

# 第一部分 C 语言上机指导

## 第 1 章 Visual C++ 6.0 上机操作

C++语言是在C语言的基础上发展而来，它增加了面向对象的编程，成为当今最流行的一种程序设计语言。Visual C++ 6.0（简称VC++ 6.0）是由微软公司开发的，面向Windows 编程的C++语言工具。它不仅支持C++语言的编程，也兼容C语言的编程。由于Visual C++ 6.0被广泛地用于各种编程，使用面很广。这里简要地介绍如何在Visual C++ 6.0下运行C语言程序。

### 1.1 使用 VC++ 6.0 调试程序一般步骤

在使用Visual C++ 6.0 调试程序之前，必须先了解项目和项目工作区两个概念。项目（Project）是一组相互关联的文件，是将一个应用程序中需要的所有文件组成一个有机的整体，一般包括源文件、头文件、资源文件等。而项目文件必须在某个项目工作区（Workspace）的管理之下，一个项目工作区可以包含一个以上的项目。

使用Visual C++ 6.0 调试程序的大致步骤是：

- (1) 创建一个项目工作区（Workspace）。
  - (2) 在项目工作区中新建一个项目（Project）。
- 也可以将上面两步合并，在建立新项目的同时创建项目工作区。
- (3) 在项目中建立源文件、头文件等（或将这些文件添加到项目中）。
  - (4) 编辑项目中的源文件、头文件。
  - (5) 连接、编译应用程序。
  - (6) 运行应用程序。

**注意：**如果不先创建项目工作区（Workspace）和项目（Project），而直接创建源文件，当编译该源文件时会提醒没有项目工作区（Workspace）和项目（Project），单击“确定”按钮会自动创建一个默认的项目工作区（Workspace）和项目（Project）。

### 1.2 启动 VC++

Visual C++ 6.0 是一个庞大的语言集成工具，经安装后将占用几百兆磁盘空间。从“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0，可启动Visual C++ 6.0，屏幕上将显示如图 1.1 所示的窗口。

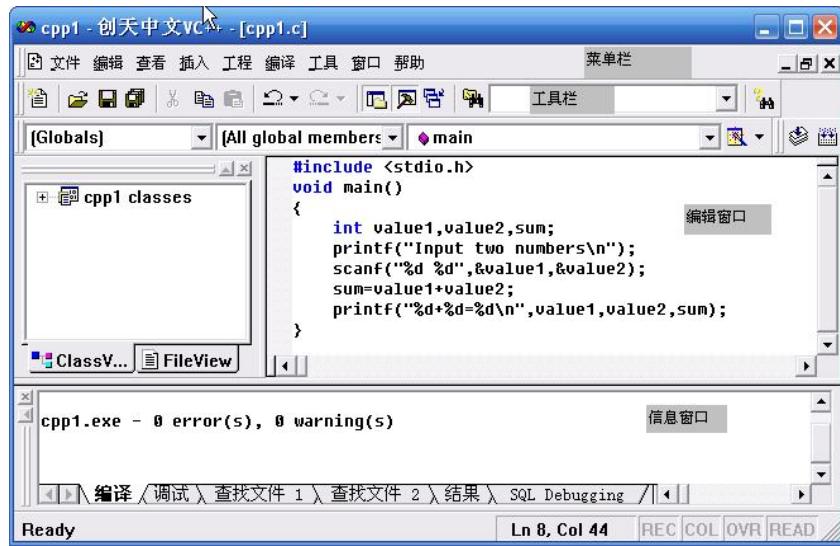


图 1.1 VC++ 6.0 窗口

### 1.3 新建或打开 C 程序文件

选择“文件”菜单中的“新建”命令，单击如图 1.2 所示的“文件”标签，选中“C++ Source File”，单击“确定”按钮，然后在编辑窗口中输入程序。



图 1.2 “新建”对话框

如果程序已经输入过，可选择“文件”菜单中的“打开”命令，并在查找范围中找到正确的文件夹，调入指定的程序文件。

### 1.4 程序保存

在打开的 VC++界面上，可直接在编辑窗口输入程序，由于完全是 Windows 界面，输入

及修改可借助鼠标和菜单进行，十分方便。当输入结束后，保存文件时，应指定扩展名为“.C”，否则系统将按 C++ 扩展名“.CPP”保存。程序保存对话框如图 1.3 所示。

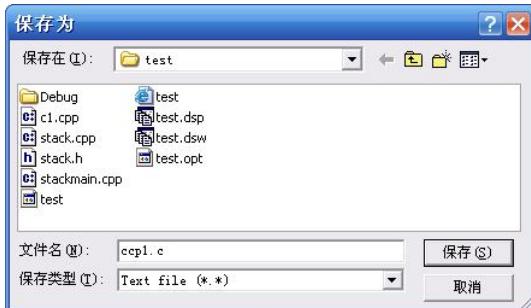


图 1.3 程序保存对话框

## 1.5 执行程序

首先要生成可执行文件。使用 VC++ “编译” 菜单中的“构件”命令，如图 1.4 所示，也可使用快捷键 F7。在编译、连接过程中 VC++ 将保存该新输入的程序，并生成一个同名的工作区。保存文件时需填入文件名，如“4-1.C”。假如不指定扩展名.C，VC++会把扩展名自动定义为.CPP，即 C++ 程序。

