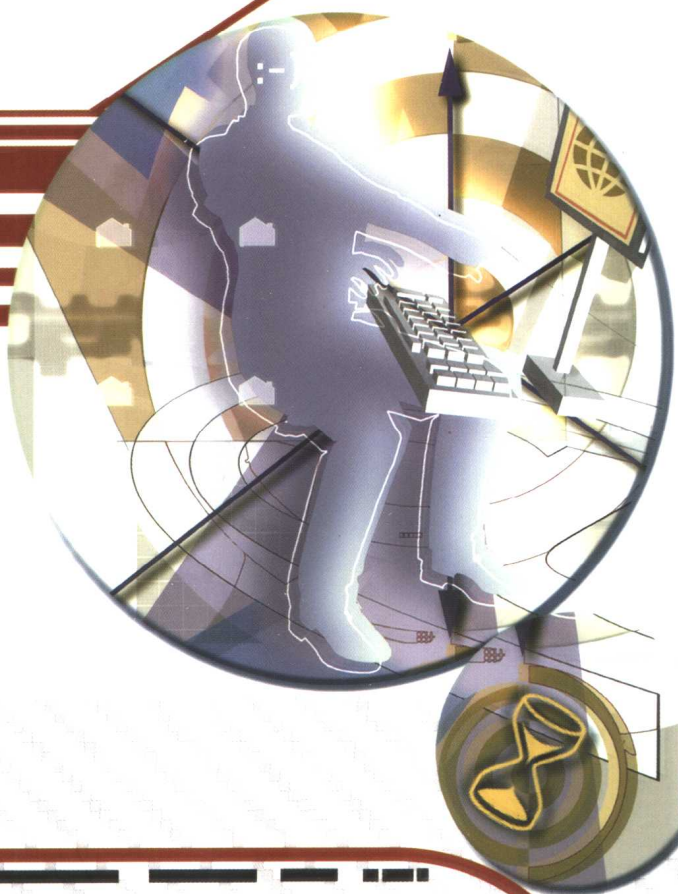


Visual C++.NET 编程实例

苏 峰 黄金双 汤 蕾 编著



- ◆ 通过精选的50个经典实例作为读者学习的模板
- ◆ 内容通俗易懂、操作步骤清晰、讲解详细
- ◆ 读者迅速掌握编程的方法，达到举一反三的目的
- ◆ 配套光盘中包括全部实例的代码和相关资源文件



清华大学出版社

Visual C++.NET 编程实例

苏 峰 黄金双 汤 蕾 编著

清华大学出版社

北 京

内 容 简 介

本书以实例讲解的形式向读者展示如何运用 Visual C++ .NET 开发实用的 Windows 应用程序。每一个实例都有详细的创建过程，一步一步引导读者完成程序的创建、调试和运行。全书分为 8 章，精选了 50 个实例，内容包含了对话框和控件、菜单和工具栏、文档和视图等编程基础，以及网络和数据库、多媒体、多线程、OpenGL、ActiveX 控件和动态链接库等编程专题。

本书内容丰富，范例典型，不仅是国内广大 Visual C++ .NET 初级开发人员很好的入门指导书，对具有丰富程序开发经验的人员同样具有参考价值。

本书所有实例程序均在 Visual C++ .NET 环境下调试通过，配套光盘中包含全部实例的代码和相关资源文件。

版权所有，翻印必究。

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

Visual C++ .NET 编程实例/苏峰，黄金双，汤蕾编著. —北京：清华大学出版社，2003

ISBN 7-302-07683-9

I.V… II.①苏…②黄…③汤… III.C 语言—程序设计 IV.TP312

中国版本图书馆 CIP 数据核字(2003)第 106051 号

出 版 者：清华大学出版社

<http://www.tup.com.cn>

社 总 机：010-62770175

地 址：北京清华大学学研大厦

邮 编：100084

客户服务：010-62776969

组稿编辑：张 瑜

文稿编辑：张彦青

封面设计：陈刘源

印 刷 者：北京市通州大中印刷厂

装 订 者：三河市兴旺装订有限公司

发 行 者：新华书店总店北京发行所

开 本：185×260 印张：33.5 字数：798 千字

版 次：2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

书 号：ISBN 7-302-07683-9/TP·5628

印 数：1~4000

定 价：52.00 元(含 1 张光盘)

