

第一章 引论

以计算机 (Computer) 为代表和核心的信息技术, 是 “ 社会信息化, 经济知识化 ” 的重要标志与核心技术, 是 20 世纪人类最伟大的科技成就与历史进步, 是 21 世纪知识经济时代最重要的知识资源与发展基础。因而学习好、应用好计算机, 已是当今社会人们必须拥有的主要通行证, 也是当代社会各类人才必须具备的基本素质之一。

1.1 计算机发展与应用概要

计算机是人脑的仿生与延伸——“人脑的物化, 物化的人脑”。计算机系统, 由计算机的硬件系统与软件系统共同构成。

计算机硬件有如计算机的躯体, 它由存储器、运算器、控制器与输入设备、输出设备、外存 (储) 设备构成。从 1946 年第一台计算机诞生以来, 计算机硬件已迅速经历了 4 代 (即: 电子管计算机、晶体管计算机、集成电路计算机、大规模和超大规模集成电路计算机) 的历史飞跃, 目前正向着第 5 代 (即: 量子计算机、激光计算机、生物计算机、分子计算机、智能计算机、……) 稳步发展。计算机硬件技术进步的社会影响是: 新一代计算机的问世, 意味着老一代计算机过时的开始。

计算机软件 (即所有计算机程序及其相关文件的总称) 恰似计算机的灵魂, 它由系统软件 (如: 操作系统、计算机语言、数据库系统、……) 与应用软件 (如: 工具软件、杀毒软件、用户应用软件、……) 构成。没有思想的人, 只能是不会生活的植物人; 没有软件的计算机, 只配称不能工作的 “ 植物机 ”。可见, 计算机的软件系统的确比硬件系统更为重要。

计算机的应用, 正日益广泛地渗入和深入到人类社会、经济、生活的各个方面, 它已经并将不断极大地影响和改变人类的生产方式、工作方式和生活方式。通常, 计算机应用可分为: 数值计算, 数据处理, 自动控制, 辅助工作 (如: 辅助设计 CAD、辅助制造 CAM、辅助教育 CAI、辅助测试 CAT、……), 人工智能等五大类。

1.2 计算机语言、算法、程序概要

常言道: 人有人言, 兽有兽语。计算机, 从诞生之日起, 就有自己的专用 “ 语言 ”。不过, 计算机的 “ 语言 ” 是人们为了控制计算机正确工作而专门为它设计的人为的 “ 形

式语言”。计算机语言，实际上是沟通“电脑”与“人脑”的相互联系纽带，是“人”与“计算机”进行双向信息交流的共同语言桥梁；实质上是当今人类社会新文化——“计算机文化”的精神文明基础。

计算机语言即程序设计语言（**Programming language**），已走过了 4 代（即：机器语言、汇编语言、高级语言、非过程语言）的发展历程，现在正向着第 5 代（即：智能语言）积极迈进。然而，应当强调指出：计算机语言的科技进步，与计算机硬件的科技进步，其性质全然不同：新一代计算机语言的产生，只表明它为更多的计算机用户提供了又一种更为便捷的新语言工具，而丝毫不意味着其它计算机语言历史使命的结束，因为每一代计算机语言（哪怕是机器语言）至今仍各有其特殊专长与用武之地。

计算机算法简称算法（**Algorithm**）。简单地说，算法是人们为了借助计算机解决给定问题，而命令计算机按照“人—（计算）机”系统所认可的操作方式和控制方式，一步一步具体施行的有穷操作过程的描述。算法恰似计算机的“解题思想”，它颇类似于人们解决给定问题的特殊方法、想法、思路、规程、处方等等。

计算机程序简称程序（**Program**）。通俗地讲，程序是人们在所设计出来的、能控制计算机正确解决给定问题的算法的基础上，进一步用某种选定的计算机语言，把算法翻译成计算机可接受、读懂和执行的“计算机语言文章”。换言之，程序是人交给计算机具体实施和执行的工作任务书与行为指南针。应当强调指出：有如“同一事物，可用不同自然语言（例如：中文、英文、法文、俄文、……）表述而有不同说法”那样，同一个算法，完全可以根据用户的实际需要和编程（即用计算机语言编写程序的简称）爱好而采用不同的计算机语言来进行编码（即翻译），并得到同一算法的不同计算机语言表现形式的程序；这些不同计算机语言的程序，其执行效果是完全一致的（例如，同一算法可以采用 **C** 语言、**BASIC** 语言、**PASCAL** 语言进行翻译，而其产物是同一“原文”——算法的三种语言的不同“译文”：**C** 程序、**BASIC** 程序、**PASCAL** 程序）。由此可见，算法远比程序深刻得多、根本得多、重要得多，并且两者的关系是“算法是程序之母，程序是算法之子”。

显然，人们为计算机设计算法及其程序的创造性劳动——程序设计（**Program design**），是赋给计算机以灵魂和生机的知识型创作。因而，程序设计是“人”驾驭“计算机”的基础环节和核心工作，是使“计算机”服务于“人类”的工作重心和成败关键，是人类新文化——计算机文明的基础建设和知识工程。因此，学习并掌握至少一种计算机语言的程序设计，对构筑当今人才和未来英才的完整知识结构，具有根本重要性。

1.3 C 语言与 Turbo C 概要

第 3 代计算机语言（如：**C** 语言、**BASIC** 语言、**PASCAL** 语言、……），是应用极为广泛、至今不断改进和发展的一类重要计算机语言，同时也是许多系统软件和应用软

件的主要开发语言。它可分为早期的非结构化语言和后期的结构化语言。20 世纪 70 年代初,为高效开发 UNIX 操作系统,美国贝尔实验室的 Dennis Ritchie 完成了“源于 ALGOL (ALGOrit hm Language, 1960 年) 始于 CPL (Combined Programming Language, 1963 年)、继于 BCPL (Basic Combined Programming Language, 1967 年)、承于 B (Basic combined programming language, 1970 年)”的技术进步,成功开发出“把高级语言根本优势与汇编语言主要特长巧集于一身(故有‘中级语言’美称)”的结构化程序设计语言——C (basic Combined programming language, 1972 年)。1988 年,美国国家标准局(ANSI,即 American National Standards Institute)以它为蓝本推出了 C 语言标准——ANSI C(以下简称标准 C 语言)。C 语言(简称 C),是第 3 代计算机语言的优秀代表之一,已成为当今计算机主流语言之一。(顺便指出:虽然各厂商相继推出的不少各有特色的 C 语言不同版本,但是都毫无例外地向下兼容作为其公共内核的标准 C 语言。)

