

# 第 1 章 程序设计基础

## 1.1 程序设计语言的发展

**机器语言：**计算机可以直接识别、执行的二进制代码。如 Z-80 机“加寄存器 B 以累加器”指令为 1000 0000。其优点是执行速度快，但难写、难读、难理解和难修改。

**汇编语言：**使用助记符来代替机器语言中复杂、繁琐的指令的语言。如 Z-80 机中加法指令为 ADD。其特点是过分依赖于机器码，即仅适合某类相同或相似的计算机，因此不易移植。

**高级语言：**独立于计算机硬件结构，让用户使用面向问题的形式，更加方便编程人员的语言。高级语言分为如下两类

(1) 编译型：“先编译后执行”，过程为：源代码→目标代码→机器代码，其执行与源程序相互独立。比如 PASCAL, C 等。

(2) 解释型：“边解释边运行”，程序执行离不开源代码的支持，不利于源程序的保护和保密。比如 BASIC, FoxBASE+ 等。

面向结构（面向过程）的设计语言有：

BASIC (1964 年), FORTRAN (1956 年), COBOL (1960 年);

面向事务处理的高级语言，适合于编写管理信息系统方面的程序有：

PASCAL (1971 年), C 语言 (1974 年), PROLOG (1970 年), FoxBASE+ (1986 年) 等。

面向对象 (Object-Orientation) 和可视化 (Visual) 语言：

C++ (1983 年), Java, Visual BASIC, Visual C++, Visual Foxpro, Visual J++, Delphi, PowerBuilder 等。

网页 (Web) / 网络 (Network) 编程语言：

HTML, DHTML, JavaScript/Jscript, VBScript, JavaApplet, Java 等。

## 1.2 算法与程序设计

### 1.2.1 算法及其表示

“算法”的概念出现在公元 825 年左右，但算法思想在 3500~5000 年前就诞生了。算法是计算机科学中的重要概念之一，它指明了问题的计算过程，是对给定问题解题方案的准确而完整的描述。

计算机没有软件的支持将无法工作，软件是计算机的灵魂，而软件的核心是算法。用计算机解决问题的方法和步骤就是算法，计算机执行算法，用计算机解决各种数据处理问题的主要方式就是寻找和设计各种算法，将算法变成代码的过程即程序设计。

程序员编制的程序包含两个方面的内容：数据结构和算法。数据结构中指定数据的类型和数据的组织形式，而在算法中指定解决问题的操作步骤和描述。

### 1.2.1.1 算法的特征

算法具备如下 5 个特征：

(1) 有穷性。算法的“有穷性”是指某个算法在有限的操作步骤和用户可接受的时间内完成其执行过程。

(2) 确定性。算法的每一步操作都是含义确切且具体可行的，对于相同的输入则有确定的且类似的输出结果。

(3) 输入。算法可以接受零个或多个输入，允许算法从外界取得相应的数据。

(4) 输出。算法执行后有一个或多个输出。

(5) 可行性。算法中的操作可以通过已经实现的基本操作执行有限次来完成。

### 1.2.1.2 算法的表示

算法可以用多种方法描述，常见的表示方法有：自然语言描述法、伪代码表示法、传统流程图表示法、N-S 流程图表示法等。

#### 1. 使用自然语言表示算法

自然语言就是人们常用的语言，可以是汉语、英语或其他语言。

用自然语言表示通俗易懂，但因为描述不够确切，容易出现“歧义”，而且，使用自然语言描述分支结构和循环结构比较累赘，不很方便。用户在算法不够清晰具体的时候，使用自然语言描述算法可以帮助自己理清思路。

**【例 1-1】**已知杯 A 中盛放的是油，杯 B 中盛放的是水，另有一个空杯 C，三个杯容积相同，请用自然语言表示算法。

#### 【解答】

第 1 步：将 A 中的油全部倒入 C 中，使 A 成为空杯。

第 2 步：将 B 中的水全部倒入 A 中，使 B 成为空杯。

第 3 步：将 C 中原在 A 中的油，倒入 B 中，完成 A 与 B 中液体的交换。

#### 【例 1-2】描述计算并输出 $z=y/x$ 的流程。

#### 【解答】

第 1 步：输入 x, y。

第 2 步：判断 x 是否为 0：

    若  $x=0$ ，则输出错误信息；

    否则计算  $y/x \Rightarrow z$ ，且输出 z。

上述两个示例，都是比较简单的问题，已经占用了相当的文字篇幅，如果问题稍复杂一些，需要更多的文字才可以描述清楚。

对于同一个问题可以有不同的解题方法和步骤，例 1-1 还有其他的算法。同一个问题的解法有优劣之分，有些算法简捷，容易实现，有些算法冗长，执行的效率比较低，因此，在算法的选取和设计中，要考虑算法的质量。

