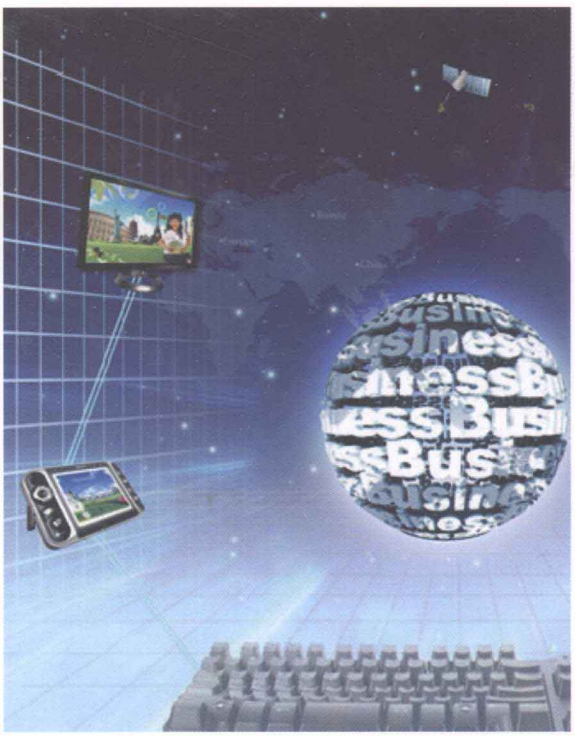
高等学校计算机应用规划教材

Visual C++程序设计

- C++概述
- 数据类型及基本运算量
- 结构化程序设计
- 数组
- 用户自定义函数
- 指针
- 类与对象
- 继承与多态性
- 对话框
- 常用控件



梁海英　主　编

张文波　孙静　于萍　刘哲　副主编

清华大学出版社

高等学校计算机应用规划教材

Visual C++程序设计

梁海英 主编

张文波 孙静 于萍 刘哲 副主编

清华大学出版社

内 容 简 介

本教材以面向应用型人才培养为目标；以非传统的组织结构为创新点；以全程伴随上机实践为特色；简洁、通俗、直观、易懂地讲述 C++程序设计。第 1~3 章讲述 C++的基础知识，包括上机环境、数据类型、常量、变量和表达式以及顺序、分支、循环三大结构及其编程。第 4~6 章介绍 C++的重点知识，包括数组、函数和指针。第 7~10 章介绍 C++的提高知识，包括类与对象、类的继承和派生、多态和虚函数、对话框和标准控件等知识。

本书从实用角度出发，内容选取先进精准、内容组织循序渐进、内容讲解文字精练；知识讲解辅助图表、理论讲述结合实例；典型实例精挑细选、算法分析流程图化、程序结构错落有致、程序结果真实有效；各章习题针对性强、题型丰富；免费提供电子课件、源程序及习题答案；详细介绍开发环境 Visual C++ 6.0 的使用方法，全部例题均在此环境中成功运行。

本书可作为高等学校非计算机专业本科生的计算机通识教材；也可作为计算机相关专业的程序设计入门教材、计算机技术的培训教材；或者作为全国计算机等级考试的参考用书和编程爱好者的自学教材。

本书对应的电子教案、习题答案和实例源文件可以到 http://www.tupwk.com.cn/downpage 网站下载。

图书在版编目(CIP)数据

Visual C++程序设计 / 梁海英 主编. —北京：清华大学出版社，2013.10

(高等学校计算机应用规划教材)

ISBN 978-7-302-33919-9

Ⅰ. ①V…　Ⅱ. ①梁…　Ⅲ. ①C 语言－程序设计－高等学校－教材　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2013)第 220371 号

责任编辑：胡辰浩　袁建华
装帧设计：牛静敏
责任校对：成凤进
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62794504

印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：19.5　　**字　　数**：450 千字
版　　次：2013 年 10 月第 1 版　　**印　　次**：2013 年 10 月第1 次印刷
印　　数：1～3500
定　　价：35.00 元

