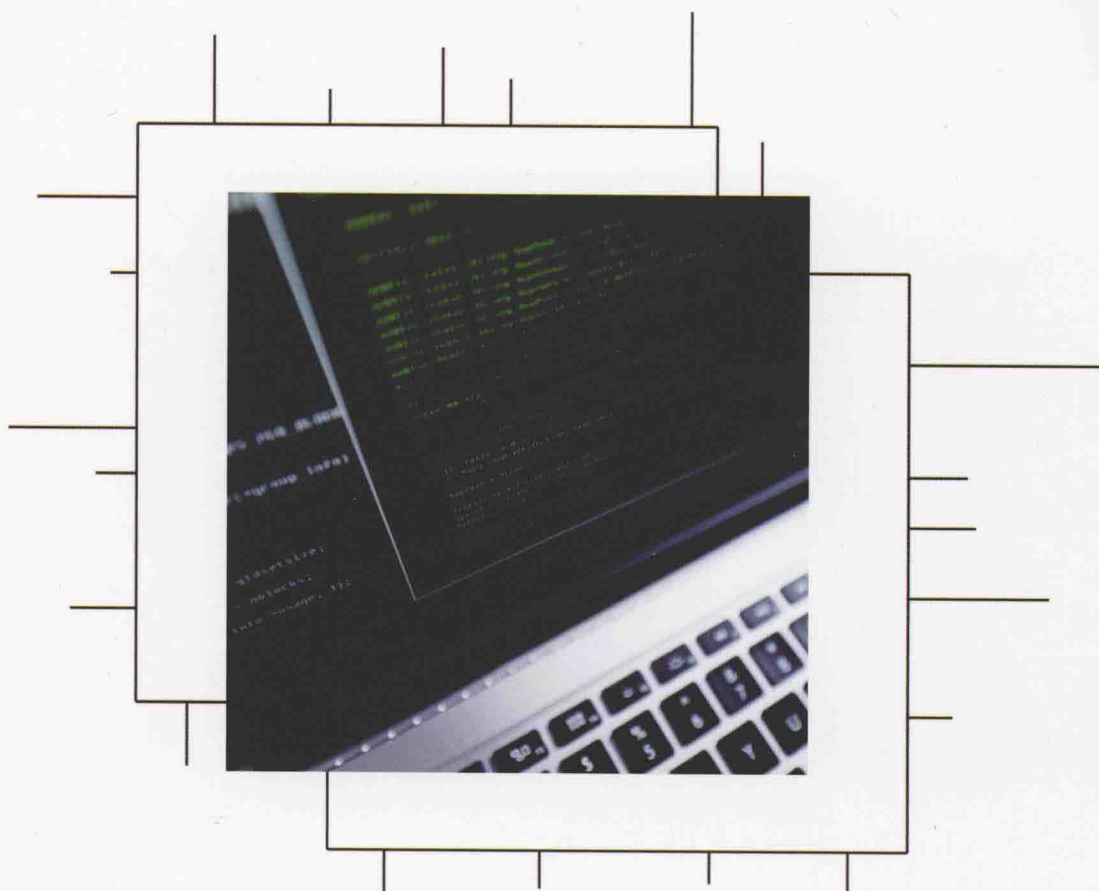




高等院校计算机类规划教材



Visual C++ 程序设计

实验教程与学习指导

主 编 赵 敏
副主编 方 芳 高建波



北京邮电大学出版社
www.buptpress.com



高等院校计算机类规划教材

Visual C++ 程序设计实验教程 与学习指导

主 编 赵 敏
副主编 方 芳 高建波



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书分为两篇:面向对象编程实验和面向对象编程习题。第1篇包括第1~10章,分别为 Visual C++ 2010 简介、数据类型和表达式、流程控制语句、函数、面向对象编程基础、面向对象编程进阶、MFC 编程、Windows 窗体应用程序开发、数据库应用编程、网络编程。第2篇包括第11~15章,分别为数据类型和表达式习题、流程控制语句习题、函数习题、面向对象编程基础习题、面向对象编程进阶习题。