C 属于编译型语言,故只有编译(包括连接)通过后其程序才能运行。它具有“语言精干,功能强大;数据丰富,使用灵活;结构简洁,易于共享;硬件控制,简单可靠;软件开发,简练高效;执行快速,便于移植”的独特优点;但与其它高级语言相比也有“语言过活,容易出错;规则过多,不易纠错;出错自纠,尚逊不足;编程要求,似嫌偏高”的弱点。因此,国内外计算机教育界和计算机科技界普遍认为:C,是计算机系统软件、应用软件开发的最主要工具语言之一,也是学习计算机程序设计的常用教学语言之一。

Turbo C,是美国 Borland 公司于 20 世纪 80 年代末开发成功、90 年代不断改进的著名 Turbo 系统软件(包括:Turbo BASIC, Turbo PASCAL, Turbo C, Turbo C++ 等)的系列重要产品之一。它科学地继承和发扬了标准 C 语言的全部优势,而且新增了“开发环境友好,语言功能更强;设计风格优雅,编程调试简便;执行效率高速,兼容性能优良;可扩充新设计方法(如 Turbo C++ 已支持“面向对象程序设计”),符合软件工程专业规范”等一系列新优势,不愧是 C 的新一代佼佼者。为此,本书特以广为流行的 Turbo C 2.0 作为 C 语言家族的典型代表是恰当而明智的,因为它不仅利于表述简洁统一,且其硬件要求不高而便于读者上机操作。

1.4 Turbo C 的进入、工作、退出

开始使用 Turbo C 之前,必须首先进入它的工作状态;结束使用 Turbo C 之时,必须及时退出它的工作状态。

1.4.1 Turbo C 的进入

当计算机处于 DOS (若要使用中文,则需处于中文 DOS) 工作状态时,首先进入

Turbo C 所在子目录（通常为\TC），然后直接执行主文件——可执行文件“TC.EXE”，便从 DOS 工作状态进入 Turbo C 工作状态，并使计算机立即出现 Turbo C 的集成开发环境（Integrated Development Environment，以下简称 IDE，它由一系列菜单与窗口构成）的初始编辑窗口，如图 1.4.1 所示。（请读者注意：为了查阅简便快速和表述统一规范，本书各章节示例的图号、表号、例号，一律采用以章号、节号为主索引的索引命名法。例如：图 1.4.1、表 1.4.1、例 1.4.1 分别表示“第一章第 4 节的第一幅图”、“第一章第 4 节的第一张表”、“第一章第 4 节的第一个例”。）

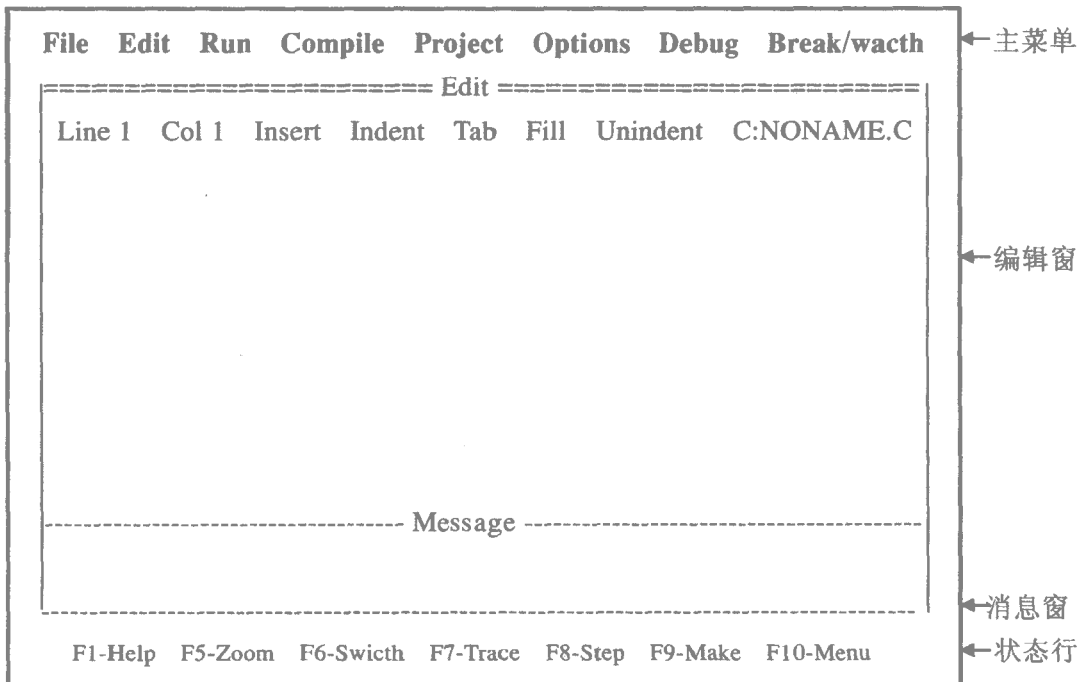


图 1.4.1 Turbo C 的集成开发环境 IDE 主菜单窗口示意图

图中：

第 1 行（即顶行）是 IDE 主菜单行。它共有 8 个菜单项：File（文件服务）、Edit（编辑服务）、Run（程序运行）、Compile（程序编译）、Project（项目服务）、Options（选项服务）、Debug（程序调试）、Break/watch（断点/监视服务）。