产品编号：055050-01

本书编委会

主　编： 梁海英

副主编： 张文波　孙　静

于　萍　刘　哲

参　编： 朱　宏　张　伟

王发斌　陈晓明

前　言

Visual C++是一门优秀的计算机程序设计语言，其功能相当强大，既适合面向过程的程序设计，又适合面向对象的程序设计。本教材目的是使学生理解和掌握 C++程序设计的基本概念和基本思想，力求通过简单算法的讲解学习 C++的编程方法，并能够利用 Visual C++编写解决实际问题的程序。

本教材面向应用型人才培养，内容安排由简到难，逐步深入，以免学习者失去学习信心。本教材组织结构共分三大部分，第一部分主要讲解 C++语言的基本知识，包括：C++的基本词法和语法规则、基本数据类型、常量、变量和表达式、程序控制 3 种结构；第二部分主要讲解 C++的数组、函数和指针；第三部分主要讲解面向对象程序设计思想和 Windows 可视化编程。主要内容包括：类与对象的概念和定义格式、对象的赋值和引用、友元、类的继承和派生、多态和虚函数、对话框和标准控件等。书中的所有例题都在 Visual C++ 6.0 版本的编译系统下运行通过，每章后面都附有习题，以便及时对所学知识进行巩固。

全书直接采用 C++的 cin 和 cout 进行输入输出，摒弃了 C 的 printf 和 scanf 函数调用。从实用的角度出发，内容选取先进精准、组织循序渐进、讲解文字精练；各部分辅助图表、结合实例、深入浅出、结构清晰；典型实例精挑细选、算法分析流程图化、程序结构错落有致、程序结果真实有效；各章习题针对性强、题型丰富。

在本书编写过程中，得到了同事们的热心帮助和支持，参加本书内容编写、程序调试、课件制作、习题收集、答案制作、内容审校等工作的老师还有朱宏、张伟、王发斌和陈晓明等，在此表示衷心的感谢！

在编写过程中，本教材参考了有关 Visual C++教材、文献和网站内容，并引用了一些材料，在此对这些作者表示衷心的感谢。由于书稿涉及的内容多、范围广，尽管作者已尽最大努力，由于时间仓促，书中难免存在不妥之处，请读者原谅并提出宝贵意见。我们的电话是 010-62796045，信箱是 huchenhao@263.net。

编　者

2013 年 6 月

目　录

第1章　C++概述

C++是一种应用广泛的面向对象程序设计语言，由C发展而来，其保留了C语言的所有优点，既可以用于面向过程的结构化程序设计，又可以实现面向对象程序设计。本章主要讲述C++的特点和语法、C++程序的实现过程和Visual C++集成开发环境。

1.1　程序设计语言

人们与计算机进行交流是通过程序实现的，只有解决一定问题的程序才能指挥计算机自动地进行工作，而程序又是通过程序设计语言开发的。程序是指人们使用编程语言开发，为解决一定问题，能够被计算机执行的指令代码。计算机程序设计语言是计算机可以识别的、编程人员应遵守的程序代码规则，是人指挥计算机进行工作、与计算机进行交流的工具。

计算机程序设计语言是不断发展的。纵观其历史，可以将其分为低级语言和高级语言两大类。

1.1.1　低级语言

低级语言又称为面向机器的语言，因 CPU 的不同而不同，可移植性差。使用低级语言可以编出效率高的程序，但对程序设计人员的要求也很高。其不仅要考虑解题思路，还要熟悉机器的内部结构，所以非专业人员很难掌握这类程序设计语言。

低级语言又分为机器语言和汇编语言。

1. 机器语言

机器语言是 CPU 可以直接识别的一组由 0 和 1 序列构成的指令代码。用机器语言编写程序，就是从所使用 CPU 的指令系统中挑选合适的指令，按照解决问题的算法组成一个指令序列。这种程序可以被机器直接理解并执行，速度很快，但由于它们不直观、难记、难写、不易查错、开发周期长，所以现在只有专业人员在编制对于执行速度有很高要求的程序时才采用。

