析(Semantic Analysis)。语义分析器对各句子的语义做检查:运算符两边类型是否兼容;该做哪些类型转换,例如,实数向整数赋值要“取整”;控制转移是否转移到了不该去的地方;是否有重名或者语义含糊的记号;等等。如果有错误,则转出错处理;否则可以生成执行代码。

④ 中间代码生成。中间代码是向目标码过渡的一种编码,其形式尽可能靠近机器的汇编语言,以便下一步的代码生成,但中间码不涉及具体机器的操作码和地址码。采用中间码的好处是可以在中间码上做优化。

⑤ 优化。对中间码程序做局部优化和全局(整个程序)优化,目的是使程序运行更快,占用空间最小。局部优化是合并冗余操作,简化计算。例如, $x := 0$ 可用一条“清零”指令替换。全局优化包括改进循环,减少调用次数和采用快速地址算法等。

⑥ 代码生成。由代码生成器生成目标机器的目标码(或汇编)程序,其中包括数据分段、选定寄存器等工作,以及生成机器可执行的代码。

高级语言源程序经编译得到目标码程序后,还不能立即装入机器执行,因为程序中如果用到标准函数(它们生成的目标码已存放在模块库中),还需对编译后得到的目标模块进行连接。连接程序(Linker)找出需要连接的外部模块,然后到模块库中找出被调用的模块,调入内存并连接到目标模块上,形成可执行程序。执行时,把可执行程序加载>Loading)到内存中的合适位置(此时得到的是内存中的绝对地址)就可执行了。这一过程如图1.1所示。

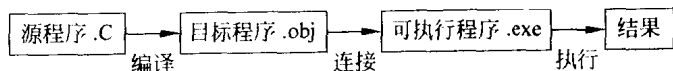


图 1.1 编译、连接和执行程序的过程

2. 高级语言程序的解释执行

编译型语言由于可进行优化(有的编译器可做多次优化),目标码效率很高,因此是目前软件开发的最主要编程语言。常见的程序设计语言,如C/C++、Pascal和FORTRAN等都是编译型语言,用这些语言编写的源程序,都需要进行编译和连接,然后才能生成可执行程序。这对于大型程序、系统程序和支持程序来说是十分有利的,虽然编译时花费了不少时间,但程序的执行效率很高。不过,在对程序的执行效率要求不高的场合,没有必要在编译上花费大量的时间,可以对高级语言源程序采取解释执行的方式。



解释执行需要有一个解释器(Interpreter),它将源代码逐句读入。第一步先作词法分析,建立内部符号表;再作语法和语义分析,并作类型检查(解释语言的语义检查一般比较简单,因为它们往往采用无类型或动态类型系统)。完成检查后即把每一语句压入执行堆栈,并立即解释执行。因为解释执行时只看到一个语句,无法对整个程序进行优化。但是解释执行占用空间很少。

操作系统的命令、Visual Basic、Java 和 JavaScript 都是解释执行的(其中有些语言也可以编译执行)。解释器不大,工作空间也不大。不过,解释执行难于优化、效率较低,这是解释类语言的致命缺点。

3. 高级程序设计语言的基本特征

下面介绍程序设计语言最基本的特征。

1) 变量、表达式和赋值

高级语言程序最基本的功能包括计算和输出计算结果。在编程时,计算的数据对象特别是计算结果的对象一般用变量表示。表达式是常量、变量、函数调用或者是由它们和运算符组成的序列。

赋值和函数调用是程序语言改变变量值的基本手段。

2) 程序的控制结构

高级语言程序最基本的控制结构包括如下部分。

① 条件分支语句: 条件分支语句使用条件判断语句 if...then...else...end if。

② 迭代语句: 迭代语句使用当循环语句 while ($I \leq \dots$) do...end do, 或者使用直到循环语句 do...until (E) ..., 也可以使用计数循环语句 for($I = E_0; E_1; E_2$) do...end do。

