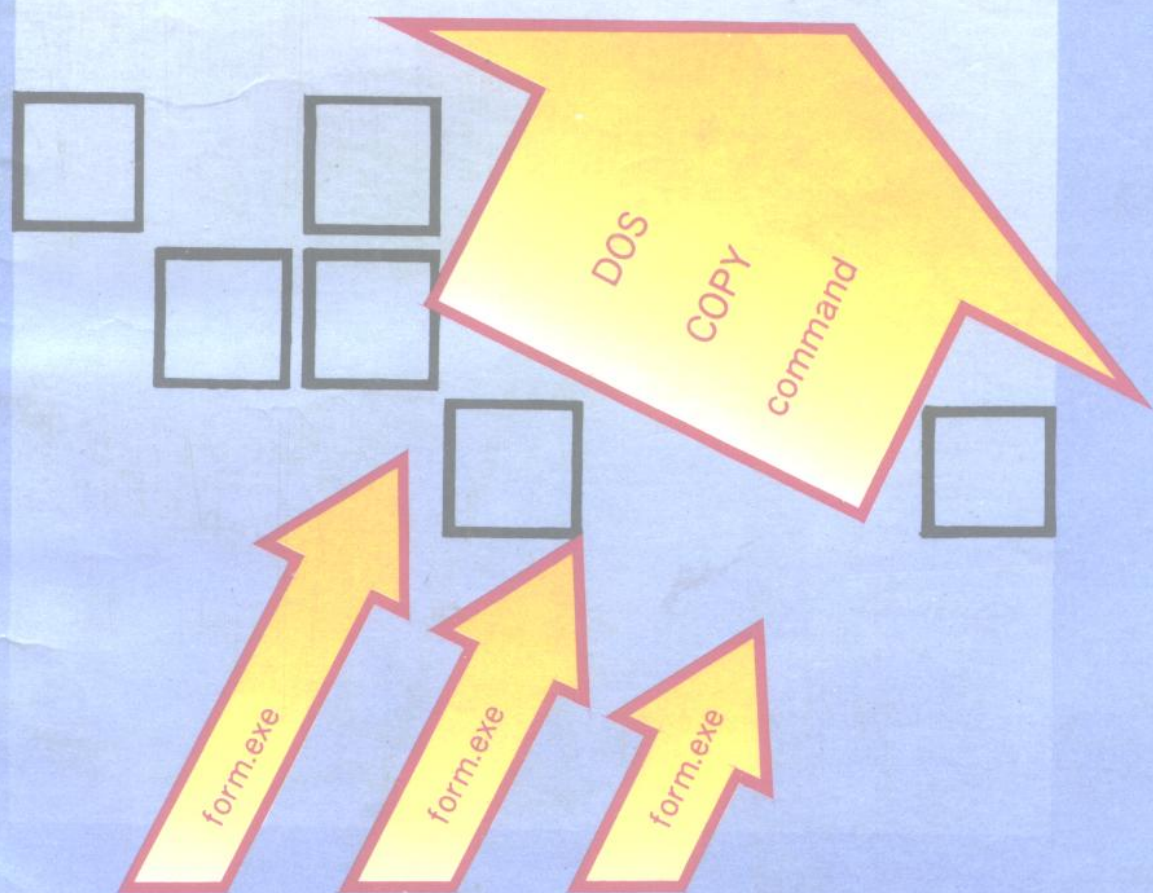


最新 Borland C++ 实用教程 3

Borland C++ 编程实例集锦

求实编著



科学出版社

最新 Borland C ++ 实用教程 3

Borland C ++ 编程实例集锦

求 实 编 著

科 学 出 版 社

1 9 9 4

(京)新登字 092 号

内 容 简 介

本书包括 Borland C++ 和 Turbo C++ 基础子程序,用 C++ 开发动画、卡通制作、高级图形设计、鼠标、键盘的实例程序,以及窗口设计,交互式绘图,二、三维图形技术,CAD 程序实例。

