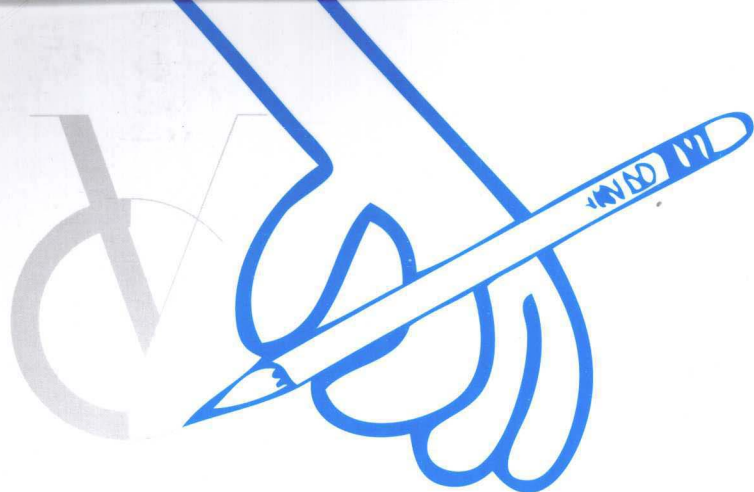




从零开始，手把手教会你用Visual C++编程

本书特色：

基础知识→核心技术→典型应用→综合练习→项目实践

107个典型实例、70个练习题、2个项目开发案例

超值、大容量DVD

17小时多媒体视频教学

本书源代码、本书教学PPT

赠送40小时相关知识多媒体视频

本书技术支持

QQ群：21948169

论坛：<http://www.rzchina.net>

• 从零开始学编程 •

从零
开始学



Visual C++

时多媒体教学视频

◎ 刘雪洁 刘永纯 等编著

循序渐进·由浅入深 内容充实·层次清楚 实例丰富·步骤清晰 对比讲解·理解深刻 习题指导·巩固学习 案例精讲·深入剖析

• 从零开始学编程 •

从零
开始学

Visual C++

◎ 刘雪洁 刘永纯 等编著

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

要想学好 Visual C++ 开发, 一本适合自己的入门书是非常重要的。本书充分考虑 Visual C++ 的难度, 合理安排章节, 由浅入深, 通过生动的范例程序和详细的代码注释, 带领读者掌握 Visual C++ 软件开发的技巧。

本书共分 16 章, 由浅入深, 循序渐进地介绍了 Visual C++ 编程的各个知识点。本书共分为 5 篇, 内容包括 C++ 基本语法、面向对象程序设计、Visual C++ 开发环境、常用控件、对话框、GDI 图形编程、单文档应用程序、视图风格、切分窗口、多文档应用程序、文件编程、数据库编程、DataGrid 控件、OpenGL 三维编程、五子棋游戏、公交换乘软件等。

本书配 DVD 光盘 1 张, 内容为本书的实例文件和作者专门为本书录制的全程多媒体语音教学视频。

本书内容全面, 论述翔实, 适合 Visual C++ 的初学者, 也可作为大、中专院校师生的培训教材, 对于 Visual C++ 爱好者, 本书也有很大的参考价值。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有, 侵权必究。

图书在版编目 (CIP) 数据

从零开始学 Visual C++ / 刘雪洁等编著. —北京: 电子工业出版社, 2011.2

(从零开始学编程)

ISBN 978-7-121-12247-7

I. ①从… II. ①刘… III. ①C 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 220700 号

责任编辑: 高洪霞

印 刷: 北京中新伟业印刷有限公司

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 20.25 字数: 488 千字

印 次: 2011 年 2 月第 1 次印刷

印 数: 5000 册 定价: 39.80 元 (含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。



C++有着极美好的未来，用它你能写出伟大的代码。

——Bjarne Stroustrup (C++之父)

Visual C++是 Windows 下的桌面软件开发利器，结合 C++语言的高效灵活和 MFC 框架的强大实用，开发者可以尽显才智，实现各类特定功能，如自己电脑桌面上的软件，大部分都是使用 Visual C++开发实现的。在学习 Visual C++的开发过程中，可以深入了解 Windows 程序的实现原理，为成为一名高级软件工程师打下坚实基础。