#### 2. 使用传统流程图表示算法

传统流程图用规定的一组图形符号和文字说明来表示各种操作的算法，如表 1-1 所示。

表 1-1 中符号由美国国家标准化协会 ANSI (American National Standard Institute) 规定，用流程图表示算法，直观形象，易于理解，但画起来比较困难，由于绘制出的流程图幅面偏大，造成绘制和修改比较困难。

表 1-1 传统流程图常用符号

符号	符号名称	含义
	起止框	表示算法的开始和结束
	输入/输出框	表示输入/输出操作
	判断框	表示对框内的条件进行判断
	处理框	表示对框内的内容进行处理
	流程线	流程的方向
	连接点	避免流程线的交叉或过长，使流程图清晰

【例 1-3】用传统流程图表示对两个数按从小到大的顺序输出的算法。

【解答】传统流程图如图 1-1 所示。

在结构化程序设计中，流程图只包括 3 种基本结构：

(1) 顺序结构。顺序结构是结构化程序设计中最简单的一种结构，这种结构有一个入口和一个出口，中间的若干操作按照顺序依次执行，如图 1-2 所示，先执行操作 A，然后执行操作 B。

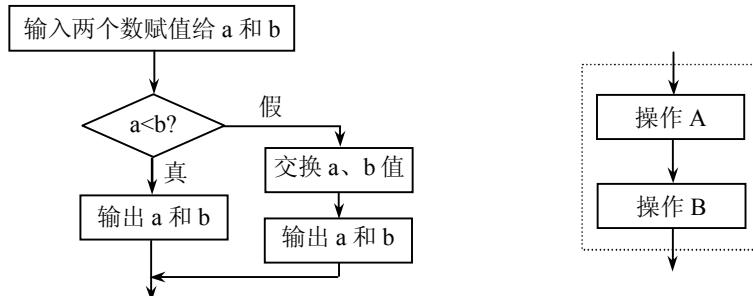


图 1-1 由小到大输出两个数

图 1-2 顺序结构

(2) 选择结构。选择结构与分支结构属同一结构，它由一个条件和两组语句组成，计算机根据条件的真假来选择执行的分支，如果判断成立则执行操作 A，否则执行操作 B。选择结构的流程图如图 1-3 所示。

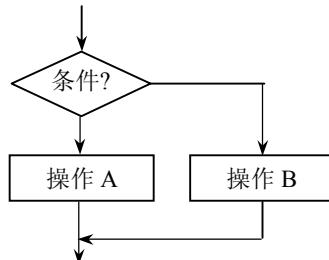


图 1-3 选择结构

(3) 循环结构。循环结构是指重复执行同一组程序段的结构。循环结构由两部分组成，其一是循环条件；其二是循环体，被重复执行的程序段称为循环体。循环体能否继续执行由循环条件决定。根据循环条件出现的不同位置，循环结构又分为“当型循环”和“直到型循环”两类。

“当型循环”中，当逻辑条件成立时，反复执行循环体，直到逻辑条件不成立时结束循环执行，如图 1-4 所示。

“直到型循环”中，反复执行循环体，直到循环条件不成立时结束循环，如图 1-5 所示。

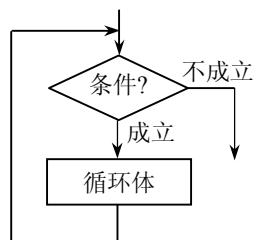


图 1-4 “当型循环”结构

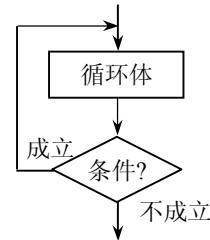


图 1-5 “直到型循环”结构

### 3. 使用 N-S 流程图表示算法

由于传统流程图存在不足，提出一种新的结构化流程图形式，这种形式的流程图的主要特点是取消了流程线，不再显示控制流，整个算法的流程写在一个矩形框内，这种形式的流程图称为 N-S 流程图。根据结构化程序的 3 种基本结构，N-S 流程图有 3 种基本形式，其中循环结构 N-S 流程图又分为“当型循环”和“直到型循环”两种。N-S 流程图表示的 3 种基本结构如下所示：