如果程序没有错误，将在如图 1.5 所示的信息窗口中显示内容：

```
0 error(s), 0 warning(s)
```

表示没有任何错误。有时出现几个警告性信息 (warning)，不影响程序执行。假如有致命性错误 (error)，如图 1.6 所示，双击某行出错信息，程序窗口中会指示对应的出错位置，可根据信息窗口的提示分别予以纠正。然后用“编译”菜单中的“执行”命令 (或快捷键 Ctrl+F5) 执行程序。



图 1.4 编译菜单

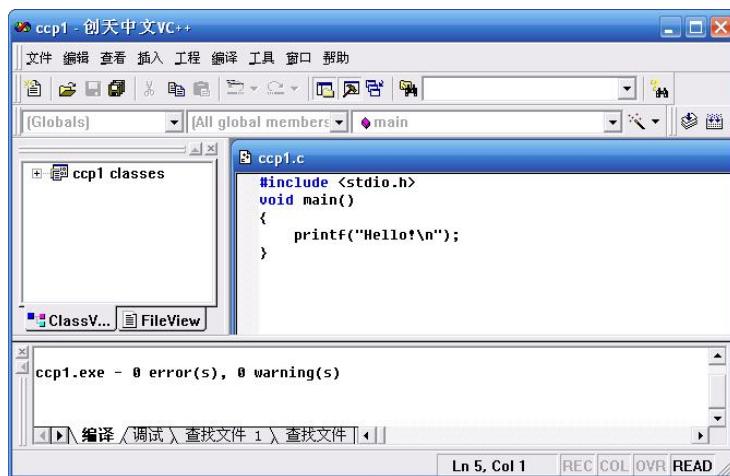


图 1.5 编译连接正确

当运行 C 程序后，VC++ 将自动弹出程序运行窗口，如图 1.7 所示。按任意键将关闭该窗口。

对于编译、连接、执行操作，VC++还提供了一组工具按钮，如图 1.8 所示。

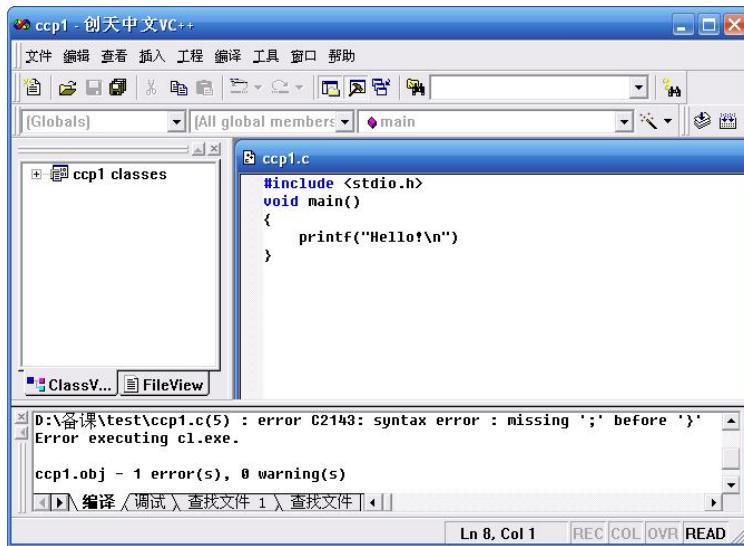


图 1.6 编译连接出错



图 1.7 程序运行窗口

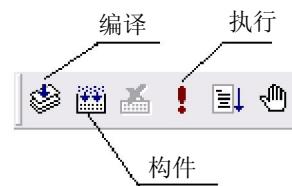


图 1.8 编译连接执行工具按钮

## 1.6 关闭程序工作区

当一个程序编译连接后，VC++系统自动产生相应的工作区，以完成程序的运行和调试。若想执行第二个程序时，必须关闭前一个程序的工作区，然后通过新的编译连接，产生第二个程序的工作区。否则，运行的将一直是前一个程序。“文件”菜单提供关闭程序工作区功能，如图 1.9 所示，执行“关闭工作区”命令，然后在如图 1.10 所示的对话框中单击“否”按钮。如果单击“是”按钮将同时关闭源程序窗口。



图 1.9 “文件”菜单

图 1.10 文件关闭对话框

## 1.7 命令行参数处理

VC++是一个基于窗口操作的 C++系统，没有提供命令行参数功能，需要在 Windows 的“MS-DOS 方式”窗口里以命令方式实现。具体参考如下步骤：

- (1) 正确编译连接，生成可执行程序。
- (2) 通过“我的电脑”或“资源管理器”找到所运行的 C 源程序（设为 a.c）。
- (3) 进入 debug 文件夹（它包含 a.c 程序的可执行文件 a.exe）。
- (4) 执行“开始”菜单的“运行”命令，填入 a，然后单击“确定”按钮。
- (5) 在打开的“MS-DOS 方式”窗口中输入：a 参数 1 参数 2 ..，带参数运行程序。

