第3章 程序结构

本章学习目标

- 熟练掌握简单语句的使用方法
- 熟练掌握 C++的输入/输出方法
- 了解程序的运行步骤和结构

本章先介绍简单语句，然后介绍 C++预处理命令、输入/输出的各种函数和输入/输出流对象，最后介绍程序的运行步骤和结构。

3.1 简单语句

3.1.1 表达式语句

在 C++中，所有对数据的操作和处理都是通过表达式来实现的，在表达式后面加分号就构成了表达式语句。

格式：

```
表达式;
```

例如：

“i++;”语句由自增表达式 $i++$ 后面加分号组成。

“sum=a+b;”语句由赋值表达式 sum=$a+b$ 后面加分号组成。

“3;”语句由常量表达式 3 后面加分号组成。

3.1.2 空语句

当程序中的某个位置在语法上需要一条语句，而在语义上又不要求执行任何动作时，可放上一条空语句。空语句不进行任何操作，仅是一个分号。

格式：

```
;
```

3.1.3 复合语句

当程序中的某个位置在语法上只允许一条语句出现，而在语义上又要执行一条以上的

语句才能完成某个操作时需要使用复合语句。复合语句是由左、右大括号{ }括起来的一条或多条语句。

格式：

```
{语句组 }
```

例如，复合语句{ int a,b;a=4;b=7;}由 3 条语句组成。

3.2 预处理命令

在程序编译之前进行的处理称为预处理，预处理一般以＃开头。C++中提供了 3 种预处理命令，即宏定义、文件包含和条件包含。预处理命令所在行的结尾不能有分号，而且一行只能书写一条预处理命令。

3.2.1 “文件包含”命令

“文件包含”将一个文件的内容包含到当前文件中，其作用是节省程序设计者的重复工作。C++语言除了有自己的“文件包含”处理命令方式外，还保留了 C 语言中的“文件包含”处理命令方式。

1. C 语言中的“文件包含”命令

C 语言提供了＃include 命令实现“文件包含”的操作，被＃include 命令包含的文件称为头文件，头文件的扩展名为.h。

格式 1：

```
＃include <文件名.h>
```

功能：从系统指定的文件夹中寻找所给定的头文件，并把头文件的信息包含到当前源程序文件中。

格式 2：

```
＃include "文件名.h"
```

功能：先从当前文件夹中寻找给定的头文件，若找不到，自动在系统指定的文件夹中寻找所给定的头文件，并把头文件的信息包含到当前源程序文件中。

例如：

```
＃include <iostream.h>
＃include <math.h>
```

其含义是分别把头文件 iostream.h 中的输入/输出对象和 math.h 中的数学函数包含到当前源程序文件中。

2. C++语言中的“文件包含”命令

C++标准库中的类和函数都在命名空间 std 中声明，因此，C++语言除提供了#include命令实现“文件包含”的操作外，还要使用命令“using namespace std;”才能把指定的头文件包含到当前源文件中。“using namespace std;”的意思是“使用命名空间 std”。被#include命令包含的文件称为头文件，C++头文件没有扩展名。

格式 1：

```
#include <文件名>
using namespace std;
```

功能：从系统指定的文件夹中寻找所给定的头文件，并把头文件的信息包含到当前源程序文件中。

格式 2：

```
#include "文件名"
using namespace std;
```

功能：先从当前文件夹中寻找给定的头文件，若找不到，自动在系统指定的文件夹中寻找所给定的头文件，并把头文件的信息包含到当前源程序文件中。

例如：

```
#include "cmath"
using namespace std;
```

其含义是把头文件 cmath 中的数学函数包含到当前源程序文件中。

3.2.2　宏定义

格式 1：

```
#define  标识符  常量
```

功能：用标识符替换常量，其中，标识符称为符号常量。

格式 2：

```
#define  宏名(参数表) 常量
```

功能：用宏名(参数表)代替常量。

注意：预处理命令在编译之前被处理，处理时宏定义中的常量只做替换，不做计算，也不对常量进行求解。

例如：

```
#define a 3 * 4
#define b 5 + 6
#define S(a,b) a * b
```

经过预处理后，S(a,b)已经替换为 3 * 4 * 5+6，程序执行时，S(a,b)的值为 66。

3.3 数据的输入/输出

在C++语言中,没有专门的输入/输出语句,C++的输入/输出是通过输入/输出函数和流对象来完成的。

3.3.1 标准输入/输出函数

C++语言保留了C语言中用于输入/输出单个字符的函数。在C++语言中,标准输入/输出是通过输入/输出库“stdio.h”提供的库函数实现的。

1. 字符输出函数

格式:

```
putchar(字符)
```

功能:在标准输出设备(即显示器)中输出一个字符。
例如:

```
putchar('a');
```

执行该语句后,输出字符a。

```
char d = '3'; putchar(d);
```

执行这两条语句后,输出字符3。

2. 字符输入函数

格式:

```
getchar()
```

功能:从标准输入设备(即键盘)中获取一个字符。
例如:

```
char x; x = getchar();
```

执行这两条语句后,从键盘输入一个字符赋给字符变量x,如果输入a,则x='a'。

3.3.2 格式化输入/输出函数

C++语言保留了C语言中的格式化输入/输出函数。在C++语言中,格式化输入/输出函数是通过输入/输出库“stdio.h”提供的库函数实现的。

1. 格式化输出函数

格式:

```
printf("格式控制",输出表)
```

功能：按给定的格式控制输出表中的各个输出项。

说明：

(1) 输出表如果有多个输出项，各输出项之间用逗号分隔。

(2) 输出项可以是表达式、字符和字符串。若是表达式，则先计算表达式的值，然后将该值输出。若是字符或字符串，则按字符或字符串原样输出。

(3) 格式控制用于指定数据项的输出格式，它包含格式控制符和普通字符。

(4) 格式控制符以%开头，其格式为“%<标志><域宽><.精度><转换说明符>”。

其中，<标志>、<域宽>和<.精度>是任选项。常用的<标志>如表 3.1 所示；<域宽>是整型表达式，表示输出数据项的宽度(包括小数点)；<精度>是整型表达式，表示输出数据项中小数位的宽度；常用的<转换说明符>如表 3.2 所示。

