

面向 21 世纪

高等学校计算机类专业规划教材

Visual C++ 基础教程

A First Course in Visual C++

郭文平 王跃存 杨晓光 编 著

胡建伟 主 审

西安电子科技大学出版社

<http://www.xduph.com>



面向 21 世纪高等学校计算机类专业规划教材

Visual C++基础教程

郭文平 王跃存 杨晓光 编著

胡建伟 主 审

西安电子科技大学出版社

2006

内 容 简 介

本书全面、系统地介绍了 Visual C++ 面向对象程序设计的基础知识和 MFC Windows 程序设计的主要基础内容。

全书共 13 章, 分为两个部分。前 6 章为 C++ 语言基础部分, 内容包括: C++ 概述, C++ 面向过程的程序设计, 类和对象, 继承和派生类, 多态性和虚函数以及模板。这部分对 C++ 兼容 C 语言的内容只作了简单的总结, 而着重于对面向对象程序设计的介绍。后 7 章为 MFC Windows 程序设计部分, 内容包括: Windows 编程基础, 创建应用程序框架, 对话框和控件, 图形输出, 菜单、工具栏和状态栏, 文档/视图和单文档界面以及文件的读/写。

本书可作为高等院校各相关专业“面向对象程序设计”课程的教材, 也可作为 Visual C++ 初、中级读者的自学教材或培训教材。本书适合具有一定 C 语言基础的读者使用。

★本书配有电子教案, 需要者可与出版社联系, 免费提供。

图书在版编目 (CIP) 数据

Visual C++ 基础教程 / 郭文平等编著. —西安: 西安电子科技大学出版社, 2006.9

(面向 21 世纪高等学校计算机类专业规划教材)

ISBN 7-5606-1738-7

I. V… II. 郭… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 106173 号

策 划 毛红兵

责任编辑 阎 彬 毛红兵

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

http: //www.xduph.com

E-mail: xdupfbx@pub.xaonline.com

经 销 新华书店

印刷单位 陕西天意印务有限责任公司

版 次 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 23.75

字 数 560 千字

印 数 1~4000 册

定 价 29.00 元

ISBN 7-5606-1738-7/TP·0432

XDUP 2030001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜, 谨防盗版。

C++ 语言继承了 C 语言的全部优点和功能,同时它还支持面向对象程序设计。通过 C++ 语言的学习,可以深刻理解和领会面向对象程序设计的精髓和程序设计思路。因此,目前许多高等院校都将 C++ 语言作为“面向对象程序设计”课程的首选语言。

学习一门程序设计语言,除了需要掌握语言本身的语法规则和使用特点外,更重要的是要掌握程序设计的方法和思路并利用它们解决实际的问题。目前,C++ 面向对象程序设计方面的教材大多注重 C++ 基础,主要介绍 C++ 语言的基本概念和知识以及面向对象程序设计的方法,缺少 C++ 语言的具体应用。因而出现了这样一种现象:学生学习完 C++ 后感觉 C++ 比 C 语言并没有什么大的优势。而本书最大的特点是注重实用性。

本书在 C++ 基础和 MFC Windows 程序设计间取得了较好的平衡。本书首先全面、详细地介绍了 C++ 语言的主要内容,对实际应用中很少使用的内容进行了删除,着重于 C++ 语言的基础,为实际应用做准备。然后在 MFC Windows 程序设计部分介绍了 Windows 程序设计的原理,并对 MFC Windows 程序设计的主要基础内容作了详细介绍,着重于 MFC 的应用。对于初学 Windows 程序设计的读者来说,程序设计和实现的步骤非常重要,因此,在介绍 MFC Windows 程序设计时,本书不仅给出了实现程序功能的代码,而且详细说明了程序的实现过程和思路。

