

第5章 C 程序设计基础

引言

计算机程序是一系列的指令,它能使计算机执行特定的任务。编程语言,如 Visual C++ 用于将人们能懂得的指令翻译成计算机可以理解和执行的步骤。深入计算机的最底层,作为计算机“心脏”的微处理器,只能理解数字指令(机器指令)。这种处理器所能理解的指令是极简单的命令,这些命令大多数只能在存储地址之间移动数据。这些处理器懂得的命令称为机器语言,即 PC 可使用的最基本的语言。

机器语言称为低级语言,因为它是处理器能够懂得的最低级的方式。用机器语言编写程序是一项极其烦琐的任务。幸运的是,用户不必用它来编写计算机程序。有更高级的编程语言已经开发出来,使一般用户能够编写程序。这些高级编程语言允许程序员以类似英文的方式编写指令,然后将指令转化成处理器能懂的含有机语言指令的程序。

教学目的

- 掌握 C 程序的基本结构。
- 掌握变量、表达式及运算符的基本用法。
- 熟悉编写 C 程序的方法。
- 掌握 C 的流程控制语句。

5.1 程序设计基础

5.1.1 什么是程序设计

计算机之所以比电视机、DVD 机、计算器等其他电子设备功能更灵活,是因为计算机软件的可编程”,也就是说,同样的硬件配置,加载不同的软件就可以完成不同的工作。

当用户使用计算机来完成某项工作时,会面临两种情况:一种是可以借助现成的应用软件完成,如文字处理可使用 Word,表格处理可使用 Excel,科学计算可选择 Matlab,

绘制图形可使用 Photoshop 等;另一种情况是,没有完全合适的软件可供使用,这时就需要使用计算机语言来编制完成某个特定功能的程序,这就是程序设计(programming)。

考虑这样一个任务:统计大学中一个班学生的考试成绩,并选出优秀学生。

可以看出,这是一个很简单的任务。如果由人去做,是一件比较轻松的事,只需要几张纸和一支笔,一个小时就完成了。对这样简单的任务,可以不需要计算机去做。但如果将任务修改为:要求对一个大学中的 4 万名学生的英语成绩进行管理,分别统计出 100 至 60 各个分数段的人数,并对过去 5 年中的数据进行对比分析。这个任务由人工完成会比较麻烦,完全值得设计计算机程序。

功能完善的商业程序一般都是比较小的,一个字处理软件可能包含 75 万行左右的代码,而按照美国国防部的标准,少于 10 万行代码称为小程序,超过 100 万行才是大程序。为便于理解,下面还是以微小的程序作为例子来介绍程序设计的概念。

计算机程序设计是用计算机语言编写一些代码(指令)来驱动计算机完成特定的功能,是问题求解过程的关键步骤之一。我们已经知道,对复杂的系统性问题的求解,需要经过需求分析、软件设计、程序编制和调试、系统测试、文档编写等一系列活动,甚至还包括硬件系统配置、人员配置、开发方法及开发工具选择等相关的各种任务管理。而对于小型问题(或称算法类问题),这个过程可以简化为问题描述、算法设计、代码编制以及调试运行。

5.1.2 程序设计语言

在过去的几十年里,人们根据描述问题的需要而设计了数千种专用和通用的计算机语言,有的语言是为了编写系统软件而重在提高效率(如 C 语言);有些是为了提高程序设计速度,面向商业应用(如 COBOL 和 dBASE);还有些语言是为了用于教学(BASIC 和 Pascal 都属于此类)。这些语言中只有少部分得到了比较广泛的应用。

1. 计算机语言的分类

对程序设计语言的分类可以从不同的角度进行,如面向机器的程序设计语言、面向对象的程序设计语言、面向过程的程序设计语言等。其中,最常见的分类方法是根据程序设计语言与计算机硬件的联系程度,可以将其分为机器语言、汇编语言和高级语言 3 种类型。前两种类型的语言紧密依赖于计算机硬件,有时也统称为低级语言;而高级语言与计算机硬件关系较小。可以说,程序设计语言的演变经历了由低级向高级的发展过程。

1) 机器语言

机器语言(machine language)是直接机器指令的集合作为程序设计手段的语言,而机器指令是以计算机所能理解和执行的以 0 和 1 组成的二进制编码表示的命令,它是所有语言中唯一能够被计算机直接理解和执行的指令。如第 2 章中所述,机器指令由操作码和操作数组成,其具体的表现形式和功能与计算机系统的结构相关联。

