

趣味
程序

导学

Visual C++

董未名 杨柳 汤筠 编著



清华大学出版社

趣味程序导学 Visual C++

董未名 杨 柳 汤 筠 编著

187513/CP

清华大学出版社

(京)新登字 158 号

内 容 提 要

本书通过编写趣味游戏程序来引导读者学习 Visual C++编程的方法和技巧,形式新颖活泼,别具一格。

全书从 Visual C++语言基础知识和编制简单的程序入手,将 Visual C++编程的知识点有机地分散在“幸运 52”,“速算 24”,“俄罗斯方块”,“拼图游戏”,“属于你的 OICQ”等多个趣味游戏的程序设计示例中,引导读者轻松学习 Visual C++编程的相关知识、编程思想及技巧,其中包括 Visual C++中消息处理、多媒体、图形图像、数据库处理以及网络编程等内容。

本书以示例教学方式组织内容,集趣味性、直观性和可操作性于一体,适用于 Visual C++初学者及对游戏程序感兴趣的电脑爱好者。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名:趣味程序导学 Visual C++

作 者:董未名 杨 柳 汤 筠

出版者:清华大学出版社(北京清华大学校内,邮编 100084)

印刷者:北京朝阳科普印刷厂

发行者:新华书店总店北京科技发行所

开 本:787×1092 1/16 印张:18.625 字数:453 千字

版 次:2002 年 6 月第 1 版 2002 年 7 月第 2 次印刷

印 数:5001~8000

盘 号:ISBN 7-900643-07-9

定 价:29.50 元(含光盘)

丛书总序

电脑游戏

“我喜欢游戏！”

“游戏是我生命的一部分”

“我是游戏的一部分”

这是许多玩家从开始玩电脑游戏，到喜欢，直到痴迷的三段自我写照。

当计算机技术给游戏提供了强有力的支持后，一个陌生而又似曾相识的新奇世界展示在人们面前：这里有逝去的童年梦想，有心头压抑已久的情感，有疯狂、神秘，有脑力和技巧的挑战，也有可以轻松获得的志得意满的“虚拟”成就感。游戏里有一个别样的人生，有一个神奇的世界。

娱乐、游戏是人的天性。无论关于游戏的各种观点怎样碰撞，年轻一代对电脑游戏的痴迷已经无法逆转。在不久的将来，我们将面对“玩游戏长大的一代”，甚至人们的思维方式也将受到游戏的很大影响。

程序设计

Java, JavaScript, Delphi, VB, VC, C++Builder……，窗口，图形界面，事件驱动，数据库，多媒体，网络编程……当我们编写的代码通过编译运行（或解释执行）产生奇妙的动态效果，当我们成功地编写了一个窗口程序，当我们亲自编写了一个哪怕是很粗糙的聊天工具，那一刻的成功、喜悦、振奋和激动都会让人无以言表。

计算机程序设计给我们带来了另一个精彩的别样世界。掌握和使用新的程序设计语言，学习和操作新的程序设计工具，认识和思考新的“信息世界”，不断吸收信息新知，是信息时代弄潮儿永远不知疲倦的一件赏心乐事。

熟悉一些流行的程序开发工具，掌握一定的程序设计方法，已经成为年轻一代所必须的素质，也是时代的要求。也许你还是一名中学生，也许你是一名大学生，或许你已经就业工作，作为一个跨世纪的现代人、21世纪的主人翁，我们有必要了解、掌握、驾驭一定的程序设计工具和程序设计语言。

通过趣味游戏程序学习程序设计

学习程序设计，并不是一件艰苦、枯燥的事情，它能像电脑游戏那样让你充满好奇、富有乐趣。这正是本丛书的编写目的！

本丛书面向初、中级用户，精选了目前全球最流行、最常用的程序设计语言和程序开发工具，通过趣味游戏示例，以目标式教学为主，引导读者学习、掌握程序设计思想和编程技巧。

本丛书努力做到如下几点：

- **趣味性：**以趣味游戏程序为例，形式新颖活泼，读者在学习的过程中能自己动手设计电脑游戏，感受学习的乐趣，保持学习的兴趣。本丛书均带有光盘，在光盘中给出了全部示例的源代码和各种资源文件，读者可以分析、参考和学习。
- **直观性：**将程序设计的知识点有机地分散在多个趣味游戏的设计示例中，使得程序设计语言众多的对象、属性、方法以及程序开发工具的各种设置和操作都变得具体、

