第1章

预备知识

本章知识点：

- C++编程语言
- 计算机数据表示方法
- C++开发工具

基本要求：

- 了解 C++编程语言的发展历程
- 掌握计算机表示数据的方法
- 理解 C++开发工具的使用方法

能力培养目标：

通过本章的学习，掌握计算机中数据表示方法以及进行 C++编程语言开发的工具。

1.1 C++简介

C++编程语言是在 C 语言的基础上发展而来的。C 语言是一种通用的、过程式的编程语言，广泛用于系统与应用软件的开发，具有高效、灵活、功能丰富、表达力强和较高的移植性等特点，在程序员中备受青睐，是最近 25 年使用最为广泛的编程语言。C 语言是在由 UNIX 的研制者丹尼斯·里奇（Dennis Ritchie）和肯·汤普逊（Ken Thompson）于 1970 年所研制出的 B 语言的基础上发展和完善起来的。目前，C 语言编译器普遍存在于各种不同的操作系统中，如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。但是随着软件规模的增大，用 C 语言编写程序就显得吃力了。C 语言是结构化和模块化的语言，它是基于过程的，在处理较小规模的程序时，程序员用 C 语言还比较得心应手。但是问题比较复杂、程序的规模比较大时，结构化程序设计方法就显出它的不足。C 程序的设计者必须细致地设计程序中的每一个细节，准确地考虑到程序运行时每一时刻发生的事情。这对程序员的要求是比较高的，如果面对的是一个复杂问题，程序员往往感到力不从心。当初提出结构化程序设计方法的目的是解决软件设计危机，但这个目标并未完全实现。

为了解决软件危机，在 20 世纪 80 年代提出了面向对象的程序设计（Object Oriented Programming，OOP），需要设计出能支持面向对象的程序设计方法的新的语言。在实践中，由于 C 语言是深入人心、使用广泛、面对程序设计方法的革命，最好的办法不是另外发明一种新的语言去代替它，而是在它的原有基础上发展。在这种形势下，C++应运而生。

C++这个词通常被读作“C 加加”，而西方的程序员通常读作“C plus plus”、“CPP”。它是

一种使用非常广泛的计算机编程语言。C++是一种静态数据类型检查的、支持多重编程范式的通用程序设计语言。它支持过程化程序设计、数据抽象、面向对象程序设计、泛型程序设计等多种程序设计风格。C++扩充和完善了 C 语言，成为一种面向对象的程序设计语言。相对于 C 语言，C++提出了一些更为深入的概念，它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间，为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法。因而也增加了整个语言的复杂性，掌握起来有一定难度。

C++由美国 AT&T 贝尔实验室的本贾尼·斯特劳斯特卢普博士在 20 世纪 80 年代初期发明并实现（最初这种语言被称作“C with Classes”，即带类的 C）。开始，C++是作为 C 语言的增强版出现的，从给 C 语言增加类开始，不断地增加新特性。虚函数（virtual function）、运算符重载（operator overloading）、多重继承（multiple inheritance）、模板（template）、异常（exception）、RTTI、命名空间（name space）逐渐被加入标准。

C++编程语言的优点如下：

（1）C++设计成静态类型，是和 C 语言同样高效且可移植的多用途程序设计语言。

（2）C++直接和广泛地支持多种程序设计风格（程序化程序设计、资料抽象化、面向对象程序设计、泛型程序设计）。

（3）C++给程序设计者更多的选择。

（4）C++尽可能与 C 兼容，借此提供一个从 C 到 C++的过渡。

（5）C++避免平台限定或没有普遍用途的特性。

（6）C++不使用会带来额外开销的特性。

（7）C++设计无须复杂的程序设计环境。

（8）出于保证语言的简洁和运行高效等方面的考虑，C++的很多特性都是以库（如 STL）或其他的形式提供的，而没有直接添加到语言本身里。

（9）C++在一定程度上可以和 C 语言很好地结合，甚至目前大多数 C 语言程序是在 C++的集成开发环境中完成的。C++相对众多的面向对象的语言，具有相当高的性能。

C++引入了面向对象的概念，使得开发人机交互类型的应用程序更为简单、快捷。很多优秀的程序框架包括 MFC、QT、wxWidgets 使用的就是 C++。

然而 C++也有其自身的约束性：C++由于语言本身过度复杂，其语义甚至难于理解。由于本身的复杂性，复杂的 C++程序的正确性相当难以保证。

1.2 计算机数据表示方法

计算机数据表示是指计算机硬件能够辨认并进行存储、传送和处理的数据表示方法。一台计算机的数据表示方法是计算机设计人员规定的，尽管数据的来源和形式有所不同，但输入这台计算机并经它处理的全部数据都必须符合规定。软件设计人员还可以依此来规定各数据类型（如虚数、向量等）和组织复杂的数据结构（如记录、文件等）。