本书面向具备一定 Visual C++ 编程基础、对面向对象编程感兴趣的读者。

图书在版编目(CIP)数据

Visual C++ 程序设计实验教程与学习指导 / 赵敏主编. -- 北京:北京邮电大学出版社, 2022. 4
ISBN 978-7-5635-6613-6

I. ①V… II. ①赵… III. ①C 语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2022)第 041896 号

策划编辑:刘纳新 姚 顺 责任编辑:刘春棠 封面设计:七星博纳

出版发行:北京邮电大学出版社

社 址:北京市海淀区西土城路 10 号

邮政编码:100876

发行部:电话:010-62282185 传真:010-62283578

E-mail: publish@bupt.edu.cn

经 销:各地新华书店

印 刷:保定市中国画美凯印刷有限公司

开 本:787 mm×1 092 mm 1/16

印 张:11.5

字 数:283 千字

版 次:2022 年 4 月第 1 版

印 次:2022 年 4 月第 1 次印刷

ISBN 978-7-5635-6613-6

定价:30.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

本书分为两篇：面向对象编程实验和面向对象编程习题。

面向对象编程实验由 Visual C++ 2010 简介、数据类型和表达式、流程控制语句、函数、面向对象编程基础、面向对象编程进阶、MFC 编程、Windows 窗体应用程序开发、数据库应用编程、网络编程共 10 章组成。

面向对象编程习题由数据类型和表达式习题、流程控制语句习题、函数习题、面向对象编程基础习题、面向对象编程进阶习题共 5 章组成。

本书第 1 篇中的例题用极简的代码，实现对基本概念的介绍，使读者能够更透彻地理解类、对象、函数重载、函数模板、类的继承、虚函数等面向对象的关键基础知识。学习面向对象程序设计最快、最有效的方式是学习现有的程序，因此编程实验能使读者全面、深入地理解并熟练地掌握所学内容。而编程离不开调试，本书补充了程序调试的方法与技巧并将之应用于部分程序的分析。第 2 篇是对第 1 篇的有效补充，能够进一步培养读者分析问题、解决问题的能力。

与其他高级语言（除了 C 语言外）相比，Visual C++ 具有更底层、运算速度更快的特点，因此至今仍是非常流行的经典编程语言。

本书之所以使用 Visual C++ 2010 是因为该版本安装方便，很多学生在安装更高版本的 Visual C++ 时会遇到各种问题，容易产生挫败感，不利于学生进一步深入学习。学生学完该课程，具备了一定的 Visual C++ 基础知识后，可以下载并使用更高版本的 Visual C++，或者学习更加便捷的 QT 窗体编程。

本书收集了大量的例题和习题，讲解深入浅出，特别适合初学者使用。

本书由赵敏担任主编，由方芳和高建波担任副主编。感谢曾经使用过本书的学生所提出的宝贵意见。

由于编者水平有限，书中难免会有疏漏，恳请读者批评指正。

目 录

第 1 篇 面向对象编程实验

第 1 章 Visual C++ 2010 简介	3
【例 1-1】 创建一个 CLR 控制台应用程序,输出“Hello World!”	3
【例 1-2】 创建一个 Win32 控制台应用程序,输出“hello”	6
【例 1-3】 创建一个 Windows 窗体应用程序,输出“大家好,欢迎使用 C++”	7
【例 1-4】 使用断点调试的方法分析程序的功能	9
第 2 章 数据类型和表达式	14
【例 2-1】 引用实例	14
【例 2-2】 测试变量 K 的值、地址及 K 的地址所存储的内容	15
【例 2-3】 引用、指针实例	15
【例 2-4】 局部、全局数组变量的认识	16
第 3 章 流程控制语句	17
【例 3-1】 输入一个整数,求该整数的位数	17
【例 3-2】 求 100 以内能同时被 3 和 7 整除的奇数	17
【例 3-3】 输入两个实数和四则运算符,输出运算结果	18
第 4 章 函数	20
【例 4-1】 将一个十进制数转换成任意进制数	20
【例 4-2】 将引用作为函数参数,求数组的最大值和最小值	23
【例 4-3】 输出两个数(函数的默认参数值)	24
【例 4-4】 函数模板与函数重载	25
【例 4-5】 求一个数的绝对值(函数的重载)	27