形象、直观，通俗易懂，深入浅出。

- **可操作性**：以示例教学、目标式学习来组织内容，将程序设计的思路、操作步骤、知识点和方法的讲解紧密结合，互相映证。本套丛书力求做到结构明晰，容易理解，便于操作，读者可以跟随书本，一边思考、体会程序设计的思路，一边一步步进行实际的操作，并及时从操作情况和程序执行的效果中得到反馈，带着目的学习，带着问题学习，有的放矢，从实际的操作、具体的设计中体会、领悟、积累程序设计的知识、技能和经验，这将极大地提高学习效率，达到更好的学习效果。
- **循序渐进**：本丛书尤其注意由浅入深，循序渐进，让读者的学习是一个轻松渐进、平衡上升的过程。每本书首先都从基础讲起，读者在一开始可以是一个完全的门外汉；随着学习的深入，将被一步步领进门，登堂入室，渐入佳境，最后从入门达到提高的目的。

我们将电脑游戏和程序设计这两个精彩世界有机地嫁接在一起，希望读者能在充满趣味的编程过程中，掌握程序设计语言，领悟程序设计的方法和技巧。

学习建议

本丛书以示例为主，注重操作性，将程序设计各方面的知识点有机地分散在各游戏的设计步骤中，在使用本书时，最好使用如下方法：

（1）在实际的操作中学习

本丛书实战性非常强，读者最好一边阅读，一边上机，两者紧密结合。一定要亲自动手，体会实际的操作过程，查看程序运行的效果反馈，并及时思考、总结。每学完一章，我们應該有自己的收获，动手编制出自己的游戏作品，同时理解、掌握程序设计过程中所用到的知识和技能。

（2）发挥主观能动性，积极思考

本书循序渐进，每个游戏侧重于程序设计的一个方面，在实际的设计过程中，又分为很多步骤。在每一步，读者应充分发挥自己的主观能动性，积极思考，尽量先有自己的思路，甚至给出自己的解决方法，然后再看书中的实现方法，并进行分析和比较，深入理解程序设计的精髓。

（3）借助于网络结成学习共同体

21世纪是一个信息社会，学习者不再是封闭、孤立的个体，而应该尽量借助网络来和其他学习者、专家进行沟通、协作，以积极寻求帮助和互助，提高学习效率。

本套丛书由北京高校计算机图书创作联盟策划、创作和编写。联盟主要由清华大学、北京大学等高校的研究生组成，成员有很强的计算机技术背景和丰富的实践经验。以团队协作、大胆创新的精神为宗旨，以认真负责、严谨细致的态度，努力创作真正切合广大电脑应用学习者需要的计算机精品图书。

最后，感谢科海培训中心夏非彼老师对联盟的关心。从联盟的最初构想、初创时起，夏老师就给予了积极的支持、热心的帮助和非常有价值的指导。感谢科海培训中心张红编辑对本套丛书提出的修改意见和建议，使我们的工作能够得以顺利地进展。

北京高校计算机图书创作联盟

前 言

Visual C++是目前PC机上最优秀的Windows应用程序开发环境。它把完全的可视化与真正的面向对象及C++的高效率、高性能完美地结合起来。面向对象是C++的核心。面向对象的软件开发方法起源于20世纪80年代初,但自20世纪90年代以来,才得到迅猛发展,成为当前最重要的一种软件开发方法。目前世界上大多数大学在讲授面向对象的课程时,一般都以C++语言作为基础。读者在阅读本书时,应紧紧抓住面向对象这一核心思想,只有真正理解了它,才能深刻领会Visual C++的编程思想,掌握其精华。

本书的内容

本书以示例为主,第1章介绍了Visual C++的基础知识,然后,每章将介绍一个或多个精心制作的趣味游戏,它们各自侧重于应用Visual C++语言的某些特性,循序渐进,讲述了应用Visual C++编程的方法和技巧。

在第2章,通过编制一个对流行的电视节目“幸运52”进行模拟的小游戏,介绍了Visual C++的初步应用,重点讲述了如何生成源代码基本框架及管理对话框的类、控件的添加、消息处理函数的定义、图片资源的引入、焦点控制及如何对用户的意外操作进行响应,让读者获取有关Visual C++程序设计的感性认识。