第 2 到 18 行是 IDE 编辑窗。它包括：第 2 行的编辑窗名（Edit）行，第 3 行的编辑状态行（例如：光标位置，插入/覆盖、文件名等），第 4 到 18 行的程序编辑区（注意：C 语言源程序必须在此窗口内进行输入、编辑、编译、调试、运行等处理）。

第 19 到 23 行是 IDE 消息窗。它包括：第 19 行的消息窗名行（其名称，当显示一般消息时，为 Message；当显示调试信息时，为 Watch），第 20 到 23 行的消息/信息显示区。IDE 的编辑窗、消息窗，可用 F6 进行切换。

第 24 行（即底行）是 IDE 状态行，其显示内容随计算机当前状态而变化。初始屏幕状态下为热键提示：F1（在线帮助）F5（当前窗缩扩）F6（编辑窗/消息窗切换）F7（跟踪调试）F8（单步执行）F9（生成并执行可执行文件*.EXE）、F10（到主菜单）

1.4.2 Turbo C 的工作

进入 Turbo C 工作状态后，就可以在由它的 IDE 菜单系统构成的工作环境下，方便地进行指定的相应工作。Turbo C 的各项工，大都在其 IDE 菜单系统下，通过“点菜”来挑选所需菜单项（即指定所需工作性质），并按照其要求进行操作来实现的。该菜单系统有一系列的特殊窗口，每一个窗口是屏幕中的一个工作区域，用户可以根据需要而切换、放大、缩小相关窗口。IDE 菜单系统的“点菜”方式，既可以用键盘，也可以用鼠标。

1. 键盘点菜方式

其操作如下：

① 激活主菜单行：必须也只需键入（即按下）热键 F10 即可。

② 选中所需菜单项，可用以下两种方法之一：

用光标上、下、左、右行键，移到所需菜单项处（将以高亮度的显著颜色突出之），然后键入回车键。

直接键入所需菜单项的代表字符（已以特别颜色突出之）。

2. 鼠标点菜方式

其操作如下：

① 激活菜单条：使鼠标指针移到所需菜单名处，然后单击（即快速按下鼠标左键）即可显示此菜单。

② 选中所需菜单项，可用以下两种方法之一：

使鼠标指针移到所需菜单项处，然后单击即可。

直接在菜单名上拖动，以选择菜单项，即：在所需菜单项处释放鼠标按钮（此时，如果用户改变主意，则可将鼠标移出该菜单外，就不会选中其任何菜单项）。

通过下述简单示例，可以初步了解 Turbo C 的实际应用方法与具体使用过程。

例 1.4.1 试设计 C 语言程序让计算机屏幕显示如下中英文对照欢迎词：

欢迎你使用 Turbo C! Welcome you to use Turbo C!

为此，人们必须首先设计出能解决该问题的算法（因太简单，故略），并用 C 语言对它进行正确翻译而得其 C 程序如下（其中，形如“/*...*/”者均为程序注释）：

```
/*程序 Eg01401      (输出欢迎词的程序)*/
#include <stdio.h> /* 预处理：指定含输出函数 printf 的标准输入输出头文件 stdio.h*/
void main() {      /*程序开始（注：无值类型 void 说明主函数 main(无返回值)*/
    printf("欢迎你使用 Turbo C! Welcome you to use Turbo C!\n"); /*欢迎词输出*/
}                  /* 程序结束*/
```

顺便说明，上述程序虽可简化为完全等效的如下最简形式（这是因为：程序注释均可略去；标准输入、输出函数 `scanf`、`printf` 可缺省用以指定需标准（或称系统）头文件 `stdio.h` 的预处理命令“`#include <stdio.h>`”（详见第六章）；主函数 `main`（只能由 C 语言系统本身所调用，且往往无需返回其函数值（简称返回值），故可省略其返回值类型 `void`），但不宜提倡之：

```
main() { printf("欢迎你使用 Turbo C! Welcome you to use Turbo C!\n"); }
```

但只设计出上述 C 程序仍然是不够的，因为：要在 Turbo C 状态下（或者在英文 DOS 或中文 DOS 状态下），让计算机输出这段欢迎词，还必须在 Turbo C 下进一步完成如下一系列操作（注：为节省篇幅，以下只以键盘操作为例）：

第一步 指定当前程序。可采用如下两种方法之一。

方法 1

Load	F3	← 开启新旧文件
Pick	Alt-F3	← 挑选已用文件
New		← 开启新文件 (NONAME.C)
Save	F2	← 以当前程序名存盘 (扩展名为 .C)
Write to		← 以自选文件名存盘
Directory		← 指定文件目录特征
Change dir		← 改变当前子目录
Dos shell		← 执行 DOS 外壳命令
Quit	Alt-X	← 退出 Turbo C (返回 DOS)

图 1.4.2 Turbo C 文件服务菜单（2 级）及其说明

(1) 首先，键入热键 `F10`，进入（或称激活）主菜单行；接着，移到文件服务菜单项“File”处；然后，回车选中该菜单项，进入文件服务菜单（2 级菜单）如图 1.4.2 所示。（注意：可用组合热键“`Alt-主菜单项代表字母`”，直接选中对应主菜单项。例如本步操作可简化为键入“`Alt-F`”——即先按住“`Alt`”键不放，再按下“`F`”键；凡双按键或称双组合键均与此类同，以后不再说明。）

(2) 在文件服务菜单内，移到开启新旧文件菜单项“Load”处，回车选中该菜单项，并立即出现“开启新旧文件对话框”。

(3) 按当前程序的存在方式，分别采用下述办法之一，以指定其文件名：

如果当前程序是已经存在的文件，则可用回车键直接进入当前目录 C 程序名显示窗口（此时，计算机将自动暂取当前路径的第一个程序名作为当前程序待定名），再用光标上、下行键在其中挑选所需当前程序名，最终以回车认定后便立即进入当前程序编辑状态。如果所需文件不在当前路径下，则需由用户在此输入其所需路径。

如果当前程序是并不存在的文件，则用户必须在“*.C”文件名输入框内自行键入所需文件名。例如，本例必须键入解决本题的当前 C 程序名“`Eg01401.C`”，也可只