但 MFC 框架的复杂也是公认的，号称“最难学的开发框架”，事实上没有什么东西是好学的，Visual C++之难在于它将所有可用资源全部罗列出来，初学时茫然无措，不知从何处下手，但一旦学成之后，就身怀绝技，从此迈入高手行列。就个人发展而言，无须追赶技术潮流，如今 IT 技术发展之快令人望尘莫及，追也是追不上的，最紧要的事是练好内功，以不变应万变。Visual C++可谓开发工具中的“少林七十二绝技”，是修炼内功最好的一门技术，即便以后工作中用不上，也能从中受益匪浅。言尽于此，还望读者静下心来，认真读完这本书，相信会大有收获。

本书的特点

不可否认，Visual C++确实有一定的难度，事实上，难学的东西太多，没有最难，只有更难，比如高等数学、大学物理，但总有人能够把这些东西学得很精通，会者不难，只要坚持学习并找到科学的学习方法，以各位的聪明才智，相信没有搞不定的东西。

Visual C++之所以难学，最主要的原因在于没有好的教材，好的教材应该包含生动的文字描述信息和丰富实用的实例程序，不少教材给出大段代码后，却少有注释，或者提供的实例程序过于简单，没有实际意义，笔者深知读者需要什么，怎样讲解最利于读者学习，简而言之，一切为读者考虑。本书的主要特点如下：

- ❑ 讲解内容全面，涵盖了 Visual C++开发的各项基础知识，如对话框、单文档、多文档三大框架，以及文件、数据库等工作中常用的开发技术，集中篇幅深入讲解这些最实用的内容。
- ❑ 实例丰富，绝大部分章节以实例程序为核心，在实际应用中讲解要介绍的内容，如介绍树控件时，将树控件常用的添加、修改、删除、选中、勾选等功能集中到一个实例程序中，便于读者掌握最实用的技术。
- ❑ 代码注释丰富，对于实例代码，提供详细的代码注释，通过注释，读者可以了解每一句代码的意义，无须担心看不懂代码。
- ❑ 通俗易懂，将很多复杂的原理用通俗的话讲出来，原理就简单了，方便读者了解。

① 知识点介绍 准确、清晰是其显著特点，一般放在每一节开始位置，让零基础的读者了解相关概念，顺利入门。

② 实例 书中出现的完整实例，以章节顺序编号，便于检索和循序渐进地学习、实践，放在每节知识点介绍之后。

③ 实例代码 与实例编号对应，层次清楚、语句简洁、注释丰富，体现了代码优美的原则，有利于读者养成良好的代码编写习惯。

14 从零基础学 Visual C++

第1章 C++基本语法 15

1.7 指针

指针是C和C++语言中的重要因素，但是也是难以控制的不稳定因素。在C#、Java语言中都取消了指针类型，不能直接操作物理内存，指针是变量的内存地址，不同类型的变量需要不同的指针类型，尽管指针的实际值是一个数字，但用了存储指针值的变量必须明确声明为指针类型。

1.7.1 指针概述

指针存储变量的内存地址，指针类型为变量类型加*，如int* p1；若p1设定为指向int类型的指针，则不能接受其他类型的指针，&用于获取变量的内存地址，如int* p1=&a，a获取指针指向的地址，使用如int b=*p1；在不同的场合*有不同的意义，应小心区分。

指针变量存储有效的地址值，和通常数据值一样，也是一种数据类型，在宏定义中，经常完成通过指针类型的转换，如int*转为void*，void*转为double*，就生类指针转为基类指针等，这些转换都是编译层次上的表面转换，实际上就是一个地址。

【实例1-25】输出指针变量存储的地址值，指针指向的值，指针本身所在的地址，指针变量占用的内存大小。

```
#include <iostream>
using namespace std;

int main()
{
    int a=123; //整型变量
    int* p=&a; //将a的地址赋给指针变量p
    cout<<"指针p的地址为:"<<"<<p<<endl; //p存储地址
    cout<<"p指向的地址为:"<<"<<*p<<endl; //p指向的地址
    int** q=&p; //取得指针p的指针
    cout<<"指针q的地址:"<<"<<q<<endl; //指针q所指向的内存地址
    cout<<"指针q指向的地址:"<<"<<*q<<endl; //指针q指向的地址
    return 0;
}
```