早期的机械式和继电式计算机都用具有 10 个稳定状态的基本元件来表示十进制数据位 0,1,2,…,9，一个数据的各个数据位是按 10 的指数顺序排列的，如 $386.45=3\times10^{2}+8\times10^{1}+6\times10^{0}+4\times10^{-1}+5\times10^{-2}$。但是，要求处理机的基本电子元件具有 10 个稳定状态比较困难，十进制运算器逻辑线路也比较复杂。二进制是计算技术中广泛采用的一种数制，由 18 世纪德国数理哲学大师莱布尼兹发现。二进制运算比较简单，能节省设备，而且二进制与处理机逻辑运算能协调一致，

便于用逻辑代数简化处理机逻辑设计。因此，二进制得到广泛应用。

1．定点表示法

在二进制中，0 和 1 分别由处理机电子元件的两个稳定状态表示，2 为数的基底。

2．字符数据表示法

用二进制位序列组成供输入、处理和输出用的编码称为字符数据。字符数据包括各种运算符号、关系符号、货币符号、字母和数字等。中国通用的是 1980 年颁布的国家标准 GB 1988—80《信息处理交换用的七位编码字符集》，它以 7 个二进制位表示 128 个字符。它包括 32 个控制字符集、94 个图形字符集、一个间隔字符和一个抹掉字符。

1.2.1　二进制、八进制、十六进制

二进制数据是用 0 和 1 两个数码来表示的数，它的基数为 2，进位规则是“逢二进一”，借位规则是“借一当二”。当前的计算机系统使用的基本上是二进制系统，所有输入计算机的信息最终都要转化为二进制。最基本的单位为bit。

编程中，常用的还是十进制。

比如：

```
int a = 100，b = 99;
```

不过，由于数据在计算机中的表示最终以二进制的形式存在，所以有时候使用二进制可以更直观地解决问题。

首先来看一个二进制数：1111，它是多少呢？由于 1111 才 4 位，所以可以直接记住它每一位的权值，并且是从高位往低位记：8、4、2、1。即最高位的权值为 2^3=8，然后依次是 2^2=4，2^1=2，2^0 = 1。则二进制数 1111 对应的十进制数就是 1×8+1×4+1×2+1×1=15。

记住 8421，对于任意一个 4 位的二进制数，都可以很快算出它对应的十进制值。

采用二进制计数制，对于计算机等数字系统来说，运算、存储和传输极为方便。然而，二进制数用来表示一个数据的时候过于烦琐，比如 int 类型占用 4 个字节，32 位。如十进制数 100，用 int 类型的二进制数表达将是：

0000 0000 0000 0000 0110 0100

面对这么长的数进行思考或操作，没有人会喜欢。因此，C 和 C++编程语言没有提供在代码中直接写二进制数的方法。

用十六进制或八进制可以解决这个问题。因为，进制越大，数的表达长度也就越短。不过，为什么偏偏是十六或八进制，而不是其他的诸如九或二十进制呢？这是因为 2、8、16，分别是 2 的 1 次方、3 次方、4 次方，这一点使得三种进制之间可以非常直接地互相转换。八进制或十六进制缩短了二进制数，但保持了二进制数的表达特点。

1．二进制数转换为十进制数

二进制数第 0 位的权值是 2^0，第 1 位的权值是 2^1，依次类推。

设有一个二进制数：0110 0100，转换为十进制数的计算如下。

用竖式计算：

0110 0100　换算成　十进制数
第 0 位　$0 * 2^0$　=　0
第 1 位　$0 * 2^1$　=　0
第 2 位　$1 * 2^2$　=　4
第 3 位　$0 * 2^3$　=　0
第 4 位　$0 * 2^4$　=　0
第 5 位　$1 * 2^5$　= 32
第 6 位　$1 * 2^6$　= 64
第 7 位　$0 * 2^7$　=　0　　+

100

用横式计算：

$$0 * 2^0 + 0 * 2^1 + 1 * 2^2 + 1 * 2^3 + 0 * 2^4 + 1 * 2^5 + 1 * 2^6 + 0 * 2^7 = 100$$

0 乘以多少都是 0，所以可以直接跳过值为 0 的位：

$$1 * 2^2 + 1 * 2^3 + 1 * 2^5 + 1 * 2^6 = 100$$

2．八进制数转换为十进制数

八进制就是逢 8 进 1。
八进制数采用 0～7 这八个数来表达一个数。
八进制数第 0 位的权值为 8^0，第 1 位权值为 8^1，第 2 位权值为 8^2，依次类推。
设有一个八进制数：1507，转换为十进制数的计算如下。
用竖式计算：
第 0 位　$7 * 8^0 = 7$
第 1 位　$0 * 8^1 = 0$
第 2 位　$5 * 8^2 = 320$
第 3 位　$1 * 8^3 = 512$　+

