

高职高专计算机系列教材

主编 谭浩强

悦语言程序设计

谭浩强 著

清华大学出版社

内 容 简 介

悦语言功能丰富,使用灵活,可移植性好,既具有高级语言的优点,又具有低级语言的许多特点,既可以用来编写系统软件,又可用于编写应用软件,是国内外广泛使用的计算机语言。悦程序设计是计算机应用人员应掌握的基本功。

针对悦语言比较难学的情况,作者对全书内容做了精心安排,分解难点,减小台阶,用通俗易懂的语言和丰富的例题解释清楚复杂的概念。

本书是高职高专学生学习悦语言程序设计的理想教材。凡具有计算机初步知识的读者都能读懂本书。本书也可作为悦语言培训教材,也可供自学参考。

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。

摇

书摇名:悦语言程序设计

作摇者:谭浩强摇著

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

课表:计算机专业必修课程

印刷者:北京国马印刷厂

发行者:新华书店总店北京发行所

开摇本:16开,1.91米×2.85米,印张:10.5,字数:250千字

版摇次:1995年11月第1版,1995年11月第1次印刷

书摇号:ISBN 7-302-01911-1

印摇数:10000~15000

定摇价:12.00元

教授计算技术的大师
普及现代科技之巨擘

敬颂谭浩强教授创立三系成就

宋健

一九九五年一月

▲ 原国务委员、国家科委主任、现全国政协副主席、
中国工程院院长宋健同志给谭浩强教授的题词

编辑委员会

摇摇摇摇主摇摇任摇摇谭浩强

副 主 任摇摇焦金生 陈 明 丁桂芝

委 员摇摇(按姓氏笔画排序)：

王智广摇摇刘荫铭摇摇朱桂兰摇摇李文英

李 琳摇摇李志兴摇摇孙 慧摇摇武绍利

张 玲摇摇张克善 郝 玲 袁 玫

訾秀玲摇摇薛淑斌摇摇谢 琛

序

《高职高专计算机系列教材》

摇摇摇摇摇

到 21世纪,计算机将成为人类的常用现代工具,每一个有文化的人都应当了解计算机,学会使用计算机,并用它来处理面临的事务。

学习计算机知识有两种不同的方法:一种是侧重知识的学习,从原理入手,注重理论和概念;另一种是侧重应用的学习,从实际入手,注重掌握其应用方法和技能。不同的人应根据其具体情况选择不同的学习方法。对大多数人来说,计算机是作为一种工具来使用的,主要以应用为目的,以应用为出发点。对于高职和高专的学生,显然应当采用后一种学习方法。

传统的理论课程采用以下的三部曲:提出概念——解释概念——举例说明,这适合前面第一种方法。对于侧重应用的学习者,我们在教学实践中摸索出新的三部曲:提出问题——介绍解决问题的方法——最后归纳出一般规律或概念。实践证明这种方法是行之有效的,减少了初学者在学习上的困难。传统的方法是:先理论后实际,先抽象后具体,先一般后个别。我们采用的方法是:从实际到理论,从具体到抽象,从个别到一般,从零散到系统。我们认为,后一种方法对高职、高专和成人高教是很合适的。

本系列教材是针对高职和高专的特点组织编写的,包括了高职高专的计算机专业和非计算机专业的教材和参考书。不同专业可以从中选择所需的部分。本系列教材包含的内容比较广,除了可作为正式教材外,还可作为某些专业的选修课或指定自学的教材。

应当指出,检查学习好坏的标准,不是“知道不知道”,而是“会不会用”,学习的目的全在于应用。因此,希望读者一定要重视实践环节,多上机练习,千万不要满足于“上课能听懂、教材能看懂”。有一些问题,别人讲半天也不明白,自己一上机就清楚了。教材中有些实践性比较强的内容,不一定在课堂上由老师讲授,而应指定学生通过上机掌握。这样做可以培养学生的自学能力,启发学生的求知欲望。

本系列教材是由“浩强创作室”组织北京和天津一些普通高校和高职大学的老师们编写的,他们对高职高专的教学特点有较多的了解,有较多的实践

经验。相信本系列教材的出版会有助于高职高专的教材建设和教学改革。

由于我国的高职教育正在蓬勃发展,许多问题有待深入讨论,新的经验将会层出不穷,对如何进行高职教育将会有更新更深入的认识,本系列教材的内容也将会不断丰富和调整。我们只是为了满足许多高职高专学校对教材的急需,才下决心抓紧编写了这套系列教材,以期抛砖引玉。清华大学出版社克服了许多困难,使本系列教材在较短的时间内得以出版。