输入当前 C 程序名的基本名“Eg01401”（因为计算机会自动为它添上 C 程序扩展名“.C”），回车认定后便立即进入当前程序的编辑状态。

方法 2 用热键 F3，直接调出“开启当前文件对话框”。其余操作同于本步方法 1 的“(3)”。

第二步输入程序内容。例如，在当前程序编辑窗口内必须正确键入本例上述 C 程序 Eg01401 的全部内容。（请初学者特别留意：每一个程序行的自身内容输入完毕时，都务必以回车键作为该行输入的结束，否则可能会出现意料不到的错误！）

为了提高程序的编辑效率，在当前程序编辑窗口内还可使用常见编辑软件的块操作热键命令：块首 Ctrl-K-B、块尾 Ctrl-K-K、拷块 Ctrl-K-C、移块 Ctrl-K-V、删块 Ctrl-K-Y、写块 Ctrl-K-W、读块 Ctrl-K-R、隐块 Ctrl-K-H 等。

第三步 编译当前程序。可采用如下三种方法之一。

方法 1

(1)键入热键 F10，进入主菜单行。

(2)在主菜单行内，移到程序编译菜单项“Compile”处；然后，回车选中该菜单项，使之出现编译菜单，如图 1.4.3 所示。

(3)在编译菜单内，可选定其菜单项“Compile”（必要时还应再选“Link”、“Make”），让计算机自动编译当前程序，以供计算机执行。（注：计算机只能实际运行 C 程序的可执行程序“.EXE”，而不能直接运行扩展名为“.C”的文本状态的 C 程序）此时，若用户当前程序出现编译错误（即 C 语言的语法错误），则系统将会自动指示错处（初学者小心：产生错误的根源可能不在所指处，但总在其前或后！）及其性质，以提醒和方便用户及时修改；改正后，再重复本步操作，以重新编译修正后的当前程序，直至编译成功为止。

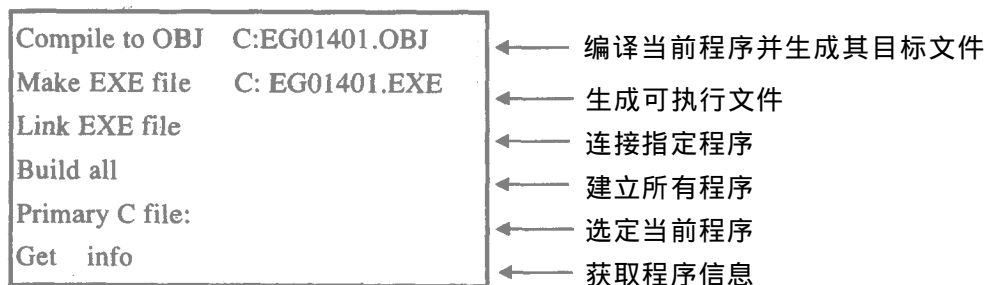


图 1.4.3 编译菜单（3 级）及其说明

方法 2 用热键 Alt-C，直接调出“编译菜单”。其余操作同于本步方法 1 的“(3)”。

方法 3 用热键 Alt-F9，直接编译当前程序。

第四步 在 IDE 下运行当前（目标）程序。可采用如下两种方法之一。

方法 1 首先，键入热键 F10，进入主菜单行；接着，移到程序运行菜单项“Run”处；然后，回车选中该菜单项，让计算机自动运行当前（.EXE）程序。即：运行用户

当前程序（注意：它实际运行的是编译通过后所得可执行程序），以解决所论问题，并获取人们所关切的有关信息。此时，倘若用户当前程序出现逻辑错误（即构造性设计错误如乘号 * 误为除号 / ）或运行错误（即运行性数据错误，如：分母误为 0）、或者运行结果尚不尽如人意，还可对它进行重新进行编辑、修改、编译、运行。

（顺便说明，对未经编译的当前程序，也可直接进入“运行当前程序”的操作。但其实质仍然是“先自动编译，后接着运行”。）

方法 2 用热键 **Ctrl-F9**，直接运行当前（.EXE）程序。

第五步 察看运行结果。

由于当前目标程序运行结束后，计算机将会毫不停顿地自动返回当前程序的原编辑屏幕状态；所以，在当前程序运行结束后，用户必须采用如下方法之一，才能进入用户屏幕，以察看其运行结果：

方法 1 用热键 **Alt-F5** 直接进入用户屏幕，然后用户方能察看其运行结果。察看完毕后，只需按任意键，即可返回当前程序的原编辑屏幕状态。

方法 2 在 C 程序的结束描述部分（即 `main` 函数的最后一个花括号）“`}`”之前，额外增加一个（或一组）“用户结束察看”的输入操作“`getchar();`”，使计算机暂时不结束其运行，以供用户察看。用户察看完毕后，只需按任意键，即可返回当前程序的原编辑屏幕状态。

例如，在用户屏幕上，可以看到本例程序 `Eg01401` 的运行结果如下：

欢迎你使用 Turbo C! Welcome you to use Turbo C!

第六步 保存当前程序（以文本文件方式存盘）。

在程序编辑的全过程中，为了当前程序的日后重复利用与防止意外丢失，必须及时将它存盘。其存盘的一般方法是：

首先，键入热键 **F10**，进入主菜单行；再移到文件服务菜单项“**File**”处并回车选中该菜单项，进入文件服务菜单。

其次，在文件服务菜单内，选定“以当前程序名存盘”的菜单项“**Save**”，或者“以自选文件名存盘”的菜单项“**Write to**”（注意：此时用户必须自行输入所需文件名），即可实现相应方式的存盘。