第 5 章 面向对象编程基础	28
【例 5-1】 编写一个 Win32 控制台应用程序(关于类、构造函数、重载)	28
【例 5-2】 静态成员、静态函数	29
【例 5-3】 求三个数中的最大数(函数内联)	30
【例 5-4】 友元函数	31
【例 5-5】 友元类	31
第 6 章 面向对象编程进阶	33
【例 6-1】 父类仅包含公有成员时的类继承	33
【例 6-2】 父类包含公有、保护和私有成员时的类继承	34
【例 6-3】 类继承中构造、析构函数的调用顺序	35
【例 6-4】 类继承中构造函数的参数传递	36
【例 6-5】 继承的重复问题	38
【例 6-6】 虚基类	39
【例 6-7】 虚基类的调用顺序和参数传递	41
【例 6-8】 虚函数	42
第 7 章 MFC 编程	45
【例 7-1】 菜单实例一	45
【例 7-2】 菜单实例二	47
【例 7-3】 时钟小程序	50
【例 7-4】 消息实例	54
【例 7-5】 计算器(小数点后仅保留一位数字)	56
【例 7-6】 创建一个基于对话框的 MFC 不规则窗口	61
【例 7-7】 画一条直线	62
【例 7-8】 创建一个单文档的 MFC 应用项目,命名为“tuxingwenzi”	64
【例 7-9】 编写一个单文档使用 CPaintDC 类的 MFC 应用程序	65
第 8 章 Windows 窗体应用程序开发	66
【例 8-1】 Windows 窗体应用程序的窗体和消息	66
【例 8-2】 在富文本框中显示在消息框中所选的按键	69

【例 8-3】 菜单、工具栏、状态栏、富文本框	70
第 9 章 数据库应用编程	77
【例 9-1】 用 SqlConnection 等控件新建一个 Windows 窗体程序	78
【例 9-2】 使用 SqlCommand 对象	83
【例 9-3】 查看学生数据库中 xsxx 表的内容	85
【例 9-4】 查看表的内容及其字段结构	87
【例 9-5】 创建使用无连接的表	90
【例 9-6】 SqlDataAdapter 对象	91
【例 9-7】 简单数据绑定	98
【例 9-8】 复杂数据绑定	99
【例 9-9】 DataGridView 控件的某个单元格单击事件	100
第 10 章 网络编程	102
【例 10-1】 使用 Socket 建立一个简单的聊天室应用程序	105

第 2 篇 面向对象编程习题

第 11 章 数据类型和表达式习题	115
第 12 章 流程控制语句习题	118
第 13 章 函数习题	123
第 14 章 面向对象编程基础习题	126
第 15 章 面向对象编程进阶习题	134
参考文献	142
附录 1 习题参考答案	143
附录 2 关于主函数中两个参数的解释	171
附录 3 对多张表的增删改查	173

第 1 篇 面向对象编程实验

第 1 章 Visual C++ 2010 简介

使用 Visual C++ 2010 可以开发两种不同的 C++ 应用程序:①本地 C++ 程序,即在本地计算机上执行的应用程序,其使用的是 ISO/ANSI 语言标准;②CLR 程序或 C++/CLI 程序,是使用 C++/CLI 的 C++ 扩充版本编写的在 CLR 控制下运行的应用程序。

1. 实验要求

- (1) 了解集成开发环境(Integrated Development Environment)。
- (2) 熟悉程序的开发过程,能够运行一个完整的程序。


2. 控制台应用程序设计实例

像开发 Windows 应用程序一样,Visual C++ 2010 为我们提供可以编写、编译并测试没有任何 Windows 程序所需元素的 C++ 程序,即基本上基于字符命令行的应用程序。这些程序在 Visual C++ 2010 中被称为控制台应用程序。用户在字符模式中通过键盘和屏幕与它们进行通信。因此,控制台应用程序可以使读者将注意力集中在语言特点上,而不必考虑操作环境(因为该语言定义中没有图形功能)。在开发 MFC 应用程序(Microsoft Foundation Classes,微软基类)或者 CLR Windows Forms 应用程序时,这两类应用程序本身会提供大量的用户界面图形。

本章将用到两种不同的控制台应用程序:Win32 控制台应用程序是本地代码,使用该类程序可以测试 ISO/ANSI C++ 的功能;CLR 控制台应用程序是针对 CLR 的,因此,在学习 C++/CLI 功能时使用该类应用程序。

CLR 控制台应用程序是利用 C++ 语言开发的应用程序类型之一,因为不涉及 Windows 系统的组成元素,所以结构比较简单。下面介绍如何创建控制台应用程序。

【例 1-1】 创建一个 CLR 控制台应用程序,输出“Hello World!”

