1.3多文件项目的开发

在之前的示例代码中，宏定义、函数Add的原型声明和定义、主函数main都在一个.c文件中。当程序规模比较大时，往往需要将宏定义、函数原型声明、源代码分别保存在不同类型的多个文件中。本节将介绍多文件项目的开发和调试方法。

1.3.1 Visual Studio 下的多文件项目开发

创建项目并添加文件

使用在 1.2.1 节中所介绍的方法，在解决方案 VS2019Demo 中增加第三个 项目 Test3，并在该项目中添加以下不同类型的文件。

1. 添加.h文件: const.h

   首先，在解决方案资源管理器中的项目 Test3 上单击鼠标右键，如图 1-69所示，在弹出的菜单中选择：“添加”/“新建项”。之后会显示如图 1-70 所示的界面，在其中选择“头文件(.h)”，输入名称 const.h 即可完成添加 const.h 的操作。

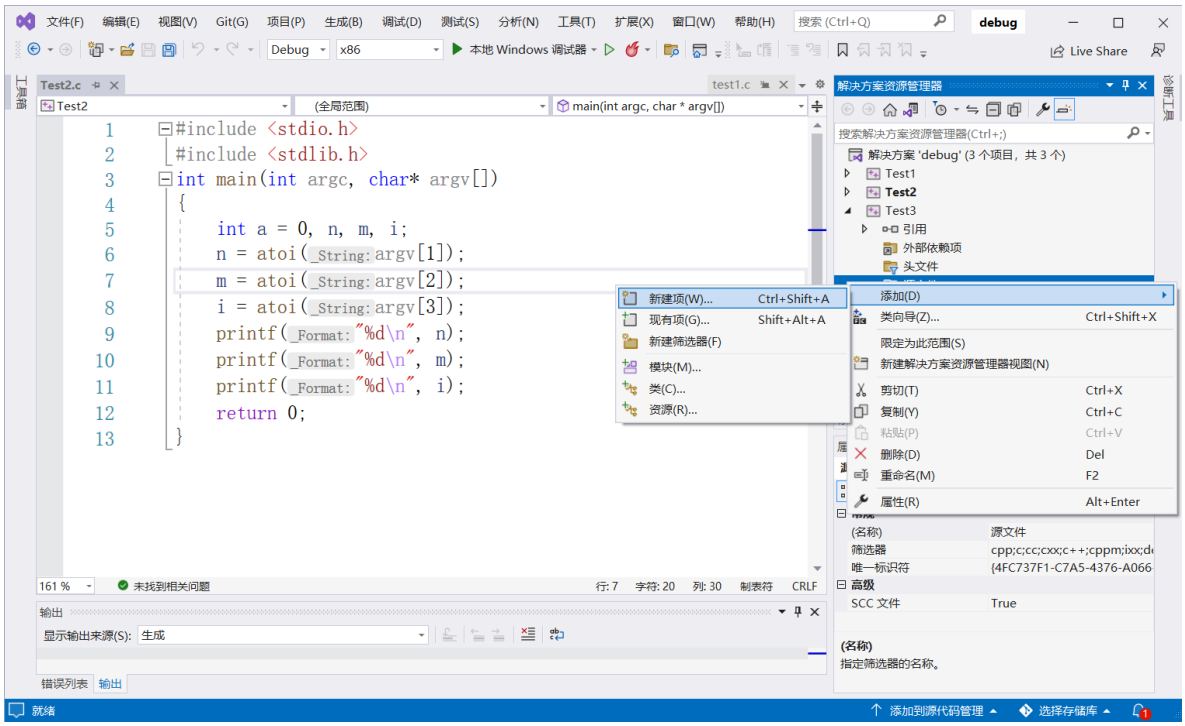

图 1-69 在项目中添加新建项的界面

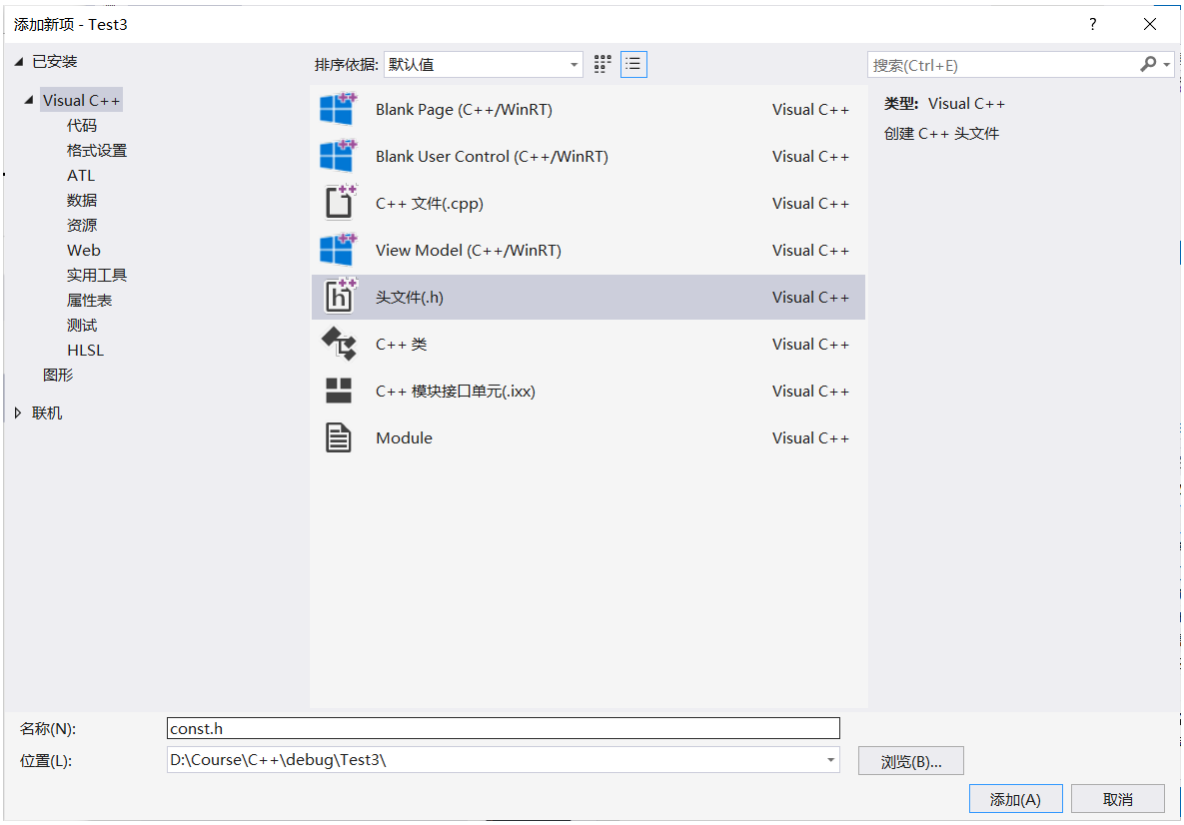

图 1-70 添加.h头文件的界面

编辑 const.h 文件，该文件内容如下：

```
#define MAXSIZE 5
#define PI 3.1415926