前 言

2002年初, Microsoft 公司发布了 Visual Studio .NET 可视化编程环境, 在众多方面比 Visual Studio 6.0 有了很大改进。其中 Visual C++ .NET 与 Visual C++ 6.0 相比, 在开发环境界面上有了很大的变化, 也增加了一些新的特性, 如托管 C++编程和 ATL Server(对 ATL 库的扩展), 开发人员可使用 C++语法创建 ISAPI Web 应用程序等。

要想掌握一门编程语言, 熟悉一种编程环境, 实战是很重要的。通过大量的实例演练来逐步提高, 是很好的掌握编程语言的途径。本书以实例讲解的形式向读者展示如何运用 Visual C++ .NET 开发实用的 Windows 应用程序。每一个实例都有详细的创建过程, 一步一步地引导读者完成程序的创建、调试和运行。通过本书所有例子的实战演练, 读者可以熟练运用 Visual C++ .NET 编程, 熟悉 Windows 编程的各个方面, 为将来的开发实践打下坚实的基础。

全书共分为 8 章, 精选了 50 个实例, 内容主要包含对话框和控件、菜单和工具栏、文档和视图等编程基础, 以及网络、数据库和多媒体等编程专题。

第 1 章通过 3 个简单的实例演示了 Visual C++ .NET 开发环境的使用, 由于 .NET 的开发环境界面跟以往的 Visual Studio 6.0 相比有很大的不同, 本章详细介绍了 Visual C++ .NET 的开发界面, 并通过一个实例讲述了 Visual C++ .NET 的新技术——托管 C++编程。

第 2 章创建了 12 个基于对话框的应用程序, 演示了在 Visual C++ .NET 环境下常用控件的添加与使用, 以及其属性的修改。

第 3 章通过 6 个实例演示了 Visual C++ .NET 应用程序中菜单、工具栏、状态栏的添加和修改。

第 4 章主要涉及了文档和视图的编程, 包括单文档、多文档编程, 视图分割, 打印预览等内容。

第 5 章演示了如何在应用程序中绘图。

第 6 章给出了 Visual C++ .NET 环境下网络和数据库方面的编程实例, 内容涉及多个网络方面的类的使用, 以及基于 ODBC 和 ADO 的数据库编程。

第 7 章演示了如何利用 Visual C++ .NET 开发小游戏, 给出了 3 个有趣的小游戏的设计步骤。

第 8 章是其他类别的实例, 包括 Visual C++ .NET 环境下的多媒体、多线程、OpenGL、ActiveX、动态连接库等内容。

本书所选实例多为 Windows 应用程序开发的基础, 通过这些实例向读者讲述了如何利用 Visual C++ .NET 这种新的开发环境创建应用程序。由于本书主要面向初级读者, 因此侧重于基本的 MFC 编程, 而没有涉及 ATL 和 ISAPI 等技术, 读者可以参考其他相关书籍。

书中每一个实例均在 Visual C++ .NET 中文版下调试通过, 附送光盘中给出了书中所有例子的源代码和所用到的资源文件。

我们假设本书的读者有一定的 C++ 语言基础,但不要求熟悉 Visual C++ 编程环境。限于篇幅,书中没有对每一段代码的意义作过多的讲解,相关的内容可以参考有关 Windows 编程的书籍。

