

21世纪高等院校计算机应用规划教材

Visual C++ 程序设计学习指导

主 编 赵 璐 李 斌 张月琴 孙 冰



 南京大学出版社

21世纪高等院校计算机应用规划教材

Visual C++ 程序设计学习指导

主 编 赵 璐 李 斌 张月琴 孙 冰



144661

广西工学院鹿山学院图书馆



d144661

 南京大学出版社

图书在版编目(CIP)数据

Visual C++程序设计学习指导/赵璐等主编. —南京:
南京大学出版社, 2011. 1 (重印)

21 世纪高等院校计算机应用规划教材

ISBN 978-7-305-06581-1

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

中国版本图书馆 CIP 数据核字(2009)第 212418 号

出版者 南京大学出版社
社 址 南京市汉口路 22 号 邮 编 210093
网 址 <http://www.NjupCo.com>
出版人 左 健
丛 书 名 21 世纪高等院校计算机应用规划教材
书 名 Visual C++程序设计学习指导
主 编 赵 璐 李 斌 张月琴 孙 冰
责任编辑 樊龙华 单 宁 编辑热线 025-83596923
照 排 南京南琳图文制作有限公司
印 刷 南京大学印刷厂
开 本 787×1092 1/16 印张 16.25 字数 400 千
版 次 2009 年 12 月第 1 版 2011 年 1 月第 2 次印刷
ISBN 978-7-305-06581-1
定 价 31.60 元
发行热线 025-83594756
电子邮箱 Press@NjupCo.com
Sales@NjupCo.com(市场部)

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购
图书销售部门联系调换

前 言

本书是针对 VC++ 程序设计课程编写的学习指导用书。全书分为四篇,分别从概念、算法和上机实验环节对 VC++ 涉及到的各知识点进行了总结和阐述。

第一篇“VC++ 语法知识基础”,总结、归纳了 VC++ 基本语法知识,程序的三种基本结构,以及函数的使用。通过对知识点的总结、例题的分析帮助读者领会、吃透最基础的语法知识,掌握最基本的程序设计方法。

第二篇“VC++ 语法知识进阶”,对 VC++ 中的数组及指针的使用进行了全面的总结和阐述,并将面向对象中复杂而纷繁的概念进行了归纳。结合第一篇的基础语法知识分析,帮助读者理清 VC++ 语法脉络,掌握面向对象的基本概念。

第三篇“常用算法及编程技巧”,通过对基本算法技巧的归纳,和具体应用问题的分类分析和讲解,帮助学生在众多的应用程序中发现和总结算法规律,掌握解题工具。

第四篇“程序编写及调试”,主要针对上机实践环节。向读者介绍常见错误及常用的调试方法,指导学生掌握调试程序的基本步骤。参考计算机等级考试的模式,本篇进行了类的编程模式介绍,帮助读者找出此类问题的编程规律,对顺利通过计算机等级考试也有一定的指导意义。

本书是 VC++ 程序设计复习用书,建议与《Visual C++ 程序设计教程》配套使用。全书对概念和知识点主要进行概括、总结和比较,并通过例题分析来强化概念。在适当的章节还将使用图表来帮助读者记忆。

本书第 1 章~第 7 章、第 16 章由张月琴老师编写,第 8 章~第 10 章由李斌老师编写,第 11 章、第 15 章由赵璐老师编写,第 12 章~第 14 章由孙冰老师编写。

限于作者的水平与时间的窘迫,书中难免有许多不当之处,敬请读者不吝赐教。

编 者
2009 年 7 月

目 录

第一篇 VC++ 语法知识基础

1 程序设计代码的组织方式	3
1.1 程序设计的书写规则	3
1.2 VC++的函数	4
1.3 VC++的头文件	4
2 常量与变量、数据类型	5
2.1 数据类型	5
2.2 标识符	5
2.3 常量与变量	6
2.4 构造数据类型	8
3 运算符、表达式与库函数	12
3.1 VC++中的运算符与表达式	12
3.2 VC++中的库函数	18
4 顺序结构	20
4.1 简单的输入输出语句	20
4.2 赋值语句	22
5 选择结构	24
5.1 if 结构	24
5.2 switch 结构语句	25
6 循环结构	28
6.1 while 结构语句	28
6.2 do ... while 结构语句	29
6.3 for 结构语句	30
6.4 控制执行顺序的语句	32