839

同样，也可以用横式直接计算：

$$7 * 8^0 + 0 * 8^1 + 5 * 8^2 + 1 * 8^3 = 839$$

结果是八进制数 1507 转换成十进制数为 839。

在 C 和 C++语言中，如何表达一个八进制数呢？如果这个数是 876，可以断定它不是八进制数，因为八进制数中不可能出 7 以上的阿拉伯数字。但如果这个数是 123、567 或 12345670，那么它是八进制数还是十进制数都有可能。所以 C/C++规定，一个数如果要指明它采用八进制，必须在它前面加上一个 0，例如，123 是十进制数，但 0123 则表示采用八进制。这就是八进制数在 C/C++中的表达方法。

3．十六进制数转换成十进制数

十六进制就是逢 16 进 1，但只有 0～9 这十个数字，所以用 A、B、C、D、E、F 这六个字母来分别表示 10、11、12、13、14、15。字母不区分大小写。

十六进制数的第 0 位的权值为 16^0，第 1 位的权值为 16^1，第 2 位的权值为 16^2，依次

类推。

所以，在第 N（N 从 0 开始）位上，如果是数 X（X 大于等于 0，并且 X 小于等于 15，即 F），表示的大小为 $X*16^N$。

假设有一个十六进制数 2AF5，那么如何换算成十进制数呢？

用竖式计算：

第 0 位：$5 * 16^0 = 5$

第 1 位：$F * 16^1 = 240$

第 2 位：$A * 16^2 = 2560$

第 3 位：$2 * 16^3 = 8192$　+

10997

直接计算就是：

$$5 * 16^0 + F * 16^1 + A * 16^2 + 2 * 16^3 = 10997$$

在上面的计算中，A 表示十进制数 10，而 F 表示十进制数 15。

现在可以看出，所有进制换算成十进制，关键在于各自的权值不同。

如果不使用特殊的书写形式，十六进制数也会和十进制数相混。比如随便一个数 9876，就看不出它是十六进制数还是十进制数。C/C++规定，十六进制数必须以 0x 开头。比如 0x1 表示一个十六进制数，而 1 则表示一个十进制数。另外，如 0xff、0xFF、0X102A 等，其中的 x 不区分大小写（注意：0x 中的 0 是数字 0，而不是字母 O）。

1.2.2　表示数据的字节和位

位（bit）又称“比特”，是存储信息的最小单位，它的值是二进制 1 或 0。计算机中所有的信息都是以“位”（bit）为单位表示、存储及传递的。早期的 DOS 操作系统就是 8 位的，后期的 DOS 操作系统是 16 位的，Win9X 操作系统是基于 DOS 的，所以也是 16 位的，NT 核心的 Windows 是 32 位的，现在也有了 64 位的 XP/Win7 操作系统等，CPU 也有 64 位的，所说的位就是 bit 的意思，即二进制数的长度。

每 8 个位（bit）组成一个字节（byte）。字节是什么概念呢？一个英文字母就占用一个字节，也就是 8 位，一个汉字占用两个字节。一般位简写为小写字母“b”，字节简写为大写字母“B”。

每一千个字节称为 1KB，注意，这里的“千”不是通常意义上的 1000，而是指 1024。即 $1KB=1024B=2^{10}B$。

每一千个 KB 就是 1MB（同样这里的 K 是指 1024），即

$$1MB=1024KB=1024\times1024B=1048576B$$

每一千个 MB 就是 1GB，即 1GB=1024MB。例如，一篇 10 万汉字的小说，如果存到磁盘上，需要占用多少空间呢？

100000 汉字=200000B=(200000B÷1024)KB≈195.3KB≈(195.3KB÷1024)MB≈0.19MB

另外一个例子就是网速，提到网速的时候经常会省略单位，往往只是说 G、M、K，其实 G、M、K 是数量的简略表示法，换算公式为：1G=1024M，1M=1024K，1K=1024，就相当于人们常说的亿、万、千，只是数量的简略表示而已，并不是单位，单位是字节/秒（Bps）。

1.2.3　内存

内存（Memory）也称为内存储器，其作用是暂时存放 CPU 中的运算数据，以及与硬盘等

外部存储器交换的数据。只要计算机在运行中，CPU 就会把需要运算的数据调到内存中进行运算，当运算完成后 CPU 再将结果传送出来，内存的运行也决定了计算机的稳定运行。

