第 1 章

C++ 概 述

C++ 是一门以 C 语言为基础发展而来的面向对象的高级程序设计语言。本章将首先介绍 C++ 语言的发展过程，让读者对 C++ 有一个基本的认识，了解 C++ 的特点和运行机制，接着详细介绍 C++ 开发环境的配置，讲解如何创建 C++ 程序并运行，并对 C++ 开发过程中一些基本的注意事项进行了说明。

学习目标：

- 了解 C++ 与 C 的关系及 C++ 语言的特点。
- 了解 C++ 语言发展历史及应用领域。
- 了解 C++ 常用的开发环境种类。
- 掌握 Visual Studio 的安装及配置过程。
- 能编写并运行简单的 C++ 程序。

1.1 C++ 语言发展

1.1.1 C++ 与 C 语言的关系

1. C 语言

1967 年剑桥大学的 Martin Richards 为编写操作系统软件和编译程序开发了 BCPL 语言。1970 年，Ken Thompson 在继承 BCPL 语言诸多优点的基础上开发了实用的 B 语言。1972 年，贝尔实验室的 Dennis Ritchie 在 B 语言的基础上做了进一步的充实和完善，开发出了 C 语言。

C 语言具有许多优点：语言简洁灵活、运算符和数据类型丰富、具有结构化控制语句、程序执行效率高，同时具有高级语言和汇编语言的优点等。与其他高级语言相比，C 语言可以直接访问物理地址，与汇编语言相比又具有良好的可读性和可移植性。因此 C 语言得到了极为广泛的应用，后来的很多软件都用 C 语言开发，包括 Windows、Linux 等。

C 语言是面向过程的，只能把代码封装到函数，没有类。所谓面向过程，就是通过不断地调用函数来实现预期的功能。在 C 语言中，我们会把重复使用或具有某项功能的代码封装成一个函数，将具有相似功能的函数放在一个源文件，调用函数时，引入对应的头文件即可，如图 1-1 所示。

但是随着 C 语言应用的推广，其存在的缺陷也逐渐暴露出来。例如，C 语言对数据类型检查的机制比较弱，缺少支持代码重用的结构；随着计算机性能的飞速提高，软件工程规模的扩大，很多软件的大小都超过 1GB，如 Flash、Visual Studio 等，用 C 语言开发这些软件就显得非常吃力了。C 语言是一种面向过程的编程语言，不能满足运用面向对象的方法开发软件的需要。为克服 C 语言本身存在的缺点，同时为支持面向对象的程序设计，1980 年贝尔实验室的 Bjarne Stroustrup 在 C 语言基础上创建、开发出了一种通用的程序设计语言——C++ 。

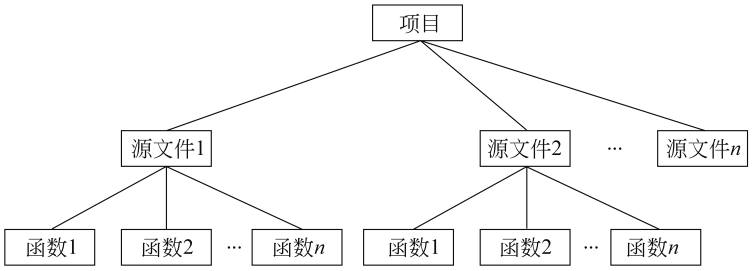

图 1-1　C 语言中项目的组织方式

2. C++ 语言

C++ 读作“C 加加”，是 C Plus Plus 的简称。C++ 是在 C 语言的基础上开发的一种集面向对象编程、泛型编程和过程化编程于一体的编程语言。

在 C++ 中，多了一层封装，就是类(Class)。类是一个通用的概念，C++ 、C♯、Java、PHP 等很多编程语言中都有类，可以通过类来创建对象(Object)。在 C++ 中，可以将一个类或多个类放在一个源文件中，使用时引入对应的类即可。封装让 C++ 多了很多特性，并成为一种面向对象的程序设计语言。C++ 中项目的组织方式如图 1-2 所示。