```

这里尽可能将所有类的定义以及宏定义放到一个文件中，在需要使用类定义和宏定义的代码文件(.h,.cpp)中，只要增加该头文件的包含语句即可，形式如下：

#include "const.h"

在大型项目中，不同的文件之间包含关系复杂，最终可能会导致一个代码文件直接、间接包含了某个.h 文件很多次（超过1次），进而导致编译错误— —宏的重复定义。为了避免这种情况，可以用条件编译语句 ifndef... #else... #endif，囊括.h 文件的全部内容。

```
#ifndef HIT_C_EXAMPLE_CONST_H
#define HIT_C_EXAMPLE_CONST_H
#define MAXSIZE 5
#define PI 3.1415926
#endif
```

添加条件编译语句后，相当于告知编译器：如果没有定义过宏 HIT_C_EXAMPLE_CONST_H，则先定义该宏，再定义 MAXSIZE、PI。如果 已经定义了 HIT_C_EXAMPLE_CONST_H，#ifndef 条件不成立，则该文件的内 容相当于空文件，也就不会导致重复宏定义了。这里，要保证 HIT_C_EXAMPLE_CONST_H 不会和别的代码文件中的宏名字重复。因此，通常采用这种较长的命名形式，甚至在末尾添加随机数串，以确保名称的唯一性。