- (1) 顺序结构的 N-S 流程图如图 1-6 所示。
- (2) 选择结构的 N-S 流程图如图 1-7 所示。
- (3) 循环结构。“当型循环”结构的 N-S 流程图如 1-8 所示，“直到型循环”结构的 N-S 流程图如图 1-9 所示。



图 1-6 顺序结构

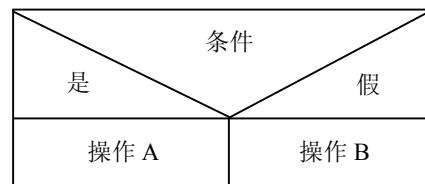


图 1-7 选择结构

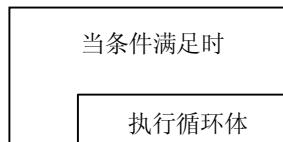


图 1-8 “当型循环”结构

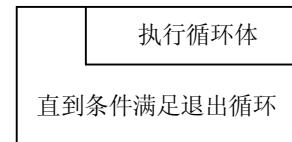


图 1-9 “直到型循环”结构

### 4. 使用伪代码表示算法

伪代码使用介于自然语言和计算机语言之间的文字和符号来描述算法。这种表示方式比

较接近于计算机语言，如果结合相应的语法，即可以修改成为程序。

**【例1-4】**使用伪代码描述数据A与数据B进行交换的算法，要求不引入临时数据。

**【解答】**使用伪代码描述如下：

- (1) A=A+B, 将A与B相加赋值给A，则A为原数据A+B的和；
- (2) B=A-B, 原两数和减去B即A的值，现赋值给B；
- (3) A=A-B, 原两数和减去最近更新的B即A的新值（原数据B）；
- (4) 两数A、B完成交换。

由于伪代码不能直接执行，但表示算法方便，容易理解。伪码的表现形式灵活自由，格式紧凑，但没有严谨的语法格式。

### 1.2.2 结构化程序设计与面向对象程序设计

编写高质量的程序，除了熟练掌握程序设计语言语法外，还要研究程序设计方法和设计技巧。在不断练习和思考中深入理解程序设计的内涵，才可以逐步提高程序设计的能力。

目前，软件开发行业中有两种典型的程序设计方法，一种为结构化程序设计方法；另一种为面向对象的程序设计方法。

“结构化程序设计”方法。传统的“结构化程序设计”（Structured Programming）最早由E.W.Dijkstra在1965年提出的，它的主要观点是采用自顶向下、逐步求精的程序设计方法，使用3种基本结构构造程序，任何程序都可由顺序、选择、循环3种基本控制结构编写出来。

大型软件的开发，需要科学管理，缺乏规划的软件开发和不完善的管理，会导致软件开发的进度被推迟，成本超过预算等问题。“结构化程序设计”方法旨在提高程序的可读性，保证软件的质量，降低软件成本，提高软件开发和维护的效率。因此，软件的开发应当按照工程的生产方式来组织和管理，每个成员都必须按照统一的规划、方法进行工作，使生产的软件有统一的标准和风格，以便于生产、推广和维护。

“面向对象程序设计（OOP）”方法。所谓面向对象的程序设计，就是把面向对象的思想应用到软件工程中，并指导开发维护软件。

对象是由数据和容许的操作组成的封装体，所谓面向对象，就是基于对象的概念，以对象为中心，类和继承为构造机制，认识了解刻画客观世界以及开发出相应的软件系统。

“面向对象程序设计”就是通过数据抽象，将许多实例中共性的数据和为操作这些数据所需要的算法抽取出来，并进行封装和数据隐藏，形成一个新的数据类型——“类”类型。

面向对象程序设计的关键在于如何进行数据的抽象和封装，以及对处理数据的过程的设计。面向对象程序设计的最大优点是允许“继承”，即在某个类的基础上可以派生出新类。目前的面向对象程序设计开发工具都提供了大量的类，用户可以直接使用这些类，或通过对这些类的扩充和重用形成新的类。与其他数据类型一样，只有类型对应的变量才是实际操作的对象。而类的变量我们称为“对象”。对“对象”的操作就是利用对象进行程序设计，也就是面向对象程序设计。