编译运行，结果如图1-20所示。通过&获取整型变量a的地址值，存放到int指针p中，通过*获取指针p所指向的变量的值，指针作为一个变量，也有其地址，通过&获取指针变量的地址值，存放到q中，int**表示一个指向int指针变量的地址，通过sizeof获取指针变量占用的内存大小，指针变量占用4个字节。

1.7.2 指针与数组

数组变量实际上就是一个指针，数组名为数组的第1个元素地址，即数组的首地址。根据下标索引确定元素地址，如a[i]的地址等于a[0]的地址加上i个元素的大小。

尽管指针实际上是个地址值，但不能把指针当做值进行数字运算，只能按照编译器设定的方式进行运算，如a[2]等同于*(a+2)，数组名a为数组首地址，2表示两个元素的大小，a[2]即为第3个元素的地址，通过*获取第3个元素的值。

【实例1-26】将数组名赋给指针变量，用指针变量访问数组元素，输出每个元素的值和地址值。

```
#include <iostream>
using namespace std;

int main()
{
    int a[5]={12,45,60};
}
```

1.7.4 指针与字符串

字符串可看做是字符串数组，如字符串"language"是长度为9的字符串数组，前8个元素存储单个字符，最后1个为结束标志"0"，表明字符串到此结束。字符串的ASCII值为0，可根据元素值是否等于0来判断是否到达字符串末尾。

个字符串数组能存放的字符串长度为数组长度减1，字符串初始化时，C++自动在字符串末尾添加"0"字符"0"，字符串数组名相当于一个指向字符串首地址的指针，通过移动指针指向的位置可完成编辑一个字符串。

【实例1-28】通过字符串指针遍历一个字符串数组，输出所有字符。

```
#include <iostream>
using namespace std;

int main()
{
    char a[]="Study Program"; //字符串数组
    char* p=a; //字符串指针，指向字符串首地址
    while(*p!='\0') //在字符串数组中不等于'\0' (ASCII 0)
    {
        cout<<"p="<<"<<p<<endl; //输出字符串
        p++; //字符串指针向后移动1个字节
    }
    cout<<endl;
    return 0;
}
```

编译运行，结果如图1-23所示。字符串数组名a同于字符串数组的首地址，将字符串数组名赋给字符串指针p后，p指向第1个元素，通过*获取p指向的元素的值，p每次递增移动一个元素大小的位置，指向下一个元素，当指向字符串结束标志"0"时，停止循环。

1.8 小结

本章讲述了C++的基本语法，Visual C++是C++的可视化开发环境，由于MFC类库采用C++语言编写，所以学习Visual C++时，掌握C++是关键。本章主要介绍了C++语言的基本数据类型、运算符、控制结构、函数、数组和指针，其中函数、数组和指针较为复杂，希望读者认真掌握，为后面Visual C++程序的编写打下坚实的基础。

1.9 习题

1. C++中基本数据类型有哪些？
2. 函数的具体构成是什么？如何定义一个函数？
3. 编写一个程序，输出"你好，中国"。
4. 什么是数组？如何定义一个二维数组？

Tips 位运算符常用于参数属性的标志位，如一个标志参数用二进制表示为10010001，每个位代表一种特定的属性，位为1表示使用该属性所代表的属性，若为0表示不使用。如用CFire类调用Open函数打开文件时指定CFire::modeCreate|File::modeWrite两个标志，用位或运算符相连，则这两个标志所对应的值为1；其余位都为0。

- ④ 运行结果 对实例给出运行结果和对应图示，帮助读者更直观地理解实例代码。
- ⑤ 习题 每章最后提供专门的测试习题，供读者检验所学知识是否牢固掌握。
- ⑥ 贴心的提示 为了便于读者阅读，全书还穿插着一些技巧、提示等小贴士，体例约定如下：