机器语言是面向机器的语言,其优点是计算机能够直接识别,执行效率高;缺点是因与具体机器相关,可移植性很差。且由于是由 0 和 1 这样的编码表示,在记忆、书写、编程、可读性等方面都比较困难。表 5-1 是分别用二进制和十六进制数表示的机器指令代

码,功能是求两数之和。由此可以看出,机器指令使用起来是很困难的。

表 5-1 求两数之和的机器语言指令

二进制表示	十六进制表示
1010 0000 0000 0000 0000 0000	A00000
0000 0011 0000 0110 1100 1000 0000 0000	0306C800
1010 0011 1100 1100 0000 0000	A3CC00

2) 汇编语言

为了克服机器语言的缺点,人们采用了助记符与符号地址来代替机器指令中的操作码与操作数。如用 ADD 表示加法操作,用 SUB 表示减法操作,且操作数可用二进制、八进制、十进制和十六进制数表示。这种表示计算机指令的语言称为汇编语言(assembly language)。汇编语言也是一种面向机器的语言,但计算机不能直接执行汇编语言程序。用它编写的程序必须经过汇编程序翻译成机器指令后才能在计算机上执行。目前,由于它比机器语言可理解性好,比其他语言执行效率高,许多系统软件的核心部分仍采用汇编语言编制。

表 5-1 中的机器语言指令代码若用汇编语言编写,可表示为如下 3 行指令:

```
MOV  AX,  DATA1
ADD  AX,  DATA2
MOV  SUM, AX
```

3) 高级语言

高级语言是更接近自然语言和数学语言的程序设计语言,是面向应用的计算机语言,与具体的计算机硬件平台无关。它的主要优点是符合人类叙述问题的习惯,而且简单易学。目前的大部分程序设计语言都属高级语言,其中使用较多的有 BASIC (Visual Basic)、Pascal(Delphi)、FORTRAN、COBOL、C、C++、Java 等。

用 Visual Basic 语言完成表 5-1 的程序功能只需要下面一行语句:

```
SUM= DATA1+ DATA2;
```

可以看出,这很接近于自然语言,便于人理解。

目前高级语言正朝着非过程化发展,即只需告诉计算机“做什么”,而“怎样做”则由计算机自动处理。高级语言的发展将以更加方便用户使用为宗旨。

2. 程序语言的语法和语义

计算机语言有多种,不同的语言在语法和语义上都存在一定的差异。通俗地讲,语义是程序语言所表示功能的描述。比如,给一个数赋值,比较两个数大小,在屏幕上显示文字等。

不同的计算机语言可能都具有某些相同的功能定义,但在表现形式上却有不同,也就是语法不同。不同计算机语言的区别主要表现在语法(词法)上,即对同一种功能(相同语义),不同的语言可能有不同的表现形式。比如给一个变量赋值,下面的两种语言就有两种不同的表示方法:

```
sum:=eng+math      (Pascal 语言的赋值语句)
sum=eng+math;       (C 语言的赋值语句)
```

又如：要表示“如果数学成绩(math)和英语成绩(eng)都大于 80 就显示姓名(name)和分数合计(sum)”。C 语言的语句是

```
if(math>80 && eng >80)
    printf("第一名: %s %d",name,sum);
```

而如果用 Basic 语言表示,则是:

```
IF math>80 AND eng >80 THEN
    PRINT name1,sum1
END IF
```

可以看出,不同的语言用不同的符号表达了完全相同的含义。但很多时候,语言的差异还体现在语义上,即功能的描述上。比如在 C 和 Pascal 中的指针,在很多语言中就没有对等的体现。

3. 程序执行的起始点

程序都会从起始点开始执行,但不同的语言对起始点的处理方法有所不同。BASIC 语言中,程序从第一条语句开始执行,不论它是什么(这应当是最直观、最容易理解的方式了);在 C 程序和 Visual Basic 控制台程序中,程序会从 main 函数的第一条语句开始执行,而不论 main 函数处于程序的什么位置(如果没有 main 函数,则无法运行)。

以下是一个 C 语言编写的在屏幕上显示“Hello World!”的程序段。

```
int main()
{
    printf("Hello World");    //程序从 main 函数的第一条语句开始执行
    return 0;
}
```