表 3.1　printf 函数常用的标志

标　志	含　义
－	输出在域宽内左对齐
＋	在正数值之前显示一个加号，在负数值之前显示一个减号
空格	在正数值之前显示一个空格
0	用 0 填充域宽

表 3.2　printf 函数常用的转换说明符

类型字符	含　义	类型字符	含　义
d	十进制整型量	s	字符串
f	实型的小数形式	e	实型的科学记数法形式
c	字符	u	无符号十进制整型量

(5) 如果格式控制中有普通字符，则输出该普通字符。

例如：

```
int a = 3;float b = 56.789; char d = '*';
printf("a = %5d,b = %5.2f,d = %c,%c,%s\n",a,b,d,'x',"apple");
```

该语句普通字符有 'a'、'='、','、'b'、'='、'd'和'\n'等，格式控制符有%5d、%5.2f、%c和%s。输出表中的第一个输出项 a 与第一个格式控制符%5d 对应，输出项 a 按%5d 输出，依此类推，输出项 b 按%5.2f 输出，输出项 d 按%c 输出，输出项'x'按%c 输出，输出项"apple"按%s 输出。

执行该语句后的输出结果如下：

```
a =     3,b = 56.79,d = *,x,apple
```

2. 格式化输入函数

格式：

```
scanf("格式控制",变量地址表)
```

功能：按给定的格式控制将从键盘输入的数据分别赋给变量地址表中对应的变量。

说明：

(1) 变量地址表中可以有多个变量地址，各变量地址之间用逗号分隔。

(2) 格式控制用于指定输入数据的格式，包括格式指示符和普通字符。

(3) 格式指示符以%开头，其格式为："%＜宽度＞＜转换说明符＞"。

常用的＜转换说明符＞如表 3.3 所示；＜宽度＞是指变量获取输入数据的宽度。

表 3.3 scanf 函数常用的转换说明符

类型字符	含 义	类型字符	含 义
d	十进制整型量	s	字符串
f	实型的小数形式	e	实型的科学记数法形式
c	字符	u	无符号十进制整型量

(4) 如果格式控制中含有普通字符，在输入数据时按对应位置输入普通字符，作为相邻非字符型数据的分隔。普通字符可以是空白字符(空格符、制表符和回车符)。

例如：

```
int a;float b; char d;
scanf("%3d %f, %c",&a,&b,&d);
```

其中，&a、&b、&c 分别表示变量 a、b、c 的地址。"%3d %f, %c "是格式控制，%3d、%f 和 %c 是格式指示符，空格" "和逗号","是普通字符。变量地址表中的输入项 a、b、c 分别与%3d、%f 和 %c 对应，该语句的意义是从键盘中输入 3 个数据分别赋给 a、b、c，第 1 个数按%3d 格式输入，第 2 个数按%f 格式输入，第 3 个数按%c 格式输入，并且，第 1 个数与第 2 个数以空格分隔，第 2 个数与第 3 个数以逗号分隔。

在执行该语句时，如果输入 123 5，* ↵，则有 a=123，b=5，d='*'。

如果输入 12 3455 ，* ↵，则有 a=12，b=3455，d='*'。

如果输入 1234，5 ↵，则有 a=123，b=4，d='5'。

3.3.3 输入/输出流对象

在 C++语言中，输入/输出是通过输入/输出库"iostream.h"提供的输入/输出流对象实现的。在程序的开头增加一行预处理 #include ＜iostream.h＞或 #include "iostream.h" 后，在程序中才能使用输入流对象 cin 和输出流对象 cout。

1. 输出语句

在 C++语言中，输出语句由输出流对象和插入运算符＜＜组成。

格式：

```
cout <<表达式 1 <<表达式 2 <<…<<表达式 n;
```

说明：

(1) 数据的输出格式由系统自动决定。

(2) 表达式可以是变量、表达式、字符和字符串。若是变量，则输出变量的值；若是表达

式，则先计算表达式的值，然后将该值输出；若是字符或字符串，则按字符或字符串原样输出。

例如，设“int a＝2；”“cout＜＜a；”表示输出 a 的值，执行该语句后在屏幕上显示 2；“cout＜＜ "abcd"；”表示输出字符串"abcd"，执行该语句后在屏幕上显示 abcd。

2. 输入语句

在 C++语言中，输入语句由输入流对象 cin 和提取运算符＞＞组成。

格式：

```
cin>>变量 1>>变量 2>>…>>变量 n;
```

说明：从键盘输入的数据用空白符(空格、回车键、制表符 Tab)分开。

例如，设“int a；　float b；”，执行“cin＞＞a＞＞b；”后，在键盘上输入用空格分开的两个数分别赋给 a 和 b，如果输入 3 45.6 ↵，则有 $a=3$，$b=45.6$。