7 函 数	33
7.1 函数的定义和调用	33
7.2 函数调用时的参数传递	38
7.3 函数的递归调用	42
7.4 标识符的作用域和生命期	45
7.5 文件包含和宏定义	52

第二篇 VC++ 语法知识进阶

8 数 组	59
8.1 数组的基本概念	59
8.2 字符数组与字符串	63
8.3 数组与函数	68
9 指针及链表	71
9.1 简单指针	71
9.2 指针与数组	72
9.3 指针与字符串	75
9.4 二级指针	78
9.5 指针与函数	78
9.6 指针所指向类型的判断	80
9.7 new 和 delete 运算符	80
9.8 结构体指针	82
9.9 链表	83
10 面向对象	90
10.1 类和对象的基本概念	90
10.2 构造函数和析构函数	96
10.3 继承与派生	105
10.4 友元与静态成员	113
10.5 多态性	116
10.6 运算符重载	120

第三篇 常用算法及编程技巧

11 算法中使用的基本技巧	129
11.1 交换、累加及累乘	129
11.2 穷举法	133
11.3 最值的求解	135

11.4 递推与递归方法.....	136
12 关于数的运算等问题.....	141
12.1 利用穷举法寻找特征数.....	141
12.2 数的各位分离.....	150
12.3 数的进制转换.....	153
12.4 数列计算.....	156
12.5 级数计算.....	158
13 排序、查找	162
13.1 排序问题.....	162
13.2 查找问题.....	170
14 字符串运算.....	173
14.1 字符串函数编程.....	173
14.2 字符串排序.....	175
14.3 子串的查找、匹配和删除	176
14.4 字符串的变换.....	181
15 矩阵处理.....	183
15.1 矩阵元素求和问题.....	183
15.2 矩阵元素的最值问题.....	187
15.3 矩阵的变换问题.....	191
15.4 矩阵的相加和相乘.....	197
15.5 矩阵相关的图形问题.....	201

第四篇 程序编写及调试

16 程序调试和程序设计.....	215
16.1 程序调试.....	215
16.2 程序设计.....	234
主要参考文献.....	250



第一篇

V C ++ 语法知识基础

程序设计,必须从数据、运算符和表达式开始,这些是构建一个程序的基本要素。不同的语言对这些基本要素的要求和规则会有差别,在程序设计开始之前,必须要熟记这些规则。

根据具体问题,使用顺序结构、选择结构或循环结构将表达式进行合理的组织就可以得到一个简单的程序。而对于一些比较复杂的问题,为了使程序结构更为清晰、条理更加清楚,往往还需要编写若干个功能独立的函数。

这一篇,就重点讨论基本的语法要素、三种语法结构和函数的使用。通过对概念的总结、归纳、比较,对例题的分析、解答,帮助读者记忆和巩固基本概念,熟悉和掌握程序设计的基本方法。

1

程序设计代码的组织方式

VC++的程序由注释、编译预处理和函数组成。平时养成良好的程序书写习惯,可以帮助读者提高阅读程序的能力,减少编程过程中发生的错误,也有利于软件的维护。本章着重讨论函数的书写格式、头文件等一些基本的概念。

1.1 程序设计的书写规则

VC++程序的书写要符合一定的规则,如:

```
/* VC++程序的基本结构 */
#include <iostream.h>
void main()
{
    //主函数开始
    float j;
    cin>>j;
    cout<<'j='<<j<<'\\n';
}
//主函数结束
```