在第3章,通过“速算24”的扑克游戏讲述了Visual C++中的数学运算,并讲述了如何发牌、如何随时显示用户的输入、如何进行运算等多方面的内容。学完这一章,读者即可熟练掌握在Visual C++中进行数学运算及相关处理的方法和技巧。

第4章编写的“拼图”游戏是和文曲星里的拼图一样的一个游戏。这一章讲述Visual C++中位图的基本结构、位图资源的读入和显示、图格的移动、游戏完成条件的判断、帮助画面的添加、提示信息的显示及游戏计时器的加入等方面的内容。

第5章讲述媒体播放器——多媒体程序的设计,内容包括:在按钮上显示位图、对话框背景的添加、媒体播放类的创建、MIDI文件、Wave文件、CD的播放和控制、音响效果的显示和音量控制及如何使用ActiveMovie控件制作媒体播放器等内容。

第6章主要介绍了Visual C++数据库编程基础、数据库的创建、数据的添加、字段的定义、Visual C++与数据库的接口、用户DSN的设置、记录集的操作、MFC基本控件响应等内容。

第7章通过编写一个“俄罗斯方块”游戏,来综合介绍Visual C++语言编程。游戏设计过程中涉及到游戏框架、游戏界面的编写、背景的显示、方块的显示和控制、如何截获键盘的操作、如何显示成绩和排名、图形按钮和游动字幕的制作、数字的特殊效果显示等内容。通过这一章,读者可以领略到Visual C++的整体编程风格。

第8章讲述如何利用Visual C++创建网上聊天程序，主要介绍了网络的基本知识、动态链接库的生成、创建基于TCP协议的Socket类、在DLL中添加CTCP Socket、Winsock、连接的建立、函数映射等内容。

本书很好地贯彻了这套丛书选题、策划时的初衷，正如丛书总序中所说，努力做到以下几点：

- 趣味性
- 直观性
- 可操作性
- 循序渐进

我们将电脑游戏和程序设计这两个精彩的世界嫁接在一起，希望读者能在充满趣味的学习过程中，轻松地入门，尝试编程的乐趣，以便尽早掌握这一现代编程工具。

本书所附光盘说明

随本书带有一张光盘。光盘中含有本书中涉及到的全部示例的源代码及各种资源文件，以方便读者在学习过程中查阅、参考。

编者
2002年5月

目 录

第1章 初识Visual C++	1
1.1 什么是Visual C++	1
1.2 C++的新特性	1
1.3 面向对象简介	3
1.3.1 基本概念	3
1.3.2 继承和多态	10
1.4 VC++集成开发环境简介	12
1.4.1 AppWizard工具	12
1.4.2 工程和工程工作区	12
1.4.3 Class Wizard工具	13
1.4.4 Wizard Bar工具栏	14
1.5 创建第一个工程	14
1.5.1 生成一个基于文本框的工程	14
1.5.2 生成一个基于对话框的工程文件	18
1.6 运行工程文件	20
1.6.1 基于文本框的程序	20
1.6.2 基于对话框的程序	22
1.7 Microsoft基本类库与应用程序框架	23
1.7.1 什么是Application Framework	23
1.7.2 为什么要用Application Framework	24
1.7.3 Microsoft Foundation Class(MFC)与VC++	24
1.7.4 纵观MFC	24
1.7.5 怎样才能学好MFC	24
1.7.6 用Application Wizard生成的程序的结构	26
1.8 本章知识点回顾	27
第2章 “幸运52”游戏——Visual C++ 初步应用	29
2.1 “幸运52”游戏简介	29
2.2 设计初始界面	31
2.2.1 生成源代码基本框架	31
2.2.2 添加控件并设置其属性	31
2.2.3 生成管理对话框的类, 定义成员变量	34
2.2.4 定义消息处理函数	35
2.2.5 引入图片资源	35