这些程序均可直接上机运行,且每一程序都附有剖析的说明,以便于读者学习和引用,减轻编程工作量。这些程序还可改编成 DLL 动态接库,供经常使用。本书与本教程 1,2 配合使用,将提高读者的学习速度,更快地掌握 Borland C++ 编程技巧。

本书可供程序设计和开发人员、高等院校软件专业的师生参考。

最新 Borland C++ 实用教程 3 **Borland C++ 编程实例集锦**

求 实 编 著

责任编辑 童安齐

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

北京市通县燕山印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1994 年 5 月第 一 版 开本:787×1092 1/16

1994 年 5 月第一次印刷 印张:33 1/2

印数:1—5 000 字数:780 000

ISBN 7-03-004044-9/TP·342

定价:29.00 元

前 言

本书是最新 Borland C++ 实用教程之三,系统地介绍了利用 Borland C++ 进行程序设计的实例和由浅入深的设计方法。

本教程分两部分,第一部分包括第一章到第九章,第二部分包括十章到第二十一章。

第一部分用大量短小实用的实例对 C++ 程序设计所要涉及的几乎所有方面进行了逐个介绍,以期给初学者更多的感性认识,同时培养读者的程序设计经验,从而使他们在短时间内达到较高的 C++ 程序设计水平。

这些内容涉及程序流程控制、数据类型定义、重载函数、类定义及对象声明、继承、虚函数、支元函数及文件操作等,都是读者最关心的内容。

在第一部分的基础上,第二部分用程序包的形式,以图形程序设计为核心介绍了高级 C++ 程序设计技术,以期将读者的程序设计能力提高到中高级水平,并使读者在读完本教程后能具有独立编写中大型应用程序的能力。

第二部分自成体系,从 BGI 图形接口入手,首先介绍各种绘图函数、字体,引导读者学习设计复杂图表,然后对动画设计技术进行深入讨论,围绕鼠标、图标和弹出式窗口程序设计,逐步构造一个大型三维 CAD 程序。读者通过对本书的研究,结合有关资料,定能独立设计优秀的三维动画系统。

作为 Borland C++ 系列教程之三的本书是按初学者学习面向对象程序设计的思路,由浅入深逐步展开的。它给出了初学者所应掌握的大量程序设计技术。

本实用教程之一、二、四分别针对 Borland C++ 入门知识、面向对象的 Windows 高级程序设计技术、类库及工具库等主题进行讨论。对希望进一步提高 C++ 程序设计技术的读者来说,本实用教程是一套不可多得的参考资料。

本书第一至第九章由李梅、李敏、朱尽染、李宗浩、刘秀芳、陆静宜、刘诚、张荣军、李蕾、李宁执笔;第十章至第十五章由李宗文、朱志岩、何大庆、阎继忠、李波、车墩仁、刘崇福、魏洪、韩亮执笔;第十六章至第二十一章及附录 A 由刘秀英、魏宁、覃社庭、朱志勇、蔡萌、欧阳琼、孟莎执笔。此外,魏忠才、程功对全书的结构作了合理的安排,诗颖、白琴为本书的编排付出了辛勤的劳动。在此一并深表谢意。

目 录

第一部分 Borland C++ 初级程序设计技术

第一章 程序流程控制	2
1.1 关系操作符	2
1.2 for 循环	3
1.3 域宽函数 setw()	6
1.4 for 循环嵌套	7
1.5 while 循环	9
1.6 while 循环嵌套	12
1.7 getche()与 getch()	13
1.8 do 循环	14
第二章 用户定义的数据类型	18
2.1 typedef	18
2.2 结构数据类型	19
2.3 嵌套结构	23
2.4 几个关于结构的例子	25
2.5 union	29
2.6 enum	31
第三章 函数	35
3.1 函数定义	35
3.2 以常量作为函数参数	37
3.3 以变量作为函数参数	39
3.4 以结构作为函数参数	40
3.5 使用 return 返回函数值	42
3.6 引用调用	43
3.7 基本数据类型的引用调用	43
3.8 以结构为引用参数的函数调用	46
3.9 重载函数	47
3.10 嵌入函数	51
3.11 缺省参数值的函数声明	52
第四章 类与对象	54
4.1 类定义与对象声明	54
4.2 构造函数	59
4.3 析构函数	63
4.4 在类外部定义成员函数	64
4.5 构造函数的重载	66