本书出版得到了徐西刚、单永杰、赵世初同志的帮助和支持,在此深表感谢。由于作者水平有限,加之时间仓促,错误疏漏之处在所难免,敬请读者批评指正。

编者

2003 年 4 月

目 录

第1章 初识 Visual C++ .NET.....1

实例 1.1 Hello.....1

1.1.1 实例简介1

1.1.2 创建过程1

1.1.3 运行效果4

实例 1.2 响应键盘和鼠标.....4

1.2.1 实例简介4

1.2.2 创建过程4

1.2.3 运行效果8

实例 1.3 托管 C++编程9

1.3.1 实例简介9

1.3.2 创建过程9

1.3.3 运行效果17

第2章 对话框和控件.....19

实例 2.1 单选和复选.....19

2.1.1 实例简介19

2.1.2 创建过程19

2.1.3 运行效果26

实例 2.2 查看内存信息.....26

2.2.1 实例简介26

2.2.2 创建过程27

2.2.3 运行效果28

实例 2.3 超链接28

2.3.1 实例简介28

2.3.2 创建过程29

2.3.3 运行效果42

实例 2.4 计算器42

2.4.1 实例简介42

2.4.2 创建过程43

2.4.3 运行效果50

实例 2.5 动画播放器.....50

2.5.1 实例简介50

2.5.2 创建过程 51

2.5.3 运行效果 56

实例 2.6 文件夹浏览器..... 56

2.6.1 实例简介 56

2.6.2 创建过程 57

2.6.3 运行效果 69

实例 2.7 滑块和进度条..... 69

2.7.1 实例简介 69

2.7.2 创建过程 69

2.7.3 运行效果 74

实例 2.8 日期和时间..... 74

2.8.1 实例简介 74

2.8.2 创建过程 74

2.8.3 运行效果 80

实例 2.9 IP 地址控件..... 81

2.9.1 实例简介 81

2.9.2 创建过程 81

2.9.3 运行效果 86

实例 2.10 ActiveX 控件..... 87

2.10.1 实例简介 87

2.10.2 创建过程 87

2.10.3 运行效果 94

实例 2.11 对话框示例..... 94

2.11.1 实例简介 94

2.11.2 创建过程 94

2.11.3 运行效果 102

实例 2.12 可扩展对话框..... 103

2.12.1 实例简介 103

2.12.2 创建过程 103

2.12.3 运行效果 109

第3章 菜单、工具栏和状态栏..... 110

实例 3.1 添加和删除菜单项..... 110

3.1.1 实例简介	110	4.4.2 创建过程	161
3.1.2 创建过程	110	4.4.3 运行效果	168
3.1.3 运行效果	116	实例 4.5 属性表和属性页	168
实例 3.2 动态菜单	117	4.5.1 实例简介	168
3.2.1 实例简介	117	4.5.2 创建过程	169
3.2.2 创建过程	117	4.5.3 运行效果	182
3.2.3 运行效果	122	第 5 章 绘图	184
实例 3.3 环境菜单	122	实例 5.1 在设备环境中绘图	184
3.3.1 实例简介	122	5.1.1 实例简介	184
3.3.2 创建过程	122	5.1.2 创建过程	184
3.3.3 运行效果	126	5.1.3 运行效果	189
实例 3.4 添加工具栏按钮	126	实例 5.2 画笔和画刷	190
3.4.1 实例简介	126	5.2.1 实例简介	190
3.4.2 创建过程	126	5.2.2 创建过程	190
3.4.3 运行效果	132	5.2.3 运行效果	198
实例 3.5 向工具栏中添加控件	132	实例 5.3 鼠标绘图	199
3.5.1 实例简介	132	5.3.1 实例简介	199
3.5.2 创建过程	133	5.3.2 创建过程	199
3.5.3 运行效果	137	5.3.3 运行效果	211
实例 3.6 状态栏示例	137	实例 5.4 字体	212
3.6.1 实例简介	137	5.4.1 实例简介	212
3.6.2 创建过程	138	5.4.2 创建过程	212
3.6.3 运行效果	145	5.4.3 运行效果	219
第 4 章 文档和视图	146	第 6 章 网络与数据库	220
实例 4.1 文本编辑器	146	实例 6.1 获取本地 IP 地址	220
4.1.1 实例简介	146	6.1.1 实例简介	220
4.1.2 创建过程	146	6.1.2 创建过程	220
4.1.3 运行效果	148	3.1.3 运行效果	222
实例 4.2 文字处理器	148	实例 6.2 网页浏览器	223
4.2.1 实例简介	148	6.2.1 实例简介	223
4.2.2 创建过程	148	6.2.2 创建过程	223
4.2.3 运行效果	152	6.2.3 运行效果	239
实例 4.3 打印和打印预览	152	实例 6.3 邮件检查器	240
4.3.1 实例简介	152	6.3.1 实例简介	240
4.3.2 创建过程	153	6.3.2 创建过程	240
4.3.3 运行效果	160	6.3.3 运行效果	253
实例 4.4 切分窗口	160	实例 6.4 FTP 客户端程序	253
4.4.1 实例简介	160	6.4.1 实例简介	253