初学者在学习编程的时候应以“结构化程序设计”为基础，“面向对象”编程中涉及到的编程思想和某些经典算法仍然来自“结构化程序设计”。

## 1.3 C 语言简介

### 1.3.1 C 语言的产生与发展

C 语言是在 20 世纪 70 年代初问世，是目前国际上广泛流行的一种结构化的程序设计语言。1978 年由美国电话电报公司开始了 C 语言的最初研发，它不仅是开发系统软件很好的工具，而且也是开发应用软件很好的程序设计语言。

当初的 C 语言是为开发 UNIX 操作系统而研制的，1983 年美国国家标准学会（ANSI）将 C 语言标准化，随着 C 语言的广泛应用又不断推出新的 C 语言版本，其性能也越来越强。到了 1975 年 UNIX（第 6 版）的推出和随着面向对象程序设计（OOP）技术的出现，C 语言的突出优点就引起了人们的普遍关注，ANSI C 标准于 1989 年被采用，该标准定义为 C89，到了 1995 年，出现了 C 的修订版，其中增加了一些库函数，出现了初步的 C++，在此基础上，C89 成为 C++ 的子集。此后，C 语言不断发展，在 1999 年又推出了 C99，C99 在基本保留了 C 的特性的基础上增加了一系列新的特性，随后又几经修改和完善，它也从面向过程的编程语言发展到面向对象的程序设计语言，目前可在微机上运行的 C 语言版本主要有 Turbo C、Quick C、Visual C++ 等版本。

### 1.3.2 C 语言的特点与应用领域

C 语言的主要特点有：

- (1) 语言简洁，紧凑，使用方便，灵活。
- (2) 运算符丰富。
- (3) 数据结构丰富，具有现代化语言的各种数据结构。
- (4) 具有现代化的控制语句（如 if…else 语句，while 语句，do…while 语句，switch 语句，for 语句）。
- (5) 语法限制不太严格，程序设计自由度大。
- (6) C 语言能进行“位”（bit）操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。
- (7) 生成目标代码质量高，程序执行效率高。
- (8) 程序可移植性好。
- (9) 绘图能力强，可完成图形图像相关的程序设计。

## 1.4 C 语言的结构与执行

### 1.4.1 C 语言源程序的结构特点

C 语言程序的一般形式如下：

```
编译预处理部分
全局变量定义
main() /*主函数*/
{
```

```

全局定义序列
语句序列
}
func1() /*自定义函数 func1*/
{
    .....
}
func2() /*自定义函数 func2*/
{
    .....
}

```

C语言源程序的结构特点：

- (1) 一个C语言源程序可以由一个或多个源文件组成。
- (2) 每个源文件可由一个或多个函数组成。
- (3) 一个源程序不论由多少个文件组成，都有一个且只能有一个 main() 函数，即主函数。
- (4) 源程序中可以有预处理命令，预处理命令通常应放在源文件或源程序的最前面。
- (5) 每一个声明，每一条语句都必须以分号结尾。但预处理命令，函数头和花括号 “{}” 之后不能加分号。
- (6) 标识符，关键字之间必须至少加一个空格以示间隔。
- (7) 可以用/\*.....\*/对 C 语言程序中的任何部分进行注释。一个好的、有使用价值的源程序都应当加上必要的注释，增加程序的可读性和可维护性。

### 1.4.2 高级语言程序的执行过程

高级语言与具体计算机无关，是一种能方便描述算法过程的计算机程序设计语言。高级语言种类千差万别，但一般包含有以下4种成分：数据、运算、控制和传输。“数据”成分用来描述程序所涉及的数据；“运算”成分用来描述运算；“控制”成分用来表达程序的控制构造；“传输”成分用来表达数据的传输。由于高级语言程序主要是描述计算机的解题过程，即描述复杂的加工处理过程，所以也称这种高级语言为“面向过程语言”。

用高级语言编写的程序称为“源程序”。计算机不能直接执行源程序的各语句，通常有“解释”和“编译”两种方法处理源程序。“解释”方式，即让计算机运行解释程序，解释程序逐句取出源程序中的语句，对它作解释执行，输入数据，产生结果。“编译”方式，即先运行编译程序，从源程序中一次性翻译产生计算机可直接执行的二进制程序（称为目标程序），然后让计算机将目标程序结合具体的计算机硬件产生相应的“可执行”程序，输入数据，产生结果。