4.6	以对象作为函数参数	68
4.7	返回值为对象	70
4.8	结构与类的区别	73
第五章	重载操作符	75
5.1	单目重载操作符	75
5.2	具有返回值的重载操作符	78
5.3	双目重载操作符	81
5.4	比较重载操作符	86
5.5	设置特殊操作符的重载	88
5.6	不同数据类型的转换	90
5.7	不同类型的类的转换	92
5.8	等号重载操作符	98
5.9	前置运算和后置运算	100
5.10	综合应用实例	102
第六章	继承	106
6.1	继承	106
6.2	基类与派生类	106
6.3	private 与 public 继承关系	111
6.4	派生类的构造函数定义	113
6.5	派生类成员函数同名定义	116
6.6	含基类构造函数的构造函数声明	118
6.7	类层次	121
6.8	多重继承	124
6.9	嵌套类	128
第七章	指针	133
7.1	指针地址、地址运算符	133
7.2	指针变量	134
7.3	存取指针变量所指的值	135
7.4	指针与数组	136
7.5	指针与函数	138
7.6	指针与气泡排序	140
7.7	指针与字符串	145
7.8	指针数组与字符串	148
7.9	内存分配函数 new 与释放函数 delete	149
7.10	new,delete 与类	150
7.11	指针与结构	152
7.12	指针与对象	153
第八章	虚函数、友元函数与 this 指针	156
8.1	静态联编与动态联编	156
8.2	虚函数	158
8.3	纯虚函数	159
8.4	纯虚函数的应用	161

8.5	友元函数	164
8.6	友元类	166
8.7	this 指针	167
8.8	用 this 指针返回值	169
第九章	文件	172
9.1	文件类简介	172
9.2	字符串 I/O	172
9.3	字符 I/O	176
9.4	fstream 输入/输出文件对象	179
9.5	对象 I/O	186
9.6	文件指针	189
9.7	其它形式的文件	192

第二部分 Borland C++ 高级程序设计

第十章	Borland 图形接口(BGI)	195
10.1	初始化 BGI	195
10.2	编写基本的 BGI 程序	196
10.3	错误检查措施	197
10.4	使用坐标	198
10.5	绘图命令	199
10.6	切割成型的风景画	209
第十一章	BGI 绘图函数	214
11.1	像素级绘图	214
11.2	绘图命令综述	217
11.3	动画基础	222
11.4	区域填充	224
第十二章	BGI 字体和正文	233
12.1	图形模式的正文	233
12.2	Borland C++ 如何存取字体	236
12.3	建立定制的字体	238
12.4	放大字符	243
12.5	显示字符和数码	247
12.6	扩展的正文处理例程	247
12.7	使用正文输入	249
第十三章	表示图	255
13.1	基本图形类型	255
13.2	动画图	270
第十四章	动画	273
14.1	间隔化	273
14.2	在背景上动画化对象	281
14.3	用调色板动画化	285
14.4	使用多重内存页面	290

第十五章 创建鼠标工具包	292
15.1 鼠标的使用	292
15.2 访问鼠标驱动程序	293
15.3 鼠标函数	294
15.4 增添键盘输入	300
15.5 测试鼠标	317
第十六章 使用图标	319
16.1 为什么使用图标	319
16.2 表示图标	319
16.3 保存图标	320
16.4 读图标文件	321
16.5 交互式编辑程序	321
第十七章 弹出式窗口	333
17.1 基本方法	333
17.2 使用窗口程序包	343
17.3 测试程序	343
第十八章 交互式绘图工具	346
18.1 交互式图形程序包	346
18.2 擦除	352
18.3 喷涂效果	352
18.4 画线	353
18.5 画多边形	354
18.6 画矩形	355
18.7 画圆	356
18.8 画椭圆	357
18.9 画弧	35
18.10 杂项绘图支援	358
第十九章 绘画程序	377
19.1 绘画程序综述	377
19.2 绘画函数	380
19.3 下拉菜单	381
19.4 改变填充类型	381
19.5 交互作用	382
19.6 使用绘画程序	382
19.7 增强绘画程序	382
19.8 测试绘画程序	383
第二十章 CAD 程序	397
20.1 绘画和画图	397
20.2 画各种对象	400
20.3 复制函数	404
20.4 旋转命令	404
20.5 修改绘图次序	404