为了突出重点,作者对书中用到的实例进行了一些简化。

本书共 13 章,分为两个部分。前 6 章为 C++ 语言基础部分,系统讲述了面向对象程序设计的概念和特点、C++ 语言的基础知识和基本规则以及编程方法。后 7 章为 MFC Windows 程序设计部分,内容包括 Windows 编程基础,创建应用程序框架,对话框和控件,图形输出,菜单、工具栏和状态栏,文档/视图和单文档界面以及文件的读/写。

本书第 1、7~13 章以及附录由郭文平编写,第 2、3 章由王跃存编写,第 4、5、6 章由杨晓光编写。全书由郭文平主编并统稿。

本书可作为高等院校各相关专业“面向对象程序设计”课程的教材,也可作为 Visual C++ 初、中级读者的自学教材或培训教材。本书适合具有一定 C 语言基础的读者使用。

在本书编写过程中得到了天津工业大学信息与通信工程学院领导和教师的大力支持,祖晓媛在书稿的校对中做了大量的工作,在此向他们表示衷心的感谢。亦向本书所列参考文献的作者表示感谢。

由于作者水平有限,加之时间仓促,书中错误在所难免,恳请读者批评指正。

作 者

于天津工业大学信息与通信工程学院

2006.6

目 录

第一篇 C++语言基础

第 1 章 C++概述.....	3
1.1 C++简史和特点.....	3
1.1.1 C++简史.....	3
1.1.2 C++的特点.....	3
1.2 C++程序的基本结构.....	4
1.2.1 C++程序实例.....	4
1.2.2 C++程序的基本结构.....	5
1.3 C++基本语法成分.....	5
1.3.1 字符集.....	5
1.3.2 关键字.....	6
1.3.3 标识符.....	6
1.3.4 运算符和表达式.....	7
1.3.5 注释.....	8
1.4 数据的输入和输出.....	8
1.4.1 屏幕显示输出.....	9
1.4.2 键盘输入.....	10
1.4.3 输入/输出格式控制.....	11
1.5 Visual C++中 C++程序的实现.....	13
1.5.1 Visual C++的集成开发环境.....	14
1.5.2 C++程序的实现.....	14
1.5.3 程序调试.....	16
习题.....	18
第 2 章 C++面向过程的程序设计.....	21
2.1 数据类型.....	21
2.1.1 基本数据类型的取值范围.....	22
2.1.2 C++的扩展.....	22
2.2 常量和变量.....	24
2.2.1 常量.....	24
2.2.2 变量.....	26
2.3 基本控制结构.....	28
2.3.1 顺序结构.....	28
2.3.2 选择结构.....	29

2.3.3 循环结构	33
2.3.4 流程的转移控制	35
2.4 数组和指针	36
2.4.1 数组	36
2.4.2 指针	38
2.4.3 指针与数组的关系	40
2.4.4 动态内存分配	41
2.5 函数	43
2.5.1 函数的定义和调用	43
2.5.2 函数原型	44
2.5.3 内联函数	45
2.5.4 函数参数的默认值	46
2.5.5 函数重载	47
2.5.6 引用	49
习题	53
第3章 类和对象	57
3.1 面向对象程序设计概述	57
3.1.1 面向对象程序设计的产生	57
3.1.2 面向对象程序设计的基本概念和特征	57
3.2 类与对象	59
3.2.1 类的定义	59
3.2.2 成员函数的定义	61
3.2.3 对象的定义与使用	62
3.2.4 内联成员函数	65
3.2.5 成员函数的重载和参数的默认值	66
3.3 构造函数和析构函数	68
3.3.1 构造函数	68
3.3.2 缺省构造函数	72
3.3.3 拷贝构造函数	73
3.3.4 析构函数	75
3.3.5 构造函数的类型转换和类型转换函数	78
3.4 对象数组和对象指针	82
3.4.1 对象数组	82
3.4.2 对象指针	85
3.4.3 this 指针	86
3.5 静态成员	88
3.5.1 静态数据成员	88
3.5.2 静态成员函数	89
3.6 友元	91