6.4.2 创建过程	253	8.3.2 创建过程	435
6.4.3 运行效果	284	8.3.3 运行效果	446
实例 6.5 访问 ODBC 数据源	285	实例 8.4 OpenGL 编程	446
6.5.1 实例简介	285	8.4.1 实例简介	446
6.5.2 创建过程	285	8.4.2 创建过程	447
6.5.3 运行效果	293	8.4.3 运行效果	460
实例 6.6 使用 ADO 访问数据库	293	实例 8.5 查看系统信息	461
6.6.1 实例简介	293	8.5.1 实例简介	461
6.6.2 创建过程	294	8.5.2 创建过程	461
6.6.3 运行效果	310	8.5.3 运行效果	466
第 7 章 小游戏	311	实例 8.6 线程示例	466
实例 7.1 扫雷	311	8.6.1 实例简介	466
7.1.1 实例简介	311	8.6.2 创建过程	466
7.1.2 创建过程	311	8.6.3 运行效果	471
7.1.3 运行效果	334	实例 8.7 多线程示例	472
实例 7.2 俄罗斯方块	335	8.7.1 实例简介	472
7.2.1 实例简介	335	8.7.2 创建过程	472
7.2.2 创建过程	335	8.7.3 运行效果	492
7.2.3 运行效果	358	实例 8.8 视频播放	492
实例 7.3 五子棋	359	8.8.1 实例简介	492
7.3.1 实例简介	359	8.8.2 创建过程	493
7.3.2 创建过程	359	8.8.3 运行效果	498
7.3.3 运行效果	419	实例 8.9 音频播放	499
第 8 章 其他实例	420	8.9.1 实例简介	499
实例 8.1 时钟	420	8.9.2 创建过程	499
8.1.1 实例简介	420	8.9.3 运行效果	508
8.1.2 创建过程	420	实例 8.10 创建 ActiveX 控件	509
8.1.3 运行效果	426	8.10.1 实例简介	509
实例 8.2 操作系统注册表	427	8.10.2 创建过程	509
8.2.1 实例简介	427	8.10.3 运行效果	516
8.2.2 创建过程	427	实例 8.11 动态链接库	517
8.2.3 运行效果	434	8.11.1 实例简介	517
实例 8.3 图片浏览器	435	8.11.2 创建过程	517
8.3.1 实例简介	435	8.11.3 运行效果	525

第 1 章 初识 Visual C++ .NET

许多人可能会认为编写 Windows 应用程序是一件很费力的事情，但是，如果选择了一种好的编程工具可以使工作简单许多。要进行可视化编程，Visual C++ 无疑是一个很好的选择。

Visual C++ 是 Microsoft 公司推出的目前使用极为广泛的基于 Windows 平台的可视化编程环境。最新版本 Visual C++ .NET 与以往版本相比，增加了许多新的特性，其中包括：Web 应用程序、新颖的 C# 编程语言以及 ATL、DCOM、MFC 和数据库等方面性能的增强，尤其是在开发环境界面上变化更大，它采用平面化的操作界面，这一点跟 Visual Basic 有点类似，它的亲和性使更多的程序开发人员乐意接受。

本章将通过 3 个简单的实例来演示 Visual C++ .NET 开发环境的使用。

实例 1.1 Hello

1.1.1 实例简介

本例将创建一个简单的单文档应用程序，程序运行时将出现一个 Windows 窗口，其中显示一行提示信息。通过本例，读者可以初步认识 Visual C++ .NET 的开发环境和应用程序向导，为将来开发更加有用的程序奠定基础。