20.6	选择和移动一个对象	405
20.7	访问 gobjlist 中的成员函数	406
20.8	扩充 CAD 程序	406
20.9	编译 CAD 程序	407
第二十一章 三维图形		443
21.1	编译 3d.cpp 程序	443
21.2	测试三维程序	443
21.3	三维程序设计实例	444
附录 A C++ 初学者必读		460
A.1	封装	461
A.2	继承	463
A.3	多态性	463
A.4	重载	464
A.5	用类来模拟现实世界	464
A.6	再论继承	472
A.7	虚函数	483
A.8	动态对象	493
A.9	C++ 中更多的灵活性	498
A.10	用户定义数据类型的 I/O	510
A.11	下一步	511
A.12	更好的 C: 从 C 过渡	512
A.13	对象支持	515
A.14	小结	525

第一部分 Borland C++ 初级程序设计技术

在这一部分中，我们将结合大量实例由浅入深地讲解一些程序设计技术。由于本教程对最基本的程序设计基础已作过大量介绍，所以本书不再赘述，但为相对独立，在必要的地方，将略微详细地介绍一些基本概念。

下面就从程序的灵魂——流程控制开始讨论。

第一章 程序流程控制

程序语句的执行很少是直接从头执行到尾，这就好像人类的行为一样，有时要执行程序语句中的条件分支或要重复执行某些程序语句。分支或重复执行某些程序片段的语句称为流程控制语句（control statements），而决定执行哪些程序片段的条件则是那些包含关系操作符（relational operator）的表达式（取决于真还是假）。

1.1 关系操作符

C 及 C++ 没有提供二进制数据类型（如 TRUE, FALSE），用以比较两种数据类型（如 char, int, float 是用户自定义类型）的值是真还是假。真假是用数值表示的，0 表示包含关系操作符的表达式为假，非 0 值（如 1, -6, 28）则表示包含关系操作符的表达式为真。C++ 提供的关系操作符如下表所示：

关系操作符	含 义	实 例	逻辑值
>	大于	3 > 5	0
<	小于	13%5 < 5	1
=	等于	'A' == 'A'	1
!=	不等于	'A' != 'B'	1
>=	大于、等于	64/7 >= 5	1
<=	小于、等于	64/7 <= 5	0

程序实例 ch1_1.cpp

(1) 程序名称：ch1_1.cpp。

(2) 计算含关系操作符的表达式。

```
// Function Name: ch1_1.cpp
// Function: demo the relational operation
#include <iostream. h>    // for cout, cin
main ( )
{
    int quantity;          // define variables
    cout << " Input a number:";
    cin << quantity;    cout << endl;
    cout << quantity << " > 10 return:"
        << (quantity > 10) << endl;
    cout << quantity << " % 13 > 10 return:"
        << ( (quantity % 13) > 10) << endl;
    cout << "'A' == 'a' return:"
```

```

        << ('A' == 'a');
    cout << endl;
}

```

输出:

```

Input a number: 29
29 > 10 return: 1
29 % 13 > 10 return: 0
'A' == 'a' return: 0

```

说明:

(1) $29 \% 13$ 的余数为 3, $3 > 10$ 不为真, 故返回 0。

(2) 以 ASCII 码的顺序为准比较字符。字符 'a' 的 ASCII 码为 97, 字符 'A' 的 ASCII 值为 65, 故 'A' < 'a', 返回逻辑值 1。