程序从起始点开始,按照程序员书写的顺序一条条执行指令。第一条语句先执行,接下来是第二条指令……一直到程序结尾。如果遇到一个子程序,则中断当前程序而转去执行子程序,执行完返回刚才的断点继续执行。

除了顺序执行外,程序执行还有分支、循环等多种控制。程序的控制结构将在 5.7 节中讨论。

5.1.3 程序的编译

在计算机语言中,用除机器语言之外的其他语言书写的程序都必须经过翻译,变成机器指令,才能在计算机上执行。因此,各种用于计算机程序设计的语言都必须配备相应的“翻译程序”。

编译是指将用高级语言编写好的程序(又称源程序、源代码),经编译程序翻译,形成

可由计算机执行的机器指令程序(称为目标程序)的过程。如果使用编译型语言,必须把程序编译成可执行代码。因此编制程序需要三步:写程序、编译程序和运行程序。一旦发现程序有错,哪怕只是一个错误,也必须修改后再重新编译,然后才能运行。幸运的是,只要编译成功一次,其目标代码便可以反复运行,并且基本上不需要编译程序的支持就可以运行。

5.1.4 C 程序基本结构

首先来看一个简单例子,以便让读者对 C 语言编写的程序有一个初步的认识。

【例 5-1】 第一个 C 程序,在计算机屏幕上显示“Hello World!”。

程序:

```
//屏幕上显示:Hello World!
#include <stdio.h>           //包含基本输入输出库文件
int main()                  //主函数名
{
    printf("Hello World!\n"); //屏幕显示语句
    return 0;                //表示程序顺利结束
}
```

输出:

```
Hello World!
```

分析:该程序非常简单,仅由一个主函数构成,在主函数中也只有两条语句。

程序的第 1 行是注释。注释以“//”开头,直到该行的末尾,用于说明或解释程序段的功能、变量的作用以及程序员认为应该向程序阅读者说明的其他任何内容。可以看到,在该程序中还有一些注释。在将 C 程序编译成目标代码时所有的注释行都会被忽略,因此即使使用了很多注释也不会影响目标码的效率。恰当地应用注释可以使程序清晰易懂、易于调试,便于程序员之间的交流与协作,所以,在编写程序时,精心撰写注释是一个良好的编程习惯。

第 2 行是编译预处理,把 `stdio.h` 库文件插入到程序中相应的位置,用来支持函数 `printf`。

从第 3 行到最后一行是主函数。主函数是该程序的主体部分,由其说明部分

```
int main()
```

和用一对花括号 `{}` 括起来的函数体构成。

通常,一个 C 程序包含一个或多个函数,但其中有且只有一个 `main` 函数,C 程序的执行就是从 `main` 函数开始的。`main` 函数左边的关键字 `int` 表示 `main` 函数返回一个整数值,这是和程序的倒数第 2 行的 `return 0` 对应的,有关函数的进一步内容将在第 6 章介绍,这里只要记住每个程序中都要包含这样的语句就行了。

在函数体内,有两条以分号结束的语句:第 1 条是输出语句,用于将字符串显示在计

计算机屏幕上;第2条是 return 语句,它放在函数的末尾,数值 0 表明程序运行成功。

5.2 使用 Eclipse 和 Visual Studio 编译 C 程序

C 语言的编译器有很多,常见的有 GNU Compiler Collection(或称 GCC)、Microsoft 的 Visual Studio C++ 以及 Turbo C++ 等。本节介绍两种编程环境的使用,其中 Eclipse 仅仅只是一个编辑器,后台需要使用其他的编译器来一起工作。本书选择的是 Eclipse+GCC。

5.2.1 使用 Eclipse 编译 C 程序

在使用 Eclipse 之前,要进行以下的安装设置工作。

(1) J2SE Development Kit(JDK)安装。Eclipse 需要 Java 环境的支持。因此,在安装 Eclipse 之前,需要确保计算机上安装了 JDK,它也是整个 Java 的核心,包括 Java 运行环境(Java Runtime Environment)、Java 工具和 Java 基础的类库。请在 Java 网站(http://www.java.com/zh_CN/)上下载最新版的 Java 并正确安装。