提示：通常是一些贴心的提醒，让读者加深印象或提供建议，或者解决问题的方法。

注意：提出学习过程中需要特别注意的一些知识点和内容，或者相关信息。

警告：对操作不当或理解偏差将会造成的灾难性后果做警示，以加深读者印象。

经作者多年的培训和授课证明，以上讲解方式是最适合初学者学习的方式，读者按照这种方式，会非常轻松、顺利地掌握本书知识。

2. 实用超值的DVD光盘

为了帮助读者比较直观地学习，本书附赠DVD光盘，内容包括多媒体视频、电子教案（PPT）和实例源代码等。

● 多媒体视频

配有长达17小时手把手教学视频，讲解关键知识点界面操作和书中的一些综合练习题。作者亲自配音、演示，手把手教会读者使用。

从零基础学 Visual C++

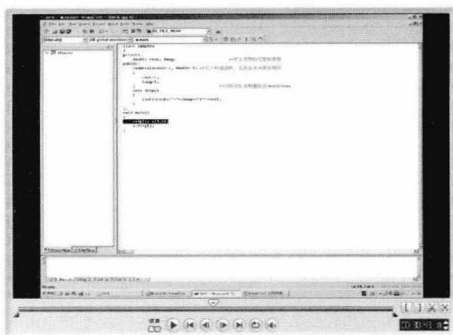
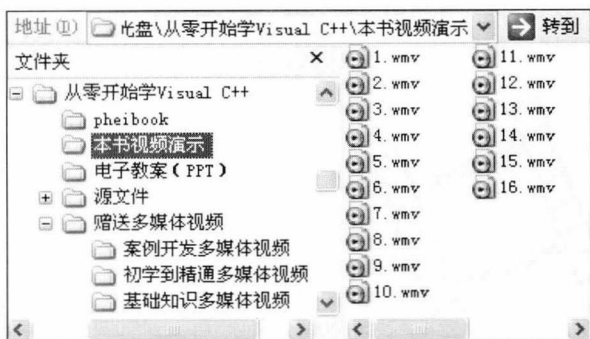
pheiBOOK

本书视频演示

电子教案（PPT）

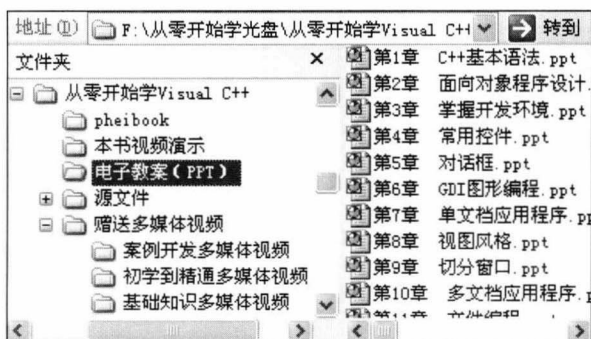
源文件

赠送多媒体视频



● 电子教案 (PPT)

本书可以作为高校相关课程的教材或课外辅导书，所以笔者特别为本书制作了电子教案 (PPT)，以方便老师教学使用。



3. 提供完善的技术支持

本书提供了论坛：<http://www.rzchina.net>，读者可以在上面提问交流。另外，论坛上还有一些小的教程、视频动画和各种技术文章，可帮助读者提高开发水平。

4. 丰富的额外素材下载

相关的开发素材文件，在 www.broadview.com.cn 提供下载。

适合阅读本书的读者

- ☐ 本书适合 Visual C++开发自学者；
- ☐ 软件开发人员；
- ☐ 大中专院校相关专业的学生。

本书作者

本书主要由吉林大学计算机科学与技术学院的刘雪洁和黑龙江省直属机关党校的刘永纯编写。其中，第 1~5 章和第 11~16 章主要由刘雪洁编写，第 6~10 章主要由刘永纯编写。其他参与本书编写的人员有昊燃、曾光、张双、朱照华、黄永湛、孟祥嵩、张贺军、李勇、关涛、王岩、李晓白、魏星、刘蕾。在此一并表示感谢！