2. 汇编语言

为了减轻编程者的劳动强度，人们使用一些帮助记忆的符号来代替机器语言中的0、1代码，使得编程效率和质量都有了很大的提高。由这些助记符组成的指令系统，称为符号语言，也称为汇编语言。汇编语言指令与机器语言指令基本上是一一对应的。因为这些助

记符不能被机器直接识别，所以用汇编语言编写的程序必须被汇编成机器语言才能被机器理解。汇编之前的程序称为源程序，汇编之后的程序称为目标程序，再使用组建程序将目标程序组建成可执行程序。可执行程序能够脱离语言环境独立运行。

1.1.2　高级语言

高级语言提供大量的与人类语言相类似的控制结构，使程序设计者可以不关心机器的内部结构及工作原理，把主要的精力集中在解决问题的思路和方法上。这类摆脱了硬件束缚的程序设计语言的出现是计算机技术发展的里程碑，使得编程不再是少数专业人员的专利。由于高级语言不依赖具体的机器，所以用高级语言编写的程序可移植性好。

根据编程机制的不同，高级语言又分为面向过程的程序设计语言和面向对象的程序设计语言。

1. 面向过程的程序设计语言

面向过程的程序设计语言由一个入口和一个出口构成，程序每次执行都必须从这个入口开始，按照程序的结构执行到这个出口为止，属于过程驱动的编程机制，由过程控制程序运行的流向。编程人员要以过程为中心来考虑应用程序的结构，执行哪一部分代码和按何种顺序执行代码都由程序本身控制。它允许将程序分解为多个函数，这使得同一个程序可以由多人分工开发，大大提高了编程效率，从而能够开发出规模越来越大、功能越来越强的应用软件和系统软件。常用的面向过程的语言有 C、Fortran、Pascal 等。

2. 面向对象的程序设计语言

面向对象的程序设计语言将整个现实世界或者其中的一部分看作是由不同种类的对象构成的，同一类型的对象既有相同点又有不同点。各种类型的对象之间通过发送消息进行联系，消息能够激发对象作出相应的反应，从而构成一个运动的整体，属于事件驱动的编程机制，由事件控制着程序运行的流向。编程人员要以对象为中心来设计模块，代码不是按预定的顺序执行，而是在响应不同的事件时执行不同的代码。当前使用较多的面向对象的程序设计语言有 Visual Basic、C++、C#、Java 等。

高级语言也不能被机器直接识别，也需要翻译才能运行。高级语言的运行方式有解释和编译两种。所谓解释是指边解释边执行，不形成目标代码，执行速度不快，源程序保密性不强，Visual Basic 属于解释方式；所谓编译是将源程序使用语言本身提供的编译程序编译为目标程序，再使用组建程序与库文件组建成可执行程序，可执行程序能够脱离语言环境独立运行。本课程所学的 C++语言程序设计属于编译方式。

1.2　C++语言的特点

C++语言于 20 世纪 80 年代由贝尔实验室设计并实现，是在 C 语言的基础上发展起来

的，既支持传统的面向过程的程序设计，又支持面向对象的程序设计。

1.2.1　C 语言的特点

C 语言的特点主要表现在：

(1) 功能强，应用广泛；

(2) 语句简洁，表达能力强；

(3) 运算符丰富；

(4) 数据结构丰富，具有现代化语言的各种数据结构；

(5) 具有结构化的控制语句；

(6) 程序设计自由度大；

(7) C语言允许直接访问物理地址，能够进行位操作，能够实现汇编语言的大部分功能，可以直接对硬件进行操作，既有高级语言的功能，又有低级语言的功能；

(8) 生成目标代码质量高，程序执行效率高；

(9) 可移植性好。

除上述优点之外，C 语言也有其局限性：

(1) C 语言的类型检查机制相对较弱，有些错误不能在编译阶段检查出来；

(2) C 语言本身几乎没有支持代码重用的语言结构；

(3) 当程序的规模达到一定程度时，程序员就很难控制程序的复杂性。

1.2.2　C++语言的特点

与 C 语言不同，C++是一种广泛使用的面向对象的程序设计语言，其包括了 C 的所有特征、属性和优点(如高效、灵活性)，同时改进了 C 的一些不足，并且支持面向对象的程序设计。