(2) MinGW 安装。MinGW (Minimalist GNU for Windows)是一个自由软件,可以将 C/C++ 撰写的程序编译为 Windows 环境下的可执行文件。它所使用的编译器(如 C 语言的 gcc 或 C++ 语言的 g++ 等)是由 GCC 移植而来。包括了 C、C++、Objective-C、FORTRAN、Java、Ada 等语言的编译器及相关的函式库等必要的文件。GCC 被誉为世界上最重要的软件之一,它由理察·马修·斯托曼(Richard Matthew Stallman,RMS)在 1987 年开始建立,以作为 GNU(GNU's Not UNIX)自由软件计划的编译器。根据 GNU 自由软件的精神,任何人都可以免费取得 GCC 与 MinGW,并且在符合 GNU 通用公共许可证(GNU General Public License,GPL)的情况下,自由地使用、复制、修改和分发 GCC 及 MinGW。MinGW 又称 MinGW32,在不需要第三方(third-party)动态链接函数库(Dynamic Link Library,DLL)支持的情况下,它可以将 C、C++ 等程序编译为可以在 Windows Win32 平台上执行的程序。可以从(<http://sourceforge.net/projects/mingw/files/>)下载最新版的 MinGW 并安装。对于本书,只需要安装和 C 语言有关的编译器和库即可。

(3) Eclipse 安装。在 Eclipse 的官方网站(<https://www.eclipse.org/downloads/>)上下载 Eclipse IDE for C/C++ Developers,无须安装,解压缩后直接运行即可。

(4) 设置环境变量。在系统变量中的 Path 添加“MinGW 安装目录\bin;”(例如 C:\MinGW\bin;)。

【例 5-2】 从键盘输入两个整数,计算两个整数的和并输出。

(1) 启动 Eclipse,如图 5-1 所示。

图 5-1 为 C/C++ Eclipse 启动后的界面,如果是第一次启动,需要先选择一个工作文件夹的存放位置。

(2) 选择菜单 File→New→C Project,弹出如图 5-2 所示的对话框,填好项目名称,选择 Empty Project 和 MinGWGCC 后单击 Finish 按钮。