目 录

第 1 篇 Visual C++基础入门

第 1 章 C++基本语法 ( 教学视频: 61 分钟)	15
1.1 了解 C++程序设计	15
1.1.1 学习 C++的好处	15
1.1.2 最简单的 C++程序——hello world	15
1.2 基本数据类型	16
1.2.1 整型 int	16
1.2.2 浮点型 float、double	17
1.2.3 字符型 char	17
1.2.4 布尔型 bool	18
1.2.5 宽字符型 wchar_t	18
1.3 运算符	18
1.3.1 算术运算符	19
1.3.2 关系运算符	19
1.3.3 赋值运算符	19
1.3.4 自增自减运算符	20
1.3.5 逻辑运算符	21
1.3.6 位运算符	21
1.4 控制结构	22
1.4.1 if/else 选择结构	22
1.4.2 while 循环结构	23
1.4.3 break 和 continue 语句	23
1.4.4 for 循环结构	24
1.4.5 switch 多选结构	24
1.5 函数	25
1.5.1 什么是函数	25
1.5.2 定义函数	25
1.5.3 变量作用域	25
1.5.4 使用函数	26
1.5.5 函数重载	27
1.6 数组	27
1.6.1 什么是数组	27
1.6.2 一维数组	27
1.6.3 二维数组	28
1.6.4 动态数组	29

1.6.5	数组排序	30
1.7	指针	31
1.7.1	指针概述	31
1.7.2	指针与数组	31
1.7.3	指针与函数	32
1.7.4	指针与字符串	33
1.8	小结	33
1.9	习题	33
第 2 章	面向对象程序设计 ( 教学视频: 57 分钟)	34
2.1	类和对象	34
2.1.1	类和对象的关系	34
2.1.2	定义类	34
2.1.3	构造函数	35
2.1.4	析构造函数	36
2.1.5	内联函数	37
2.1.6	static 成员	38
2.1.7	const 成员	40
2.1.8	友元	41
2.2	运算符重载	42
2.2.1	了解运算符重载	43
2.2.2	一元重载	43
2.2.3	二元重载	44
2.3	继承性	45
2.3.1	类的继承	45
2.3.2	访问控制	47
2.3.3	调用流程	47
2.4	多态性	48
2.4.1	多态性的实现	48
2.4.2	virtual 虚函数	49
2.4.3	抽象类	49
2.5	模板	50
2.5.1	如何定义模板	50
2.5.2	模板类	51
2.5.3	标准模板库 STL	52
2.6	异常处理	53
2.6.1	处理程序异常	54
2.6.2	自定义异常类	54
2.7	小结	55
2.8	习题	55
第 3 章	掌握开发环境 ( 教学视频: 41 分钟)	56
3.1	创建运行程序	56
3.1.1	Win32 程序	56

3.1.2	对话框程序	58
3.1.3	单文档程序	59
3.1.4	多文档程序	61
3.2	开发界面	62
3.2.1	菜单	62
3.2.2	工具条	65
3.2.3	类视图	66
3.2.4	资源视图	66
3.2.5	文件视图	66
3.2.6	类向导	66
3.2.7	输出窗口	67
3.3	使用技巧	68
3.3.1	添加类	68
3.3.2	添加类成员函数	68
3.3.3	添加类成员变量	69
3.3.4	添加消息处理函数	69
3.3.5	重写虚函数	70
3.3.6	添加资源	70
3.3.7	添加已有文件和控件	71
3.3.8	设置代码字体样式	71
3.4	小结	72
3.5	习题	72

第 2 篇 可视化编程

第 4 章	常用控件 ( 教学视频: 139 分钟)	73
4.1	了解生成类	73
4.2	静态文本	75
4.2.1	设置属性	75
4.2.2	更新内容	76
4.3	编辑框	77
4.3.1	设置属性	77
4.3.2	数据交换	78
4.4	按钮	82
4.4.1	设置属性	82
4.4.2	消息响应	82
4.5	单选按钮	84
4.5.1	设置属性	84
4.5.2	消息响应	85
4.6	复选按钮	85
4.6.1	设置属性	86
4.6.2	消息响应	86
4.7	组合框	87

4.7.1	设置属性	87
4.7.2	编辑项	88
4.7.3	消息响应	89
4.7.4	添加图像	91
4.8	列表框	93
4.8.1	设置属性	93
4.8.2	编辑项	94
4.8.3	消息响应	95
4.9	进度条	97
4.9.1	设置属性	97
4.9.2	更新值	97
4.10	滑块	99
4.10.1	设置属性	100
4.10.2	消息响应	100
4.11	列表控件	101
4.11.1	设置属性	101
4.11.2	编辑项	102
4.11.3	消息响应	104
4.11.4	添加图像	106
4.12	树控件	106
4.12.1	设置属性	106
4.12.2	编辑项	107
4.12.3	消息响应	109
4.13	日期控件	114
4.13.1	设置属性	114
4.13.2	读取设置日期	115
4.13.3	日期响应	115
4.14	高级控件	116
4.14.1	Windows Media Player 控件	116
4.14.2	Flash 控件	118
4.15	小结	119
4.16	习题	119
第 5 章	对话框 ( 教学视频: 20 分钟)	120
5.1	模态对话框	120
5.1.1	添加对话框资源	120
5.1.2	添加对话框类	121
5.1.3	初始化对话框	121
5.1.4	显示模态对话框	122
5.2	非模态对话框	124
5.3	小结	126
5.4	习题	126