3.6.1 友元函数	91
3.6.2 友元类	94
3.7 对象成员	96
习题	98
第4章 继承和派生类	104
4.1 基类和派生类	104
4.1.1 继承	104
4.1.2 派生类的定义	105
4.1.3 派生类对基类的扩充	107
4.2 继承方式	107
4.2.1 基类成员在派生类中的访问属性	108
4.2.2 派生类对象对基类成员的访问	108
4.3 派生类的构造函数和析构函数	114
4.3.1 构造函数	114
4.3.2 析构函数	117
4.4 多重继承	117
4.4.1 多重继承的定义	117
4.4.2 多重继承派生类的构造函数	120
4.4.3 二义性	122
4.5 虚基类	126
4.5.1 虚基类的概念	126
4.5.2 虚基类的初始化	128
4.6 赋值兼容规则	130
习题	132
第5章 多态性和虚函数	135
5.1 函数重载	135
5.2 运算符重载	137
5.2.1 运算符重载的规则	137
5.2.2 类以外运算符重载	138
5.2.3 运算符重载为成员函数	140
5.2.4 运算符重载为友元函数	145
5.3 虚函数	147
5.3.1 虚函数的定义	148
5.3.2 虚析构函数	151
5.4 纯虚函数和抽象类	153
5.4.1 纯虚函数	153
5.4.2 抽象类	153
习题	155

第 6 章 模板	160
6.1 函数模板	160
6.2 类模板	166
习题	170

第二篇 MFC Windows 程序设计

第 7 章 Windows 编程基础	173
7.1 Windows 编程基础知识	173
7.1.1 Windows 编程模型	173
7.1.2 Windows 编程基础知识	175
7.2 Windows 应用程序的基本结构	178
7.2.1 实例	178
7.2.2 Windows 应用程序的基本结构	181
7.3 MFC 程序设计基础	185
7.3.1 MFC 概述	186
7.3.2 MFC Windows 程序的基本结构	188
习题	194
第 8 章 创建应用程序框架	195
8.1 应用程序向导 AppWizard	195
8.1.1 MFC AppWizard[exe]的使用	195
8.1.2 基于框架构造应用程序	202
8.2 AppWizard 生成的类和源文件	202
8.2.1 应用程序类	202
8.2.2 框架窗口类	203
8.2.3 文档类	203
8.2.4 视图类	203
8.2.5 对话框类	204
8.2.6 其它文件	204
8.3 项目和项目工作区	205
8.3.1 项目	205
8.3.2 项目工作区	205
8.3.3 项目工作区窗口的使用	206
8.4 ClassWizard	208
8.4.1 添加消息处理函数	209
8.4.2 编辑消息处理函数	210
8.4.3 删除消息处理函数	211
8.4.4 重载虚函数	212
8.4.5 为项目添加新类	212
习题	213

第 9 章 对话框和控件	214
9.1 对话框类 CDialog	214
9.2 基于对话框的应用程序	217
9.2.1 简单应用程序实例	217
9.2.2 理解应用程序框架	223
9.2.3 对话框数据交换和验证	225
9.3 模态对话框与非模态对话框	226
9.3.1 模态对话框	227
9.3.2 非模态对话框	232
9.4 标准控件	237
9.4.1 控件通用属性	237
9.4.2 静态文本框	239
9.4.3 编辑框	239
9.4.4 命令按钮	240
9.4.5 单选按钮	241
9.4.6 复选框	241
9.4.7 分组框	242
9.4.8 列表框	245
9.4.9 组合框	249
9.4.10 滚动条	252
9.5 公用对话框	255
习题	258
第 10 章 图形输出	260
10.1 图形设备接口	260
10.1.1 设备环境	260
10.1.2 设备环境的属性	263
10.1.3 绘图模式	265
10.1.4 映射模式与坐标转换	266
10.1.5 颜色设置	271
10.2 画笔和画刷	273
10.2.1 画笔	273
10.2.2 画刷	276
10.3 绘图	279
10.3.1 绘制点、直线和曲线	279
10.3.2 画封闭图形	280
10.4 文本和字体	282
10.4.1 文本输出	282
10.4.2 设置文本的设备环境属性	283
10.4.3 获取字体信息	284