如果只需要以当前程序名存盘，则可用热键 **F2**（或者三组合键“**Ctrl-K-S**”），直接实现其存盘。

第七步 在 DOS 下执行所得可执行程序。

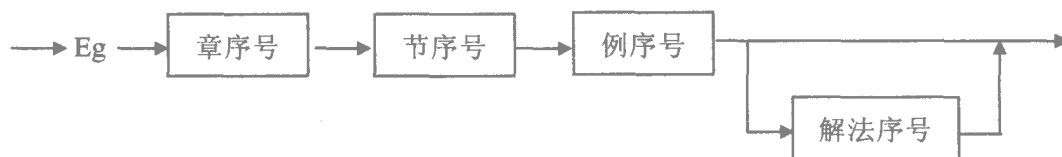
首先，采用以下“1.4.3 Turbo C 的退出”所述操作，实现从 Turbo C 状态退出并返回到 DOS 状态下。然后，在 DOS 下即可直接执行所得可执行文件（其基本名相同而扩展名为 .EXE），并得到所需执行结果。

例如，在 DOS 下执行本例的可执行文件 `Eg01401.EXE`，同样可得与上述运行结果完全一致的执行结果（略）。

在此，提请读者特别留意：

为了查阅简便快速和表述统一规范，本书各章节示例的算法及其程序的名称，都统一采用如本节例 1.4.1 算法及其程序所示的索引命名法。即算法名称及其程序名称，一律用英文单词“example”（示例）的习惯缩写“Eg”开头，其后所跟各字符分别定义为：第 1、2 两个数字表示章序号，第 3 个数字表示节序号，第 4、5 个数字表示例序号；如果同一示例因有多种解法（因而有多个算法及其程序），则在例序号后再用一个小写英文字母表示解法序号。例如：上述例 1.4.1 中的 C 程序“Eg01401”，表示该程序是“本书第一章第 4 节例 1.4.1 的 C 程序”。

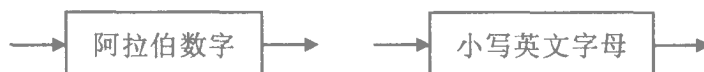
本书算法名及其程序名的索引命名法构成规则，还可形象地用如图 1.4.4 所示的语法图来描述。（注：所谓语法图，是指计算机科学中用以直观地描述有关语法实体的语法结构构成规则，即指明该语法实体是由哪些语法成分并以何种方式构成的重要形象化图形描述工具。语法图中：直接内容，表示有关语法对象的专用标志符号——保留字、标准标识符和其它符号，以及有关运算符或其它无需再定义的简单语法成分；矩形框内容，表示尚需由其它语法图来进一步定义的语法实体；而指向线所指者，则表示其有关相邻语法成分的直接后继语法成分。）



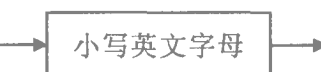
(a) 算法名及其程序名的语法图



(b) 章序号和例序号的语法图



(c) 节序号的语法图



(d) 解法序号的语法图

图 1.4.4 本书算法名及其程序名的语法图

1.4.3 Turbo C 的退出

当计算机处于 Turbo C 的工作状态时，如果确认各项工作已全部完成，而应结束 Turbo C 的使用，则应从 Turbo C 状态退出，并返回到 DOS 状态下。其实现操作，可采用如下两种方法之一：

方法 1

直接键入热键 Alt-X，便立即退出 Turbo C 工作状态而返回 DOS 工作状态。

方法 2

(1) 直接用组合热键 Alt-F（或者：先用热键 F10，进入主菜单行；再选中文件服务的菜单项“File”），调出文件服务菜单。

(2) 在文件服务菜单中，选定“退出 Turbo C(返回 DOS)”的菜单项“Quit”，便立即退出 Turbo C 工作状态而返回 DOS 工作状态。

1.5 算法与 C 程序的总体结构、注释结构概述

算法、尤其结构化算法的表现方法，20 世纪 70 年代以前常采用非结构化流程图（即以菱形框和指向线为主要特征的传统流程图），80 年代已进化为采用结构化流程图（例如：NS 图、NS 周图即 NSZ 图、PAD 图、……）。鉴于新型的结构化流程图——NS 周图即“Nissa-Shneiderman-周启海（Zhou Qihai）图”比 NS 图等其它各种结构化流程图性能更加优越，故本书一律采用 NS 周图作为算法描述工具。

算法的总体结构，通常由算法名称描述部、数据（包括类型、变量）定义描述部、算法开始描述部、中心处理描述部、算法结束描述部等五大构成部分顺次构成。算法的总体结构设计，就是从整体大局上把握好算法总体结构的各构成部分及其关系的算法（或程序）设计。

算法的注释结构，是指根据设计者需要而在算法中某些地方适当增添若干注释性内容，以利算法和程序的设计、调试、使用和维护。应当指出，尽管注释结构的有无与多寡与算法的执行和功能全然无关，但恰到好处的注释结构可增加算法的易读性、易学性、易用性。算法的注释结构，其用法较为灵活，可分为行尾（包括整行）注释与行内注释两类。这两类注释，在 NS 周图中，是用一对单花括号及其括起来的注释内容来表示（但不允许在其注释结构内部再出现注释结构）。

由于程序是算法的最终产物——计算机语言表述的算法，因而程序总体结构和注释结构，只不过是算法总体结构和注释结构在某计算机语言下的一种反映。例如：C 语言程序的注释结构（用一对注释符“/*”、“*/”及其括起来的注释内容来表示），虽与 NS 周图注释结构的描述形式有所不同，但它们的本质、意义和作用均完全相同。