2. 添加.h 文件 MyMath.h

将关于加法的一些函数原型声明写到该文件中，内容如下：

```
#include "const.h"
int Add(int a[], int n);
float Norm(int a[], int n);
```

这里为了简便，只写了一个函数。这样，如果希望加法函数 Add()等能在不同的源代码文件中被调用，也不需要在每个源代码文件中都做一次原型声明，在需要调用该函数的源代码文件中增加#include"MyMath.h"编译预处理指令即可。

3. 添加.c 文件 MyMath.c

```
#include <math.h>
#include "MyMath.h"
/*
功能：计算一维整型数组元素总和
参数：a 是一维数组，n 是数组长度（元素个数）
返回值：整型的数组元素总和数值
*/
int Add(int a[], int n)
{
int sum, i;
sum = 0;
for (i=0; i<n; i++)
{
sum += a[i];
}
return sum;
}
/*
功能：计算向量的模
参数：a 表示向量（一维数组），n 是向量的维数（数组长度、元素个数）
返回值：浮点型的向量模数值
*/
float Norm(int a[], int n)
{
int i;
float result = 0;
for (i=0; i<n; i++)
{
result += a[i] * a[i];
}
return sqrt(result);
}
```

该源代码文件负责实现 MyMath.h 中声明的函数。

C 语言中有两种方式使用#include 命令：

- 在指令#include 后用<>将头文件名括起来。这种方式用于标准或系统提供的头文件，可到保存系统标准头文件的位置查找头文件。

- 在指令#include 后用双引号""将头文件括起来。用这种方式时，编译器先查找当前目录是否有指定名称的头文件，再从标准头文件目录中查找。这种方式常用于程序员自己定义的头文件。

4. 添加.h 文件 Area.h

   将与面积计算相关的函数的原型声明写在该文件中，内容如下：

```
#include "const.h"
float CircleArea(float r);
float SphereArea(float r);
```

5. 添加.c 文件 Area.c

```
#include "Area.h"
int iCallAreaTimes = 0;
/*
功能：计算圆的面积
参数：r 是圆的半径
返回值：浮点型的面积数值
*/
float CircleArea(float r)
{
    iCallAreaTimes++;
    return PI * r * r;
}
/*
功能：计算球体的表面积
参数：r 是球的半径
返回值：浮点型的面积数值
*/
float SphereArea(float r)
{
    iCallAreaTimes++;
    return 4 * PI * r * r;
}
```

   该文件给出 Area.h 中各函数的完整定义。

6. 添加.c 文件 test.c

   该文件为主程序源代码文件，内容如下：

```
#include "const.h"
#include "Area.h"
#include "MyMath.h"
extern int iCallAreaTimes;
int main()
{
int sum;
int a[MAXSIZE] = { 5, 4, 3, 2, 1 };
sum = Add(a, MAXSIZE);
printf("sum =%d\n", sum);
```

```
printf("Area=%f,iCallAreaTimes=%d\n",CircleArea(1.2),iCallAreaT
imes);
printf("iCallAreaTimes=%d,Area=%f\n",iCallAreaTimes,SphereArea(
2));
return 0;
}
```

主程序中使用了宏 MAXSIZE、函数 Add()和函数 CircleArea()。因此，需要在 test.c 中包含它们的定义/原型声明.h 文件。

主程序中使用了 Area.c 中定义的全局变量 iCallAreaTimes。对于主程序文件test.c 来说，iCallAreaTimes 变量是“外部的”（extern）变量，即其他文件（Area.c）已经定义好的变量。因此，需要在程序前部添加外部变量声明语句：

```
extern int iCallAreaTimes;
```

这仅仅是条声明语句，形式清晰易懂，而非变量定义，也不能指定初始数值。完成上述文件添加后，在项目 Test3 的文件夹内可看到刚刚添加的.h 文件和.c 文件，如图 1-71 所示。

名称	修改日期	类型	大小
Area.c	2023/10/19 13:48	C Source file	1 KB
Area.h	2023/10/19 13:48	C++ Header file	1 KB
const.h	2023/10/19 13:44	C++ Header file	1 KB
MyMath.c	2023/10/19 13:46	C Source file	1 KB
MyMath.h	2023/10/19 13:46	C++ Header file	1 KB
test.c	2023/10/19 13:50	C Source file	1 KB
Test3.vcxproj	2023/10/19 13:33	VC++ Project	7 KB
Test3.vcxproj.filters	2023/10/19 13:33	VC++ Project Fil...	1 KB
Test3.vcxproj.user	2023/10/19 13:33	Per-User Project...	1 KB