C++语言的特点主要表现在：

(1) 保持与 C 兼容；

(2) 可读性更好，代码结构更合理；

(3) 生成代码的质量高；

(4) 可重用性、可扩充性、可维护性和可靠性有所提高；

(5) 支持面向对象的机制。

C++语言中与面向对象有关的特征如下。

(1) 类和数据封装

C++支持数据封装，将数据和对该数据操作的函数封装在一起作为一种数据类型，称为类。同时提供一种对数据访问严格控制的机制，封装体通过操作接口与外界交换信息。

(2) 结构作为一种特殊的类

在 C 语言中可以定义结构体，但是这种结构只包含数据，不包含函数。C++中的类是数据和函数的封装体，在 C++中，结构可以作为一种特殊的类。

(3) 构造函数和析构函数

构造函数是类内和类同名的成员函数，创建对象时对类的数据成员进行初始化。析构函数的功能是用来释放对象。

(4) 私有、保护和公有成员

在 C++类中可以定义 3 种不同访问控制权限的数据成员。其分别是私有(private)、保护(protected)和公有(public)成员。私有成员只有类本身定义的函数才能访问，而类外的其他函数不可以访问；保护成员只有派生类可以访问，而在类外不可以访问的成员；公有成员是在类外也可以访问的成员，是该类与外界的接口。

(5) 对象和消息

对象是类的实例，对象之间通过消息来实现合作，共同完成某一任务。每个对象根据收到消息的性质来决定需要采取的行动，以响应这个消息。

(6) 友元类和友元函数

类中的私有成员是不允许类外的任何函数访问的。但是友元打破了类的这一限制，破坏了类的封装性，它可以访问类的私有成员。友元可以是类外定义的整个类，称为友元类，也可以是类外的函数，称为友元函数。

(7) 运算符和函数名重载

运算符重载和函数名重载都属于多态，多态是指相同的语言结构可以代表不同类型的实体，或者对不同类型实体进行操作。C++允许相同的运算符或标识符代表多个不同实现的函数，这称为标识符或运算符重载，用户可以根据需要定义标识符重载或运算符重载。

(8) 派生类，继承性

一个类可根据需要生成派生类，派生类继承了基类的所有方法，同时还可定义新的不包含在父类中的方法。派生类包含从父类继承过来的数据成员和自己特有的数据成员。

(9) 虚拟函数，多态性，动态联编

C++可以定义虚函数，通过虚函数实现动态联编。动态联编是多态的一个重要特征。多态性形成由父类和其子类组成的一个树型结构。在这个树中的每一个子类可接收一个或多个具有相同名字的消息。当一个消息被这个树中的一个类的一个对象接收时，这个对象动态地决定给予子类对象的消息的某种用法。多态中的这一特性允许使用高级抽象。

1.3　C++程序结构的特点

本节通过几个例子让读者对 C++程序的结构有一个大体上的认识，并对 C++语言程序的构成有一个初步的了解。

1.3.1　C++程序结构

首先介绍一个简单的 C++程序，使读者对 C++程序有一个大概的了解。下面的例子虽然简单，但反映了一般 C++程序的特点以及基本的组成。

例 1-1　编写一个 C++程序，其功能是显示字符串"This is our first C++ program."。

其 C++程序如下所示：

```
#include<iostream.h>      //包含头文件 iostream.h
void main()               //主函数
{
  cout<<"This is our first C++ program.\n" ;//输出一行字符
}
```

程序的运行结果如图 1-1 所示。

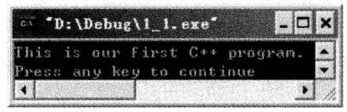


图 1-1　例 1-1 运行结果

程序的第一行：#include<iostream.h>通常称为命令行，命令行必须用符号#开头，一对尖括号中的 iostream.h 是系统提供的文件名，包含着有关输入输出函数的信息。调用不同的标准库函数，应包含不同的头文件，随着课程的深入，将在以后的章节中陆续介绍相关的头文件。