(1) 创建项目:启动 Visual Studio.NET,选择“开始”→“程序”→“Microsoft Visual Studio 2010”→“ Microsoft Visual Studio 2010”菜单项。

通常,要编写一个 C++ 应用程序,首先应该创建一个项目。在起始页面的“项目”选项卡中单击“创建项目”按钮,或者选择“文件”→“新建”→“项目(P)...”菜单项打开新建项目对话框,如图 1.1 所示。

(2) 选择 C++ 项目类型:在新建项目对话框的左侧项目类型选项组中选择 Visual C++。右侧为模板选项组,Visual Studio 为程序员提供了多种类型的模板,如 Windows 窗体应用程序、MFC 应用程序等。这里选择“CLR 控制台应用程序”选项,如图 1.2 所示。

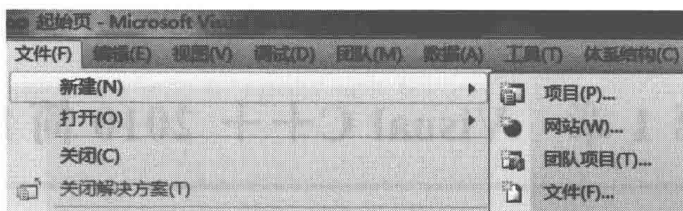


图 1.1 用菜单创建项目

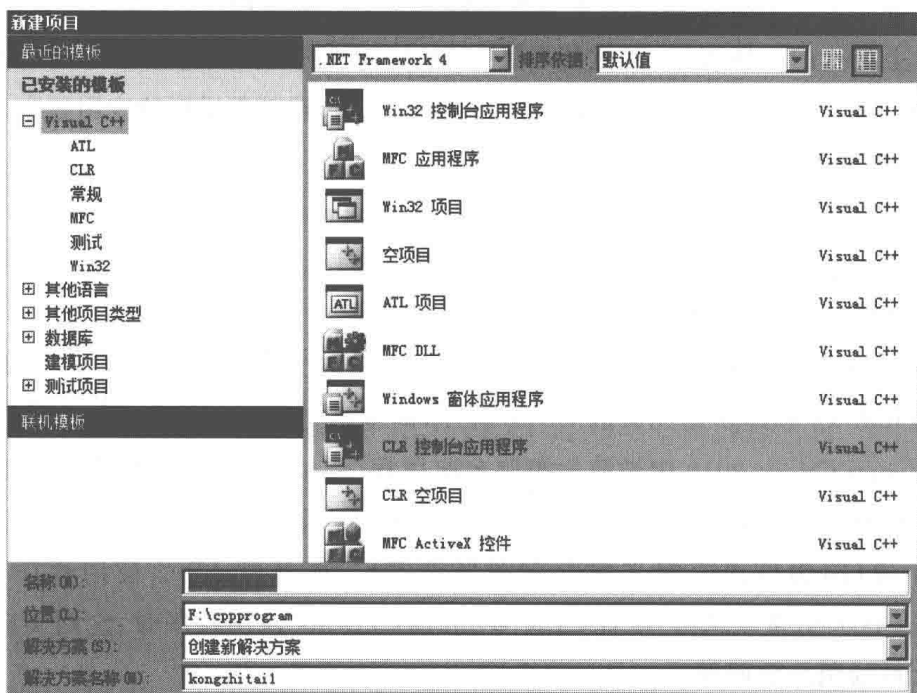


图 1.2 新建项目对话框

(3) 项目名称:在图 1.2 所示的新建项目对话框的下方输入项目的名称、位置 and 解决方案名称。该项目名称为“kongzhitail”。

设置完成后,单击“确定”按钮,Visual Studio 将在指定的文件夹中创建解决方案文件,并在该文件夹下创建以项目名称命名的子文件夹,存储项目文件。

(4) 打开项目:创建项目“kongzhitail”后,可在起始页上的“最近使用的项目”列表中看到该项目的名称,如图 1.3 所示。再次打开 Visual Studio 时,直接单击该链接即可打开此项目。或在起始页中单击打开项目,在打开项目对话框中的查找范围内选择项目所在目录,然后选中要打开的项目文件,如图 1.4 所示。