③ Case 语句:

```
Case I
I1: S1
I2: S2
...
In: Sn
End Case
```

3) 数据类型

计算机中所有程序设计语言都提供下面几种常用的基本数据类型。

① 纯量(常量、变量): 整型(Int/Integer); 实型/浮点类型(Real/Float/Double); 布尔型/逻辑型(Boolean/Logical); 字符型(Char/Character); 字符串类型(String); 枚举类型(Enum)。

② 数组: 代表单个数据值的变量称作纯量变量。如果代表多个(数组)或多种(记录)数据值的变量称作结构型数据变量。数组的特征是所有元素均为同一类型。数组维度可以扩展下去成为多维数组。

③ 记录: 相同或不同类型数据组成的结构叫记录。记录型数据记录了对对象的属性信息,记录的各个组成部分称为记录域,各个域的数据类型可以不相同。

4) 指针类型

变量表示内存中的存储单元,如果在一个变量中存放的不是对象(如数值和字符等),而是另一个变量在计算机内存中的存储地址,则通过这个地址就可以找到另一个变量,把这种变量叫做指针变量,即指向一个变量的变量。

如果给指针赋了初值(在编译时),并且该值在程序的运行期间不再改变,就称这种指针为常量指针,在 C++ 中也叫做引用。

5) 过程

将需要重复执行的内容编制成一个子程序称为过程,在主程序中可以重复调用这个过程。函数过程是参数化的程序。使用过程还可以降低程序的复杂性,使程序结构变得清晰明了。

(1) 过程的定义。

从过程关键字开始到过程结束之间的一段封闭的程序就是过程定义,例如:

```
Function FunctionName (形式参数表) As 返回类型
[类型和数据声明]
语句集
End Function
```

(2) 过程调用。

函数过程的返回值可以出现在主程序的表达式中。具体做法是以函数名引用函数值,并列出与形式参数表变元的个数、类型和次序一样的实在参数表,例如:

```
PrivateVariable= FunctionName (实在参数表)
```

过程调用执行时,形参和实参匹配后,过程的执行和主程序中其他代码的执行没什么两样。这时,主程序中声明的变量为全程量,在过程中也可以使用。但反过来不行,过程中声明的变量为局部量,在过程执行完之后,所有的数据和过程体均消失。

(3) 过程的参数传递。

过程是一段封闭的程序,数据的入口是参数表,过程只有通过形、实参数匹配才能具体执行。从主程序向过程传递数据可以有两种方式:传值和引用。

按引用传递参数时,参数表中传递的是地址,过程需要到主程序该地址中去取值,它引用了主程序中的值。同理,过程体中对该值有了改变也立刻改变了主程序该地址中的值。

按值传递参数时,即把主程序该地址的值复制到过程体定义的单元中,在过程体中无论将该单元赋何值,主程序该地址单元原有值也不会改变。局部程序不会改变全局变量的值。其缺点是需要双倍存储单元。

对于传引用和传值,各种程序设计语言都有相应的表示法,在 C 语言中,如果不加任何修饰则一律传值,要传引用可传指针值。VB 正好相反,如果不加修饰则按引用传递参数,加上 ByVal 就是传值。

6) 输入/输出

程序的输入/输出分为两大类:一类是程序之间以文件形式进行数据传递;另一类是人-机交互,把数据按一定格式输入到程序变量中,或者按用户要求的格式显示或打印输

出。程序的输入/输出一般由高级程序设计语言以过程调用（标准过程）的形式实现。

1.1.3 程序设计语言的评价标准

程序设计语言是用来开发软件的工具。根据其特性,对程序设计语言有以下 7 种基本评价标准:

- (1) 用户界面好,操作简便;
- (2) 对硬件、软件环境要求低;
- (3) 对硬件资源利用的程度高;
- (4) 代码质量高,目标程序运行速度快,可靠性高,结构化性能高,可读性好;
- (5) 应用范围广泛;
- (6) 调用其他软件容易;
- (7) 自身发展快。