第3篇 Visual C++的应用

第6章 GDI 图形编程 ( 教学视频: 64 分钟)	127
6.1 设备环境	127
6.1.1 什么是设备环境	127
6.1.2 设备环境分类	127
6.2 图形绘制	129
6.2.1 点线	129
6.2.2 多边形	132
6.2.3 文本	135
6.3 画笔	137
6.3.1 创建画笔	137
6.3.2 使用画笔	138
6.4 画刷	140
6.4.1 创建画刷	140
6.4.2 使用画刷	141
6.5 字体	143
6.5.1 创建字体	144
6.5.2 使用字体	144
6.6 映射模式	145
6.6.1 了解映射模式	145
6.6.2 窗口和视口	147
6.7 小结	150
6.8 习题	150
第7章 单文档应用程序 ( 教学视频: 206 分钟)	151
7.1 了解生成类	151
7.1.1 App 类	151
7.1.2 Doc 类	152
7.1.3 View 类	156
7.1.4 Frame 类	158
7.1.5 类联系方式	160
7.2 菜单	161
7.2.1 添加菜单资源	161
7.2.2 更新菜单	162
7.2.3 禁用和勾选菜单	167
7.2.4 右键菜单	168
7.3 工具栏	171
7.3.1 添加工具栏资源	171
7.3.2 显示工具栏	172
7.3.3 添加按钮处理函数	173
7.4 状态栏	179
7.4.1 设置分区	179

7.4.2	更新内容	182
7.5	对话框	184
7.5.1	添加对话框资源	184
7.5.2	显示对话框	185
7.5.3	添加控件处理函数	185
7.6	文档视图	189
7.6.1	文档类存取数据	189
7.6.2	视图类显示数据	191
7.7	小结	193
7.8	习题	193
第 8 章	视图风格 ( 教学视频: 20 分钟)	194
8.1	Edit 视图	194
8.2	List 视图	195
8.3	Tree 视图	197
8.4	RichEdit 视图	198
8.5	小结	201
8.6	习题	201
第 9 章	切分窗口 ( 教学视频: 26 分钟)	202
9.1	了解窗口切分	202
9.2	静态切分窗口	203
9.3	多视图切换	206
9.4	小结	210
9.5	习题	210
第 10 章	多文档应用程序 ( 教学视频: 25 分钟)	211
10.1	了解生成类	211
10.2	类联系方式	214
10.3	多文档视图	215
10.3.1	添加文档模板	215
10.3.2	更新视图	216
10.4	小结	217
10.5	习题	217

第 4 篇 Visual C++编程

第 11 章	文件编程 ( 教学视频: 76 分钟)	218
11.1	文件类	218
11.1.1	文件格式	218
11.1.2	文件对话框	219
11.1.3	文件操作	222
11.1.4	文件状态	224
11.1.5	读/写文本文件	225
11.1.6	读/写二进制文件	226