3.4　C++程序结构

例 3-1　输入圆的半径(要求大于 0)，求圆的面积。

分析：设用 r 表示圆的半径，用 s 表示圆的面积，则有 $s=\pi r^2$。其中，π 是常数，r 是因变量，这样根据给定 r 的值计算 s 的值。

程序：

```
#include <iostream.h>                          //第 1 行，预处理命令
void main( )                                   //第 2 行，主函数 main
{                                              //第 3 行，函数体开始
    double r;                                  //第 4 行，定义双精度实型变量 r
    cout <<"输入圆的半径"<<'\n';                //第 5 行，输出语句
    cin >> r;                                  //第 6 行，输入语句
    double s;                                  //第 7 行，定义双精度实型变量 s
    s = 3.14 * r * r;                          //第 8 行，赋值语句
    cout <<"圆的面积为"<< s << endl;            //第 9 行，输出语句
}                                              //第 10 行，函数结束
```

运行结果：

```
输入圆的半径                                   (本行为输出)
4.8 ↵                                          (本行为输入)
圆的面积为 72.3456                             (本行为输出)
```

程序分析：该程序共有 10 行，每一行中有注释的标志符//，它表示从“//”开始到行末的内容全部为注释。第 1 行是预处理命令，用于把对象 cin、cout 包括到本程序中。第 2～10 行定义主函数，函数名为 main，该函数的类型为 void，其中，第 2 行是函数首部，第 4～9 行是函数体。程序的执行从主函数开始，依次执行第 4、5 行，直到执行第 10 行，程序运行结束。

说明：

(1) 注释不是程序的执行部分，是为了增加程序的可读性和可理解性而添加的。注释

除了可以用“//”注释外，还可以用/*……*/对C++程序中的任何部分做注释，从/*开始到*/之间都是注释内容。在用“//”注释时，有效范围是本行。当注释内容较少时常用“//”，当注释内容较长时用/*……*/注释。

(2) 在输入或输出时，常用文字信息对用户进行提示。例如，在例3-1的程序中第9行中的"圆的面积为"。

(3) 按缩排规则书写程序。

缩排规则是指书写程序时同一层次的结构在同一列开始，下一层的结构比上一层的结构缩进两个字符的位置开始。例如例3-1中，第3行的左大括号{和第10行的右大括号}是同一列位置，第4行的首字符比第3行缩进两个字符，第4行到第9行属于同一层，这些行的首字符都在同一列位置。

在例3-1的程序中，3.14是一个常量，若程序中多次出现该常量，可以用一个符号常量代替。

例3-2 输入半径，求圆的面积的周长。

程序：

```
#include <iostream.h>                          //第1行
#include <math.h>                              //第2行,预处理命令
#define PI 3.14                                //第3行,定义符号常量PI
void main()                                    //第4行
{                                              //第5行
    double r,s;                                //第6行
    cout <<"输入圆的半径"<<'\n';                 //第7行
    cin >> r;                                  //第8行
    s = PI * pow(r,2);                         //第9行,函数pow求r^2
    cout <<"圆的面积: "<< s << endl;             //第10行
    cout <<"圆的周长: "<< 2 * PI * r << endl;    //第11行
}                                              //第12行
```

该程序的功能是计算圆的面积和周长。第2行，预处理命令，用于把数学函数包括到本程序。第3行用符号常量PI代替3.14，第9行和第11行都用到了圆周率π的值3.14。如果要调整π的值，例如将π改为3.14159，则在第3行将3.14改为3.14159就可以，这样做到了“一改全改”。

从例3-1和例3-2的程序可知，一个简单的C++程序基本上由预处理命令和函数两部分组成。其中，函数由函数首部和函数体组成，函数体由一对大括号{}、变量定义和若干个语句组成。在程序中有且仅有一个主函数，并且程序总是从主函数开始执行。

3.5 C++程序运行的步骤

在Windows系统平台，Visual C++ 6.0不仅是一个C++编译器，而且是一个基于Windows操作系统的可视化集成开发环境。Visual C++ 6.0由许多组件组成，包括编辑器、调试器、连接和程序向导AppWizard、类向导Class Wizard等开发工具。Visual C++ 6.0为用户开发C++程序提供了一个功能齐全的集成开发环境，能完成源程序的编辑、编译、连接

以及程序运行中的调试与跟踪等。

程序的运行过程经常需要重复5个步骤，即编辑、编译、连接、执行和分析结果。因此，程序的运行一般有Visual C++ 6.0的启动、编辑程序(建立源程序或打开源程序)、调试程序(编译、连接)和执行4个步骤。