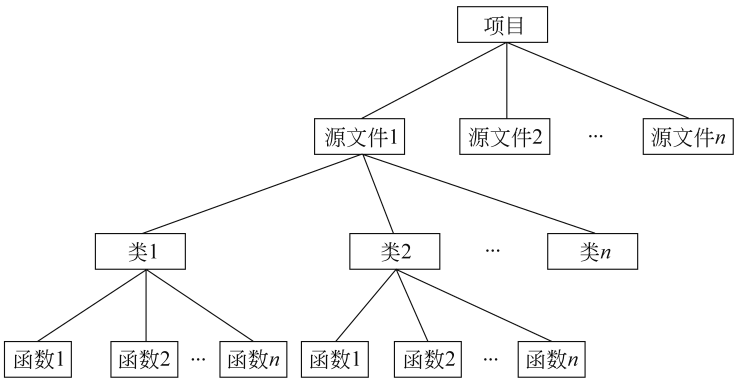

图 1-2　C++ 中项目的组织方式

注意：C 语言和 C++ 的源代码格式的区别如下。

- C 语言源文件后缀为.c。C++ 源文件后缀为.cpp。
- 通常，C++ 集成环境约定当源程序文件的扩展名为.c 时，则为 C 程序；而当源程序文件的扩展名为.cpp 时，则为 C++ 程序。
- .c 源文件会按照 C 语言的方式编译。.cpp 源文件会按照 C++ 的方式编译。C++ 几乎完全兼容 C 语言，它们的关系类似子集(C 语言)和超集(C++)的概念。

1.1.2　C++ 的特点及应用

1. C++ 的特点

(1) 保持与 C 兼容。C++ 既保持了 C 语言的优点，又对 C 的类型进行了改革和扩充，C++ 的编译系统能检查出更多的类型错误。另外，由于 C 语言的广泛使用，因而极大地促进了C++ 的普及和推广。大多数的 C 程序代码略作修改或不需修改就可在 C++ 的集成环境下

调试和运行。这对于继承和开发当前已广泛使用的软件是非常重要的,因为可以节省大量的人力和物力。

(2) C++ 语法灵活,功能强大。C++ 设计无须复杂的程序设计环境,语法思路层次分明,语法结构明确,对语法限制比较宽松,给用户编程带来书写上的方便;缺点是由于编译时对语法限制比较宽松,许多逻辑上的错误不容易被发现,给用户编程增加了难度。C++ 的很多特性都是以库或其他的形式提供的,而没有直接添加到语言本身。

(3) 面向对象的机制。C++ 是一种面向对象的程序设计语言,使开发人机交互类型的应用程序更为简单、快捷。它使程序的各个模块独立性更强,程序的可读性和可移植性更强,程序代码的结构更加合理,程序的扩充性更强,这对于设计、编制和调试一些大型的软件尤为重要。

(4) 适合大型系统的开发。C++ 可以应用于几乎所有的应用程序。从字处理应用程序到科学应用程序,从操作系统组件到计算机游戏等,C++ 都得到了广泛的应用和发展。

2. C++ 的应用

C++ 作为一门高效的程序设计语言,具体应用在以下几个领域。

(1) 系统编程。在该领域,C 是主要的编程语言,但是 C++ 凭借其 C 的兼容,可以方便嵌入汇编语言,实现底层的调用,适合开发系统级软件,编写驱动程序等。另外还可以开发基础软件和高级语言的运行时环境,如大型数据库软件、Java 虚拟机、C# 的 CLR、Python 编译器和运行时环境,以及目前见到的各种桌面应用软件,如 QQ、杀毒类软件等。

(2) 网络编程。在多线程、网络通信、分布应用、服务器端、客户端程序方面,C++ 有着其他语言不可比拟的优势。C++ 拥有很多成熟的用于网络通信的库,其中最具有代表性的是跨平台的、重量级的 ACE 库,该库可以说是 C++ 语言最重要的成果之一,在许多重要的企业、部门甚至是军方都有应用。

(3) 服务端开发。很多互联网公司的后台服务器程序都是基于 C++ 开发的,而且大部分是 Linux,UNIX 等类似操作系统,编程者需要熟悉 Linux 操作系统及其在上面的开发,熟悉数据库开发,精通网络编程。也应熟悉游戏的服务器后台,如魔兽世界的服务器和一些企业内部的应用系统等。

(4) 嵌入式开发。由于 C++ 既保持了 C 语言的优点,又对 C 的功能进行了扩充,因此具有较高的效率,同时由于它的灵活性,使它在底层开发中被极大地应用。低端嵌入式开发主要是基于汇编语言和 C 语言,中端嵌入式开发主要是使用 C 和 C++ 。