解释方式的主要优点是计算机与人的交互性好，调试程序时，能一边执行一边直接改错，能较快得到一个正确的程序。缺点是逐句解释执行，运行速度慢。

编译方式的主要优点是计算机运行目标程序快，缺点是修改源程序后必须重新编译以产生新的目标程序。

### 1.4.3 程序设计的开发期与开发习惯

程序设计的过程可以分为若干个相互关联的阶段。针对问题的要求，从分析问题的需求出发，逐步深入，到最后编制出能正确编译并可执行的程序。

第1阶段，分析问题，确定问题的需求。

接受任务后，首先要深入了解待解决问题，深刻掌握题意，知道要“做什么”。

第 2 阶段，分析问题，建立数学模型。

使用数学方法来描述实际问题的方法称为建立数学模型。建立数学模型的过程实质上就是通过一系列分析和实验找出待解决问题中涉及到的运算操作和活动的规律，然后进行归纳，并作抽象的数学描述。成功建立数学模型，找出规律离待解决问题更近一步。

第 3 阶段，选择计算方法。

对于同一个数学模型，存在多种解决方案，如例 1-1 和例 1-4 中关于两数交换就有两种处理方法，再如计算定积分的计算机求解，可以采用矩形法，也可以采用梯形法。虽然各种不同的方法均可解决问题，但在计算的精度，处理的速度和需要占用的资源都存在差异。因此，要针对具体问题，选择合适的计算方法以达到更好的处理效果。

第 4 阶段，设计算法，绘制流程图。

根据所选择的数学模型设计可行的计算方法，并把计算过程使用形象、直观的流程图清晰反映算法的基本思想和操作步骤。有了流程图，程序的编写工作就显得简单有条理，有利于程序的调试、修改和交流。

第 5 阶段，代码编写。

编程者要熟悉语言的语义和各种语法规则和规定，将算法准确而具体地表达出来，形成一个可以编译的完整的程序。

第 6 阶段，调试程序。

所谓调试程序，是将编制的程序投入实际运行前，用手工或编译程序等方法进行测试，修正语法错误和逻辑错误的过程。查错时，可以采取分段调试、逐层分析等有效的调试手段，以期待得到一个完整的能正常工作的程序。

第 7 阶段，整理资料和交付使用。

程序编写和测试成功后，为了使用户能了解程序的具体功能和掌握程序的运行操作，必须将程序设计阶段形成的资料和相关说明进行归类整理，形成程序说明书，内容包括：程序名称、任务的具体要求、给定的原始数据、算法、程序框图、程序清单、测试及运行结果、程序操作说明、程序运行环境要求和其他资料。

关于程序的开发，从书写清晰，便于阅读、理解、维护的角度出发，在书写程序时应遵循以下规则：

- (1) 程序所采用的算法要尽量简单，符合人们一般的思维方式。
- (2) 标识符的命名尽量采取“见名知义，常用从简”的原则。
- (3) 采用“缩进”的方式书写程序。低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写，以便看起来更加清晰，增加程序的可读性。
- (4) 程序中可用/\*.....\*/或//注释，以提高程序的可读性。
- (5) 在输入数据前，加一条输出语言，提醒用户输入什么样的数据。
- (6) 编译预处理部分的语句，不能加分号。
- (7) 每一行写一条语言。
- (8) 用{}号括起来的部分，通常表示程序的某一层次结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

在编程时应养成良好的编程习惯，这样可以达到事半功倍的效果。

## 1.5 C语言的运行与调试

### 1.5.1 C程序的执行过程

C源程序的扩展名为(\*.c)，经过编译程序处理后的源程序生成目标程序(\*.obj)，目标程序经过连接程序处理，生成可执行程序(\*.exe)，可执行程序能够直接运行。整个过程如图1-10所示。

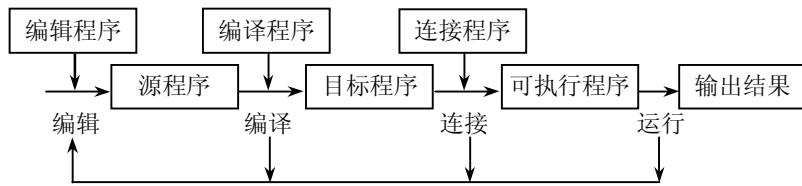


图1-10 源程序的变化过程

标准C程序的执行是从main()函数开始执行的，如果main()函数中调用了其他函数，其他函数调用结束后，返回主函数继续执行。

### 1.5.2 C程序开发的实验方法

Visual C++ 6.0是微软公司开发并广泛流行的基于Windows平台的C和C++语言集成开发环境，该环境具有良好的工程管理模式，适合团队开发大型软件，操作者可以利用它实现C语言程序的编辑、编译、连接、运行和调试。Visual C++ 6.0运行在Windows操作系统上，其基本的光标移动、插入、复制、粘贴和删除等功能的操作方法都与普通的文本编辑方法一致。下面介绍在Visual C++ 6.0集成开发环境下运行C语言程序的一般方法。