1.1.2 创建过程

1. 启动 Visual Studio.NET 开发环境

在 Windows 操作系统中单击【开始】| Microsoft Visual Studio.NET 命令启动它，开发环境初始界面如图 1.1 所示。读者计算机上显示的界面因为个人设置的不同可能会不太一样，可以在【视图】菜单选择显示相应的窗口。

2. 创建一个新的项目

单击【文件】|【新建】|【项目】命令或者按 Shift+Ctrl+N 键创建一个新的项目，将弹出【新建项目】对话框，在对话框左侧的【项目类型】列表框中选择【Visual C++ 项目】，右侧的【模板】列表框中选择【MFC 应用程序】图标，这意味着这个项目将生成一个标准的 Windows 可执行程序。

3. 为工程命名并选择存放路径

在【名称】文本框中输入将要创建的项目的名字“Hello”，单击【浏览】按钮可以改

变项目文件夹保存的路径，我们采用默认的路径，如图 1.2 所示，单击【确定】按钮进入下一步。

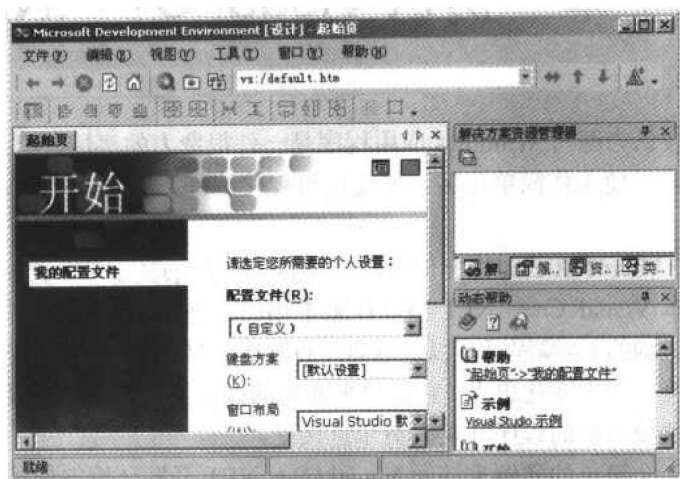


图 1.1 Visual Studio.NET 开发环境

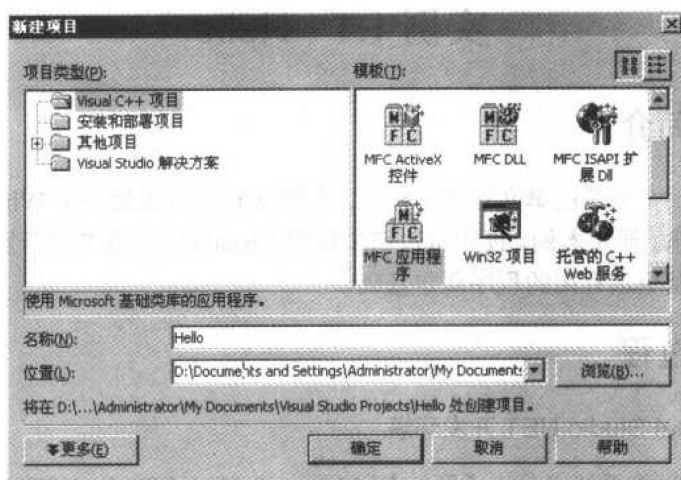


图 1.2 新建项目

4. 完成项目工程的设置

在【应用程序向导】中，单击左侧的【应用程序类型】标签，选择【应用程序类型】下的【单文档】单选按钮，如图 1.3 所示。接受应用程序向导的所有其他默认设置，关于其他设置在以后的实例中会逐步涉及到，单击【完成】按钮，开始生成一个单文档的应用程序。

5. 添加代码

切换到【解决方案资源管理器】窗口，如果开发环境中没有显示该窗口，可以单击【视图】|【解决方案资源管理器】命令来激活它。在【解决方案资源管理器】窗口中单击 Hello 左侧的“+”号展开文件列表，然后打开【源文件】文件夹，选择 HelloView.cpp 文件，如图 1.4 所示，双击打开它，在该文件的 OnDraw 函数中加入以下加粗的代码：

```
void CHelloView::OnDraw(CDC* /*pDC*/)
{
    CHelloDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: 在此处为本机数据添加绘制代码
    pDC->TextOut(20,20,"Hello! 欢迎学习Visual C++ .NET! ");
}
```