第二行的 main 是主函数名，其后的一对圆括号中间可以是空的，但是这一对圆括号不能省略，main()是主函数的起始行，一个 C++程序可以包含任意多个不同名的函数，但是必须有而且只有一个主函数，一个 C++程序总是从主函数开始执行。

主函数后面由一对花括号{}括起来的部分是主函数体，其中的语句是实现程序的具体功能。函数体用左花括号{开始，右花括号}结束。期间可以有定义部分和执行部分，定义部分主要是对要用到的变量进行说明，执行部分主要是实现程序的具体功能，执行部分的语句称为可执行语句，必须放在说明部分之后，语句的数量不限，程序中的这些语句向计算机系统发出操作命令。C++的每一条定义语句和执行语句都要以分号;作为结束，分号是 C++语句的一部分。

下面再举几个例子以便读者进一步熟悉 C++程序的结构与书写格式。

例 1-2　输入矩形的两条边，求矩形的面积。

程序如下所示：

```
#include<iostream.h>
void main()
{
    double a,b,area;
    a=1.2;
    b=1.5;
    area=a*b;
    cout<<"a="<<a<<"\tb="<<b<<"\tarea="<<area<<endl;
}
```

程序的运行结果如图 1-2 所示。

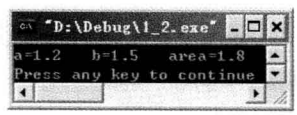


图 1-2　例 1-2 运行结果

程序中 main()后一对花括号括起来的部分称为函数体，其中，程序的第 4 行为函数的说明

部分；第 5 到 8 行是函数的执行部分。第 8 行为输出语句，功能是输出 a，b 和 area 的值。

例 1-3　从键盘输入一个直角三角形两边 a 和 b 的长度，求其斜边长度。

程序如下所示：

```
#include<iostream.h>
#include<math.h>
void triangle(double x, double y)
{
    double z;
    z=sqrt(x*x+y*y);
    cout<<"hypotenuse="<<z<<endl;
}
void main()
{
    double a,b;
    cout<<"input a and b:";
    cin>>a>>b;
    triangle(a,b);
}
```

程序运行结果如图 1-3 所示。

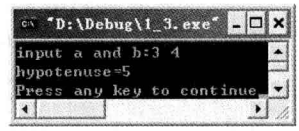


图 1-3　例 1-3 运行结果

这个程序包含两个函数：一个是用于计算直角三角形斜边长度的函数 triangle()，一个是主函数 main()。在主函数中首先要求从键盘输入两直角边 a 和 b 的值，然后调用计算斜边长度的函数 triangle()。

从上面几个例子可以看出，每个程序都包含命令行，且都以#开头，其作用是提供标准的库函数以及用户自定义类库和函数；每个程序都有一个 main()函数，它是程序的入口，每个程序都从这里开始执行，除了 main()函数外，还可以定义其他函数，整个 C++程序可以说是由若干函数组成；每个程序都包含对变量和函数的说明，同时还有输入和输出功能。

一个 C++程序的一般格式如下所示：

```
#include<……>                    //预处理命令行
函数声明                          //程序中用到的函数的说明
全局数据定义                      //程序中用到的全局数据的定义
void main()
{
函数体                          //由声明部分和执行部分组成
}
用户自定义函数的定义              //程序中用到的函数的具体实现
```