## 1.8 程序调试

VC++是一个完全基于 Windows 的系统，它的调试过程通过鼠标比较容易进行。

### 1. 程序执行到中途暂停以便观察阶段性结果

方法一：使程序执行到光标所在的那一行暂停。

- (1) 在需暂停的行上单击鼠标，定位光标。
- (2) 如图 1.11 所示，分别单击菜单“编译”→“开始调试”→Run to Cursor 命令，或按 Ctrl+F10 组合键，程序将执行到光标所在行暂停。如果把光标移动到后面的某个位置，再按 Ctrl+F10 组合键，程序将从当前的暂停点继续执行到新的光标位置，第二次暂停。



图 1.11 执行到光标所在行暂停

方法二：在需暂停的行上设置断点。

- (1) 在需设置断点的行上单击鼠标，定位光标。
- (2) 按“编译微型条”中最右面的按钮，如图 1.12 所示，或按 F9 键。被设置了断点的行前面会有一个红色圆点标志。



图 1.12 工具栏中断点设置按钮

与 TC 一样，不管是通过光标设置还是断点设置，其所在的程序行必须是程序执行的必经之路，亦即不应该是分支结构中的语句，因为该语句在程序执行中受到条件判断的限制，有

可能因条件的不满足而不被执行。这时程序将一直执行到结束或下一个断点为止。

### 2. 设置需观察的结果变量

按照上面的操作，使程序执行到指定位置时暂停，目的是为了查看有关变量的中间结果。在图 1.13 中，左下角窗口中系统自动显示了有关变量的值，其中 val1 和 val2 的值分别是 3 和 4，而变量 i、sum 的值是不正确的，因为它们还未被赋值。图中左侧的箭头表示当前程序暂停的位置。如果还想增加观察变量，可在图中右下角的 Name 框中填入相应变量名。

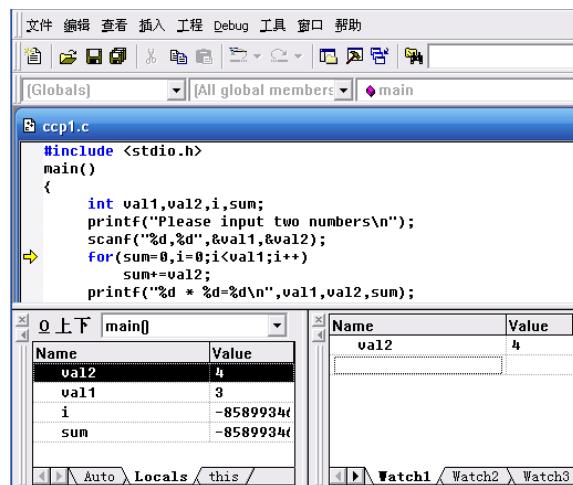


图 1.13 调试过程中观察变量的值

### 3. 单步执行

当程序执行到某个位置时发现结果已经不正确了，说明在此之前肯定有错误存在。如果能确定一小段程序可能有错，先按上面步骤暂停在该小段程序的头一行，再输入若干个观察变量，然后单步执行，即一次执行一行语句，逐行检查下来，看看到底是哪一行造成结果出现错误，从而能确定错误的语句并予以纠正。

单步执行可按“调试”工具栏中 Step Over 按钮 或 F8 键，如图 1.14 所示。如果遇到自定义函数调用，想进入函数进行单步执行，可按 Step Into 按钮 或 F11 键。当想结束函数的单步执行，可按 Step Out 按钮 或 Shift+F11 组合键。对不是函数调用的语句来说，F11 键与 F8 键作用相同。但一般对系统函数不要使用 F11 键。



图 1.14 单步调试

### 4. 断点的使用

使用断点也可以使程序暂停。但一旦设置了断点，不管是否还需要调试程序，每次执行程序都会在断点上暂停。因此调试结束后应取消所定义的断点。方法是先把光标定位在断点所在行，再按“编译微型条”中最右面的按钮或 F9 键，该操作是一个开关，按一次是设置，按两次是取消设置。如果有多个断点想全部取消，可执行“编辑”菜单中的“断点”命令，屏幕上会显示 Breakpoints 对话框，如图 1.15 所示，对话框下方列出了所有断点，单击 Remove All 按钮，将取消所有断点。断点通常用于调试较长的程序，可以避免使用 Run to Cursor（运行到光标）。

序到光标处暂停)或 Ctrl+F10 组合键功能时, 经常要把光标定位到不同的地方。而对于长度为上百行的程序, 要寻找某位置并不太方便。



图 1.15 Breakpoints (断点) 对话框

如果一个程序设置了多个断点, 按一次 Ctrl+F5 组合键会暂停在第一个断点, 再按一次 Ctrl+F5 组合键会继续执行到第二个断点暂停, 依次执行下去。

##### 5. 停止调试

使用 Debug 菜单的 Stop Debugging 命令, 或按 Shift+F5 组合键可以结束调试, 从而回到正常的运行状态。