如果一种程序设计语言能在上述 7 个方面都有较高的得分,那它就是一种比较理想的语言。

1.1.4 程序设计语言的选择标准

开发软件时,需要选择一种合适的程序设计语言作为开发工具,一般可以考虑按以下 4 个方面作为选择的标准:

1) 应用范围

各种程序设计语言都有自己的适用范围。在科学计算领域,FORTRAN 常常是首选语言。在事务处理方面,Visual Basic 语言是较理想的选择。在实时处理方面,选择汇编语言及 C 语言比较合适。在开发应用系统软件方面,可选择 Visual C++、Visual J++、Borland C++、Visual Basic、JBuilder、Delphi 及 C++ Builder 等语言。如果开发的软件中含有大量的数据操作,则可采用 SQL 和 Visual Foxpro 等数据库语言。在网络开发中,Web 服务器软件可以选择 Microsoft 公司的 IIS、Apache 组织开发的 Apache 服务器等。Web 浏览器可以选择 Netscape 公司的 Communicator、Microsoft 公司的 Internet Explorer 等。还可以选择使用文件传送工具 FTP、远程访问工具 Telnet、邮件软件如 Eudora 和 Pegasus 等。此外还有信息检索、多媒体和 Web 页创建工具等。为辅助操作系统有效地完成系统的管理和维护,可选择的软件有反病毒软件 Rising、KV3000 和 Kill 等;文件压缩工具 Winrar 和 Winzip 等;磁盘维护与诊断工具 Norton Utility 等;动态调试工具 Debug 等。

2) 算法及计算复杂性

FORTRAN、Pascal、C 和 C++ 等语言都能支持较复杂的算法与计算。而绝大多数数据库语言都只能支持较简单的算法与计算。

3) 数据结构

C、C++ 和 Pascal 语言都提供了数组、记录数据结构及带指针的动态数据结构。它们适合于设计系统程序以及需要复杂数据结构的应用程序。

4) 效率

有些实时应用系统要求具有很快的响应速度,可以选择汇编语言或 C 语言。有时应用系统中只是某一部分要求具有很快的响应速度,在这种情况下,可以选用汇编语言来编写这一部分程序,而其他部分可以选用高级语言。

此外,还要考虑语言所要求的运行环境。

1.2 程序的基本概念

1. 程序是程序设计语言的抽象符号的集合

程序设计语言有面向机器的汇编语言,以及面向过程和面向对象的高级程序设计语言。程序是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。程序员的工作就是把需要计算的问题化成高级程序设计语言的表达式、语句、过程/函数和对象。因此,程序是程序设计语言的抽象符号的集合。

2. 程序是对数据施行算法的过程

使用程序设计语言提供的符号和语法编写程序,要达到时间和空间性能的最佳,需要遵循一定的算法(即解题的算法)编定程序的执行步骤,同样的数据用不同的算法实现,其所需的时空开销大不相同。因此,在不同情况下需要选用不同算法来达到效率和质量的最佳。

从对象的角度来看,程序是对对象的行为和状态以及对象间的关系的描述。在面向对象的程序里,同样需要算法,因为,状态和属性是用数据表示的,状态改变的方法就包含了算法。只是面向对象语言提供了更抽象的对象概念。

3. 程序具有层次性

高层程序可以调用低层程序提供的服务,当需要调用低层服务时,高层程序就暂停执行(挂起)转而执行低层子程序,待低层子程序执行完后返回,又恢复上一层次程序的执行,这种程序结构称为嵌套结构。

高层程序可以利用低层系统提供的服务。层次结构由高到低的顺序为应用软件→工具集→操作系统→计算机硬件。操作系统利用计算机硬件提供的服务实现自身的功能,在操作系统之上的系统软件工具集如数据库系统等,利用操作系统提供的服务实现自身的功能。位于最高层的应用软件利用操作系统和工具集提供的服务实现自身的功能。

1.3 软件的基本概念

软件是计算机系统的重要组成部分,它是计算机程序、方法、规则、程序有关的各种文档以及在计算机上运行所必需的数据的总称。计算机程序是用计算机语言编写的,用于