其中，函数声明包括函数返回值类型、函数名和参数；函数的定义部分给出了函数的具体实现。一个 C++程序并不一定严格按照上述要求来写，有些部分可以省略。

1.3.2　程序书写规则

从书写清晰，便于阅读、理解和维护的角度出发，在书写程序时应遵循以下规则。

(1) 一行可以写多个声明或语句，但为了清晰，一个声明或一个语句最好占一行。每条声明或语句都有明确的含义，能完成一定的任务。

(2) 用{}括起来的部分，通常表示程序的某一层次结构。{}一般与该结构语句的第一个字母对齐，并单独占一行。

(3) 为了使程序便于阅读、易于调试，从而约定了锯齿形缩进的程序书写方式。将复合语句、函数体、循环体等语句用空格或 tab 键向后缩进，使得程序错落有致，具有层次感。也就是说，低一层次的语句或声明比高一层次的语句或声明缩进若干格。

(4) 标识符和关键字之间至少加一个空格以示分隔。若已有明显的分隔符，也可不再加空格。

(5) C++语言声明或语句中使用的都是西方字符(称半角字符)，所以在输入源程序时，应该将中文输入法关闭，避免输入全角字母和符号。全角字母和符号只有在字符串常量中才可以使用，而且字母是区分大小写的。

(6) 在程序中可以适当地加上注释，以增强程序的可读性。

在编程时应力求遵循这些规则，以养成良好的编程风格。

本书为了方便介绍语句、函数等的使用方法、语法格式，在命令格式中通常采用一些特殊的符号表示，包括逗号加省略号和省略号等。这些符号不是命令的组成部分，在输入具体命令时，这些符号均不可作为语句中的成分输入计算机，其只是命令的书面表示。具体含义如下所示：

,…表示同类的项目重复多项。

…表示省略了在当时叙述中不涉及的语句部分。

1.3.3　程序保留字

在 C++语言中使用的词汇分为 6 类：关键字、运算符、分隔符、注释符、标识符和常量，除标识符外，其他均为保留字，有特定的作用，不能挪为它用。如表 1-1 所示是标准 C++语言所采用的保留字。

表 1-1　C++的标准保留字

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

1. 关键字

关键字是由C++语言规定的具有特定意义的字符串。C++语言的关键字分为以下几类。

(1) 类型声明符

用于定义(或声明)变量、数组、自定义函数或自定义数据类型。如int、float、double等。

(2) 语句定义符

用于表示一个语句的功能。如if、for、while等。

(3) 预处理命令字

用于表示一个预处理命令。如前面各例中用到的include。

2. 运算符

C++语言中含有丰富的运算符。运算符与常量、变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。如算术运算符+、-、*、/等。

3. 分隔符

在C++语言中采用的分隔符有逗号和空格两种。逗号主要用在类型声明和函数参数表中，分隔各个变量；空格多用于语句各单词之间，作分隔符。在关键字、标识符之间必须要有一个以上的空格符作分隔，否则将会出现语法错误，例如，把int a;写成inta;，C++编译器会把inta当成一个标识符处理，其结果肯定出错。

4. 注释符

为了提高程序的可读性，通常在程序的适当位置加上必要的注释。C++语言的注释符有两种：一种是块注释，是以/*开头并以*/结尾的字符串；另一种是行注释，从//开始到行尾的字符串。注释可出现在程序中的任何位置，注释主要用来解释语句或函数的功能，用来向用户提示或解释程序的意义，以便读者或开发者日后能够读懂程序。程序编译时，不对注释作任何处理。在调试程序时可以对暂不使用的语句先用注释符括起来，使编译程序跳过处理，待调试结束后再去掉注释符。

5. 标识符

用来标识符号常量名、变量名、函数名、数组名、类型名、文件名等有效字符序列统称为标识符。除库函数的函数名由系统定义外，其余都由用户自己定义。

C++规定，标识符由字母(a~z，A~Z)、数字(0~9)、下划线(_)组成，并且第一个字符必须是字母或下划线，即标识符的命名规则是以字母或下划线开头的，后面跟着字母、数字或下划线的字符串。

在使用标识符时还必须注意以下几点。

(1) 标识符的长度受各种版本的C++语言编译系统限制，同时也受具体机器的限制。

(2) 在标识符中，区分大小写。例如，b和B是两个不同的标识符。

(3) 标识符虽然可由程序员随意定义，但最好遵循见名知义的原则，便于阅读和理解。