1．整数的存储方式

一个十进制整数先转换为二进制形式，如整数 10，以二进制形式表示是 1010，直接把它存放在存储单元中。如果用一个字节来存储，存储单元中的情况如下。

0	0	0	0	1	0	1	0

一个存放整数的存储单元，左面第一位用来表示符号，当最高位为 0 时表示是正数。其他 7 位都用来存放数值，则它的最大值是 01111111，即 2^7-1，它相当于十进制数的 127。如果数值大于 127，一个字节就放不下了。这显然是不满足要求的。有的编译系统以两个字节表示一个整数。这时，它的最大值是 0111111111111111，即 $2^{15}-1$，它相当于十进制数的 32767。实际中使用的整数往往超过 32767，显然两个字节也放不下，因此有的系统以 4 个字节表示一个整数，这时，它的最大值是 31 个二进位的值都是 1，即 $2^{31}-1$，约为 21 亿，一般情况下能满足使用要求了。

2．实数的存储方式

一个实数，如 123.456，就不能采用上面的办法。对于实数，一律采用指数形式存储，如 123.456 可以写成标准化指数形式 0.123456×10^3，它包括前后两部分，前面部分是数值部分，后面部分是指数部分，如下所示。

$$0.123456\times10^3$$

数值部分　指数部分

所谓“标准化指数形式”是指这样的指数：其数值部分是一个小数，小数点前的数字是零，小数点后的第一位数字不是零。一个实数可以有多种指数形式，但只有一种属于标准化指数形式。如 123.456 可以表示为 123456×10^{-3}、12345.6×10^{-2}、1234.56×10^{-1}、123.456×10^{0}、12.3456×10^{1}、1.23456×10^{2}、0.123456×10^{3} 等，在数学上它们是等价的，其中只有 0.123456×10^3 符合上面的条件，是标准化指数形式，而其他的都不是标准化指数形式。在计算机中，一般以 4 个字节存储一个实数。这 4 个字节可分为两部分：一般以 3 个字节存放数值部分，以 1 个字节存放指数部分。

3．字符的存储方式

计算机并不是将字符本身存放到存储单元，而是将字符的代码存储到相应的单元中。附录 A 是字符与代码的对照表，这是国际通用的 ASCII 代码。例如，在附录 A 中可以查到大写字母 A 相应的 ASCII 代码是 65，而 65 的二进制形式是 1000001，所以在存储单元中的信息是 01000001（第一位补 0，凑足 8 位）。

0	1	0	0	0	0	0	1

其他字符类似。以上转换工作由编译系统自动完成，不必用户自己转换。只需要输入字母 A，在计算机的相应存储单元中就会存入 01000001 的信息。

1.3　C++开发工具

什么是编程

1．编译器

编译器就是将高级语言翻译为机器语言（低级语言）的程序。一个现代编译器的主要工作流程为：源代码（source code）→预处理器（preprocessor）→编译器（compiler）→汇编程序（assembler）→目标代码（object code）→链接器（linker）→可执行程序（executables）。C++编译器是一个与标准化 C++高度兼容的编译环境。这点对于编译可移植的代码十分重要。下面介绍三种 C++编译器。

（1）Borland C++

它是 Borland C++ Builder 和 Borland C++ Builder X 这两种开发环境的后台编译器。正如 Delphi7 到 Delphi8 的转变，是革命性的两代。Borland C++由老牌开发工具厂商 Borland 倾力打造。该公司的编译器素以速度快、空间效率高著称，Borland C++ 系列编译器秉承了这个传统，属于非常优质的编译器。标准化方面早在 5.5 版本的编译器中对标准化 C++的兼容就达到了 92.73%。目前最新版本是 Borland C++ Builder X 中的 6.0 版本，官方称 100%符合 ANSI/ISO 的 C++标准及 C99 标准。

（2）Visual C++

Visual C++是 Microsoft 公司提供的在 Windows 环境下进行应用程序开发的 C/C++编译器。相比其他的编程工具而言，Visual C++在提供可视化的编程方法的同时，也适用于编写直接对系统进行底层操作的程序。随 Visual C++一起提供的 Microsoft 基础类库（Microsoft Foundation Class Library，简写为 MFC）对 Windows 9x/NT 所用的 Win32 应用程序接口（Win32 pplication Programming Interface）进行了十分彻底的封装，这使得 Windows 9x/NT 应用程序的开发可以使用完全的面向对象的方法来进行，从而能够大量地节省应用程序的开发周期，降低开发成本，也使得 Windows 程序员从大量的复杂劳动中解脱出来，而且，并没有因为获得这种方便而牺牲应用程序的性能。

（3）Borland C++ Builder X

正如前文所述，虽然版本号上和第一个 IDE（Integrated Development Environment，集成开发环境）非常相像，但是其实它们是完全不同的两个集成开发环境。C++ Builder 更多地是一个和 Delphi 同步的 C++版本的开发环境，C++ Builder X 则是完全从 C++的角度思考得出的一个功能丰富的 IDE。其最大的特点是跨平台、跨编译器、多种 Framework 的集成，并且有一个 wxWindows 为基础的 GUI 设计器。尤其是采用了纯 C++来重写整个 Framework，改进了工具的性能。