提示:

1) 包含文件或编译预处理指令:一般写在程序的开始部分,如
#include <iostream.h>

2) 注释和说明信息:对部分语句注释说明,增加程序的可读性。方法有:

```
/* <注释信息> */
```

或者 // <注释信息> (从此开始到本行结束为止)

3) 主函数 main():任何一个 C++源(程序)文件都是由一个主函数(main 函数)和若干个其他函数构成,且一个 C++程序总是从 main 函数开始执行,而不管该函数在整个程序中的具体位置。其余函数(包括库函数和用户自定义函数)可有可无,为保证程序的可读性,建议任何函数的定义均从第一列开始书写。

4) 任一函数体均要以“{”开始,以“}”为结尾,另外注意大括号要配对使用。书写时可采取左对齐的方式保证括弧的配对。

5) 程序的书写规则:程序由语句组成,每条语句后均要以分号结束。语句书写时要注意两个规则:



- 对齐规则:同一层次的语句必须从同一列开始,同一层次的大开括号必须与对应的大闭括号在同一列上。
 - 缩进规则:同一层次的语句必须缩进相同个字符,即保持左对齐。
- 6) 严格区分字母的大小写。

【例 1-1】 下列四个叙述中,错误的是_____。

- A. VC++语言中的关键字必须小写
- B. VC++语言中的标识符必须全部由字母组成
- C. VC++语言不提供输入输出语句
- D. VC++语言中的注释行可以出现在程序的任何位置

分析:选项 A、C 和 D 由前面叙述可知均是正确的,而选项 B 由本书 2.2 节内容可知,不符合标识符的命名规则。所以,本题正确答案为 B。

1.2 VC++的函数

函数是具有特定功能的模块。一个 C++ 源程序(文件)由一个主函数和其他函数组成。函数之间是调用和被调用的关系。C++ 程序执行总是从主函数开始,之后调用其他函数,最后在主函数中结束整个程序的运行。

函数包括用户自定义函数和系统库函数。库函数是 C++ 编译系统中已经预先定义的函数,使用它必须先包含该库函数所在的头(.h)文件,请参见本书第 3.2 节。自定义函数请参见本书第 7 章的内容。

1.3 VC++的头文件

头文件是一种包含功能函数、数据接口声明的载体文件。C++ 程序的头文件以“.h”为后缀,它是用户应用程序和函数库之间的桥梁和纽带,是 VC++ 语言不可缺少的组成部分。打个比喻,头文件就像是一本书中的目录,读者(用户程序)通过目录,可以很方便就查阅其需要的内容(函数库)。

常用的头文件如表 1-1 所示:

表 1-1 常用头文件一览表

头文件语句	说 明
#include <iostream. h>	需要用到输入输出(cin 或 cout)时
#include <iomanip. h>	需要用 setw(n)指定输出的宽度
#include <math. h>	需要用到数学函数时
#include <string. h>	需要用到字符串函数时
#include <fstream. h>	需要用到流文件时
#include <stdlib. h>	需要用 exit()和 abort()来终止程序运行时

2 常量与变量、数据类型

程序中基本的元素是数据类型,每一数据都有其数据类型,只有确定了数据类型,才能确定数据在内存中的空间大小和其可进行的操作。VC++具有丰富的数据类型,包括基本的数据类型和导出的数据类型。本章对各类数据的数据类型分别加以阐述。

2.1 数据类型

VC++的数据类型可用图 2-1 来描述:

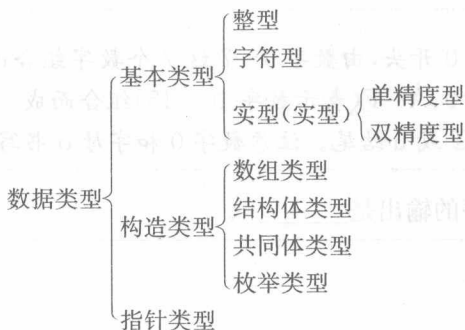


图 2-1 VC++数据类型分类图



提示:

1) 基本类型:

- 整型,说明符为 int,占用 4 个字节;
- 字符型,说明符为 char,占用 1 个字节;
- 单精度型,说明符为 float,占用 4 个字节;
- 双精度型,说明符为 double,占用 8 个字节;
- 无值型,说明符为 void,占用 0 个字节。

2) 在四种基本类型前还可以加上以下修饰符:

signed(有符号型),unsigned(无符号型),short(短型),long(长型)

3) 注意各种数据类型在内存中所占用的字节数。

2.2 标识符

合法标识符的命名规则:以字母或下划线开始,由字母、数字以及下划线组成的字符序



列。其有效长度为1~247个字符,标识符不允许是VC++的关键字。

【例2-1】下列符号中能用作C++标识符的是_____。

A. 5abc

B. if

C. -abc

D. _abc

分析:由标识符的定义可判断:选项A以数字开始,选项B是关键字,选项C以短横线为开始,所以上三个选项均不符合标识符的命名规则,只有选项D是合法的标识符。所以,该题的正确答案为D。

2.3 常量与变量

2.3.1 常量

在程序执行过程中,值保持不变(也不能被改变)的量称为常量。根据类型的不同可分为整型常量、实型常量、字符型常量、字符串型常量和标识符常量。

(1) 整型常量:可用十进制、八进制、十六进制来表示。



提示:

八进制整数以0开头,由数字0~7这7个数字组合而成;十六进制整数以0x开头,由0到9、A~F(表示数字10~15)组合而成。长整型以L或l结尾,无符号整数以U或u结尾。注意数字0和字母o书写上的区别。

【例2-2】下面程序的输出是_____。

```
#include <iostream.h>
oid main()
{
    int m=0xa,n=2;
    m+=n;
    cout<<m<<endl;
}
```

分析:注意语句 $m=0xa$, 右边 $0xa$ 是常量的十六进制表示,即 m 的值为10,因此经过 $m+=n$ 计算后,可以得出 m 的值是12的结论。

(2) 实型常量:十进制小数形式和指数形式。

十进制小数形式由数字0~9和小数点组成,指数形式(科学表示法)以 $1e<\text{指数}>$ 的形式来表示10的多少次方。



提示:

在E或e的前面必须有数字,且在E或e之后的指数部分必须是整数,如:123E12, 0.23e-2都是合法的实型常量。而e10, .e5, 1.0e3.5都是不合法的实数。

【例2-3】以下选项中,不能作为合法常量的是_____。

A. 0X2A

B. 015

C. 2.8e9

D. 123.6e0.4

分析:选项 A 是十六进制常量;选项 B 是八进制常量;选项 C 和 D 是实数的指数表示方法,根据前面内容的阐述,可以知道选项 C 是正确的表示形式,而选项 D 中指数部分不是整数,因此是错误的。所以,该题的正确答案为 D。

(3) 字符常量:单引号括起来的单个字符。如 'a', '&' 都是合法的字符常量, 'ab', 'a' 都是不合法的字符型常量。



提示:

C++ 还提供了另一种表示字符型常量的方法,即所谓的“转义序列”。转义序列就是以转义符“\”开始,后跟一个字符或一个整型常量(字符的 ASCII 值)的办法来表示一个字符。若转义字符后边是一个整型常量,则必须是一个八进制或十六进制数,其取值范围必须在 0~255 之间。该八进制数可以以 0 开头,也可以不以 0 开头;而十六进制数必须以 0X 或 0x 开头。常用的转义序列字符有:“\\”表示字符“\”,“\”表示单引号,“\”表示双引号,“\n”表示换行符等。