11.2	文件查找	229
11.3	文件序列化	233
11.3.1	如何实现序列化	233
11.3.2	创建可序列化类	234
11.3.3	序列化对象	236
11.4	小结	240
11.5	习题	241
第 12 章	数据库编程 ( 教学视频: 94 分钟)	242
12.1	了解数据库	242
12.1.1	安装 SQL Server 2000	242
12.1.2	企业管理器	243
12.1.3	查询分析器	245
12.1.4	数据查询语言	246
12.1.5	数据更新语言	247
12.1.6	ADO 数据库访问技术	248
12.2	ADO 封装类	249
12.2.1	类头文件定义	249
12.2.2	数据库连接函数	250
12.2.3	SQL 命令函数	252
12.2.4	相关辅助函数	255
12.3	ADO 访问数据库	258
12.3.1	连接数据库	258
12.3.2	添加记录	263
12.3.3	更新记录	266
12.3.4	删除记录	267
12.3.5	导出记录	267
12.4	小结	268
12.5	习题	268
第 13 章	DataGrid 控件 ( 教学视频: 30 分钟)	269
13.1	添加 DataGrid 控件	269
13.2	读取 Excel 数据表	270
13.3	添加删除数据	272
13.4	计算并更新数据	274
13.5	小结	276
13.6	习题	276
第 14 章	OpenGL 三维编程 ( 教学视频: 44 分钟)	277
14.1	了解 OpenGL	277
14.1.1	OpenGL 三维绘图	277
14.1.2	OpenGL 库文件	278
14.2	MFC 框架下使用 OpenGL	279
14.2.1	创建 MFC 框架	279

14.2.2	使用 OpenGL	280
14.2.3	读取坐标文件数据	282
14.2.4	绘制三维图形	283
14.2.5	鼠标交互式浏览	286
14.3	小结	287
14.4	习题	287

第 5 篇 案例篇

第 15 章	五子棋游戏 ( 教学视频: 6 分钟)	288
15.1	界面设计	288
15.2	算法设计	289
15.3	功能实现	291
15.4	小结	296
第 16 章	公交换乘软件 ( 教学视频: 72 分钟)	297
16.1	数据库设计	297
16.2	界面设计	298
16.3	算法设计	299
16.3.1	直达路线	299
16.3.2	一次换乘	299
16.3.3	两次换乘	300
16.4	智能提示编辑框	300
16.5	功能实现	304
16.6	小结	312
附录 A	Win32 API 开发	313
附录 B	程序调试技巧	319

第 1 篇 Visual C++基础入门

第 1 章 C++基本语法

程序设计语言经历了四十多年的发展,从传统的汇编、C、C++、Visual Basic 到现今的 C#、Java,以及其他各类编程语言共有百余种,数目之多令人眼花缭乱。C++作为众多编程语言中的经典之作,于 20 世纪 80 年代由 AT&T 贝尔实验室的 Bjarne Stroustrup 设计而成。C++兼容 C 语言,并具备面向对象的能力,是一种适合编写大型软件的高级语言。Visual C++是 C++的可视化开发环境,其核心 MFC 类库采用 C++语言编写,C++的面向对象特性充分体现在 MFC 类库里,在学习 Visual C++之前,必须掌握基本的 C++语法。

1.1 了解 C++程序设计

C++是一门同时具备面向过程和对象特性的语言,学习 C++语言既可以了解底层 API 函数的使用方法、控制变量的内存生存周期,编写功能强大高效的算法,也可以从对象的角度组织各个模块,掌握面向对象的编程思想。在学习 C++之后能够对程序设计有深刻的体会,为进一步学习打下坚实基础。

1.1.1 学习 C++的好处

在软件技术快速更新的今天,初学者面临的第一个问题就是:我该学哪种语言。刚接触编程的初学者容易被 ASP.NET、Java、Visual Basic、PHP 等名字弄得晕头转向,诚然这是一个让开发人员疯狂的年代,曾经引以为豪的技术可能不久就被新技术替代,或者由于公司项目需要不得不从自己擅长的技术平台转向新平台。心中不免思考:究竟什么东西是一劳永逸的?答案是没有,没有一门技术可以解决所有问题,但有一样东西可以帮助你快速转入其他相关领域,那就是编程的思想。

所谓编程思想,即从程序的角度思考解决问题,了解程序代码的运行原理、变量的生存周期、窗口的诞生与销毁、消息的传递流程等。当深入了解这些底层原理后,再去学习任何一门开发技术,都会有居高临下之感,从而快速转型。

和 C#、Java 等托管代码不同,C++可以直接操作物理内存,自由控制变量的生存周期,学习 C++你可以了解程序运行时,哪些要占用内存、内存分配方式、变量内存布局、内存释放、内存泄漏等其他语言无法涉及的深层内容。

C++面向对象的特性,可以让你了解如何构造类、创建类对象、释放类对象、类继承机制、虚函数实现原理等,C#、Java 都是由 C++演变而来的,学习 C++可以掌握语言最核心的内容,其他语言则可触类旁通。事实证明,有 C++背景的程序员学习其他语言周期比较短,而且理解得更加深刻,记住,真正的程序员学习 C++!