2．开发环境

开发环境对于程序员的作用非常重要，特别是在 IDE 如此丰富的情况下。下面是一些常见的 C++开发环境，可以用作日常开发使用。

（1）Visual Studio 6.0

这个虽然是 Microsoft 公司的老版本的开发环境，但是鉴于其后继版本 Visual Studio.NET 的庞大身躯，以及初学者并不那么高的功能要求，所以推荐这个开发环境给 C++的初学者，供

其学习 C++的最基本的部分，比如 C 的那部分子集。在日常的开发中，仍然有很多公司使用这个经典稳定的环境。

（2）Visual Studio.NET

作为 Microsoft 公司官方正式发布的开发环境，结合其最新的 C++编译器，对于机器配置比较好的开发人员来说，使用这个开发环境将能满足其大部分的要求。包括多种语言编译器：C++、C#、Visual Basic、F#等。

3．Visual C++ 6.0 开发环境的使用方法

考虑到目前大多数初学者使用的都是 PC 和 Windows 操作系统，以 Visual C++作为推荐的 C++编译器，但本书的绝大多数程序都可以在任何支持标准 C++的编译器中编译通过。

Visual C++软件包包含了许多单独的组件，如编辑器、编译器、链接器、生成实用程序、调试器，以及各种各样为开发 Microsoft Windows 下的 C/C++程序而设计的工具。一般情况下，Visual C++既指整个产品，又指它的开发环境。

在编程之前，必须先了解工程 Project 的概念。工程又称为项目，它具有两种含义，一种是指最终生成的应用程序；另一种则是为了创建这个应用程序所需的全部文件的集合，包括各种源程序、资源文件和文档等。

具体使用步骤如下。

（1）启动并进入 Visual C++ 6.0 的集成开发环境

Visual C++ 6.0 的集成开发环境窗口如图 1-1 所示。窗口大体上可分为四部分。上部：菜单和工具条；中左：工作区（workspace）视图显示窗口，这里将显示处理过程中与项目相关的各种文件种类等信息；中右：文档内容区，是显示和编辑程序文件的操作区；下部：输出（Output）窗口区，程序调试过程中，进行编译、链接、运行时输出的相关信息将在此处显示。

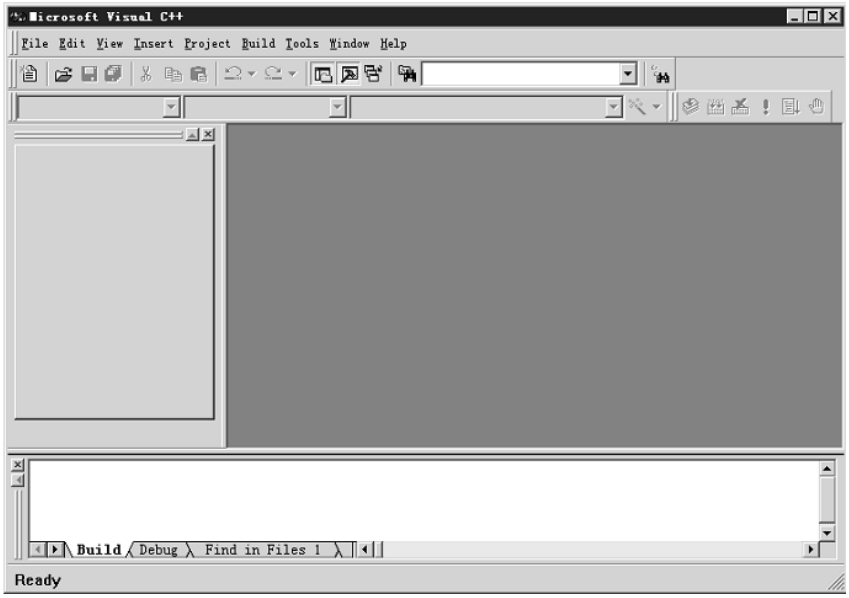

图 1-1　Visual C++ 6.0 的集成开发环境窗口

（2）创建工程并输入源程序代码

为了把程序代码输入计算机，需要使用 Visual C++ 6.0 的编辑器来完成。首先要创建工程

以及工程工作区，而后才能输入具体程序完成所谓的“编辑”工作。

执行菜单命令“File→New”，会出现一个选择界面，在属性页中选择“Projects”标签后，会看到近 20 种的工程类型，只需选择其中最简单的一种：“Win32 Console Application”，而后往右上处的“Location”文本框和“Project name”文本框中填入工程相关信息、所存放的磁盘位置（目录或文件夹位置）以及工程的名字，此时的界面信息如图 1-2 所示。