然后将 OnDraw 函数名括号中的 pDC 两侧的注释符号去掉。

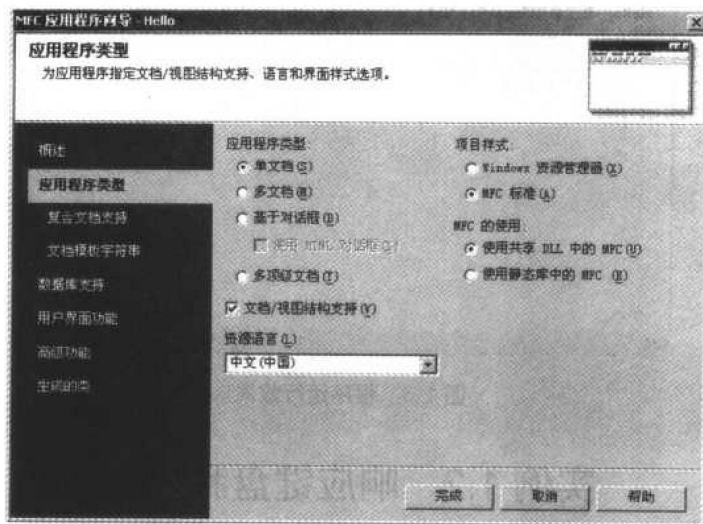


图 1.3 应用程序类型选择



图 1.4 解决方案资源管理器

6. 编译链接并运行程序

单击【生成】|【生成 Hello】命令，然后单击【调试】|【开始执行】命令，或者直接按 Ctrl+F5 键运行程序。

1.1.3 运行效果

程序运行时显示一个窗口，里边显示文字：“Hello! 欢迎学习 Visual C++ .NET! ”，如图 1.5 所示。

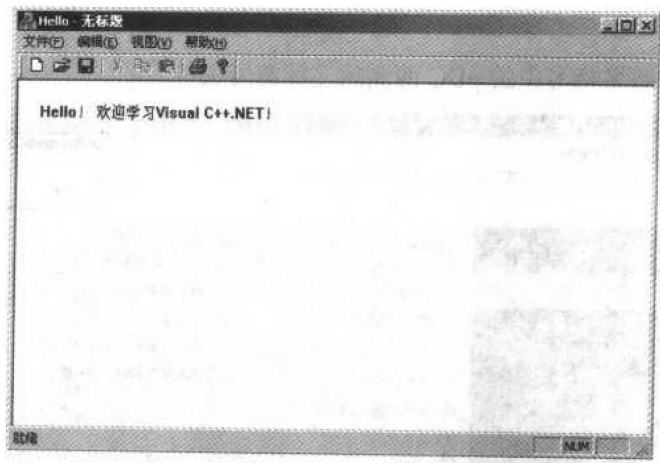


图 1.5 程序运行效果

实例 1.2 响应键盘和鼠标

1.2.1 实例简介

本例将创建一个简单的程序，响应键盘按下和鼠标单击事件。当用户按下键盘上相应的键或单击鼠标左右键时，均会弹出一个消息框，显示按下了哪一个键；当鼠标在窗口中移动时，将显示其相对于窗口左上角的横坐标和纵坐标数值。通过本例，读者可以学会如何在 Visual C++ .NET 中进行消息处理。

1.2.2 创建过程

1. 创建一个 MFC 应用程序

项目名称取为 KeyMouse，在【应用程序向导】中单击【应用程序类型】标签，选择程序类型为【单文档】，单击【完成】按钮，生成应用程序框架。

2. 响应键盘和鼠标事件

打开【类视图】窗口，如果开发环境中没有显示该窗口，可以单击菜单【视图】|【类视图】命令将其激活。在【类视图】窗口中，单击 KeyMouse 左侧的“+”号，将程序生成的类展开，如图 1.6 所示。右击类 CKeyMouseView，在快捷菜单中单击【属性】命令，则显示该类的【属性】窗口，如图 1.7 所示。