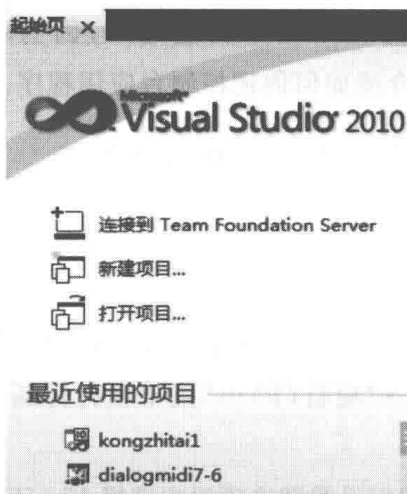


图 1.3 起始页窗口

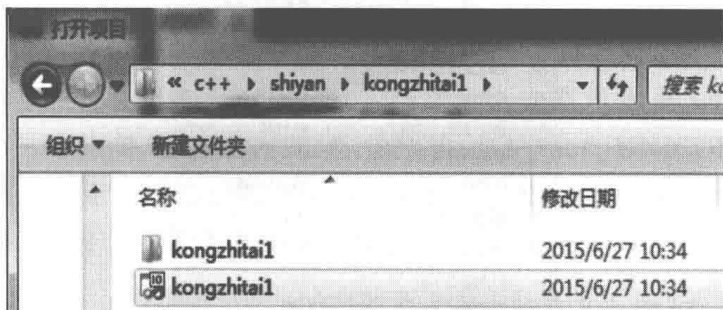


图 1.4 打开项目对话框

(5) 代码编辑器:打开项目文件,进入代码编辑器窗口,如图 1.5 所示。

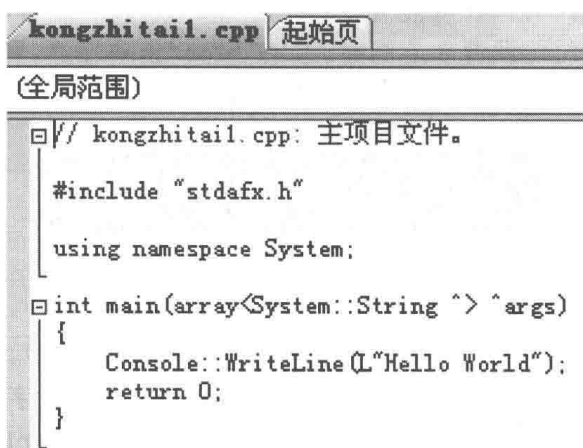


图 1.5 代码编辑器窗口

(6) 编写代码:打开 Program.cpp 文件,系统自动添加如下代码:

```
// kongzhitai1.cpp:主项目文件
#include "stdafx.h"
using namespace System;
int main(array<System::String ^> ^args)
{
    Console::WriteLine(L"Hello World");
    Console::ReadLine(); //增加使结果在屏幕停留的代码
    return 0;
}
```

(7) 保存项目

在“文件”菜单中,单击“全部保存”选项,或在工具栏中单击“保存”按钮。

(8) 运行程序

选择“调试”→“开始执行(不调试)(H)”菜单项,如图 1.6 所示。

运行结果如图 1.7 所示。

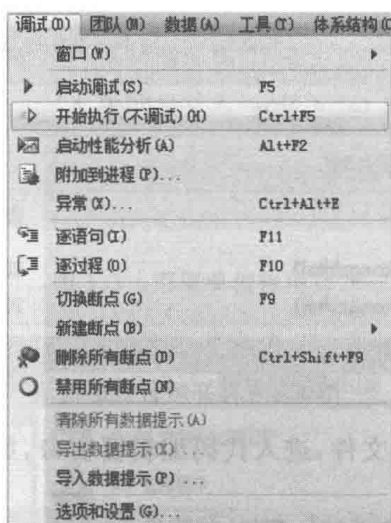


图 1.6 调试菜单界面



图 1.7 运行结果界面

这里的函数名称是 `main`, 有的名称是 `_tmain` (如例 1-2 所示)。事实上, 所有的 ISO/ANSI C++ 都是在 `main()` 函数中开始执行的。当使用 Unicode 字符时, 微软公司还提供了相应的 `wmain` 函数。而名称 `_tmain` 被定义为 `main` 或 `wmain`, 这取决于程序是否将使用 Unicode 字符。所有的 ISO/ANSI C++ 例子都使用 `main()` 函数。