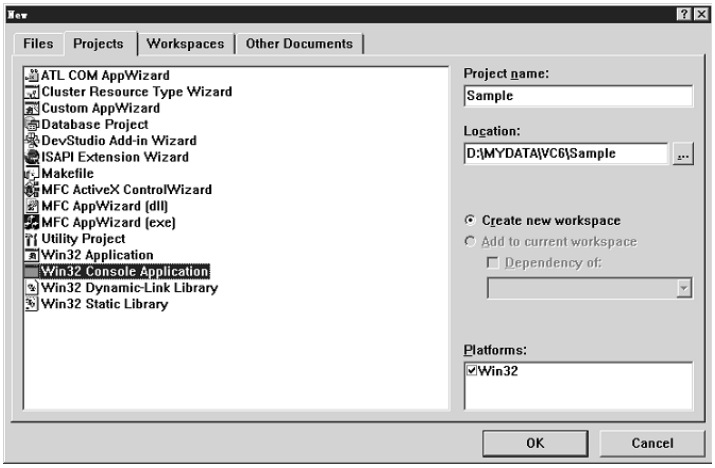

图 1-2　新建一个名为 Sample 的工程（同时自动创建一工作区）

在图 1-2 中，“Location”文本框中可填入如“D:\myData\VC++ 6.0”，这是假设准备在 D 磁盘的\myData\VC++ 6.0 文件夹即子目录下存放与工程工作区相关的所有文件及其相关信息；当然也可通过单击其右部的“…”按钮去选择并指定这一文件夹即子目录位置。“Project name”文本框中填入如“Sample”的工程名（注意，名字根据工程性质确定，此时 Visual C++ 6.0 会自动在其下的“Location”文本框中用该工程名“Sample”建立一个同名子目录，随后的工程文件以及其他相关文件都将存放在这个目录下）。

单击“OK”按钮进入下一个选择界面。 这个界面主要是询问用户想要构成一个什么类型的工程，其界面如图 1-3 所示。

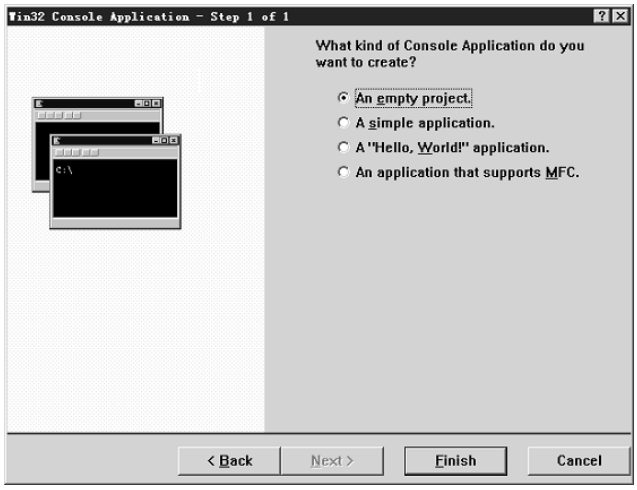

图 1-3　选择创建一个什么样的工程

若选择“An empty project”项将生成一个空的工程，工程内不包括任何东西。若选择“A simple application”项将生成包含一个空的 main 函数和一个空的头文件的工程。选择“A"Hello World!"application”项与选择“A simple application”项没有什么本质的区别，只是前者包含有显示出“Hello World!”字符串的输出语句。若选择“An application that supports MFC”项的话，可以利用 Visual C++ 6.0 所提供的类库来进行编程。

为了更清楚地看到编程的各个环节，选择“An empty project”项，从一个空的工程来开始工作。单击“Finish”按钮，这时 Visual C++ 6.0 会生成一个报告，报告的内容是刚才所有选择项的总结，并且询问是否接受这些设置。如果接受单击“OK”按钮，否则单击“Cancel”按钮。单击“OK”按钮进入真正的编程环境，界面情况如图 1-4 所示。