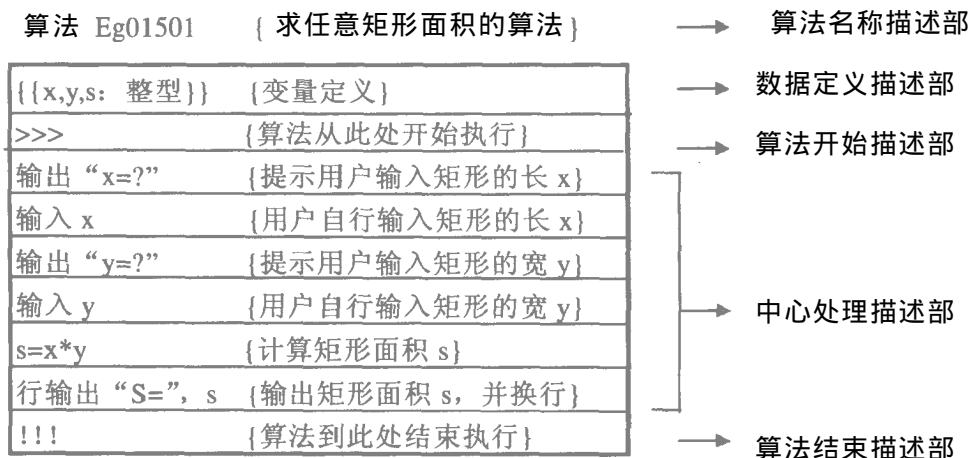


图 1.5.1 NS 周图描述的算法总体结构

例 1.5.1 如图 1.5.1 所示的算法 Eg01501，是一个用以“求任意矩形面积”的通用算法：（注：图 1.5.1 中，箭头线起、至行尾止的内容，均不是算法的应有部分，而只是作者为便于读者理解而另外增加的解释性内容。）

算法 Eg01501 很简单，它所对应的、用 C 语言翻译成的程序 Eg01501（已省略其扩展名“.C”；后同，略）如下所示：（注：其中，箭头线起、至行尾止的内容，均不是 C 程序的应有部分，而只是作者为便于读者理解而另外增加的解释性内容。）

*程序 Eg01501	(求任意矩形面积的程序) */	→	程序名称描述部
#include <stdio.h>	/*预处理：指定标准输入输出头文件*/	→	说明定义描述部
int x,y,s;	/*定义整型变量*/		
void main() {	/*主函数（执行部分）从此开始	→	程序开始描述部
printf("x=? ");	/*提示用户输入矩形的长 x*/	}	中心处理描述部
scanf("%i",&x);	/*用户自行输入矩形的长 x*/		
printf("y=? ");	/*提示用户输入矩形的宽 y*/		
scanf("%i",&y);	/*用户自行输入矩形的宽 y*/		
s=x*y;	/*计算矩形面积 s 即 x 乘 y*/		
printf("S=%i\n",s);	/*输出矩形面积 s 并换行*/		
}	/*主函数（执行部分）到此结束*/	→	程序结束描述部

该程序虽然简单，但已不难看出：

(1) C 程序的总体结构，通常应由程序名称描述部（即描述程序名的注释结构）、说明定义描述部（它由 C 语言特有的预处理部分，以及常量、变量、子函数定义等若干成分顺次构成）、程序开始描述部（它以主函数 main() 的“{”为标志）、中心处理描述部、程序结束描述部（它以主函数 main() 的最后一个右花括号“}”为标志）等五大部分顺次构成；显然，C 程序中必有且仅有一个函数名可取作主程序名“main”。

(2) C 程序的注释结构是用一对注释符“/*”、“*/”（注意：不可直接移用 NS 周图注释结构的花括号“{”、“}”，因为 C 语言已把花括号定义为复合语句起、始标志符）及其括起来的注释内容来表示，但其使用颇为灵活——可在任何需要的地方出现。

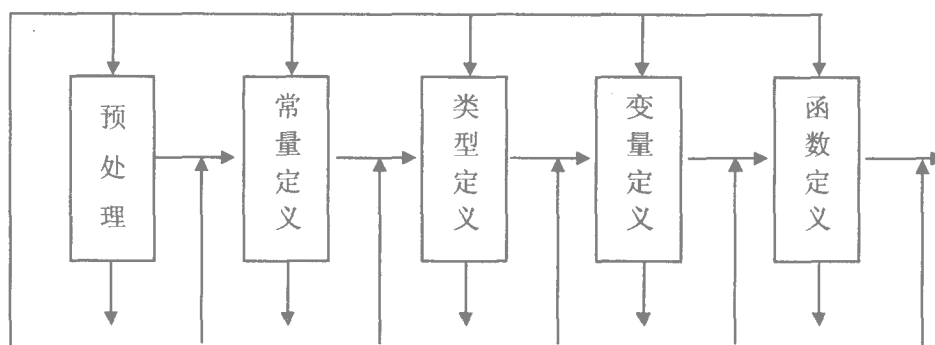
(3) C 程序的注释结构的有无与多寡，虽与程序的运行和功能无关，但有助于人们理解。

(4) C 各程序行的书写虽较自由，但应采用本书示范的错落有致、伸缩齐整的规范缩排格式，而不应采用“语句左边总对齐，成分主次全不分”的一刀切格式。

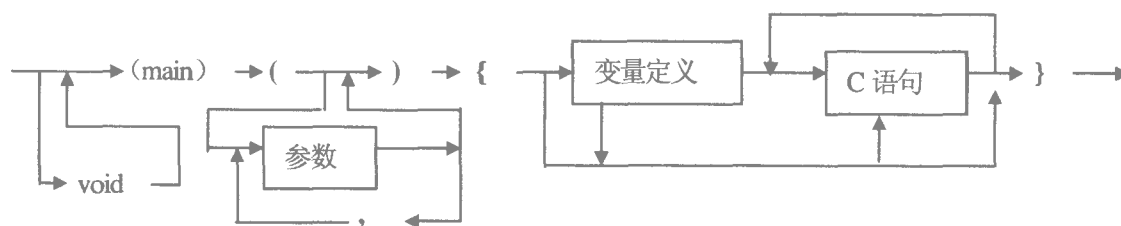
诚然，C 程序形式自由度较大；但为了养成程序设计好习惯，应倡导采用结构良好、风格优雅、形式规范的 C 程序。全面描述这种规范化 C 程序构成基本规则（注：不包括其注释结构）的语法图，如图 1.5.2 所示。



(a) C 程序的语法图



(b) C 程序说明部分的语法图



(c) C 程序主函数的语法图

图 1.5.2 C 程序构成基本规则的语法图

顺便指出：图 1.5.2 表明，理论上合法（指符合 C 语言语法规定）、实际上无用（因它什么事也不会干）的最简 C 程序必然是

```
main(){}
```