1.1.2 最简单的 C++程序——hello world

【实例 1-1】新建一个控制台程序 hello world,输出一段文字。

(1)启动 VC 6.0,选择 File/New 命令,打开 New 窗口,在 Projects 选项卡里选择 Win32 Console Application 项(Win32 控制台应用程序),在 Project name 文本框中输入 hello world。在 Location 文本框里选择工程存放路径,如图 1-1 所示。

(2) 单击 OK 按钮, 打开 Win32 Console Application 窗口, 选择 An empty project 项 (空工程), 单击 Finish 按钮, 再单击 OK 按钮, 完成工程的创建。

(3) 选择 FileNew 命令, 打开 New 窗口, 在 Files 选项卡里选择 C++ Source File 项 (C++ 源文件), 在 File 文本框中输入 hello world, 如图 1-2 所示。

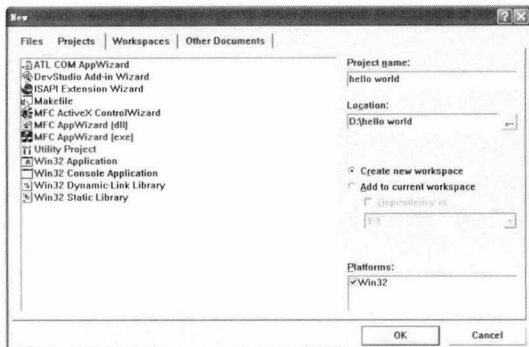


图 1-1 新建工程窗口

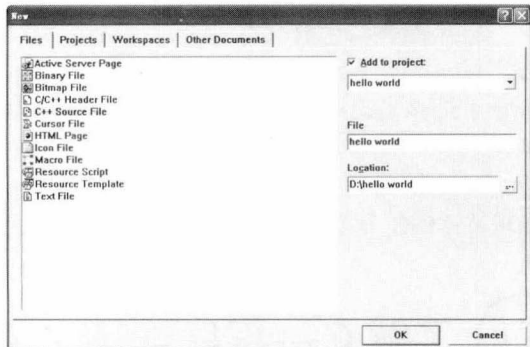


图 1-2 添加 C++源文件

(4) 单击 OK 按钮后, 在工作区窗口选择 FileView 标签, 展开 Source Files 节点, 双击添加的源文件 main.cpp, 打开代码编辑窗口, 输入以下代码:

```
#include <iostream>           //包含输入/输出流头文件
using namespace std;         //使用标准命名空间 std
int main()                   //主函数
{
    cout<<"hello world"<<endl;
    return 0;                //返回 0
}
```

(5) 按 F7 键或单击  按钮, 生成 EXE 可执行程序, 单击  按钮执行 EXE 程序, 结果如图 1-3 所示。

```
hello world
Press any key to continue
```

图 1-3 输出 hello world

1.2 基本数据类型

C++ 是一种数据类型严格的语言, 计算机根据数据类型分配对应大小的内存空间, C++ 内置有几种基本的数据类型, 掌握这些基本类型后, 可以通过结构体 (struct) 和类 (class) 创建自定义类型。

1.2.1 整型 int

int 类型用来存放整数 (integer), 一个 int 类型变量通常占用 4 个字节 (byte), 具体大小在不同平台上可能不同。计算机使用二进制方式存储数据, 类似 10001000, 一个 0 或 1 占用 1 位 (bit), 一个字节占用 8 位, int 类型变量占用 32 位, 范围为 $-2^{31} \sim 2^{31}$ 。

修饰符 short 表示短整型, 如 short int a=123; 占用 2 个字节, 范围为 $-2^{15} \sim 2^{15}$, short 等同于 short int。修饰符 long 表示长整型, 如 long int a=12345678; 占用 4 个字节, long 等同于 long int。

默认 int 类型有正负符号, 使用修饰符 unsigned 表示无符号, 如 unsigned int a=12345678; 范围为 $0 \sim 2^{32}$, 无符号修饰符 unsigned 对 float、double、char 同样适用。

【实例 1-2】 输出未赋值和赋值后的整型变量的值。

```
#include <iostream>
using namespace std;
int main()
```