本系列教材肯定会有不足之处,请专家和读者不吝指正。

《高职高专计算机系列教材》主编
全国高等院校计算机基础教育研究会会长

谭浩强

1999年 11月 11日

前言

摇摇摇摇摇

程序设计是软件工作人员的基本功。在高等学校,一般都开设程序
摇摇**程**设计课程,根据不同的需要选用不同的计算机语言。悦语言由于其功能强、使用灵活、可移植性好、目标程序质量好而受到广泛的欢迎。所有的计算机专业和许多理工科专业都开设悦语言程序设计课程。悦语言已成为软件工作者必须掌握的一个工具。

悦语言既具有高级语言的优点,又具有低级语言的许多特点,既可以用来编写系统软件,又可以用来编写应用软件。悦语言语法限制不严格,程序设计自由度大,在使用时会出现一些“副作用”,因此掌握悦语言要比掌握一门其他高级语言相对难一些。所以,学习悦语言最好有学习过一门其他语言(如**程序**)的基础。

针对许多人想学悦语言而悦语言又难学的情况,作者在**1985**年编写出版了一本《悦程序设计》教材,由清华大学出版社出版。该书采用了新的体系结构,分散难点,减小台阶,用人们易于理解的方式叙述清楚复杂的概念,受到广大读者的欢迎。许多人说,看了这本书,感到悦语言不难学了。许多高职和大专学校选用这本书作为教材,收到很好的效果。

为了满足高职高专教学改革的需要,我们决定组织出版“高职高专计算机系列教材”。在讨论编写教材时,许多学校的同志一致推荐《悦程序设计》作为高职高专的教材。作者根据大家的意见,在《悦程序设计》一书的基础上,做了必要的修改与补充,编写出这本《悦语言程序设计》,供高职高专的师生使用。

在这本书中,保留了《悦程序设计》一书的特点:体系合理、逻辑清楚、例题丰富、通俗易懂。同时又根据悦语言新标准的规定,对该书进行了改写,使之符合当前的需要。

学习悦语言时有一点应当注意:应该把精力放在最基本、最常用的内容上。开始时不要钻牛角尖,在一些具体细节上死抠。有一些细节,随着对悦语言的了解逐步深入和实践经验的逐步丰富,会自然地掌握的,而有一些细节则要通过长期的实践才能真正熟练掌握。开始时切忌滥用悦的某些容易引

起错误的细节(如不适当地使用 垣垣和 原原)。

程序设计是一门实践性很强的课程,不可能只靠听课和看书就能掌握 悦语言程序设计。应当十分重视自己动手编写程序和上机运行程序。上机的时间愈多愈好。

为了帮助读者学习本书,作者还编写了一本《悦语言程序设计题解与上机指导》,提供本书中各章习题的参考答案以及上机实习指导,由清华大学出版社于 1995 年出版。

本书如有不足之处,请读者指正。

谭浩强

1995 年 1 月于北京

目录

第1章 C语言概述	1
1.1 C语言出现的历史背景	1
1.2 C语言的特点	2
1.3 简单的C程序介绍	4
1.4 C程序的上机步骤	7
习题	12
第2章 数据类型、运算符与表达式	13
2.1 C的数据类型	13
2.2 常量与变量	14
2.2.1 常量和符号常量	14
2.2.2 变量	15
2.3 整型数据	16
2.3.1 整型常量的表示方法	16
2.3.2 整型变量	17
2.3.3 整型常量的类型	21
2.4 实型数据	21
2.4.1 实型常量的表示方法	21
2.4.2 实型变量	22
2.4.3 实型常量的类型	23
2.5 字符型数据	24
2.5.1 字符常量	24
2.5.2 字符变量	25
2.5.3 字符数据在内存中的存储形式及其使用方法	26
2.5.4 字符串常量	28

2.6	变量赋初值	29
2.7	各类数值型数据间的混合运算	30
2.8	算术运算符和算术表达式	30
2.8.1	C 运算符简介	30
2.8.2	算术运算符和算术表达式	31
2.9	赋值运算符和赋值表达式	35
2.10	逗号运算符和逗号表达式	39
	习题	41

▶ 第3章 最简单的 C 程序设计——顺序程序设计 43

3.1	C 语句概述	43
3.2	程序的三种基本结构	45
3.3	赋值语句	47
3.4	数据输入输出的概念及在 C 语言中的实现	47
3.5	字符数据的输入输出	48
3.5.1	putchar 函数(字符输出函数)	48
3.5.2	getchar 函数(字符输入函数)	49
3.6	格式输入与输出	50
3.6.1	printf 函数(格式输出函数)	50
3.6.2	scanf 函数(格式输入函数)	56
3.7	顺序结构程序设计举例	60
	习题	62