图 1-71 Test3 的程序文件（阴影部分的文件是代码文件）

除上述方法外，还可以用文本编辑器逐个创建并编辑这些代码文件，将其保存成纯文本格式的文件。最后，在 VS 中可用如下方法将这些文件添加到项目中：鼠标右键单击项目名称 Test3，选择“添加（D）”/“现有项（G）”，或按 Shift+Alt+A 组合键，在之后弹出的窗口中单击 Ctrl+鼠标左键点选要添加的文件，再单击“添加”按钮即可。一次可以添加一个或多个文件。

编译和调试

由于项目包含多个文件，因此在编程和调试过程中会涉及多个文件的修改。在程序编译时，仅对修改的文件进行编译，可以节省编译时间。但在项目规模不算庞大（例如代码行数不到一万行）时，完全重新编译的速度也比较快。如果仅编译修改的文件，在调试运行时，VS 有时会提醒项目过期或者出现断点无法停止的问题，此时需要重新完整编译。因此，在多文件项目编译时， 推荐采用“重新生成”的方法对整个项目进行完整的编译。

在解决方案资源管理器中的项目 Test3 上单击鼠标右键，选择“重新生成”，如图 1-72 所示；或者如图 1-73 所示，先在解决方案资源管理器中将预编译的项目设定为启动项目：在项目名称 Test3 上单击鼠标右键，在弹出的菜单 上选择“设为启动项目”，然后在 VS 的主菜单中单击“生成”/“重新生成解决方案”。