10.4.4 字体	287
习题	292
第 11 章 菜单、工具栏和状态栏	294
11.1 菜单	294
11.1.1 菜单基础	294
11.1.2 创建菜单	295
11.1.3 添加菜单命令处理函数	297
11.1.4 更新菜单项	300
11.1.5 键盘快捷键	301
11.1.6 快捷菜单	302
11.1.7 动态修改菜单	303
11.1.8 用代码创建菜单	303
11.1.9 使用菜单资源	307
11.2 工具栏	309
11.2.1 工具栏编辑器	309
11.2.2 创建工具栏	311
11.2.3 停靠和浮动工具栏	312
11.2.4 显示和隐藏工具栏	315
11.2.5 给按钮添加文字	316
11.2.6 在工具栏中添加非按钮控件	316
11.3 状态栏	321
11.3.1 创建和初始化状态栏	321
11.3.2 创建自定义状态栏窗格	322
习题	325
第 12 章 文档/视图和单文档界面	327
12.1 文档/视图体系结构基础	327
12.1.1 对象之间的关系	327
12.1.2 对象的创建	328
12.2 文档对象和视图对象	330
12.2.1 文档对象	330
12.2.2 视图对象	331
12.3 文档的序列化	338
12.3.1 序列化	338
12.3.2 编写 Serialize 函数	339
12.3.3 编写可序列化类	340
12.4 文档模板资源	348
12.5 滚动视图	349
12.5.1 建立滚动视图	349
12.5.2 普通视图转换为滚动视图	353

习题	355
第 13 章 文件的读/写	356
13.1 CFile 类	356
13.1.1 打开和关闭文件	356
13.1.2 文件读/写	359
13.1.3 CFile 类的其它操作	360
13.2 CFile 的派生类	362
习题	366
附录	367
参考文献	368

第一篇 C++语言基础

第1章 C++概述

1.1 C++简史和特点

1.1.1 C++简史

C++是在C语言的基础上发展演变而来的。

为了编写UNIX操作系统,1972年,美国贝尔实验室的D.M.Ritchie在B语言的基础上设计并实现了C语言。此后经过多次修改和完善,C语言得到了广泛的应用并出现了各种不同的版本。1983年,美国国家标准化协会(ANSI)制定了C语言的标准,称为ANSI C。目前的各种C语言版本基本上都是以ANSI C为基础的。

C语言有许多突出的特点,如:语言简洁,使用方便灵活;提供了丰富的运算符和数据类型;具有结构化的控制语句;生成的目标代码质量高,程序执行效率高;能直接访问物理地址,可以进行位操作,实现汇编语言的大部分功能;与汇编语言比较,具有较高的可移植性。

虽然具有许多优点,但C语言本身也存在一些局限性:C语言的类型检查机制相对较弱,这使得程序中的一些错误不能在编译阶段被检查出来;C语言本身没有支持代码重用的机制,当程序达到一定的规模后,程序员很难控制程序的复杂度,对程序的维护比较困难,因此,C语言不适合开发大型应用程序。