【例 2-4】 设有声明语句:char a='\72';则变量 a _____。

A. 包含 1 个字符

B. 包含 2 个字符

C. 包含 3 个字符

D. 声明不合法

分析:转义字符常量“\xx”可以把“\”后面的数字 xx(如本题中的 72)转换为对应的 ASCII 字符。所以,该题的正确答案为 A。

(4) 字符串常量:用双引号括起来的若干个字符。例如:"ab3d", "a"都是合法的字符串常量。字符串常量与字符常量的存放格式不一样,在字符串常量的尾部,存储时都要存放一个 '\0', '\0' 是字符串的结束标记。



提示:

1) 当双引号要作为字符串中的一个字符时,必须采用转义序列表示法;单引号作为字符串中的一个字符时,可直接出现在字符串常量中,也可以采用转义序列表示法。

2) '\0' 也可写成 0,但不能写成 '0'。

(5) 标识符常量:定义标识符常量后,可以像字面常量一样使用该标识符。可用两种方法定义标识符常量:

① 使用编译预处理命令方法,如: #define PI 3.1415926 //一般无分号结尾

② 利用常量说明语句,如:const float PI=3.1415926; //一定要以分号结尾



提示:

在程序中标识符常量必须先定义后引用,并且标识符常量在程序中只能引用,不能改变其值。

2.3.2 变量

根据其取值的不同,分为不同类型的变量:整型变量、实型变量、字符型变量、构造型变

量和指针型变量等。

变量在使用前必须先用变量说明语句进行定义,定义的过程是向系统申请相应的存储空间,以准备存放数据。变量的定义格式如下:

<数据类型> <变量名>

int a; //定义整型变量 a,并为 a 分配 4 个字节的存储单元

变量要求说明在前,使用在后。变量说明语句可以出现在程序中语句可出现的任何位置,但同一变量只能作一次定义性说明。变量定义后,存储单元为一任意数,故首次引用变量时,变量必须有一个确定的值。

对变量赋初值的三种方法:

(1) 在变量说明的同时对变量赋初值,例如:char c1='b',char c2='c';

(2) 使用赋值语句赋值,例如:float s,t;s=3.4;t=5.6;

(3) 通过输入流 cin 对变量赋初值,例如:int b;cin>>b;

2.4 构造数据类型

2.4.1 结构体类型

前面描述的基本的数据类型(如整型、实型、字符型等)的特点是:当定义某一特定数据类型,就限定该类型变量的存储特性和取值范围。但是有些数据(如通讯录、住宿登记表等)是不能用同一种数据类型描述的;在通讯地址表中有姓名、电话号码、通讯地址、邮编、E-mail 等内容;在住宿表中通常有姓名、性别、身份证号码等内容。也就是说这些表中集合了各种不同数据类型的数据,无法用前面的任一基本的数据类型完全描述,因此 VC++ 引入一种能集不同数据类型于一体的数据类型,即结构体类型。

结构体是一种导出的数据类型,它由若干结构成员组成的,每个结构成员有自己的名称和数据类型。编译程序并不为任何数据类型分配存储空间。只有定义了结构体类型的变量时,系统才为这种变量分配存储空间。要定义结构体变量,必须首先说明结构体类型。

结构体数据类型说明格式为:

struct <结构体类型名>

{

<类型名> <变量 1>;

<类型名> <变量 2>;

.....

};

//此行分号不可缺少

结构体的成员可以是任何一种类型的数据,包括基本类型和构造类型。不能指定成员的存储类型为 auto、register、extern,但是可以指定其为 static。

结构体类型变量的定义格式为:

一般格式:<结构体类型名> <变量名 1>,<变量名 2>,...,<变量名 n>;



提示:

1) 注意结构体类型和结构体类型变量的概念。

2) 结构体类型与结构体类型变量:

必须先定义结构体类型,再定义结构体类型变量。定义结构体类型时不给成员分配存储空间,定义结构体类型变量时才真正为其分配存储空间。使用时必须通过变量来完成具体的操作。