【例 1-2】 创建一个 Win32 控制台应用程序, 输出“hello”

```
#include "stdafx.h"           //包含预编译头文件
#include "iostream"           //包含 C++ 的标准输入输出头文件 iostream
using std::cout;             //调用命名空间 std 中的 cout() 函数
using std::endl;             //调用命名空间 std 中的 endl() 函数
int _tmain(int argc, _TCHAR* argv[]) //主函数开始
{cout << "hello" << endl;} //输出 hello 并换行, endl 表示换行, << 表示输出流
```

`#include "stdafx.h"`——包含预编译头文件。所谓预编译头, 就是把头文件事先编译成一种二进制的中间格式, 供后续的编译过程使用。编译器预编译头的 `.h`、`.c`、`.cpp` 文件在整个编译过程中只编译一次, 如果预编译头所涉及的部分不发生改变, 那么在随后的编译过程中此部分不重新进行编译, 进而大大提高了编译速度。

`#include "iostream"`——包含 C++ 的标准输入输出头文件 `iostream`, 也就是编译器先把头文件 `iostream` 中的所有内容复制到 `#include` 的位置, 再进行编译。注意 C++ 的这个标准输入输出头文件的名称就是 `iostream`, 没有 `.h` 的后缀。

`using std::cout`——相同的函数名可以存在于不同的命名空间中, 而 `std` 命名空间中有许多常用的函数, 如 `cout()`、`cin()` 等。

主函数有两个形参:第一个参数“argc”表示程序运行时,命令行参数的个数,默认为1;第二个参数“*argv[]”表示指向字符串数组的指针,每个字符串对应一个参数。_TCHAR是一种通用字符类型,它有可能是char型(ASCII码),也有可能是wchar_t型(Unicode码),决定它是哪一种的关键是程序是否定义了#define _UNICODE,若没有定义,_TCHAR就是char型,定义了之后就是wchar_t型。_TCHAR有这种特性是因为微软制定的字符映射规则。关于这两个参数的更多解释见附录2。

【例 1-3】 创建一个 Windows 窗体应用程序,输出“大家好,欢迎使用 C++”

(1) 运行 Visual Studio 2010,创建一个 Windows 应用程序项目,在新建项目的模板中选中 Windows 窗体应用程序,在“名称”文本框中将项目改名为 exwin_01,在“位置”文本框中输入项目保存的目录位置,本例保存在本地硬盘的“F:\cppprogram”文件夹,选中“创建解决方案的目录”复选框,然后单击“确定”按钮,如图 1.8 所示。



图 1.8 创建 Windows 窗体应用程序的新建项目界面

(2) 在窗体设计器中选中窗体,然后单击鼠标右键,打开属性窗口(也可以在菜单中选择“视图”菜单下的属性窗口,从而打开属性窗口),如图 1.9 所示。在属性窗口中分别修改窗体的 Text 属性、Size 属性和 FormBorderStyle 属性。图 1.10 所示为修改窗体的 Text 属性值,观察修改后窗体显示的效果。

(3) 从工具箱中拖放一个 Label 控件到设计窗体中,如图 1.11 所示。选中该控件,修改其 Text 属性为“大家好,欢迎使用 C++”,修改完毕后窗体如图 1.12 所示。

选择 Label1 对象,设置其 Font 属性为“隶书,5 号字,斜体”,ForeColor 属性为“红色”,修改 AutoSize 属性为“False”,拖动该控件右下角改变其大小,观察变化。