为了解决上述问题和适应程序设计的需要,1979年,美国贝尔实验室的Bjarne Stroustrup博士对C语言进行了改进和扩充,并从Simula 67中引入了面向对象程序设计的内容。改进后的C语言最初被称为“带类的C”,1983年更名为C++。之后经过了三次主要的修订,每一次修订都对C++进行了一些修改并增加了部分新的内容。第一次修订在1985年,第二次修订在1990年,第三次修订发生在C++的标准化过程中。C++语言的标准化工作从1989年开始,于1994年制定了第一个C++标准草案,1998年正式发布了ANSI/ISO C++标准,即目前常称的标准C++。所有主流的C++编译器都支持该版本的C++,例如Borland公司的C++ Builder和Microsoft公司的Visual C++,同时它们对标准C++都有不同程度的扩展。与此同时,各公司还为C++编写了各种不同的类库,其中Borland公司的OWL(Object Window Library)和Microsoft公司的MFC(Microsoft Foundation Class)就是比较优秀的代表,尤其是Microsoft公司的MFC,在国际上得到了较为广泛的应用。

1.1.2 C++的特点

C++语言具有以下特点:

(1) C++是C语言的超集。它既保持了C语言的简洁、高效和接近汇编语言等特点,又克服了C语言的缺点,其编译系统能检查更多的语法错误,因此,C++比C语言更安全。

(2) C++保持了与 C 语言的兼容。绝大多数 C 语言程序可以不经修改直接在 C++环境中运行, 用 C 语言编写的众多库函数可以用于 C++程序中。

(3) 支持面向对象程序设计的特征。C++既支持面向过程的程序设计, 又支持面向对象的程序设计。

(4) C++程序在可重用性、可扩充性、可维护性和可靠性等方面都较C语言得到了提高, 使其更适合开发大中型的系统软件和应用程序。

1.2 C++程序的基本结构

1.2.1 C++程序实例

下面通过两个程序实例来说明 C++程序的基本结构。

【例 1.1】 在屏幕上输出 “Hello, C++ World!”。

```
//This is my first C++ program
#include <iostream.h>                //文件包含命令
void main()                          //主函数
{
    cout<<"Hello,C++ World!"<<endl;    //在屏幕上输出 “Hello, C++ World!”
}
```

在 Brian Kernighan 和 Dennis Ritchie 合著的经典著作《The C Programming Language》中, 他们用在屏幕上输出 “HELLO, WORLD” 的程序作为介绍 C 语言的第一个程序。例 1.1 是这个程序的 C++版本。

程序经过编译、链接后, 运行结果如下:

```
Hello,C++ World!
```

程序中以 “//” 开始的内容为 C++的注释, 用于对程序或语句进行说明, 以提高程序的可读性。在编译时, 注释将被忽略。

程序的第 2 行 “#include <iostream.h>” 为编译预处理命令, 其作用是指示 C++编译器将头文件 iostream.h 的内容插入到 #include 命令所在的源程序中, 其中包含了输入流对象 cin 和输出流对象 cout 的定义。第 3 行为主函数头部。第 5 行是输出语句, 其功能是在屏幕上输出 “Hello, C++ World!”, endl 为控制符, 表示换行, 即输出上述信息后回车换行。

【例 1.2】 从键盘上输入两个整数, 求它们的和并在屏幕上输出。

```
#include<iostream.h>                //文件包含命令
int add(int a,int b);               //函数原型说明语句
void main()                          //主函数
{
    int x,y;                        //定义两个用于存放整数的变量
    cout<<"Enter two integer numbers:"<<endl;    //显示提示信息, 提示输入两个整数
    cin>>x>>y;                      //输入两个整数存于变量 x 和 y 中
    int sum;                        //定义用于存放结果的变量
```

```
sum=add(x,y);           //调用函数 add 求和并赋值给 sum
cout<<"The sum is:"<<sum<<endl;    //输出两个数的和
}
int add(int a,int b)      //定义函数 add
{
    return a+b;          //计算两个数的和并返回结果
}
```

程序运行结果如下:

Enter two integer numbers:

59✓

The sum is:14

其中,第二行是从键盘输入的两个数,两个数间可以使用空格键、Tab 键或回车键分隔。

说明:运行结果中带下划线的是用户从键盘输入的数据,“✓”表示输入后按回车键。

1.2.2 C++程序的基本结构

通过分析上面两个实例,可以总结出 C++程序的基本结构。

C++程序是由函数构成的。通常,一个 C++程序包含一个或多个函数,其中必须有且只有一个 main()函数。main()函数可以有参数也可以没有参数,可以有返回值也可以没有返回值。如果没有返回值,则函数类型应指明为 void 类型。函数 main()是程序执行的入口点,无论此函数在程序中的什么位置,每个程序都从 main()开始执行,main()函数返回时程序结束。

一个 C++函数由函数说明部分和函数体构成。函数说明部分包括了函数的返回值类型、函数的名称、圆括号、形参及形参的类型说明。函数体由包含在一对花括号(“{”和“}”)内的语句组成,函数体中的语句完成函数的功能。

C++中的每条语句都以分号(“;”)结束,分号是语句的必要组成部分。

为了增加对程序的理解和可读性,可在程序的适当位置添加注释,用来描述程序或语句的功能、变量的作用等。

C++程序的书写格式自由,一行内可以写多条语句,一条语句也可以分写在多行上。但为了提高程序的可读性和检查程序错误的效率,一般一行写一条语句,采用层次缩格书写的方式,同一层次的语句对齐。

一个 C++程序可以只有一个源程序文件,也可以由多个文件组成。C++源程序文件的扩展名为 .cpp 或 .cxx。

最后需要说明的是:本书前 6 章中的程序都是基于 DOS 或控制台的,而非 Windows 程序,即这些程序都在 DOS 或 Windows 命令提示符下运行。

1.3 C++基本语法成分

1.3.1 字符集

构成 C++程序的基本元素是各种字符。C++字符集规定了在程序中使用哪些字符。

在C++中,除字符型数据外,其它所有成分都只能由字符集中的字符构成。

C++字符集包括以下三类字符:

(1) 大小写英文字母: a~z 和 A~Z。

(2) 数字字符: 0~9。

(3) 特殊字符:

空格 ! # % ^ & * _ (下划线) + = - ~
< > / \ | . , : ; ? ' " () [] { }

说明: 在程序中使用这些特殊字符时应使用其英文半角形式,不能使用中文全角形式。

1.3.2 关键字

关键字是系统中已经预先定义的单词,在程序中有特殊的含义和用法。C++中常用的关键字如下:

auto	bool	break	case	char	class
const	continue	default	delete	do	double
else	enum	explicit	extern	false	float
for	friend	goto	if	inline	int
long	mutable	namespace	new	operator	private
protected	public	register	return	short	signed
sizeof	static	static_cast	struct	switch	template
this	throw	try	typedef	union	unsigned
virtual	void	while			

1.3.3 标识符

标识符是程序中用来定义变量名、函数名、对象名、常量名、类型名和语句标号等的单词。

C++中标识符分为系统预定义标识符和用户自定义标识符两类。系统预定义标识符是系统中已经预先定义好的函数名、常量名、对象名等,可以被用户直接使用,如对象 cin 和 cout 等。与关键字不同的是,系统预定义标识符允许用户赋予新的含义,即成为用户自定义标识符,但为了避免误解,一般自定义标识符不应与系统预定义标识符同名。

用户自定义标识符是用户根据需要自己定义的标识符,通常用作变量名、函数名、对象名、常量名、类型名和语句标号等。

C++标识符的构成规则如下:

- (1) 标识符以字母或下划线开始,由字母、数字字符和下划线组成。
- (2) 标识符中的大小写字母有区别,如 Sum 和 sum 是两个不同的标识符。
- (3) 不能与 C++关键字相同。

关于标识符的长度,不同的编译系统有不同的规定,如有的系统只识别前 32 个字符,Visual C++ 编译系统允许的最大标识符长度为 247 个字符。在定义标识符时最好能做到“见名知义”,以提高程序的可读性,如: min 表示最小值, sum 表示和, average 表示平均值等。