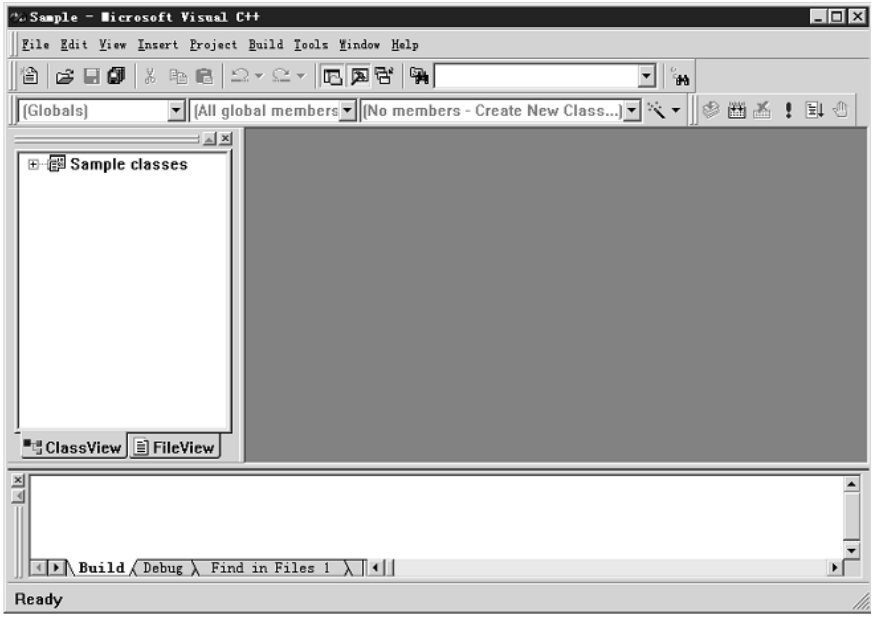

图 1-4　完成创建工程 Sample 的 Visual C++ 6.0 集成开发环境窗口

注意屏幕中的 Workspace 窗口，该窗口中有两个标签，一个是“ClassView”，一个是“FileView”。“ClassView”中列出的是这个工程中所包含的所有类的有关信息，当然程序将不涉及到类。单击“FileView”标签后，将看到这个工程所包含的所有文件信息。单击“+”图标打开所有的层次，会发现有三个逻辑文件夹：Source Files 文件夹中包含了工程中所有的源文件；Header Files 文件夹中包含了工程中所有的头文件；Resource Files 文件夹中包含了工程中所有的资源文件。所谓资源就是工程中所用到的位图、加速键等信息，在编程中不会牵扯到这一部分内容。现在“FileView”中也不包含任何东西。

逻辑文件夹是逻辑上的，它们只是在工程的配置文件中定义的，在磁盘上并没有物理地存在这三个文件夹。也可以删除自己不使用的逻辑文件夹；或者根据项目的需要，创建新的逻辑文件夹，来组织工程文件。这三个逻辑文件夹是 VC 预先定义的，就编写简单的单一源文件的 C++程序而言，只需要使用 Source Files 一个文件夹就够了。

（3）在工程中新建 C++源程序文件并输入源程序代码

执行菜单命令“Project→Add To Project→new”，在出现的对话框的“Files”标签（选项卡）中，选择“C++ Source File”项，在右中处的“File”文本框中为将要生成的文件取一个名字，取名为 Hello（其他遵照系统隐含设置，此时系统将使用 Hello.cpp 的文件来保存所输入的源程

序），此时的界面情况如图 1-5 所示。

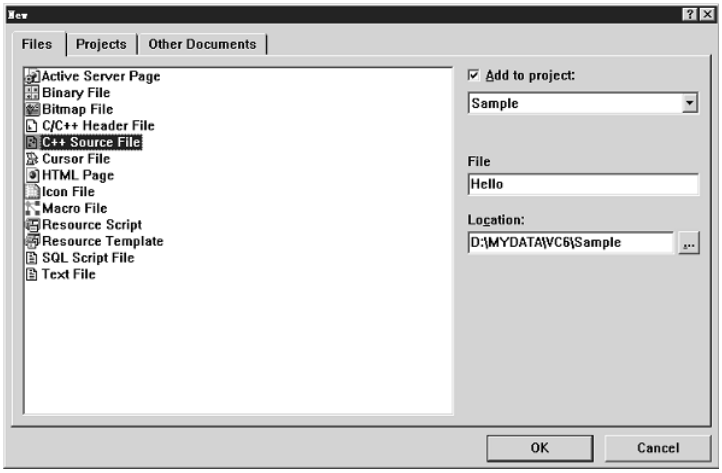

图 1-5　在工程 Sample 中新建一名为 Hello.cpp 的 C 源程序文件

单击“OK”按钮，进入输入源程序的编辑窗口（注意所出现的呈现“闪烁”状态的输入位置光标），此时只需通过键盘输入所需要的源程序代码：

```
#include <iostream>
using namespace std;
int main()
{
    cout<<"Hello World! ">>endl;
    return 0;
}
```

可通过 Workspace 窗口中的“FileView”标签，看到 Source Files 文件夹下文件 Hello.cpp 已经被加了进去，此时的界面情况如图 1-6 所示。