开发程序之初，需要建立一个工作区，工作区的主要功能是管理多个工程，在某个工程的管理下，可以编写和调试C语言程序。适应该编程环境的程序员将会受益匪浅。

- (1) 安装Microsoft Visual C++ 6.0，安装完成后不需要配置环境。
- (2) 建立工作区。
- (3) 打开工作区，并在该工作区下建立工程。
- (4) 编写源程序。
- (5) 编译源程序。
- (6) 组建源程序。
- (7) 调试和执行源程序。

### 1.5.3 Visual C++ 6.0 编写程序与调试程序的方法

以下部分介绍在Visual C++ 6.0环境下开发程序的操作步骤：

- (1) 启动Visual C++ 6.0。

单击“开始”→“程序”→“Microsoft Visual C++ 6.0”→“Microsoft Visual C++ 6.0”菜单项，启动Visual C++ 6.0。如果在桌面上已创建Visual C++ 6.0的快捷方式，双击该快捷方式的图标，也可启动Visual C++ 6.0。

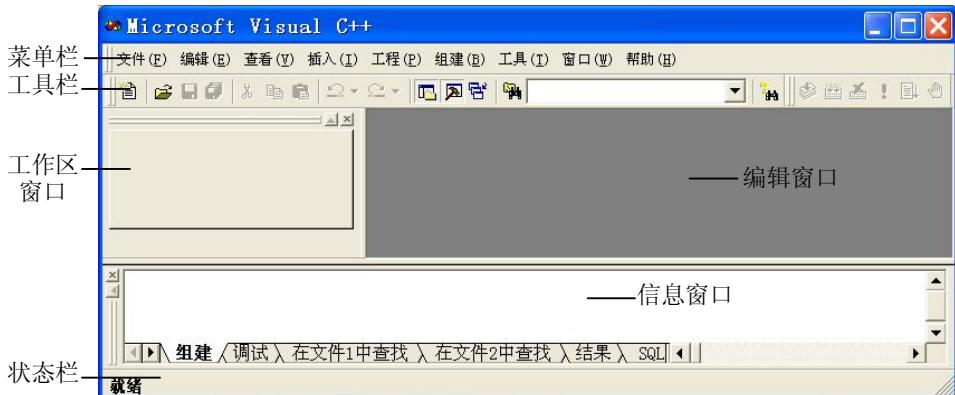


图 1-11 Visual C++ 6.0 的主窗口

如图 1-11 所示，是 Visual C++ 6.0 的主窗口，主要包括以下部分：

- 菜单栏。包括 9 个菜单项：文件（File）、编辑（Edit）、查看（View）、插入（Insert）、工程（Project）、组建（Build）、工具（Tools）、窗口（Window）和帮助（Helper）。
- 工具栏。Visual C++ 提供了几个工具栏。可以在菜单栏或工具栏的空白处单击右键，在弹出的菜单中选择要显示或关闭工具栏的名称。
- 工作区窗口。用来显示所设定的工作区的信息。
- 编辑窗口。用来输入和编辑源程序，每个源文件将显示在一个独立的编辑窗口中。
- 信息窗口。显示编译、连接和调试等信息。
- 状态栏。显示操作提示信息和编辑状态。

### (2) 建立项目（Project 工程）工作区。

在 Visual C++ 6.0 开发环境中，工程又是置于工作区的管理下，所以工作区又称为工程工作区。一个工作区可以管理多个工程，甚至是不同类型的工程。同一个工作区中的工程之间相互独立，但共用一个工作区的设置环境。在 Visual C++ 6.0 的集成环境中专门设置了一个工作区窗口，用来显示当前工作区的内容。在新建一个工程时，可以选择是为该工程新建一个工作区还是加入当前工作区。