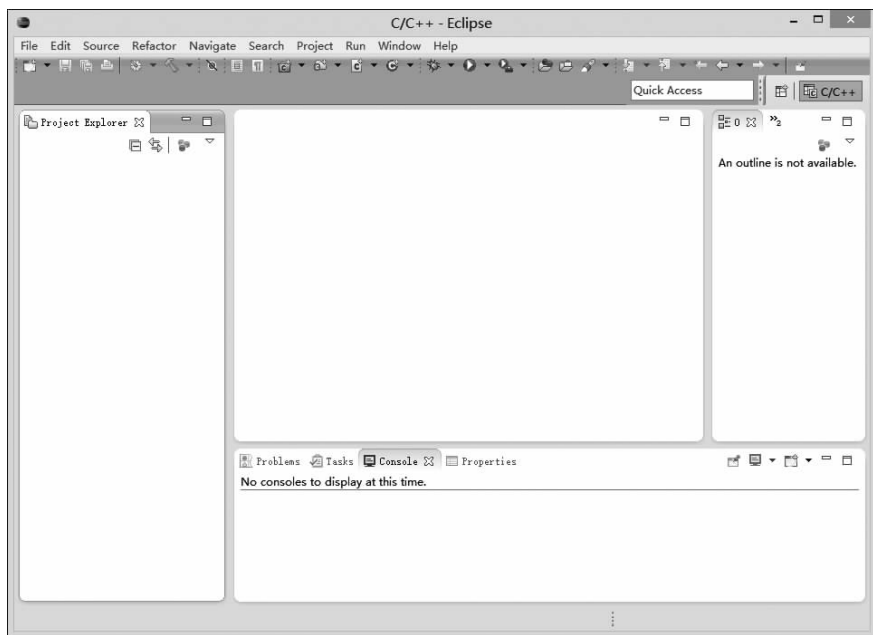


图 5-1 C/C++ Eclipse 启动后的界面

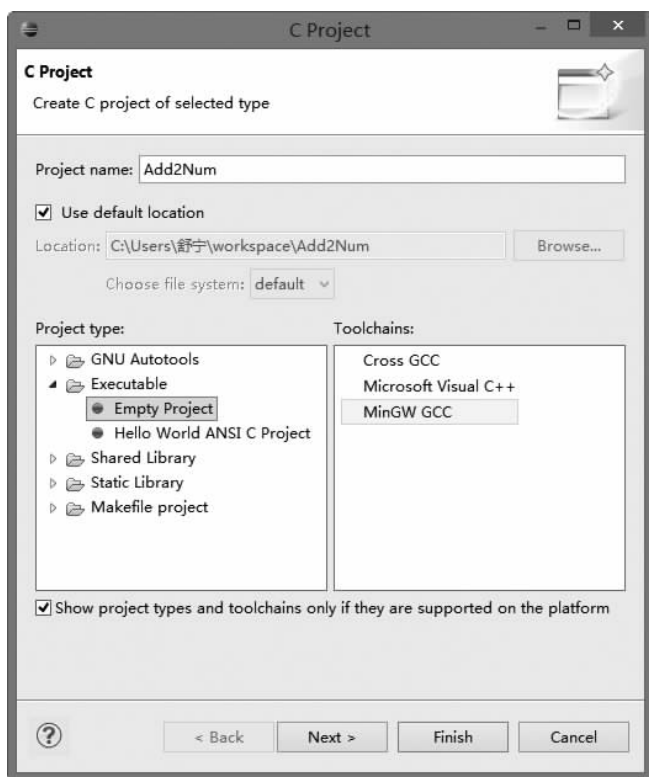


图 5-2 C Project 对话框

图 5-2 Toolchains 中出现的选项和具体计算机安装了哪些 C/C++ 编译器有关。

(3) 添加一个源代码文件。在 Project Explorer 中的 Add2Num 项目上右击, 选择 New→Source File, 如图 5-3 所示。在随后弹出的对话框中给源代码文件起一个文件名, 注意需要给出文件的后缀名.c。单击 Finish 按钮, 如图 5-4 所示。