▶ 第4章 选择结构程序设计 64

4.1	关系运算符和关系表达式	64
4.1.1	关系运算符及其优先次序	64
4.1.2	关系表达式	65
4.2	逻辑运算符和逻辑表达式	65
4.2.1	逻辑运算符及其优先次序	66
4.2.2	逻辑表达式	66
4.3	if 语句	68
4.3.1	if 语句的三种形式	69
4.3.2	if 语句的嵌套	71
4.3.3	条件运算符	73
4.4	switch 语句	75



4.5 程序举例	77
习题	82

▶ 第5章 循环控制 83

5.1 概述	83
5.2 goto 语句以及用 goto 语句构成循环	83
5.3 while 语句	84
5.4 do-while 语句	85
5.5 for 语句	87
5.6 循环的嵌套	90
5.7 几种循环的比较	91
5.8 break 语句和 continue 语句	92
5.8.1 break 语句	92
5.8.2 continue 语句	92
5.9 程序举例	93
习题	98

▶ 第6章 数组 99

6.1 一维数组的定义和引用	99
6.1.1 一维数组的定义	99
6.1.2 一维数组元素的引用	100
6.1.3 一维数组的初始化	100
6.1.4 一维数组程序举例	101
6.2 二维数组的定义和引用	103
6.2.1 二维数组的定义	103
6.2.2 二维数组的引用	104
6.2.3 二维数组的初始化	104
6.2.4 二维数组程序举例	105
6.3 字符数组	107
6.3.1 字符数组的定义	107
6.3.2 字符数组的初始化	107
6.3.3 字符数组的引用	108
6.3.4 字符串和字符串结束标志	109
6.3.5 字符数组的输入输出	110
6.3.6 字符串处理函数	112

6.3.7 字符数组应用举例	115
习题	118
 第7章 函数	120
7.1 概述	120
7.2 函数定义的一般形式	121
7.3 函数参数和函数的值	123
7.3.1 形式参数和实际参数	123
7.3.2 函数的返回值	125
7.4 函数的调用	127
7.4.1 函数调用的一般形式	127
7.4.2 函数调用的方式	128
7.4.3 对被调用函数的声明和函数原型	129
7.5 函数的嵌套调用	132
7.6 函数的递归调用	135
7.7 数组作为函数参数	141
7.8 局部变量和全局变量	146
7.8.1 局部变量	146
7.8.2 全局变量	147
7.9 变量的存储类别	150
7.9.1 动态存储方式与静态存储方式	150
7.9.2 auto 变量	150
7.9.3 用 static 声明局部变量	151
7.9.4 register 变量	153
7.9.5 用 extern 声明外部变量	154
7.9.6 用 static 声明外部变量	156
7.9.7 关于变量的声明和定义	156
7.9.8 存储类别小结	157
7.10 内部函数和外部函数	159
7.10.1 内部函数	159
7.10.2 外部函数	160
7.11 如何运行一个多文件的程序	162
习题	163
 第8章 预处理命令	165
8.1 宏定义	165

8.1.1	不带参数的宏定义	165
8.1.2	带参数的宏定义	168
8.2	“文件包含”处理	172
8.3	条件编译	175
	习题	178

▶ 第9章 指针

9.1	地址和指针的概念	180
9.2	变量的指针和指向变量的指针变量	182
9.2.1	定义一个指针变量	182
9.2.2	指针变量的引用	183
9.2.3	指针变量作为函数参数	186
9.3	数组的指针和指向数组的指针变量	190
9.3.1	指向数组元素的指针	190
9.3.2	通过指针引用数组元素	191
9.3.3	用数组名作函数参数	195
9.3.4	指向多维数组的指针和指针变量	203
9.4	字符串的指针和指向字符串的指针变量	212
9.4.1	字符串的表示形式	212
9.4.2	字符串指针作函数参数	215
9.4.3	对使用字符指针变量和字符数组的讨论	218
9.5	函数的指针和指向函数的指针变量	220
9.5.1	用函数指针变量调用函数	220
9.5.2	用指向函数的指针作函数参数	222
9.6	返回指针值的函数	225
9.7	指针数组和指向指针的指针	228
9.7.1	指针数组的概念	228
9.7.2	指向指针的指针	231
9.7.3	指针数组作 main 函数的形参	233
9.8	有关指针的数据类型和指针运算的小结	235
9.8.1	有关指针的数据类型的小结	235
9.8.2	指针运算小结	235
9.8.3	void 指针类型	237
	习题	238