(5) 游戏工具开发。目前很多游戏客户端都是基于 C++ 开发的。除了一些网页游戏,C++ 凭借先进的数值计算库、泛型编程等优势,在游戏领域应用非常广泛。

1.2 开发环境

1.2.1 C++ 开发环境介绍

C++ 的开发工具很多,而且各有优缺点,初学者往往不知道有哪些常用的开发工具,或者由于面临的选择比较多而产生困惑。下面介绍几款常用的开发工具,帮助初学者了解 C++ 开发工具并做出选择。

1. Turbo C

Turbo C 使用了集成的开发环境,采用一系列下拉式菜单,将文本编辑、程序编译、连接以及程序运行一体化,极大地方便了程序的开发。

2. Visual Studio

Visual Studio 是一套完整的开发工具集，用于生成 ASP.NET Web 应用程序、XML Web Services、桌面应用程序和移动应用程序。Visual Basic、Visual C++ 、Visual C# 和 Visual J# 全都使用相同的集成开发环境 (IDE)，利用此 IDE 可以共享工具且有助于创建混合语言解决方案。

3. C++ Builder

C++ Builder 具有快速的可视化开发环境，包含一个专业 C++ 开发环境所能提供的全部功能：快速、高效、灵活的编译器优化，逐步连接，CPU 透视，命令行工具等。它实现了可视化的编程环境和功能强大的编程语言(C++)的完美结合。

4. VC++

VC++ 6.0 是 Microsoft 公司推出的运行在 Windows 操作系统中的交互式、可视化集成开发软件，它不仅支持 C++ 语言，也支持 C 语言。VC++ 6.0 集程序的编辑、编译、连接、调试等功能于一体，为编程人员提供了一个既完整又方便的开发平台。

5. Eclipse

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。这个组件主要针对希望扩展 Eclipse 的软件开发人员，因为它允许构建与 Eclipse 环境无缝集成的工具。

1.2.2 Visual Studio 2015 开发环境

Visual Studio 简称为 VS，Visual C++ 简称为 VC。最初 Visual C++ 发布时还没有 Visual Studio，Visual C++ 是一个独立的开发工具，与 Visual Basic 等并列，最后微软将它们整合在一起组成了 Visual Studio。

Visual Studio 是微软开发的一套工具集，它由各种各样的工具组成，Visual C++ 就是 Visual Studio 的一个重要的组成部分。在 Visual Studio 中，除了 VC，还有 Visual C#，Visual Basic，过去还有 Visual J#，现在还有 Visual F# 等组件工具。Visual Studio 可以用于生成 Web 应用程序，也可以生成桌面应用程序。

1.3 C++ 程序框架及运行过程

1.3.1 建立 C++ 程序

1. VS2015 下建立相关页面

(1) 打开 VS2015 后会出现主界面，如果没有“解决方案资源管理器”，可以从视图里打开，如图 1-3 所示。

图 1-3 解决方案资源管理器

(2) 从“文件”中，新建“项目”，在弹出对话框左侧“项目类型”中选择 Win32，在“模板”中选择“Win32 控制台应用程序”。然后，在“名称”文本框中输入项目名称 exe1，并在“位置”中输入储存位置，如图 1-4 所示。单击“确定”按钮，选择随后窗口里的“空项目”(其他项不做修改)，单击“完成”按钮，如图 1-5 所示。