下面介绍用户自己建立工作区的方法：

选择“文件”→“新建”菜单项或者快捷键 Ctrl+N，打开“新建”对话框（如图 1-12 所示），选择“工作区”标签，在“位置”文本框中输入 D:\ (D 盘根目录)，然后在“工作空间名称”文本框中输入工作区名称（例如：2009，你的工作区名称最好为你的学号）。Visual C++ 将自动在“位置”文本框中目录后面加上该工作区名称。单击“确定”按钮，返回 Visual C++ 主窗口。

此时，Visual C++ 已在 2009 目录中建立工作区文件 2009.dsw。选择“文件”→“保存工作空间”菜单项，可以保存当前工作区文件。选择“文件”→“关闭工作空间”菜单项可以关闭当前工作区文件。选择“文件”→“打开工作空间”菜单项，在“打开工作区”对话框中选择扩展名为.dsw 的文件，可以打开指定的工作区文件。

### (3) 建立工程文件。

1) 选择“文件”→“新建”菜单项，打开“新建”对话框，单击“工程”标签（如图 1-13 所示）在左边工程类型列表框中选择“Win32 Console Application (Win32 控制台应用程序)”。控制台是指 Windows XP 的命令提示符窗口，在其中可以运行 DOS 程序。



图 1-12 新建工作区对话框



图 1-13 新建工程对话框

2) 在右边的“工程名称”文本框中输入工程名称，例如：p1，然后选择“添加到当前工作空间（Create new workspace）”选项，此时 Visual C++ 6.0 会在存放位置文件夹下自动加上相同的子文件夹名 p1，有关 p1 工程的文件就会自动保存到 p1 文件夹中。最后单击“确定”按钮。

3) 出现如图 1-14 所示的“Win32 Console Application – 步骤 1 共 1 步”对话框，即创建 Win32 控制台应用程序向导的第 1 步。选择“一个空工程（An empty project）”（默认值）后，单击“完成（Finish）”按钮。



图 1-14 控制台应用程序创建向导

4) 如图 1-15 所示，在“新建工程信息”对话框，可以看到新建工程的类型和位置（工程文件的位置为 D:\200912345\p1）。单击“确定”按钮，返回到 Visual C++ 6.0 的主窗口，如图 1-16 所示。



图 1-15 新建工程信息对话框



图 1-16 新创建的工程



图 1-17 新建文件对话框

在主窗口左边的工作区窗口中单击“FileView（文件视图）”标签，窗口内有一个树型结构，根节点显示新创建工作区名称 2009，并且包含一个工程。工作区的子节点是其所包含的工程及工程中的文件，例如：现在有一个工程 p1。

单击工程节点 p1 files 前面的“+”展开该节点，可以看到工程中包含 3 个文件夹：“Source Files（源文件）”、“Header Files（头文件）”和“Resource Files（资源文件）”，现在工程不包含任何文件。

#### 提示：

①在一个工作区内可以新建、添加或清除多个工程，可以利用“工程”菜单下面的相应菜单选项来操作。每个工程只能对应一个程序。

②在同一个工作区内，如果有多个工程，只能有一个工程处于活动状态，可以利用“工程”→“设置活动工程”菜单选项来切换不同的工程处于活动状态。

③如果在图 1-13 中选择“创建新的工作空间”选项，则 Visual C++ 将关闭当前工作区，在指定的目录中创建一个与工程同名的工作区。

#### (4) 建立、编辑、编译、运行 C 源程序文件。

1) 选择“文件”→“新建”菜单项或者按快捷键 Ctrl+N，打开“新建”对话框，如图 1-17 所示。

2) 单击“文件”标签，在左边列表框中选择“文本文件”。然后，选择“添加到工程”选项，指定工程名称（如 p1）。在“文件名”文本框中输入文件名，例如：p1.c，其中.c 是 C

语言源文件的扩展名，一定要加上。

在“位置（Location）”文本框中输入源文件的存放路径（如：D:\2009\p1），要确保该路径已存在，否则会导致创建文件失败。也可以单击该文本框右边的“...”按钮，打开“选择目录（Choose Directory）”对话框直接选择存放位置。