按“F5”键运行程序,运行界面如图 1.13 所示。可以看到,没有编写一句代码,就可以实现这样一个功能。

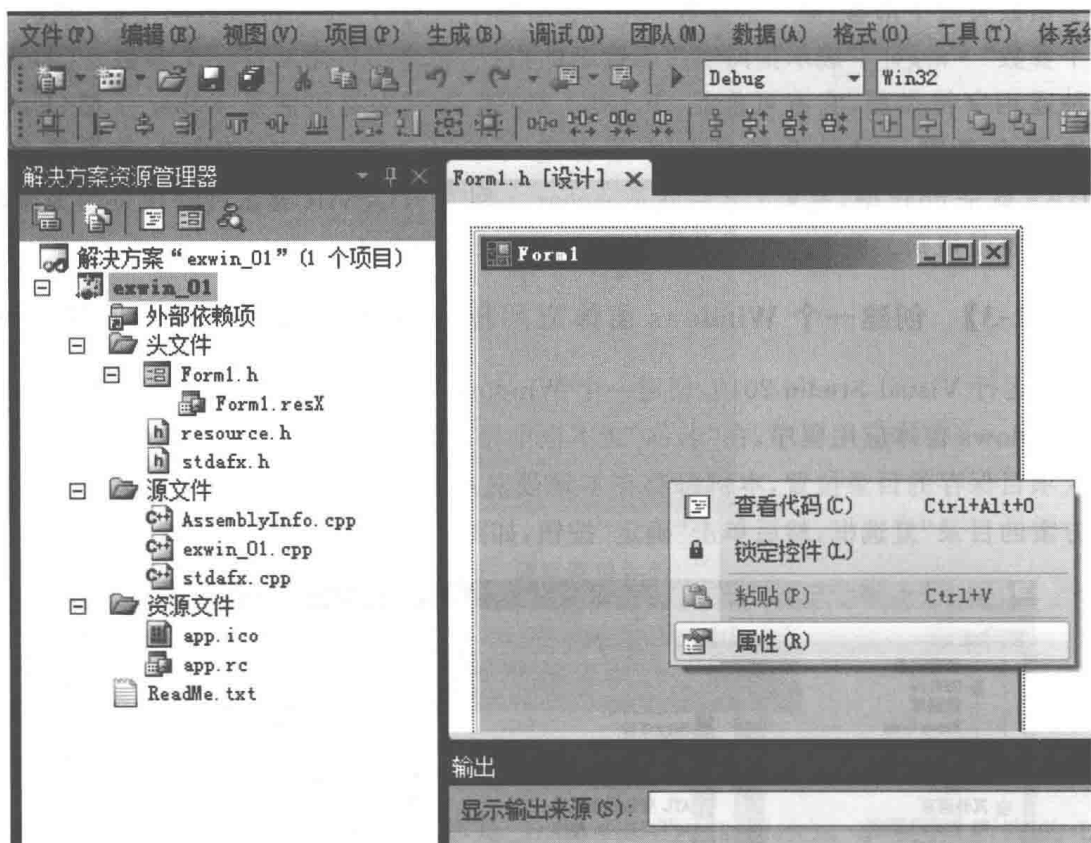


图 1.9 在窗体空白处单击鼠标右键打开属性窗口

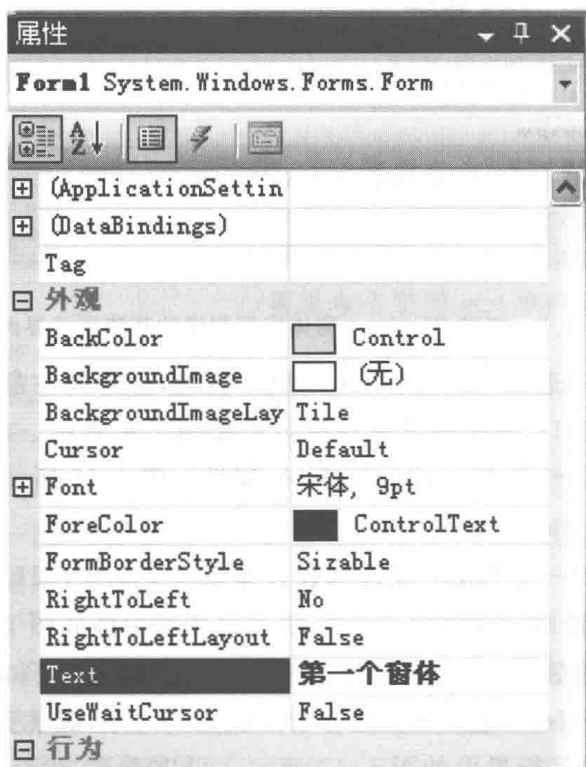


图 1.10 修改窗体属性的值

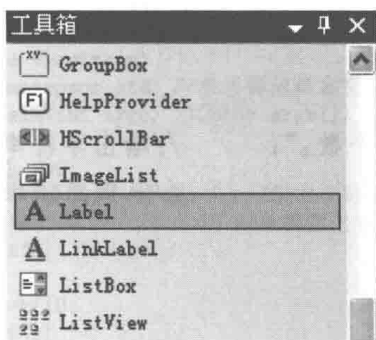


图 1.11 从工具箱中拖放一个 Label 控件到设计窗体中



图 1.12 修改 Label1 控件的属性值

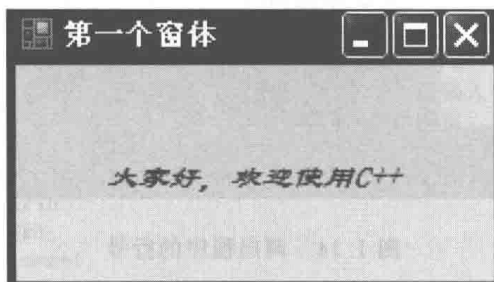


图 1.13 例 1-3 的运行界面

【例 1-4】 使用断点调试的方法分析程序的功能

1. #include "stdafx.h"
2. #include <iostream>
3. using namespace std; //表示释放命名空间 std 内的函数和变量

```

4.  int _tmain(int argc, _TCHAR * argv[])
5.  {
6.      int k,n= 0;
7.      cout<<"请输入一个数:";      //输出字符串"请输入一个数:"
8.      cin>>k;                      //从键盘上输入k 的值
9.      while(k)
10.     {