1. 启动Visual C++ 6.0

方法一：单击“开始”按钮，选择“程序”→Microsoft Visual Studio 6.0→Visual C++ 6.0命令。

方法二：双击桌面上的Visual C++ 6.0的快捷方式。

启动Visual C++ 6.0后会出现如图3.1所示的界面。

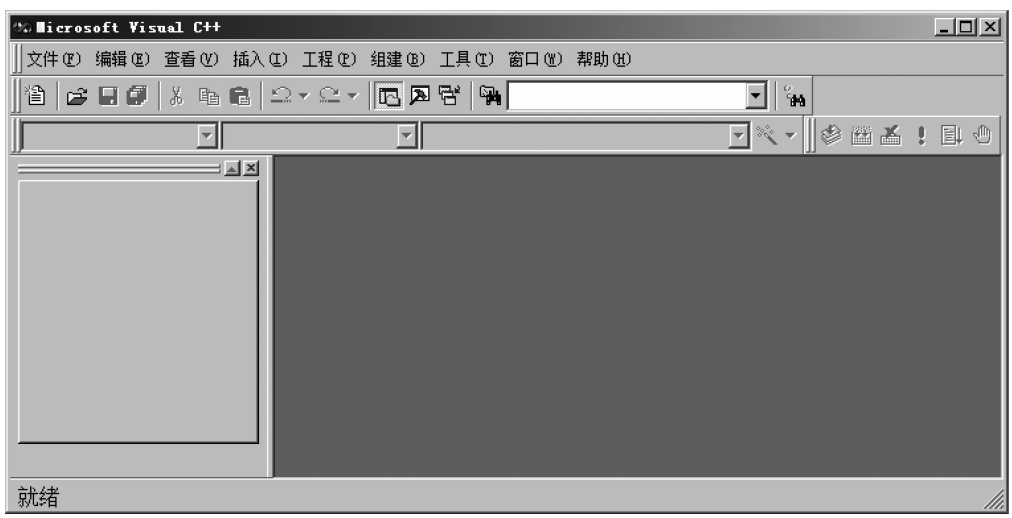

图3.1　Visual C++ 6.0界面

2. 编辑程序

编辑程序指使用C++文本编辑器编写程序，并保存到文件中，该文件被称为源程序，源程序的扩展名为.cpp。

1) 创建工程

选择“文件”→“新建”命令，弹出“新建”对话框，切换到“工程”选项卡，选择Win32 Console Application，在“位置”下面的文本框中选择驱动器(例如f:)，在“工程名称”下面的文本框中输入工程名(例如test)，选择“创建新的工作空间”单选按钮(如图3.2所示)，然后单击“确定”按钮。

在建立工程后，在f盘中会自动建立文件夹test，该文件夹中还有子文件夹debug。

2) 建立程序文件

选择“文件”→“新建”命令，弹出“新建”对话框，选择C++Source File(如图3.3所示)，在“文件名”下面的文本框中输入文件名(例如tt)，单击“确定”按钮会出现如图3.4所示的程序编辑窗口，在该窗口中可以输入程序行。

图 3.2 创建工程的界面

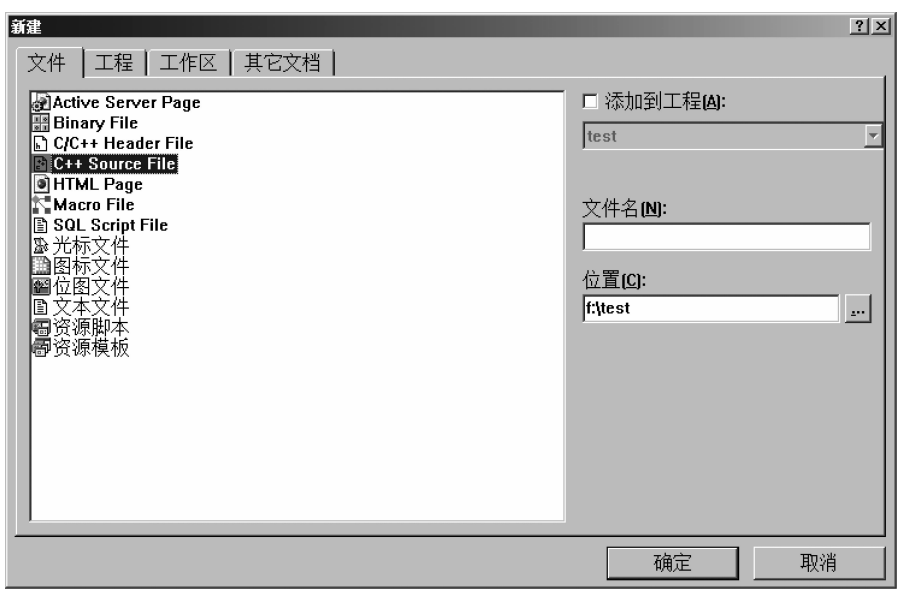

图 3.3 建立文件的界面

3. 保存程序

在输入程序后，为了避免程序不小心丢失，在编译之前最好进行保存。

方法：选择“文件”→“保存”命令。

注意：若创建了工程，保存的源程序将自动存放在文件夹 test 中。

图 3.4　程序编辑窗口

4. 编译与连接

1）编译

编译主要是通过编译器检查源程序是否有语法错误，若没有语法错误，则把源程序翻译成计算机能识别的目标程序。

编译的方法：

(1) 使用菜单命令。

选择“组建”→“编译”命令。

(2) 使用“调试”工具栏中的按钮

单击程序编辑窗口上方的“调试”工具栏中的“编译”按钮 。

在编译时，系统自动检查源程序是否有语法错误，然后在主窗口下部的调试信息窗口中输出编译信息，如果有错，就会指出错误所在的行和错误的信息，图 3.5 表明该程序出现了语法错误。