1.2 for 循环

C++ 提供三种重复执行控制结构, 即 for 循环、while 循环以及 do 循环。while 循环及 do 循环请参考下节说明。

对于 for 循环, 其触发的条件为已知的或者固定的循环次数。以下说明 for 循环的语法。

(1) 单一语句结构。

1) 语法:

```

for ( 初始表达式; 条件表达式为真时执行循环; 循环增量 )
    语句; <— 循环体

```

2) 例子:

```

for ( int i = 1; i < 10; i++ )
    语句;

```

(2) 多重语句结构。

1) 语法:

```

for ( 初始表达式; 条件表达式为真时执行循环; 循环增量 )
{
    语句 1;    <— 循环体
    语句 2;    <— 循环体
    ...
    语句 N;    <— 循环体
}

```

2) 例子:

```

for ( int i = 1; i < 10; i++ )
{
    x = y + z;
    x += ++y - z--
    ...
}

```

整个 for 循环可视为一条语句。只有整个 for 循环执行完毕，或当表达式不成立时才能执行 for 循环的下一条语句。

程序实例 ch1_2.cpp

(1) 程序名称: ch1_2.cpp。

(2) 在屏幕上的同一行显示 1 到 9 的三次方，算出的数值之间隔 3 个空白字符。

```
// Function Name: ch1_2.cpp
// Function: demo single FOR LOOP statement
#include <iostream. h>    // for cout
main ()
{
    for (unsigned int i=1; i<10; i++) // loop 1 to 9
        cout << i*i*i << "    "; // body of loop
    cout << endl;
}
```

输出:

1 8 27 64 125 216 343 512 729

说明:

(1) for 循环中的循环初始表达式为:

```
unsigned int i=1
```

数据类型 unsigned int 也可以在 for 循环之前定义，unsigned int（无符号整数）的数值表示范围为 0 到 65535。

(2) 除了类型声明之外，程序中不可重复对同一个变量的数据类型进行声明。例如，for 循环之前已定义 i 为整数，在 for 循环的循环初始表达式中不可另外定义其数据类型，故以下的语句是错误的:

```
int i;
for (unsigned i=1; i < 10; i++)
```

(3) 计算结果之间的空白可用含三个空白字符的字符串表示。

程序实例 ch1_3.cpp

(1) 程序名称: ch1_3.cpp (for 多重语句结构)。

(2) 由用户输入数值，在屏幕上依次列出 11 个数及其平方、立方。

```
// Function Name: ch1_3.cpp
// Function: demo multiple FOR LOOP statement
#include <iostream. h>    // for cout
main ()
{
    int n;
    cout << " Enter a number : ";
    cin >> n;

    // print cube list
```

```

for (int i=n; i<=n+10; i++)
{
    cout << i;
    cout << "      " << long (i) * i;
    cout << "      " << long (i) * i * i;
    cout << endl;
}
}

```

输出:

```

Enter a number : 1000
1000      1000000      1000000000
1001      1002001      1003003001
1002      1004004      1006012008
1003      1006009      1009027027
1004      1008016      1012048064
1005      1010025      1015075125
1006      1012036      1018108216
1007      1014049      1021147343
1008      1016064      1024192512
1009      1018081      1027243729
1010      1020100      1030301000

```

说明:

(1) 为尽可能容纳数值的三次方, 程序中使用 long 型整数, 例如:

```
cout << long (i) * i * i;
```

或

```
cout << long (i * i * i);
```

(2) for 循环体中的

```
cout << endl;
```

用来控制显示的换行。

(3) 注意 long 长整型数容纳的数值范围为:

-2147483648 到 2147483647

程序实例 ch1_4.cpp

(1) 程序名称: ch1_4.cpp。

(2) 计算第 n 个 (由用户输入赋值) fibonacci 数。

(3) fibonacci 数列为:

1 1 2 3 5 8 13 21 34 55 ...