▶ 第 10 章 结构体与共用体	240
10.1 概述	240
10.2 定义结构体类型变量的方法	241
10.3 结构体变量的引用	243
10.4 结构体变量的初始化	244
10.5 结构体数组	245
10.5.1 定义结构体数组	245
10.5.2 结构体数组的初始化	246
10.5.3 结构体数组应用举例	246
10.6 指向结构体类型数据的指针	248
10.6.1 指向结构体变量的指针	248
10.6.2 指向结构体数组的指针	249
10.6.3 用结构体变量和指向结构体的指针作 函数参数	251
10.7 用指针处理链表	253
10.7.1 链表概述	253
10.7.2 简单链表	254
10.7.3 处理动态链表所需的函数	255
10.7.4 建立动态链表	256
10.7.5 输出链表	259
10.7.6 对链表的删除操作	259
10.7.7 对链表的插入操作	262
10.7.8 对链表的综合操作	264
10.8 共用体	267
10.8.1 共用体的概念	267
10.8.2 共用体变量的引用方式	268
10.8.3 共用体类型数据的特点	269
10.9 枚举类型	271
10.10 用 typedef 定义类型	274
习题	277

▶ 第 11 章 位运算

11.1 位运算符和位运算	278
11.1.1 “按位与”运算符(&)	278
11.1.2 按位或运算符()	279

11.1.3	“异或”运算符(^)	280
11.1.4	“取反”运算符(~)	281
11.1.5	左移运算符(< <)	282
11.1.6	右移运算符(> >)	282
11.1.7	位运算赋值运算符	283
11.1.8	不同长度的数据进行位运算	283
11.2	位运算举例	283
11.3	位段	285
	习题	288

▶ 第12章 文件 290

12.1	C文件概述	290
12.2	文件类型指针	291
12.3	文件的打开与关闭	292
12.3.1	文件的打开(fopen 函数)	292
12.3.2	文件的关闭(fclose 函数)	294
12.4	文件的读写	295
12.4.1	fputc 函数和 fgetc 函数(putc 函数和 getc 函数)	295
12.4.2	fread 函数和 fwrite 函数	299
12.4.3	fprintf 函数和 fscanf 函数	302
12.4.4	其他读写函数	303
12.5	文件的定位	305
12.5.1	rewind 函数	305
12.5.2	fseek 函数和随机读写	305
12.5.3	ftell 函数	306
12.6	出错的检测	307
12.6.1	ferror 函数	307
12.6.2	clearerr 函数	307
12.7	文件输入输出小结	307
	习题	308

▶ 第13章 常见错误和程序调试 309

13.1	常见错误分析	309
13.2	程序调试	322

附录 I 摇常用字符与 ASCII 代码对照表	324
附录 II 摇C 语言中的关键字	325
附录 III 摇运算符和结合性	325
附录 IV 摇C 语言常用语法提要	327
附录 V 摇C 库函数	331
参考文献	338

第1章

悦语言概貌

悦语言出现的历史背景

悦语言是国际上广泛流行的计算机高级语言,既用来写系统软件,也可用来写应用软件。

以前的操作系统等系统软件主要是用汇编语言编写的(包括操作系统在内)。由于汇编语言依赖于计算机硬件,程序的可读性和可移植性都比较差。为了提高可读性和可移植性,最好改用高级语言,但一般高级语言难以实现汇编语言的某些功能(汇编语言可以直接对硬件进行操作,例如,对内存地址的操作、位操作等)。人们设想能否找到一种既具有一般高级语言特性,又具有低级语言特性的语言,集它们的优点于一身。于是,悦语言就在这种情况下应运而生了。

悦语言是在月语言的基础上发展起来的,它的根源可以追溯到1964年出现的ALGOL W。ALGOL W是一种面向问题的高级语言,它离硬件比较远,不宜用来编写系统程序。1968年英国的剑桥大学推出了ALGOL W语言。ALGOL W语言在ALGOL W的基础上接近硬件一些,但规模比较大,难以实现。1970年英国剑桥大学的Dennis Ritchie对ALGOL W语言做了简化,推出了B语言。B语言以ALGOL W语言为基础,又做了进一步简化,设计出了很简单的而且很接近硬件的月语言(取ALGOL W的第一个字母),并用月语言写了第一个操作系统,在PDP-7上实现。1971年在PDP-7上实现了月语言,并写了操作系统。但月语言过于简单,功能有限。1972年至1973年间,贝尔实验室的Richard M. Stallman在月语言的基础上设计出了悦语言(取ALGOL W的第二个字母)。悦语言既保持了ALGOL W和月语言的优点(精练,接近硬件),又克服了它们的缺点(过于简单,数据无类型等)。最初的悦语言只是为描述和实现操作系统提供一种工作语言而设计的。1974年,Stallman和Ritchie两人合作把操作系统的上层用悦改写,即第3版。原来的操作系统是1970年由美国的贝尔实验室的Stallman和Ritchie开发成功的,是用汇编语言写的。

后来,悦语言多次做了改进,但主要还是在贝尔实验室内部使用。直到1974年操作系统第3版公布后,悦语言的突出优点才引起人们的普遍注意。1975年出现了不依赖于具体机