1.6 C 的字符集与标识符

任何计算机语言，都有一整套自己专用的字符集与标识符。因此，C 用户只能不折不扣地使用 C 所规定的字符集和标识符，否则计算机视为非法错误（特指其用户不慎误用了不符合 C 规定的字符、标识符、数据、运算等）。

1.6.1 C 字符集

C 字符集，是构成 C 语言的最基本字符的集合。它由下列字母、数字、特殊字符和其它字符共 4 类组成。

1. 字母

它包括：大写和小写英文字母（即“ A, B, ..., Z’和“ a, b, ..., z”）。

2. 数字

它包括：阿拉伯数字（“0, 1, 2, ..., 9”）。

3. 特殊字符

它分为如下 4 类：

(1) 数据描述符。它包括：小数点符、记录字段（或称结构分量）标志符、共用（旧称联合）成员（或称分量）标志符“.”（对指向记录的指针，其记录字段标志符“.”还可用由减号、大于符号构成的“->”代替）；输入输出数据格式描述符“%”转义字符标志“\”（注意：它是斜杠符，而不是除号符“/”）；指针说明符“*”、数组说明符（下标变量）下标界定符“[”和“]”（即左、右方括号）枚举型常量说明符“{”和“}”（即左、右花括号）；以及必须成双配对使用的字符常数界定符（即单撇号“'”）、字符串常数界定符（即双撇号“””）。

(2) 基本运算符。它包括：取正、负数运算符“+”、“-”取变量地址运算符“&”，取指针元素（即指针所指内容，亦即指针所指变量）运算符“*”，算术运算符“+”、“-”、“*”、“/”、“%”与圆括号“（”和“）”，增量、减量运算符“++”、“--”按位运算符“&”、“|”（即斜杠符“\”，的上档“虚竖线符”注意：不是实竖线符）“^”（即数字键“6”的上档“小箭头符”）、“~”、“<<”、“>>”；赋值运算符“=”（借用等号来表示），复合赋值运算符“+=”、“-=”、“*=”、“/=”、“%=”、“<<=”、“>>=”、“&=”、“|=”、“^=”；关系运算符（“>”、“==”、“<”、“>=”、“!=”、“<=”）；逻辑（也称布尔）运算符“&&”、“||”、“!”；条件选值运算符“条件：真式：假式”中“和：逗号运算符“，”。

(3) 注释描述符“/*”和“*/”（它们必须成双配对地把有关注释内容括起来）。

(4) 语法成分分隔符。它包括：

程序行尾分隔符：每一程序行末尾必须是程序行分隔符（分号“；”）和行结束符（回车符）。

程序行内分隔符：C 语言规定一个程序行内的任何两个相邻保留字、标识符、表达式、数据等基本语法成分两两间，必须插入一个规定的分隔符（空格符“ ”是最常用的分隔符，并且一个空格与多个空格的作用相同，而行结束符、逗号“，”、分号“；”、圆括号“(”和“)”、跳格符“Tab ”和注释等，也可按规定而起分隔作用）。但是，每一个基本语法成分（即每一个保留字、标识符、表达式、数据等）内部，不得出现分隔符。每个语句必须用分号“；”结尾。

4. 其它字符

它包括：其它可用于构成字符串，或进行特殊控制的某种编码（最常用的是 ASCII 码，但可采用其它编码——如 EBCDIC 码）体系的字符（例如 ASC 码字符）和扩展码字符（例如扩展 ASC 码字符，即中文区位码字符）。提请读者注意：为了叙述简便，本书以下将仅以 ASC 码为例进行阐述。

1.6.2 C 标识符

C 标识符，只能由英文状态下的字母或下划线（即减号键的上档键符“_”，但必须严防与下档键的减号“-”相混淆）开头，后跟若干个字母、数字、下划线（注意：不包括汉字！）的任意组合所构成的字符序列。其语法图，如图 1.6.1 所示。C 语言是严格

区分大、小写英文字母的计算机语言，故应特别注意“相同字母的大、小写构成全然不同的标识符”（例如：MY、My、mY、my，构成互不相同的 4 个标识符）

按 C 标识符定义方法和使用方式的不同，可分为两大类：保留字和标准标识符，与用户标识符。

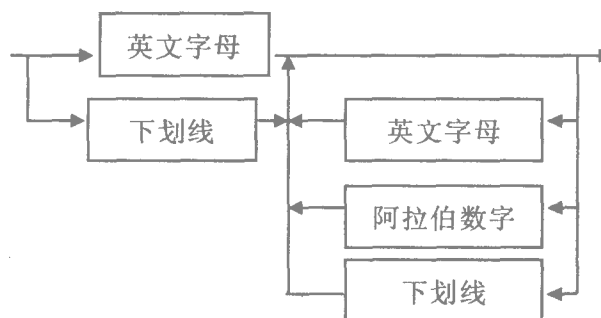


图 1.6.1 Turbo C 标识符的语法图

一、C 保留字和标准标识符

保留字和标准标识符，是计算机语言系统预定义的、专门用来控制计算机正常工作的特殊命令或者特殊数据的一类专用标识符。因此，C 保留字和标准标识符，只准用户严格使用于规定场合，而不准用户擅自另行定义或挪作它用。

1. C 保留字。

标准 C 语言共有如下 32 个保留字（均为小写英文字母）：

```

auto    break    case    char    const    continue    default    do    double
else    enum    extern    float    for    goto    if    int    long
register    return    short    signed    sizeof    static    struct    switch    typedef
union    unsigned    void    volatile    while
  
```


1.7 C 数据的类型及其形态与运算

现实生活中各种数据，是 C 数据的产生渊源与处理原型；而 C 数据，则是计算机科学用以描述现实生活中各种数据的 C 科学方法与技术手段。

C 数据是 C 程序的处理基础和运算对象。熟练掌握 C 数据的类型、形态与运算之特点和使用，是正确设计、使用 C 程序的基本前提，也是合理开发、应用 C 语言的重要基础。