(3) 做完上一步骤后，“解决方案资源管理器”会

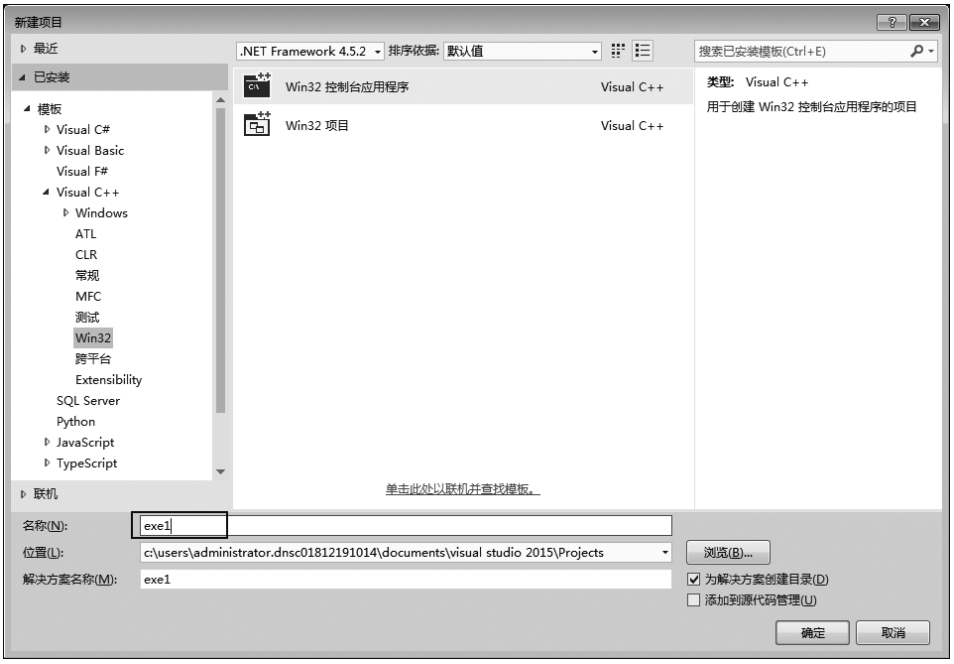

图 1-4　新建项目 1

图 1-5　新建项目 2

发生改变，右击其中的“源文件”，选择“添加”→“新建项”，如图 1-6 所示。随后会出现图 1-7，按图选择各项，并在“名称”中输入程序名 exe1_1.cpp，最后单击“添加”按钮。

2. 输入程序

【例 1-1】 输入如下符合 C++ 语法规范的语句，提示“input your name:”，用户输入名字***后，在屏幕输出 welcome!!! ***，代码如下所示。

图 1-6　新建.cpp 源文件 1

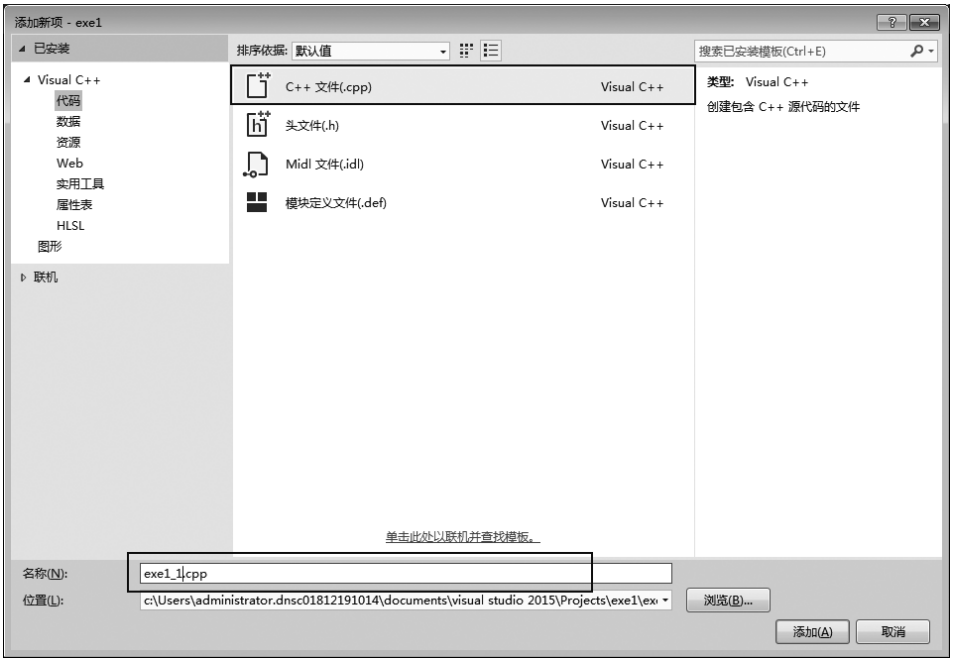

图 1-7　新建.cpp 源文件 2

```
1.  /*程序名:exe1_1*/
2.  #include <iostream>
3.  #include<string>
4.  using namespace std;                          /*预处理文件*/
5.  void main()                                    //主函数
```

```
6.  {
7.    string name;
8.    cout<<"input your name:" <<endl;                    //提示输入姓名
9.    cin>>name;                                          //输入
10.   cout<<" welcome!!! "<<name<<endl;                   //输出
11. }
```