图 1.6 类视图

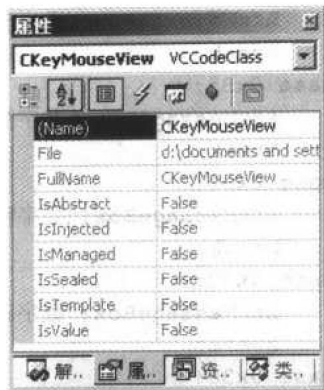


图 1.7 属性视图

在类 CKeyMouseView 的【属性】窗口中，单击【消息】按钮(左侧第 5 个)，在下面的消息列表中选择 WM_KEYDOWN，然后单击其右侧下拉列表框中的下拉箭头，选择添加 OnKeyDown 选项，如图 1.8 所示。

用同样的方法添加鼠标左键单击、右击和鼠标移动事件，分别为 WM_LBUTTONDOWN、WM_RBUTTONDOWN 和 WM_MOUSEMOVE，生成的事件处理函数分别为 OnLButtonDown、OnRButtonDown 和 OnMouseMove。



图 1.8 添加事件

3. 添加消息处理函数代码

在【解决方案资源管理器】中打开 CkeyMouse View.cpp 文件，在其中加入消息处理函数代码(加粗显示为加入代码)：

```
void CKeyMouseView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    switch (nChar)
    {
    case 0x30:
        AfxMessageBox("数字键0被按下");
        break;
    case 0x31:
        AfxMessageBox("数字键1被按下");
        break;
    case 0x32:
        AfxMessageBox("数字键2被按下");
        break;
    case 0x33:
        AfxMessageBox("数字键3被按下");
        break;
    case 0x34:
```

```
AfxMessageBox("数字键4被按下");  
break;  
case 0x35:  
    AfxMessageBox("数字键5被按下");  
    break;  
case 0x36:  
    AfxMessageBox("数字键6被按下");  
    break;  
case 0x37:  
    AfxMessageBox("数字键7被按下");  
    break;  
case 0x38:  
    AfxMessageBox("数字键8被按下");  
    break;  
case 0x39:  
    AfxMessageBox("数字键9被按下");  
    break;  
case 0x41:  
    AfxMessageBox("字母键A被按下");  
    break;  
case 0x42:  
    AfxMessageBox("字母键B被按下");  
    break;  
case 0x43:  
    AfxMessageBox("字母键C被按下");  
    break;  
case 0x44:  
    AfxMessageBox("字母键D被按下");  
    break;  
case 0x45:  
    AfxMessageBox("字母键E被按下");  
    break;  
case 0x46:  
    AfxMessageBox("字母键F被按下");  
    break;  
case 0x47:  
    AfxMessageBox("字母键G被按下");  
    break;  
case 0x48:  
    AfxMessageBox("字母键H被按下");  
    break;  
case 0x49:  
    AfxMessageBox("字母键I被按下");  
    break;  
case 0x4A:  
    AfxMessageBox("字母键J被按下");  
    break;  
case 0x4b:  
    AfxMessageBox("字母键K被按下");  
    break;
```

```
case 0x4c:
    AfxMessageBox("字母键L被按下");
    break;
case 0x4d:
    AfxMessageBox("字母键M被按下");
    break;
case 0x4e:
    AfxMessageBox("字母键N被按下");
    break;
case 0x4f:
    AfxMessageBox("字母键O被按下");
    break;
case 0x50:
    AfxMessageBox("字母键P被按下");
    break;
case 0x51:
    AfxMessageBox("字母键Q被按下");
    break;
case 0x52:
    AfxMessageBox("字母键R被按下");
    break;
case 0x53:
    AfxMessageBox("字母键S被按下");
    break;
case 0x54:
    AfxMessageBox("字母键T被按下");
    break;
case 0x55:
    AfxMessageBox("字母键U被按下");
    break;
case 0x56:
    AfxMessageBox("字母键V被按下");
    break;
case 0x57:
    AfxMessageBox("字母键W被按下");
    break;
case 0x58:
    AfxMessageBox("字母键X被按下");
    break;
case 0x59:
    AfxMessageBox("字母键Y被按下");
    break;
case 0x5a:
    AfxMessageBox("字母键Z被按下");
    break;
default:
    return;
}
CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