1.7.1 计算机数据的类型、形态、操作概述

如同现实生活中“物以类聚，人以群分”，在计算机科学中，任何数据总是“数（据）以类聚，（运）算以型分”。换言之：

(1) 所有数据，必须严格划分为不同的数据类型；每一数据，总属于某一确定数据类型。

(2) 不同类型数据，总有其互不相同的取值方式和运算方式；同一类型数据，才能施行本类型特有的取值方式和运算方式。

(3) 同一类型数据称为同型量，它可有不同的数据形态；而不同类型数据称为非同型量，其数据形态种类往往不同。

(4) 数据的形态，分为常数、常量、变量、函数和表达式五种。其简况概述如下：

常数，是一种“不经定义，即可使用”的最基本的数据形态。它用于描述算法或程序执行过程中其值稳恒不变的量。它是其余各种数据形态的取值基础和范围。

常量，是一种必须“先定义，后使用”的数据形态。它也用于描述算法或程序执行过程中其值稳恒不变的量。如果某数据类型（例如枚举型）没有常数，则常量还是它的其余各种数据形态的取值基础和范围。

变量，是最主要和最重要的数据形态，它用于描述算法或程序执行过程中其值“既可长时保持不变，又可及时动态改变”的量，是人们必须认真掌握的计算机科学最基本、最常用的重要概念之一。变量名，是用以表征变量名称的用户标识符。变量的赋值和输入操作（它所对应的计算机语言形式是语句或命令），是使变量能获取其值的关键操作；变量的输出操作，是显示（或打印）变量所获值的重要操作。“先定义，后使用”，是变量使用的基本原则（即必须“先进行合法定义，再让它得到其值，然后才可使用之”）。

应当强调指出：凡变量，都有两个极为重要的根本性质：一是内存性，即它与计算机程序的运行“共存在，同取消”（即伴计算机程序的运行而使用，随计算机程序的停用而取消）；二是灵活性，即其值既可保持不变、又可另取它值。因而，变量是比其它各种数据形态应用更广、作用更大、使用更活的一种重要数据形态，必须首先熟练掌握之。

函数，是为方便用户使用而直接提供的一些普通常用函数和系统基本函数，因而

也是一种颇常用的数据形态。它的根本用途，是用于描述由自变量所决定的函数值。由计算机系统预定义的函数称为标准函数，由用户自定义的函数（子算法）称为自定义函数。本章将只讲述标准函数。

表达式（即算法表达式的简称），由相同类型（简称同型，有时也包括相容，即“原类型虽不同，但计算机自动使之转化为同一类型”；对此，以后不再说明）的运算对象（通常是同型的常数、变量、函数）和相关运算符所构成的有意义的运算式，其结果为一个值。它是功能最强的一种数据形态，其根本特点是用于描述各种需要对同型常数、常量、变量、函数进行运算并求得其值的量。其最大特长是可进行计算。

1.7.2 C 数据的类型与形态概述

C 数据的类型，其分类如图 1.7.1 所示；图中，各类型末尾括号内是其类型名（保留字）或其类型标志符。应当申明：尽管“准逻辑型”并不是 C 的一种实有数据类型，但实践表明假设它将有利于对“C 语言用数值类型（即整数类型和实数类型的合称）数据表征和处理逻辑值（真、假）”实质的理解和运用。

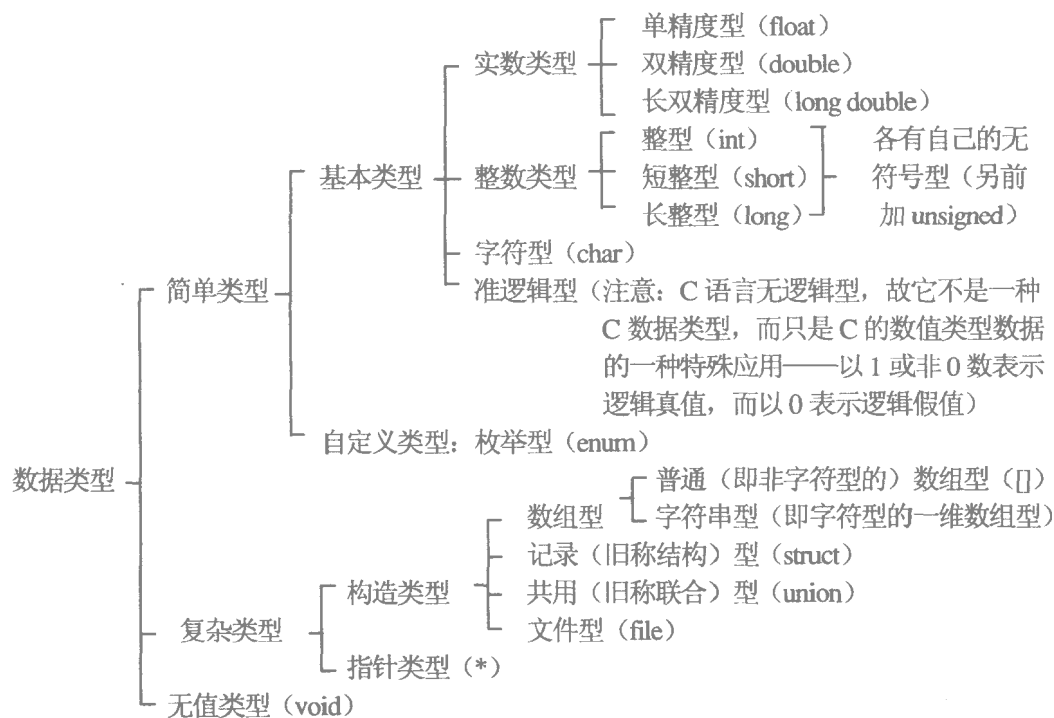


图 1.7.1 C 数据类型的分类

C 数据的形态，也有常数、常量、变量、函数、表达式五种形态。但是，不同类型的 C 数据，有各自不同的表现形态。例如：图 1.7.1 中，整型、单精度型、字符型和字符串型（即字符型的一维数组型）数据，可有常数、常量、变量、函数和表达式五种数据形态；枚举型、普通数组型（即非字符型的数组型）、记录型、指针型数据，可有变