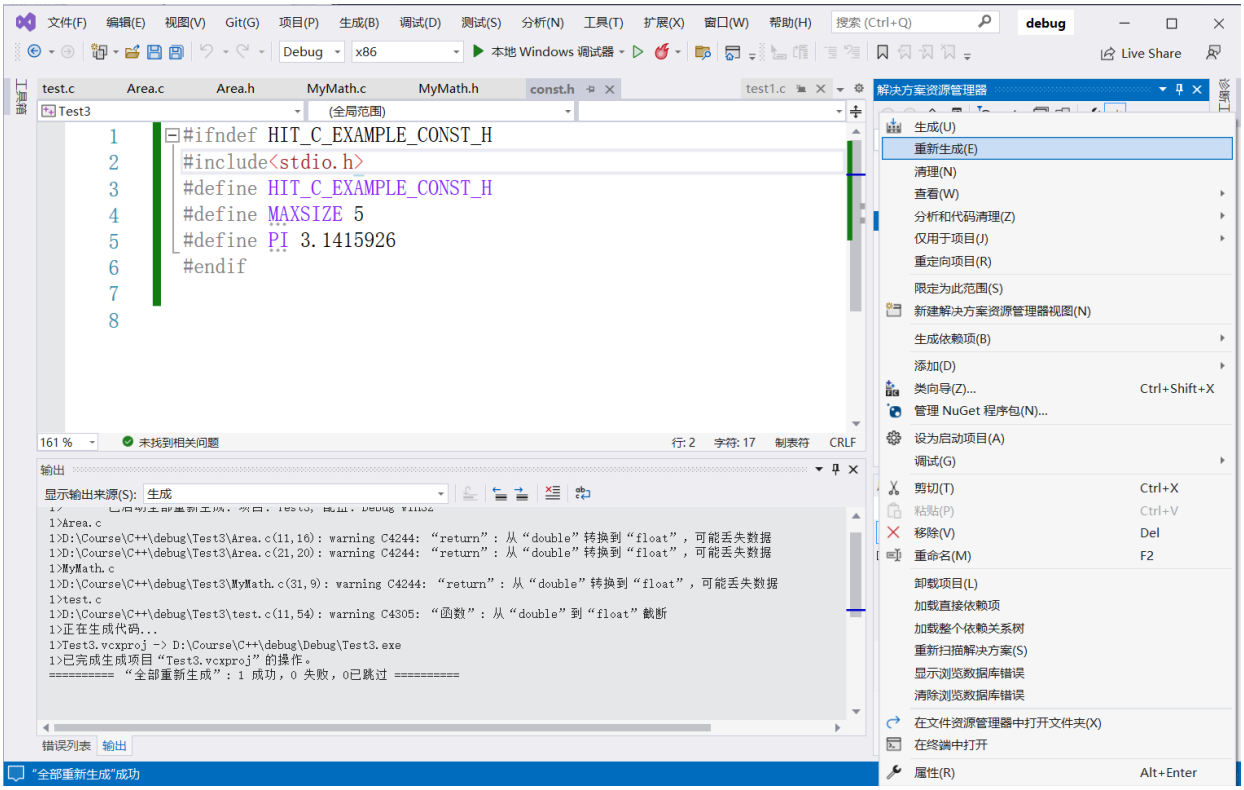

图 1-72 编译项目的方法：在项目名称上单击鼠标右键菜单

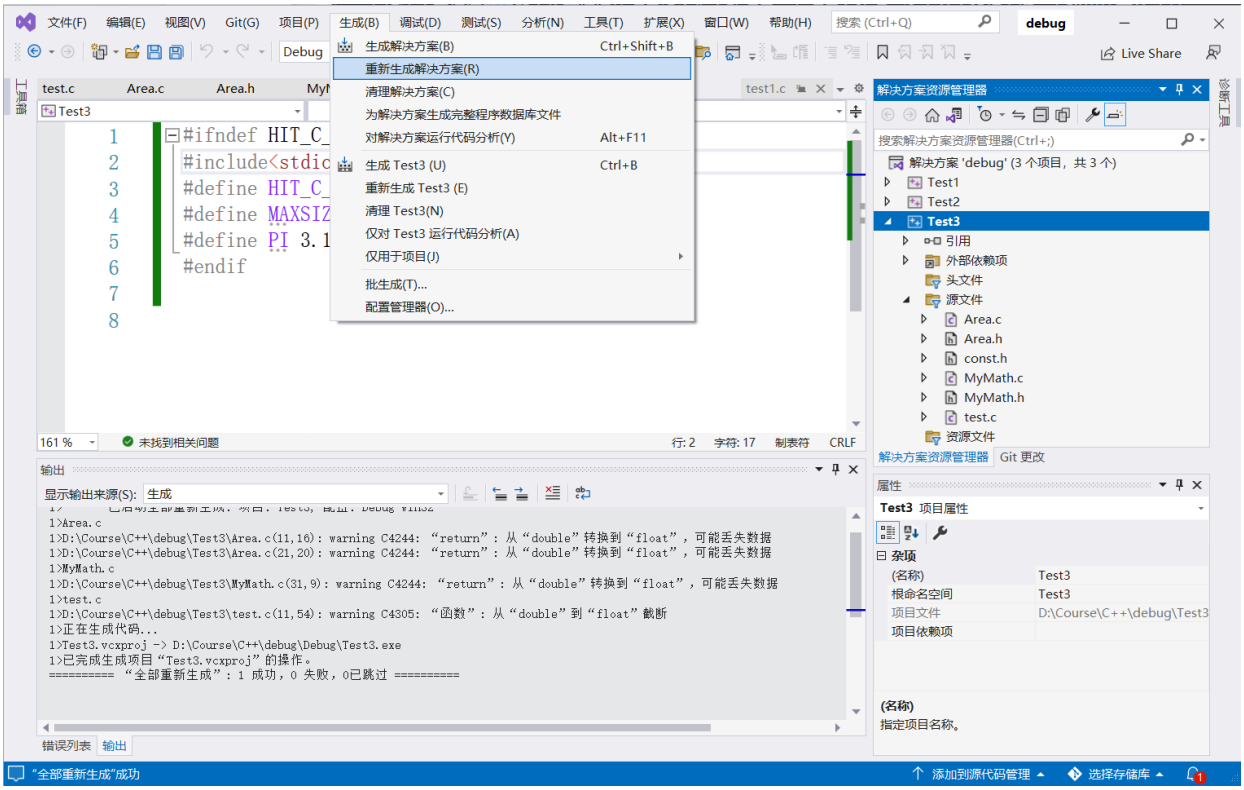

图 1-73 编译项目的方法：主菜单“生成”

多文件项目的调试过程和基本方法与普通单文件项目并无区别，只是在调试过程中，需要格外注意变量、函数、类或宏的重复定义、外部声明问题。