11.         k = k/10;
12.         if(k)
13.             n = n + 1;
14.     }
15.     n = n + 1;
16.     cout << n; //输出 n 的值
17.     return 0;
18. }

```

选择“工具”→“选项”菜单项,在弹出的对话框中展开文本编辑器,再选择 C/C++,勾选右侧的“行号”复选框,然后单击“确定”按钮,如图 1.14 所示。调出行号后,分别在程序的第 11 行和第 13 行左边灰色区域单击,从而设置 2 个断点,如图 1.15 所示。设置好断点后就可以开始调试程序了。

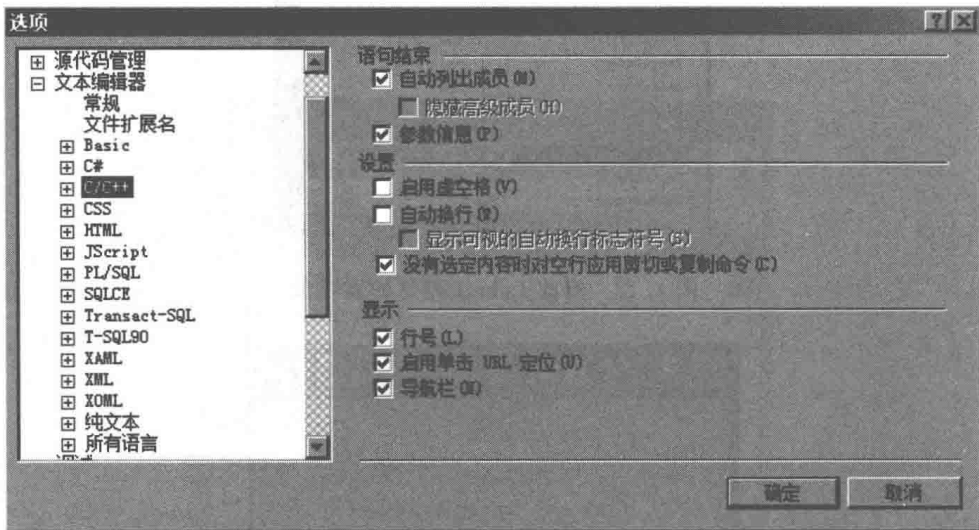



图 1.14 调出程序的行号

开始调试程序时,可以单击菜单栏上的“”快捷图标,也可以选择“调试”→“启动调试”菜单项,还可以直接按“F5”键。按“F5”键后,弹出如图 1.16 所示界面,在该界面任意输入一个整数,如“4321”,然后按“Enter”键。这时程序停留在第 11 行(该行红色断点内出现一个向右的黄色箭头),并且在下方的自动窗口中显示了 k 的值为 4321,类型为 int,如图 1.17 所示。接着使用“调试”菜单中的子菜单项“逐语句”或“逐过程”进行调试(也可以通过快捷键“F11”或“F10”进行调试)。其中,“逐语句”调试是指在遇到函数调用语句的时候进入函数内部执行;“逐过程”调试是指在遇到函数调用语句时把函数当作一条语句执行。在这里