定义:

fib (n) = 1 if n = 1 or n = 2

fib (n) = fib (n-1) + fib (n-2) if n > 2

// Function Name: ch1_4.cpp

// Function: demo multiple FOR LOOP statement

```

// calculate fibonacci number
#include <iostream. h>      // for cout
main ()
{
    int n;
    unsigned long lowfib = 0, hifib = 1;
    cout << " Enter a number : ";
    cin >> n;

    // calculate fibonacci (i)
    for (int i=2; i<=n; i++)
    {
        long x = lowfib;
        lowfib = hifib;
        hifib = x + lowfib;
    }
    cout << " fibonacci : " << hifib << endl;
}

```

输出:

D: \TCP\CHAP3> ch1_4

Enter a number: 7

fibonacci: 13

D: \TCP\CHAP3> ch1_4

Enter a number: 10

fibonacci: 55

D: \TCP\CHAP3> ch1_4

Enter a number: 45

fibonacci: 1134903170

1.3 域宽函数 setw()

cout 输出的数值或字符串均占有恰好能容纳该数值或字符串的域。为增强屏幕显示的美观性, 可以使用 setw (n) 来指定其后输出的数值或串所占用的域宽。若设置的域宽大于显示数值或字符串所需的域宽, 则由向右对齐显示。使用 setw 域宽操作符必须引用其原型定义文件<iomanip. h>。

程序实例 ch1_5. cpp

(1) 程序名称: ch1_5. cpp。

(2) 由用户输入的数值起显示 11 个数的平方及立方值。

(3) 使用 setw 操作符向右对齐显示各数值。

```
// Function Name: ch1_5.cpp
// Function: demo multiple FOR LOOP statement
// demo setw manipulator

#include <iostream.h>      // for cout, cin
#include <iomanip.h>        // for setw ()
main ()
{
    int n;
    cout << " Enter a number : ";
    cin >> n;
    // print cube list
    for (int i=n; i<=n+10; i++)
    {
        cout << setw (4) << i;
        cout << setw (12) << long (i) * i;
        cout << setw (12) << long (i) * i * i;
        cout << endl;
    }
}
```

输出:

```
Enter a number : 5

5      25      125
6      36      216
7      49      343
8      64      512
9      81      729
10     100     1000
11     121     1331
12     144     1728
13     169     2197
14     196     2744
15     225     3375
```

说明:

若不包含<iomanip. h>, 编译阶段将显示“未引用 setw 原型说明”的错误信息。

1.4 for 循环嵌套

C, C++ 允许 for 循环中嵌套其它 for 循环。一个完整的 for 循环可以视为单一的语句。以

下说明 for 循环的嵌套格式。

(1) 格式一。

```
for ( 初始表达式; 条件表达式; 循环增量表达式 )  
for ( 初始表达式; 条件表达式; 循环增量表达式 )  
    语句;
```

(2) 格式二。

```
for ( 初始表达式; 条件表达式; 循环增量表达式 )  
{  
    ...  
    for ( 初始表达式; 条件表达式; 循环增量表达式 )  
    {  
        ...  
        语句;  
        ...  
    }  
}
```

循环嵌套时, 必须注意每个内层循环的执行范围, 成对大括号内的语句即为 for 循环执行的主体。编写程序时使用缩进以及适当的程序注释就不容易产生逻辑错误。

程序实例 ch1_6.cpp

(1) 程序名称: ch1_6.cpp。

(2) 产生乘法表。

```
// Function name: ch1_6.cpp // Function: generates the multiplication table  
  
#include <iostream.h>  
#include <iomanip.h>  
  
main ()  
{  
    for (int rows=1; rows<10; rows++)    // outer loop  
    {  
        for (int cols=1; cols<10; cols++)    // inner loop  
            cout << setw (4) << cols * rows;  
        cout << endl;  
    }    // end outer  
}
```

说明:

- (1) 当外层循环的 rows 变量值为1时, 在第1行显示内层循环的执行结果, 即 rows 的值乘 cols (由1到9)。
- (2) 内层 for 循环体仅有一条语句, 即