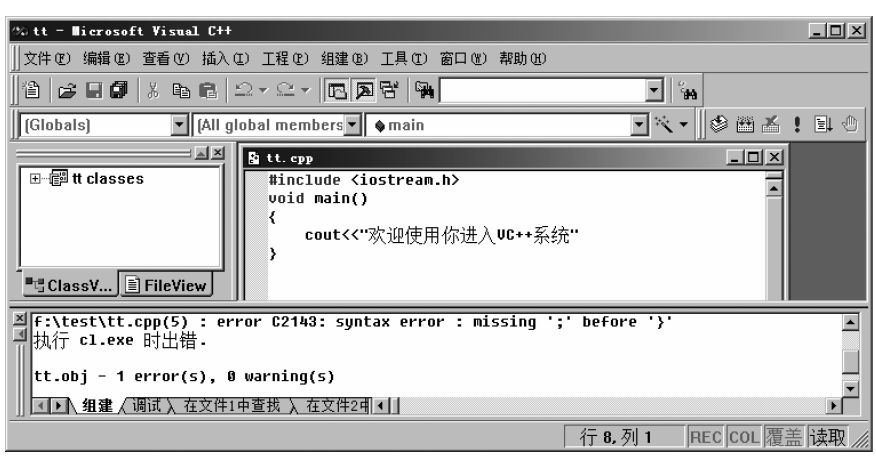

图 3.5　程序编译的错误结果

在图 3.5 中，“tt.obj - 1 error(s)。0 warning(s)”表示目标文件 tt.obj 中有一个错误，“f:\test\tt.cpp(5) : error C2143: syntax error : missing ';' before '}'”表示 f 盘的文件夹 test 中的程序文件 tt.cpp 的第 5 行“}”之前丢失了分号“;”。因此，在程序的第 4 行的后面加上一个分号“;”，再单击“编译”按钮，会得到如图 3.6 所示的信息，该信息表示程序没有语法错误，也就是说，程序通过了编译。

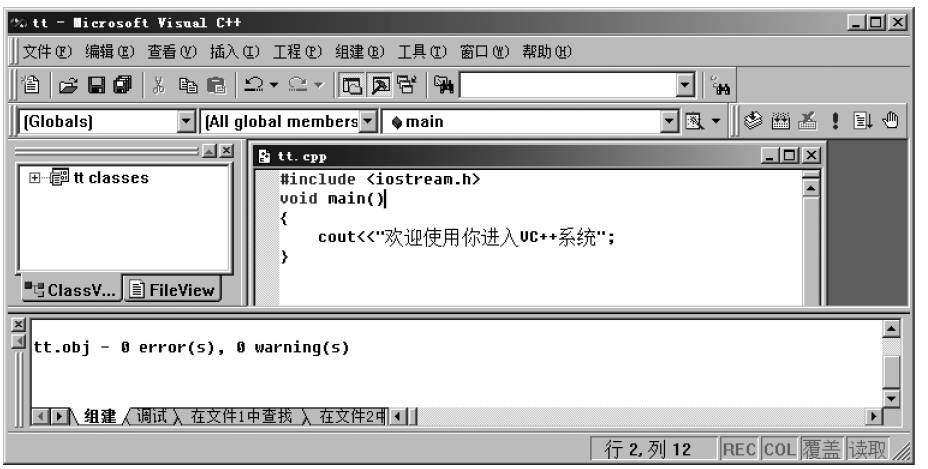

图 3.6 程序编译的正确结果

修改语法错误的方法：在调试信息窗口中双击某一个错误信息，光标会自动移到编辑窗口中该错误信息对应的所在行，然后根据错误信息提供的错误原因进行修改。

注意：*程序在编译时会自动生成一个与源程序同名的目标文件，其扩展名为.obj。*

2) 连接

程序通过编译后才能连接，连接是将目标程序与其他代码(例如库函数的目标代码和其他源程序的目标代码)连接起来，最终形成一个可执行的文件，该文件名与源程序同名，其扩展名为.exe。简单地说，连接的作用主要是将源程序生成可执行文件。

连接的方法：

(1) 使用菜单命令。

选择“组建”→“组建”命令。

(2) 使用“调试”工具栏中的按钮。

单击程序编辑窗口上方的“调试”工具栏中的“连接”按钮 。

程序连接后，在当前工程(或文件夹)的 debug 文件夹中会自动生成一个与源程序同名的可执行文件，其扩展名为.exe。

5. 执行程序

程序在连接后就可以运行并显示结果。

运行程序的方法：

(1) 使用菜单命令。

选择“组建”→“执行”命令。

(2) 使用“调试”工具栏中的按钮

单击程序编辑窗口上方的“调试”工具栏中的“运行”按钮 ！。

6. 打开程序文件

方法：选择“文件”→“打开” 命令，在弹出的对话框中选择文件夹，并输入程序文件名。

7. 关闭工作区

若在运行一个程序之后运行另一个程序文件，则在运行另一个程序文件之前一定要关闭当前的工作区，否则会出现错误。

方法：选择“文件”→“关闭工作区”命令。

8. 调试程序

如果在程序编译或连接过程中出现了错误，需要对程序进行调试。调试程序分为两类，一类是改正编译系统检查出来的语法错误，语法错误有两种，分别是错误(error)和警告(warning)；另外一类是检查程序的逻辑是否出错。