程序输出结果如图 1-8 所示。

图 1-8　exe1_1.cpp 运行结果图

3. 程序说明

(1) 程序的 1～4 行是多行注释块。注释能够帮助读者理解程序,并为后续测试维护提供明确的指导信息。一般来说,C++ 的注释部分有以下两种编写风格。

- 多行注释:一对符号"/ * "与" * / "之间的内容。这是从 C 语言中继承来的一种注释形式。一般来说,这种方法用在程序及程序模块的开头,对整个程序或模块进行注释;

```
/* 注释行 1
       ...
   注释行 n */
```

- 单行注释:一行中符号"//"之后的内容。它只能占一行,是 C++ 语言特有的一种注释形式。注释行一般是用在程序中间,对某一行进行注释。

```
//注释行
```

(2) C++ 的编译预处理命令以"#"开头,如上述代码中的#include,即为一个编译预处理命令,它是一个文件包含命令。预处理指令通常在编译时由预处理器执行,没有编译成执行指令,通常也称作伪指令,预处理指令以换行结束即可。

(3) main 主函数,程序的入口点。

main 函数的主体内容用{}括起来,形成一个程序块,其中{表示函数开始,}表示函数结束。一个 C++ 程序只能有一个入口,所以如果一个程序含有多个源程序模块,也只能允许其中包含一个 main 函数。

(4) 第 7 行"string name;"定义一个字符串类型的变量,必须加入包含语句#include <string>。用";"表示语句结束。如果比较长可以分多行写,多个语句也可以写在一行。

(5) 第 8 行"cout<<"input your name:" <<endl;"输出一条提示语句,输出的内容用""引起来。其中 cout 标准输出流设备,通常代表显示设备,一般与插入操作符<<连用,endl 为换行符号。

```
cout<<对象 1<<对象 2<<...<<对象 n;
```

表示将对象 1、对象 2 等插入标准输出流设备中,从而实现对象在显示器中的输出。

(6) 第 9 行"cin>>name;"为标准输入流对象,通常代表键盘,与提取操作符>>连用,使用格式为:

```
cin>>对象 1>>对象 2>>...>>对象 n;
```

表示从标准输入流对象键盘中提取 n 个数据分别给对象 1、对象 2……对象 n。

思考：将 #include <iostream> 改为 #include <iostream.h> 是否可以运行？有什么不同？

答：不可以。<iostream>和<iostream.h>格式不一样，两者是两个不同的文件。

(1) C 语言与早期的 C++ 头文件名为<iostream.h>。新的 C++ 标准为了对 C 语言以及早期 C++ 文件提供支持，附带了这些头文件。因此，当使用<iostream.h>时，相当于在 C 中调用库函数，使用的是全局命名空间，也就是早期的 C++ 实现。

(2) <iostream>没有后缀，使用<iostream>时，由于该头文件没有定义全局命名空间，必须使用 using namespace std 语句，这样才能正确使用 cout。命名空间 std 封装的是标准程序库的名称，标准程序库为了和以前的头文件区别，一般不加.h。

4. std 和 namespace

std：标准命名空间。C++ 标准程序库中的所有标识符都被定义于名为 std(standard)的 namespace 中。

namespace：命名空间，是标准 C++ 中的一种机制，可以在不同的空间内使用同名字的类或者函数，用来控制不同类库相同名字的冲突问题。库或程序中的每一个 C++ 定义集被封装在一个命名空间中，如果其他的定义中有相同的名字，但它们在不同的命名空间，则不会产生命名冲突。

格式：

```
namespace 命名空间名
    {
        //命名空间成员
    }
```

使用方法如下。

(1) 直接指定标识符。

```
std::cout <<"hello"<<std::endl;
```

(2) 使用 using 关键字。以上程序可以写成：

```
using std::cout;
using std::endl;
cout <<"hello" <<endl;
```