```
void CKeyMouseView::OnLButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    AfxMessageBox("鼠标左键被单击");
    CView::OnLButtonDown(nFlags, point);
}

void CKeyMouseView::OnMouseMove(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    CClientDC ClientDC( this );
    CString strInfo;
    strInfo.Format("鼠标位置: (%d,%d)",point.x, point.y);
    ClientDC.TextOut( 0, 0,strInfo, strInfo.GetLength() );
    CView::OnMouseMove(nFlags, point);
}

void CKeyMouseView::OnRButtonDown(UINT nFlags, CPoint point)
{
    // TODO: 在此添加消息处理程序代码和/或调用默认值
    AfxMessageBox("鼠标右键被单击");
    CView::OnRButtonDown(nFlags, point);
}
```

4. 编译链接运行程序

按 Ctrl+F5 键开始执行程序。

1.2.3 运行效果

程序运行后，在窗口内移动鼠标，将看到显示的鼠标在窗口中的位置坐标。单击鼠标左键，弹出一个消息框，如图 1.9 所示。随便按下字母键或者数字键，则弹出消息框，显示哪一个键被按下，如图 1.10 所示。

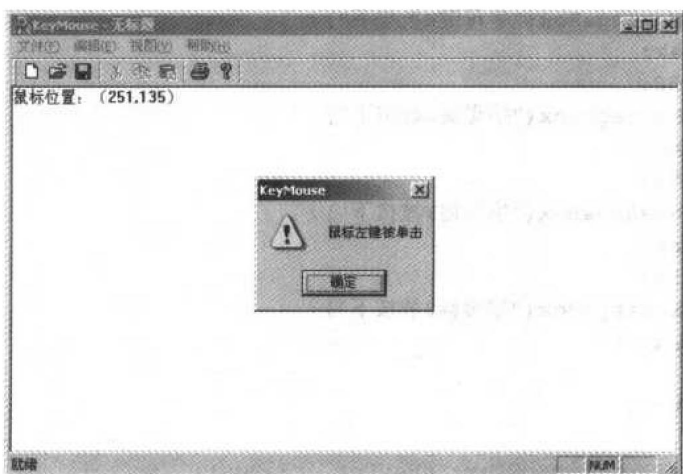


图 1.9 鼠标单击(即单击左键)时的运行效果



图 1.10 键盘字母键按下时的运行效果

实例 1.3 托管 C++编程

1.3.1 实例简介

从 Visual C++ .NET 开始, 标准 C++ 语言已经过扩展, 从而提供对托管编程的支持。C++ 托管扩展(也称为“托管 C++”)是为 Visual C++ 程序员开发 .NET 框架应用程序而设计的, 使用托管 C++ 意味着代码可以被“通用语言运行时”所管理, 并能开发出具有最新特性(如垃圾自动收集、程序间相互访问等)的 .NET 框架应用程序。使用托管 C++ 是 C++ 程序员编写 .NET 框架应用程序最好的一种选择, 通过集成在 Visual Studio.NET 开发环境的托管 C++ 向导, 可以创建几种开发 .NET 框架的项目类型。本例将学习如何利用应用程序向导创建托管 C++ 应用程序。

1.3.2 创建过程

1. 新建一个托管 C++ 应用程序

单击【文件】|【新建】|【项目】命令, 在【新建项目】对话框中, 选择【项目类型】为【Visual C++ 项目】, 选择【模板】为【托管的 C++ 应用程序】, 在【名称】文本框输入“Progress”, 如图 1.11 所示, 单击【确定】按钮, 完成程序框架的创建。

在【解决方案资源管理器】中打开程序文件 Progress.cpp, 可以看到以下程序代码:

```
// 这是使用应用程序向导生成 VC++  
// 应用程序项目的主项目文件。  
#include "stdafx.h"  
  
#using <mscorlib.dll>  
#include <tchar.h>  
  
using namespace System;
```