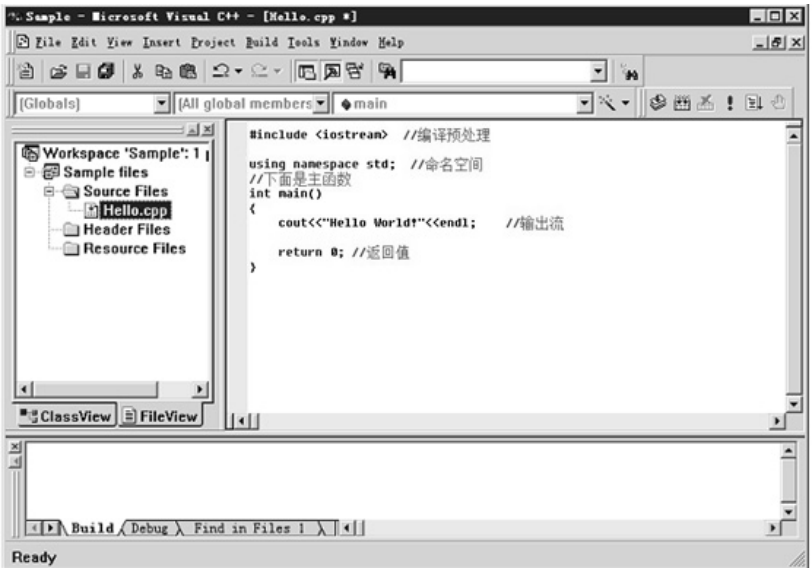

图 1-6　在 Hello.cpp 中输入 C 源程序代码

（4）编译、链接及运行程序

程序编辑工作完成，保存之后，就可以进行编译、链接与运行了。所有的命令项都处在菜单“Build”之中。注意，在对程序进行编译、链接和运行前，最好首先选择执行菜单第一项“Compile”，此时将对程序进行编译。若编译中发现错误（error）或警告（warning），将在 Output 窗口中显示出它们所在的行以及具体的出错或警告信息，可以通过这些信息的提示来纠正程序中的错误或警告（注意，错误是必须纠正的，否则无法进行下一步的链接；而警告则不然，它并不影响下一步操作，当然最好还是能把所有的警告也“消灭”掉）。当没有错误与警告出现时，Output 窗口所显示的最后一行应该是：“Hello.obj-0 error(s), 0warning(s)”。

编译通过后，可以选择菜单的第二项“Build”来进行链接生成可执行程序。在链接中出现的错误也将显示到 Output 窗口中。链接成功后，Output 窗口所显示的最后一行应该是：“Sample.exe-0 error(s), 0 warning(s)”。

最后就可以运行（执行）所编制的程序了，选择“Execute”项（该选项前有一个深色的感叹号标志“!”，实际上也可通过单击窗口上部工具栏中的深色感叹号标志“!”来启动执行该选项），Visual C++ 6.0 将运行已经编好的程序，执行后将出现一个结果界面（所谓的类似于 DOS 窗口的界面），如图 1-7 所示，其中的“Press any key to continue”是由系统产生的，使得用户可以浏览输出结果，直到按下了任一个键盘按键时为止（那时又将返回到集成界面的编辑窗口处）。

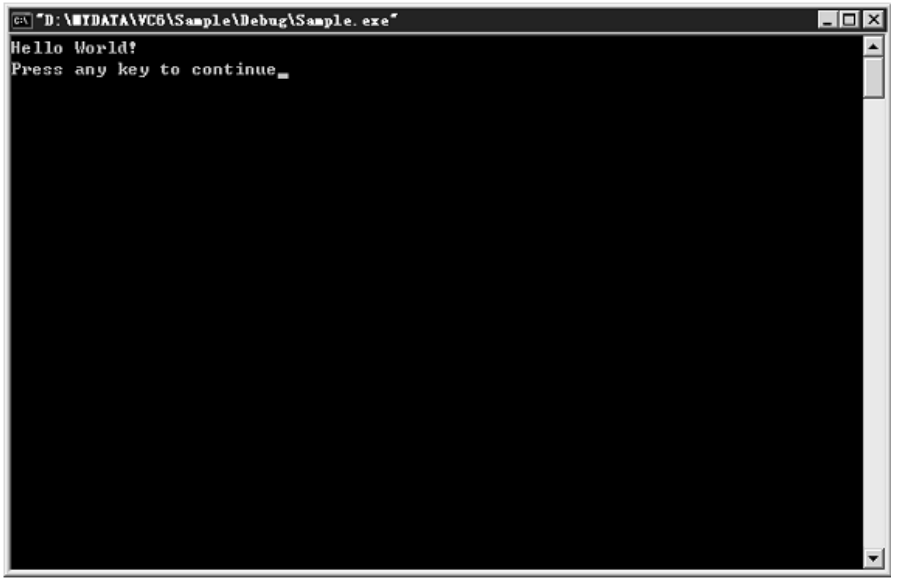

图 1-7　程序 Hello.cpp 的运行结果界面

至此已经生成并运行（执行）了一个完整的程序，完成了一个“回合”的编程任务。此时应执行菜单命令“File→Close Workspace”，待系统询问是否关闭所有的相关窗口时，回答“是”，则结束一个程序从输入到执行的全过程，回到刚刚启动 Visual C++ 6.0 的初始画面。

C++程序结构