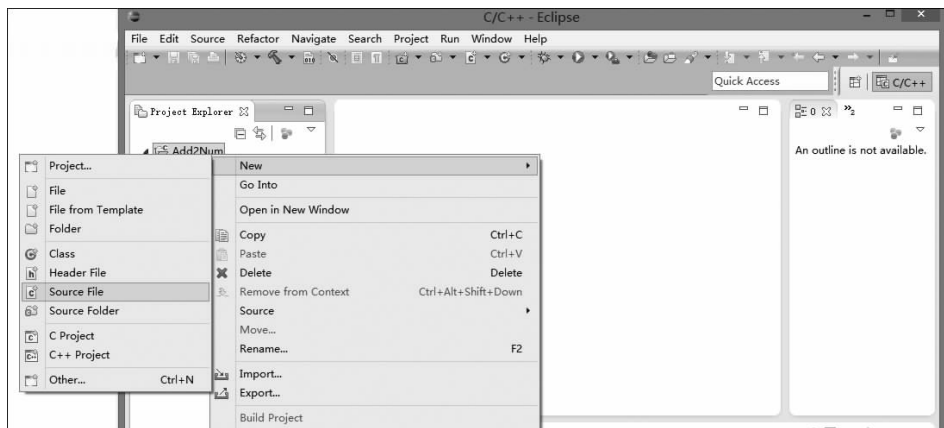


图 5-3 添加一个新的源代码文件

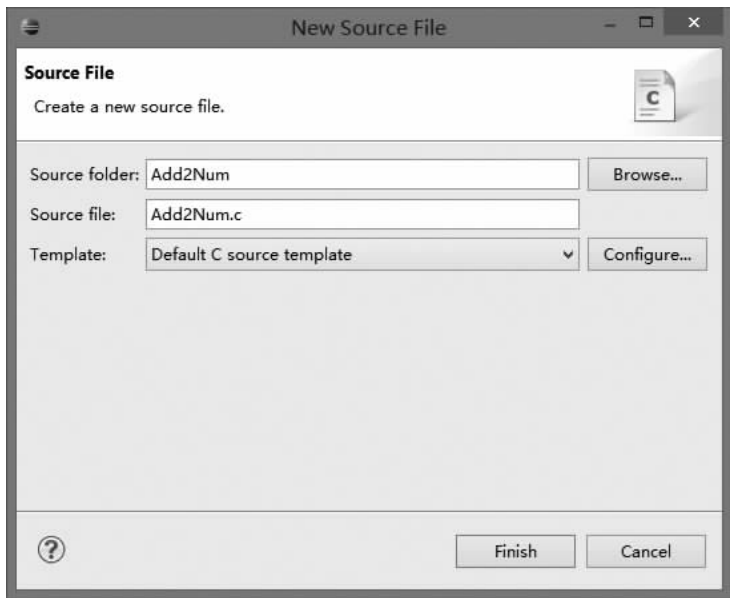


图 5-4 为新的源代码文件命名

(4) 输入代码, 如图 5-5 所示。输入完成后单击工具栏上的保存按钮, 或者选择菜单 File→Save。

(5) 选择菜单 Project→Build All, 将源代码文件编译为可执行文件。然后选择菜单 Run→Run 运行程序, 并从 Console 窗口输入两个数, 验证程序, 如图 5-6 所示。

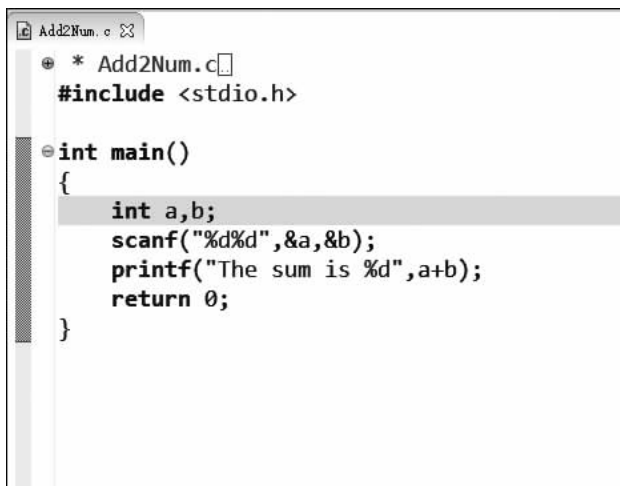


图 5-5 输入程序的源代码

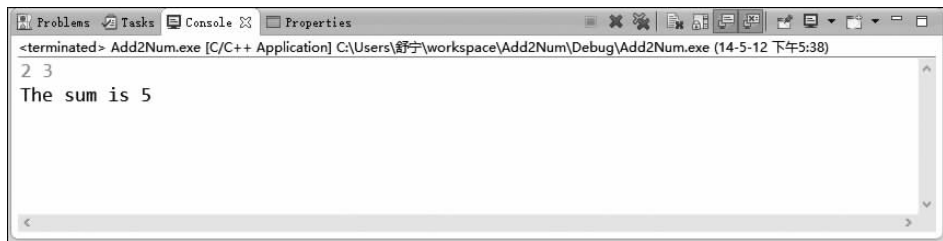


图 5-6 运行并验证程序的正确性

5.2.2 使用 Visual Studio 编译 C 程序

Visual Studio 是微软公司推出的商业版的编译器，支持多种语言的编译。无论安装 Visual Studio 6.0 还是 Visual Studio 2013，也无论是安装免费的速成版，还是付费的团队版，都可以完成本书的学习。具体的下载安装指导参看微软公司的网站 (<http://www.visualstudio.com/>)。下面以同样的例子使用 Visual Studio 2013 再做一遍。

【例 5-3】 从键盘输入两个整数，计算两个整数的和并输出。

(1) 启动 Visual Studio，选择菜单“文件”→“新建”→“项目”。在弹出的对话框中的左列，依次展开并选择“模板”→Visual C++→Win32，在右侧选择“Win32 控制台应用程序”。在项目名称中填入 Add2Num，然后单击“确定”按钮，如图 5-7 所示。

(2) 在弹出的应用程序向导中，在左侧选择“应用程序设置”，然后在右侧勾选“空项目”复选框，最后单击“完成”按钮，如图 5-8 所示。

(3) 在解决方案资源管理器上展开 Add2Num 项目，在源文件上右击，在快捷菜单中



图 5-7 新建一个项目



图 5-8 应用程序向导的设置

选择“添加”→“新建项”，如图 5-9 所示。

(4) 在弹出的对话框中，在“名称”栏填入 Add2Num.c。注意默认的是 C++ 文件，后缀名是 .cpp，这里要写入正确的 C 程序的后缀 .c，如图 5-10 所示。

(5) 在编辑窗口输入代码，如图 5-11 所示。

(6) 选择菜单“调试”→“开始执行(不调试)”或者按快捷键 Ctrl+F5，运行程序。