3) 结构体变量的存储空间为其所有成员占用的空间之和。

【例 2-5】 结构体变量在程序执行期间_____。

A. 所有成员一直驻留在内存中

B. 只有一个成员驻留在内存中

C. 部分成员驻留在内存中

D. 没有成员驻留在内存中

分析:如前所述,该题的正确答案为 A。

2.4.2 共同体类型

和 2.4.1 所述类似,共同体类型也是一种导出的数据类型,由若干结构成员组成的,每个结构成员有自己的名称和数据类型。也同样需要先定义共同体类型数据类型,再定义共同体变量,共同体类型每个成员使用同一个存储空间。

共同体数据类型定义格式:

union <共同体类型名>

{

<类型名> <变量 1>;

<类型名> <变量 2>;

.....

};

//此行分号不可缺少

共同体类型变量的定义格式:

<共同体类型名> <变量名 1>,<变量名 2>,...,<变量名 n>;

【例 2-6】 共同体类型变量在程序执行期间_____。

A. 所有成员一直驻留在内存中

B. 只有一个成员驻留在内存中

C. 部分成员驻留在内存中

D. 没有成员驻留在内存中

分析:如前所述,该题的正确答案为 B。

【例 2-7】 以下程序的输出结果是_____。

#include <iostream.h>

union

{

char i[2];

int k;

};

void main()

```

{
    r.i[0]=2;
    r.i[1]=0;
    cout<<r.k<<endl;
}

```

分析:根据共用体的定义可知:共用体变量 r 的成员 k 和成员 $i[2]$ 是共用同一段内存空间,所以,当程序给 $r.i[0]$ 、 $r.i[1]$ 赋值后,共用体成员 k 的值也确定了;又因为 k 占 2 个字节,加上分别是一个字节的 $i[0]$ 和 $i[1]$ 也共同使用这 2 个字节,所以 k 的值为 2。

共同体和结构体的区别见表 2-1 所示:

表 2-1 共同体和结构体的异同比较一览表

	相同点	不同点
结构体变量	它们都是一种新的数据类型,都必须先定义这种新的数据类型,再定义具有这种数据类型的变量。并且系统不会为这种导出的数据类型分配内存空间,只会为具有这种数据类型的变量分配内存空间。	其分配的存储空间为其所有成员所占的空间之和,各个成员都有自己的存储空间,在任一时刻,所有成员都是有效的。
共同体变量		其分配的存储空间为其成员中最大的一个所占的空间,所有成员使用同一个存储空间,因此在某一时刻,只有一个成员是有效的。

2.4.3 枚举类型

类似 2.4.1 和 2.4.2,枚举类型也是一种导出的数据类型,基本格式:

```
enum <枚举类型名>{ <枚举量表> };
```

<枚举量表>由逗号隔开的标识符组成,这些标识符称为枚举元素或枚举常量。

枚举类型变量的定义,与结构体变量的定义方法类似,如:

```
enum weekday{Sun,Mon,Tue,Wed,Thu,Fri,Sat};
```

/* weekday 有 7 个元素,这 7 个元素的名字分别为 Sun、Mon、Tue、Wed、Thu、Fri、Sat,但实际存储整型值,默认从 0 开始,依次加 1。如本例中的元素的值分别为 0、1、2、3、4、5、6 */

定义枚举类型时也可以指定元素的值为一整型数。如:

```
enum weekday{Sun=9,Mon=2,Tue,Wed,Thu,Fri,Sat};
```

/* 这时 Sun、Mon、Tue、Wed、Thu、Fri、Sat 对应的整型值分别为 9、2、3、4、5、6、7 */

【例 2-8】 以下对枚举类型名的定义中正确的是_____。

- A. enum a={one,two,three}
- B. enum a{one=9,two=-1,three}
- C. enum a={"one","two","three"}
- D. enum a{"one","two","three"}

分析:如前所述,声明枚举类型用 enum 开头,且对枚举元素按常量处理。所以,该题的