(3) 最方便的就是使用 using namespace std，这样命名空间 std 内定义的所有标识符都有效，可直接使用。

1.3.2 C++ 运行过程

C++ 程序的开发步骤如下，开发过程如图 1-9 所示。

- 分析问题。根据实际问题，分析确定解决方法，选用适当的数学模型，并用自然语言或

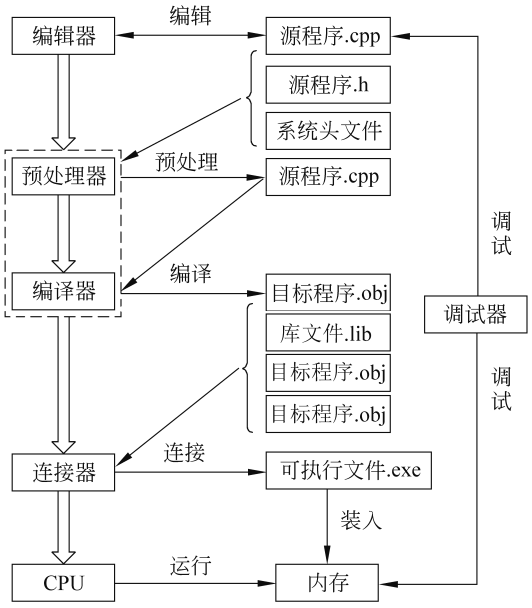

图 1-9　C++ 程序的开发步骤

流程图的方式描述解决问题的逻辑和算法。

- 编辑程序。根据上一步的算法利用编辑器编写 C++ 源程序，建立.cpp 文件。
- 程序预处理。当对一个源文件进行编译时，系统将自动引用预处理程序对源程序中的预处理部分作处理，处理完毕自动进入对源程序的编译。
- 编译程序。编译源程序，产生目标程序。文件的扩展名为.obj。
- 连接程序。将一个或多个目标程序与库函数进行连接后，产生一个可执行文件。文件的扩展名为.exe。
- 运行调试程序。运行可执行文件，分析运行结果。若有错误进行调试修改。

1. 编译预处理

编译预处理是 C++ 的一个重要功能。当对一个源文件进行编译时，系统将自动引用预处理程序对源程序中的预处理部分作处理，处理完毕自动进入对源程序的编译。

现在使用的 C++ 编译系统都包括了预处理、编译和连接等部分。不少用户误认为预处理命令是 C++ 语言的一部分，甚至以为它们是 C++ 语句，这是错误的，必须正确区分预处理命令和 C++ 语句。

C++ 提供的预处理功能主要有宏定义命令、文件包含命令、条件编译命令 3 种，主要处理 # 开始的预编译指令，如宏定义(#define)、文件包含(#include)、条件编译(#ifdef)等。这些命令以符号“#”开头，而且末尾不包含分号。合理使用预处理功能编写的程序便于阅读、修改、移植和调试，也有利于模块化程序设计。下面重点介绍宏定义和文件包含。

(1) 宏定义。宏定义的作用是实现文本替换。有两种格式：不带参数的宏定义和带参数的宏定义。

① 不带参数的宏定义，格式如下：

```
#define  宏名 字符串
```

define 是关键字,“宏名”是一个标识符,“字符串”是字符序列。该语句的意思是“宏名”代表“字符串”。执行该预处理代码时,编译系统将对程序语句中出现的“宏名”全部用“字符串”替代。

【例 1-2】 利用宏表示圆周率,求圆面积。

```
/*程序名:exe1_2*/
#include <iostream>
#define PI 3.14159      //宏定义。宏名是 PI,PI 将用 3.14159 替代
using namespace std;
void main( )
{ double s,r;
  cout<<"input r :";
  cin>>r;
  s=PI*r*r;
  cout<<"the result is :"<<s<<endl;
}
```

程序输出结果如图 1-10 所示。

图 1-10 exe1_2 运行结果图

程序说明:

- 在定义宏时,“宏名”和“字符串”之间要用空格分开。而“字符串”中的内容不要有空格。
- 宏名通常用大写字母定义。
- 宏定义后,可以在本文件各个函数中使用。也可以使用 # undef 取消宏的定义。宏定义可以嵌套,已被定义的宏可以用来定义新的宏。

```
#define  M  3
#define  N  M+2     //宏定义嵌套定义,用已定义的宏 M 来定义新的宏 N
```

② 带参数的宏定义,格式如下:

```
#define  宏名(参数列表)  字符串
```

带参数宏定义不只是简单的文本替换,还要进行参数替换。

【例 1-3】 利用带参数的宏求圆面积。

```
/*程序名:exe1_3*/
#include <iostream>
#define PI 3.14159
#define S(r) PI*r*r
using namespace std;
void main( )
{   double s,r;
    r=3.0;
    s=S(r);
```