1) 调试程序的功能键

F9：在当前光标所在的行断点，如果当前行已经有断点，则取消断点。

F5：调试状态运行程序，程序执行到有断点的地方会停下来。

F10：单步执行程序。

Ctrl＋F10：运行到光标所在的行。

Shift＋F5：停止调试。

F11：如果当前黄色箭头所指的程序行含有函数的调用，则按 F11 键进入函数体进行调试跟踪

2) 调试程序的方法

(1) 在工程文件夹中建立程序文件；

(2) 在程序的结尾处按 F9 键设定断点；

(3) 按 F10 键，在调试窗口中输入要观察值的变量；

(4) 连续按 F10 键，观察程序的运行过程和变量的值；

(5) 当程序运行到断点处时按 Shift＋F5 组合键结束调试。

3) 简单程序的调试实例

调试程序的过程：

(1) 在文件夹 test 中建立、编译和运行程序，如图 3.7 中的程序。

(2) 第一次按 F10 键时，在主函数 main 下方的“{”的左边会出现一个黄色箭头，表示程序从该处开始执行，并且在屏幕上还会出现一个运行窗口，在编辑窗口的左下方会出现变量窗口，在编辑窗口的右下方会出现观察窗口，在观察窗口中可以观察变量的当前值。例如，在观察窗口 Watch1 中分别输入程序中的变量 x 和 y，就可以观察这两个变量的值，如图 3.8 所示。

(3) 继续按 F10 键，当黄色箭头移到 cin＞＞x＞＞y 所在的行时，按 F10 键箭头不动，此时，光标应切换到运行窗口，并输入数据(例如 3 4)，按回车键后，光标应切换到编辑窗口，

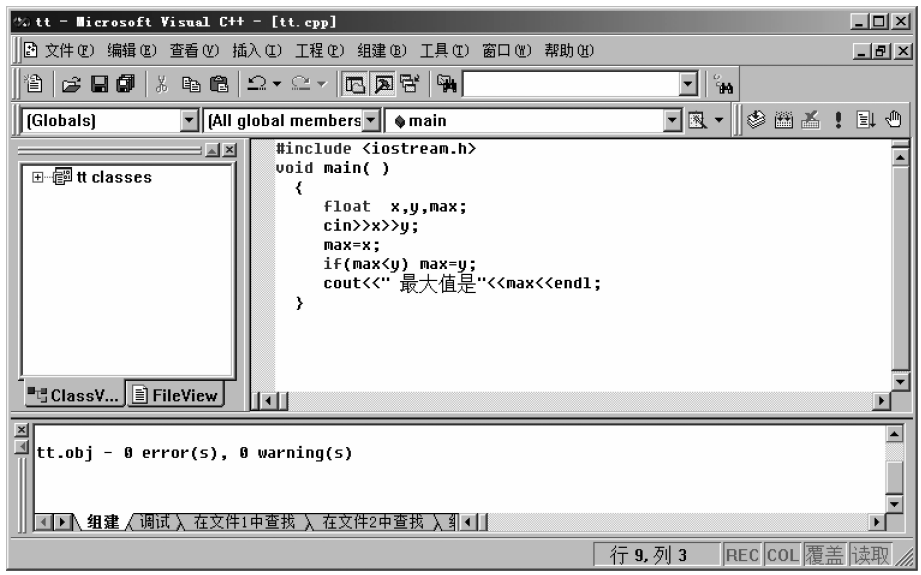

图 3.7 程序编辑窗口

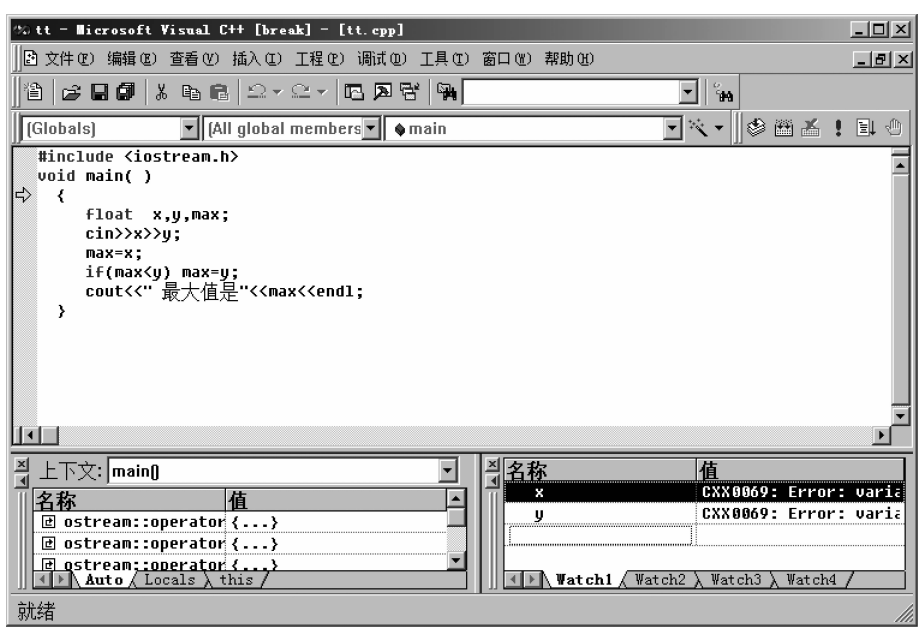

图 3.8 程序调试窗口

如图 3.9 所示。

(4) 连续按 F10 键,当黄色箭头移到最后一行的}时,单击“调试”命令,然后按 Shift+F5 组合键,终止单步调试程序,返回到编辑窗口。

4) 含函数程序的调试实例

调试程序的过程:

(1) 在文件夹 test 中建立、编译和运行程序,如图 3.10 中所示的程序。

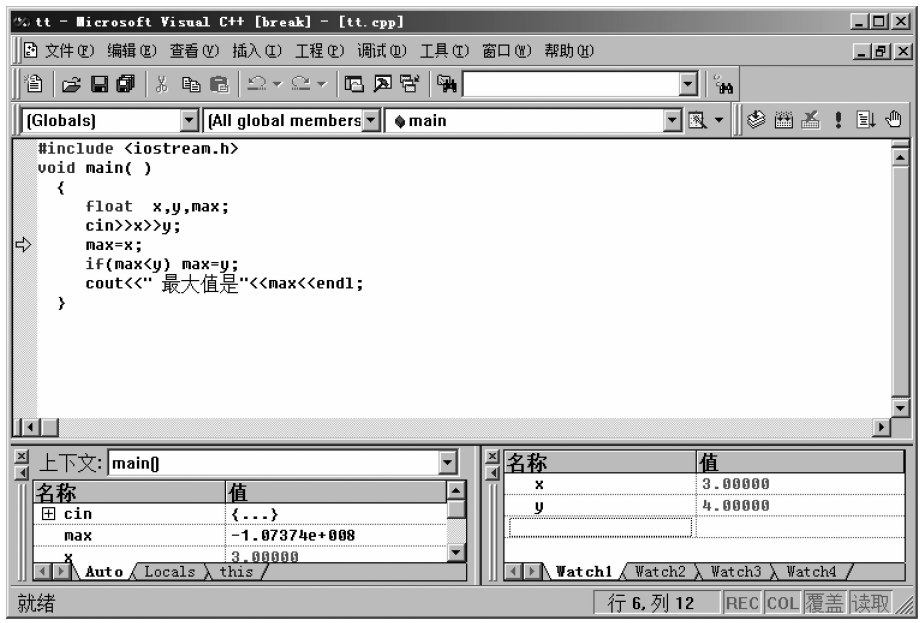

图 3.9　程序调试窗口

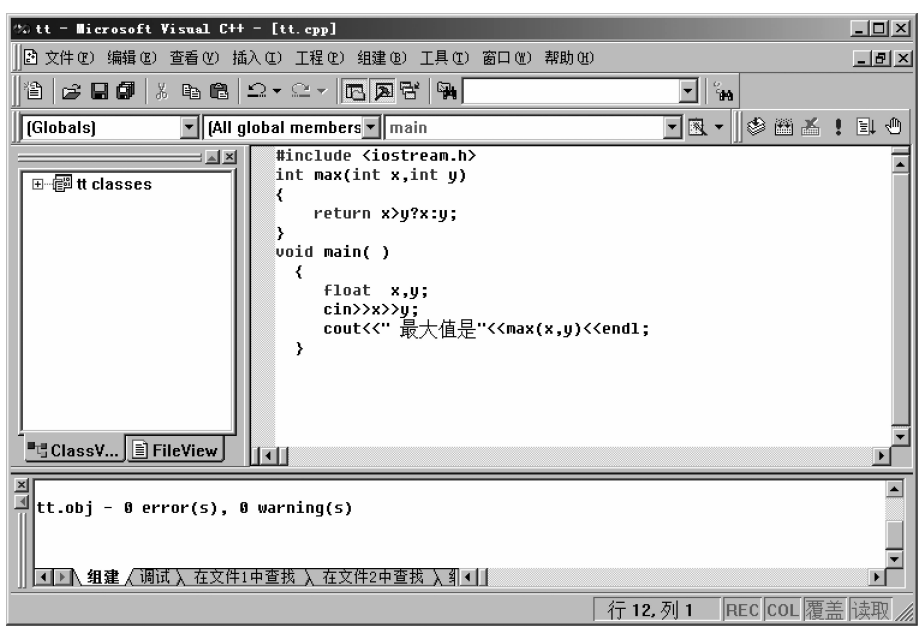

图 3.10　程序编辑窗口

(2) 第一次按 F10 键时,在主函数 main 下方的"{"的左边会出现一个黄色箭头,表示程序从该处开始执行。

(3) 继续按 F10 键,当黄色箭头移到 cin>>x>>y 所在的行时,按 F10 键箭头不动,此时,光标应切换到运行窗口,并输入数据(例如 3 4),按回车键后,光标应切换到编辑窗口。