单击“确定”按钮，即可在指定位置创建一个新的源文件，并打开一个编辑窗口，如图1-17所示。

3) 编辑文件。在文件编辑窗口中输入C语言程序代码，如图1-18所示。

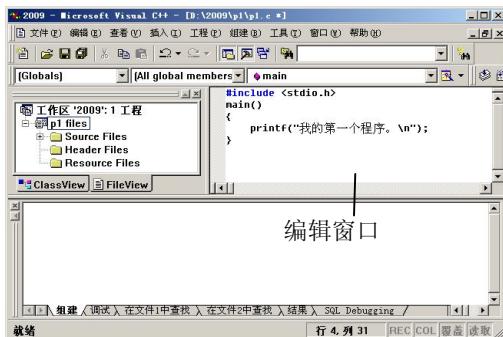


图1-18 新建文件的编辑窗口



图1-19 编译信息

4) 保存文件。在编辑文件后选择“文件（File）”→“保存（Save）”菜单项或者按快捷键Ctrl+S，保存所作的修改。

5) 编译程序。选择“组建（Build）”→“编译[p1.c]”菜单项或者按快捷键Ctrl+F7，对源文件p1.c进行编译。如图1-19所示，在信息窗口中可以看到编译的信息。如果编译系统检查源程序中有语法错误，错误信息将显示在信息窗口中，指出错误的位置和性质。例如，将源文件p1.c中，第4行的“；”号删除，然后编译该文件，出现如图1-20所示的编译输出信息。在一条错误提示信息处双击，Visual C++将在信息窗口高亮显示该行提示信息，并且切换到出错的源文件的编辑器窗口，在发现错误的代码行的前面作上标记，如图1-20所示。修改有关的错误之后，再重新编译，直至没有错误为止。

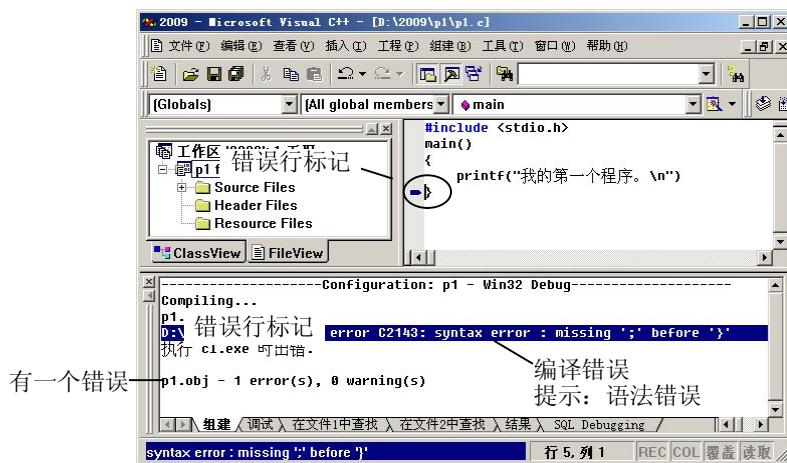


图1-20 编译错误信息

6) 连接(组建)程序。

选择“组建(Build)”→“组建[p1.exe]”菜单项或者按快捷键F7，对文件p1.c进行连接，生成可执行文件(p1.exe)等。

**提示：**第4、5步骤也可以合并成一个步骤进行。在完成编辑程序后，直接选择“组建(Build)”→“组建[p1.exe]”或者按快捷键F7，Visual C++ 6.0将先对当前文件进行编译，然后连接、生成可执行文件(如p1.exe)等。

7) 运行程序。选择“组建(Build)”→“执行[p1.c]”菜单项或者按Ctrl+F5组合键，Visual C++ 6.0将打开一个控制台窗口(或命令提示符窗口)，在其中运行当前程序，如图1-21所示。程序运行结束后，Visual C++ 6.0自动加上最后一行提示信息“Press any key to continue(按任意键继续)”，即按任意键后关闭该窗口，返回主窗口。



图1-21 运行程序

## 习题1

### 1. 填空

- (1) C程序是由\_\_\_\_\_构成的，每个函数由\_\_\_\_\_和\_\_\_\_\_两部分构成。
- (2) 一个C源程序有且只能有一个\_\_\_\_\_函数，即主函数。
- (3) 在C语言程序中，每个声明、每一条语句都必须以\_\_\_\_\_结尾。

### 2. 简答

- (1) 回答你对C语言的认识，写出其主要特点。
- (2) 写出一个C语言的构成。
- (3) 简述编辑、编译、连接和运行一个C语言程序的步骤。

### 3. 编程

- (1) 编写一个C语言程序，输出以下信息：

```
*****
Visual C++ 6.0 下 C 程序设计
*****
```