(4) 继续按 F10 键,当黄色箭头移到"cout<<"最大值是"<<max(x,y)<<endl;"所

在的行时，按 F11 键，黄色箭头移到函数 max 中的“{”处，表示进入函数 max 体中执行，如图 3.11 所示。

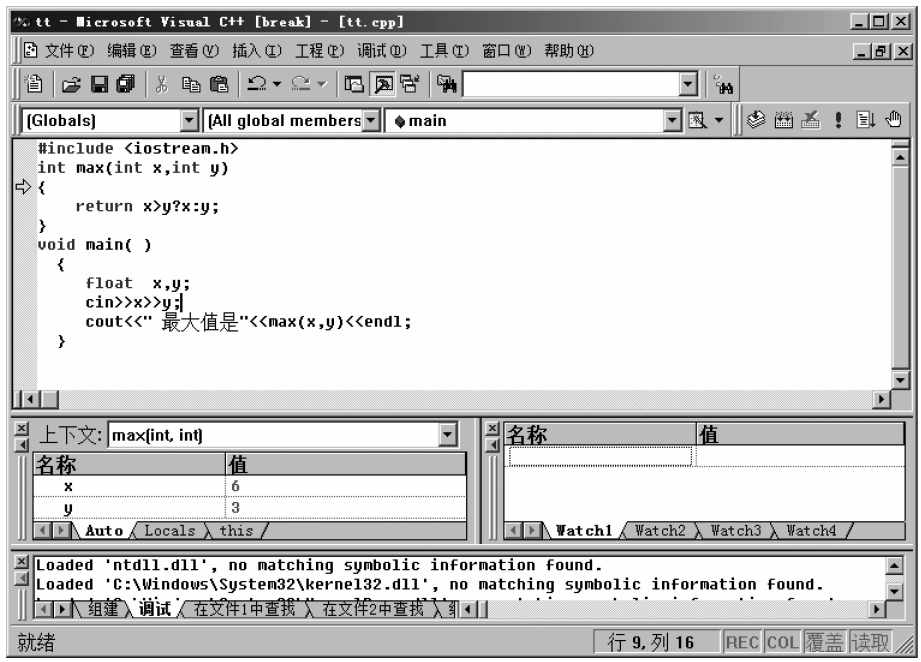

图 3.11 程序调试窗口

(5) 按 F10 键，当黄色箭头移到“return x>y? x:y; ”所在的行时，要结束函数 max 的调用并返回到函数 main 中，因此按 Shift+F11 组合键返回调用位置，即黄色箭头移到程序行“cout<<"最大值是"<<max(x,y)<<endl;”所在的位置。

(6) 按 F10 键，当黄色箭头移到最后一行的}时，单击“调试”命令，然后按 Shift+F5 组合键，终止单步调试程序，返回到编辑窗口。

说明：在程序调试窗口的观察窗口 Watch1 中可以输入要观察的变量。

习题 3

一、选择题

1. 下列关于预处理命令的描述错误的是_____。

(A) 预处理命令最左边的标识符是#　　(B) 宏定义可以定义符号常量

(C) 文件包含命令只能包含.h 文件　　(D) 宏定义中的符号常量是标识符

2. C++规定，每条语句以_____符号结束。

(A) 逗号　　(B) 感叹号　　(C) 分号　　(D) 双引号

3. 在 C++语句中，变量声明语句可以放在_____。

(A) 只能放在程序的开头部分　　(B) 程序的任何一个位置

(C) 只能放在函数的开头部分　　(D) 必须放在头文件

4. 下列选项中，属于C++语句的是_____。

（A）x * y　　（B）i=1　　（C）；　　（D）cout<<\'n'

5. C++程序的main函数的位置是_____。

（A）必须在程序的后面　　（B）必须在程序的开头

（C）可以在程序的任何位置　　（D）必须在其他函数之前

6. C++源程序文件默认的扩展名为_____。

（A）.c　　（B）.cc　　（C）.cpp　　（D）.c++

7. C++源程序文件经过编译后生成的目标文件的扩展名是_____。

（A）.cpp　　（B）.c　　（C）.exe　　（D）.obj

8. 表示C++的一条预处理命令开始的符号是_____。

（A）//　　（B）}　　（C）#　　（D）；

9. 每个C++程序必须有且仅有一个_____。

（A）函数　　（B）语句　　（C）预处理命令　　（D）主函数

10. 下面_____是合法的C++语句。

（A）# define SUM 100　　（B）m = 15；

（C）x = y = 30　　（D）/ * a =3；* /

11. 一个C++语言程序由_____。

（A）函数组成　　（B）若干过程组成

（C）若干子程序组成　　（D）一个主程序和若干子程序组成

12. 执行以下语句后，*b* 的值为_____。

```
int a=6,b=5,w=1,x=2,y=3,z=4;(a=y>z)&&(b=w>x);
```

（A）6　　（B）5　　（C）0　　（D）1

二、填空题

1. 复合语句是由_____条以上的语句加上_____组成的。

2. 表达式语句是_____加上分号组成的。

3. 标准输入设备是_____，标准输出设备是_____。

三、分析下列程序的输出结果

1.

```
#include<iostream.h>
void main()
{
 int x=10,y=10;
 cout<<x--<<' '<<--y<<endl;
}
```

2.

```
#include<iostream.h>
#define n 10
void main()
{
```

```
 int x = 5,a = 4;
 int p = (++a < 0)&&!(x -- < 0);
 cout << x << ' ' << p + n << ' ' << a << endl;
}
```

3.

```
#include < iostream.h >
#define SUB(X,Y) (X) * Y
void main()
{  int a = 3,b = 4;
   cout << SUB(a++,b++)<< endl;
}
```

4.

```
#include < iostream.h >
void main()
{ int x = 5,a = 4;
 cout << x << ' ' << a << endl ;
    {
     float x = 4/5;
     cout << x << ' ' << a++<< endl ;
    }
 cout << x << ' ' << a << endl ;
}
```

四、改错题

改正下列程序中 err 处的错误,使得程序能得到正确的结果。

注意:不要改动 main 函数,不得增行或删行,也不得更改程序的结构。

1. 输入 x、y、z 的值,输出 3 个数的最大值。

```
#include < iostream.h >
void main()
{ int x,y,z;
 cin >> x >> ' ' >> y >> ' ' >> z;                    //err1
 m = x > y?x:y;                                       //err2
 cout << m > z?m:z << endl ;                          //err3
}
```

2. 输入一个华氏温度 F,要求输出摄氏温度 C,$C=5/9(F-32)$。

```
#include < iostream.h >
void main()
{
 float F,C;
 cin >> F >> '\n';                                    //err1
 C = 5/9 * (F - 32);                                  //err2
 print(" % f\n",C);                                   //err3
}
```

五、编程题

1. 编写程序输出下列图案。

```
    *
   * * *
  * * * * *
 * * * * * * *
```

2. 编写程序输出下列图案。

```
*****************************
*        欢迎您进入 VC++环境 *
*****************************
```