2.3 编写程序代码	35
2.4 完善游戏界面	39
2.4.1 焦点控制: SetFocus方法	39
2.4.2 对用户的意外操作进行响应	42
2.5 本章知识点回顾	44
第3章 “速算24”游戏——Visual C++中的数学运算	46
3.1 设计初始界面	47
3.1.1 生成基本框架源代码	47
3.1.2 生成管理对话框的类, 定义成员变量	48
3.1.3 定义消息处理函数	48
3.1.4 引入图片资源	48
3.2 编写程序代码	48
3.3 完善游戏界面	56
3.3.1 不同时期在按钮上显示不同文字	56
3.3.2 增加计时功能	57
3.4 本章知识点回顾	58
第4章 拼图游戏——Visual C++位图操作	59
4.1 游戏效果说明	59
4.2 创建初始界面	59
4.3 位图的读入	62
4.3.1 Windows位图的基本结构	62
4.3.2 位图资源的读入	64
4.3.3 自定义位图文件的读入	66
4.4 用Static控件显示位图	69
4.4.1 设置Static控件的初始位置	69
4.4.2 图格的显示	74
4.5 图格的移动	80
4.6 编写游戏的启动代码	86
4.7 游戏完成条件的判断	88
4.8 游戏的进一步完善	90
4.8.1 添加帮助画面	90
4.8.2 用Status Bar显示提示信息	94
4.8.3 游戏计时器的加入	98
4.9 本章知识点回顾	99
第5章 媒体播放器——多媒体程序设计	103
5.1 程序效果说明	103
5.2 创建初始界面程序	104

5.2.1 在按钮上显示位图	105
5.2.2 菜单项位图的显示	107
5.2.3 对话框背景图的添加	108
5.3 媒体播放类的创建	109
5.3.1 高级音频函数	109
5.3.2 Windows MCI与多媒体软件开发	111
5.4 MIDI文件的播放和控制	114
5.4.1 MIDI简介	114
5.4.2 MIDI文件格式	115
5.4.3 MIDI文件的播放	116
5.4.4 MIDI文件的控制	122
5.5 Wave文件的播放和控制	125
5.5.1 Wave文件格式简介	125
5.5.2 Wave文件的播放和录音	127
5.6 CD的播放和控制	128
5.7 AVI文件的播放	130
5.7.1 AVI数字视频的格式	130
5.7.2 AVI数字视频的特点	131
5.7.3 AVI文件的播放	132
5.8 其他媒体文件简介	132
5.9 媒体播放类的使用	133
5.10 音响效果显示和音量控制	137
5.10.1 音响效果的显示	137
5.10.2 音量的控制	143
5.11 用ActiveMovie控件制作媒体播放器	145
5.11.1 建立工程	146
5.11.2 添加代码	146
5.12 DirectSound简介	148
5.13 本章知识点回顾	149
第6章 北京市公交查询系统——数据库编程基础	153
6.1 系统使用说明	153
6.2 数据库基础知识	154
6.3 使用Microsoft Access创建数据库	155
6.3.1 初识Access	156
6.3.2 选择关系并定义字段	157
6.3.3 添加数据	158
6.4 VC与数据库接口	159
6.4.1 用户DSN设置	159

6.4.2 ODBC标准	162
6.4.3 接口实现	163
6.5 记录集操作	169
6.5.1 使用ODBC记录集	169
6.5.2 用SELECT打开一个ODBC记录集	174
6.6 MFC基本控件消息响应与系统完善	177
6.6.1 在组合框内选择车次并显示路线信息	177
6.6.2 在编辑框内输入需要查询的车站并显示路线信息	184
6.6.3 完善界面	187
6.6.4 其他	188
6.7 主要部分源代码	188
6.8 本章知识点回顾	191
第7章 俄罗斯方块游戏——Visual C++应用深入	193
7.1 游戏效果说明	193
7.2 创建界面的主框架	194
7.2.1 用ClassWizard生成CPropertySheet	195
7.2.2 CPropertySheet类成员	196
7.2.3 成员函数	197
7.3 显示背景	201
7.4 方块的显示和控制	215
7.4.1 显示窗口	215
7.4.2 定义方块的数据结构	216
7.4.3 方块的显示	222
7.4.4 截获键盘操作	223
7.4.5 计时器	224
7.5 显示成绩和排名	226
7.6 制作图形的按钮	230
7.7 数字的特殊效果显示	238
7.8 用ActiveX美化界面	242
7.9 游动字幕About Box和说明的制作	245
7.10 本章知识点回顾	254
第8章 属于你的OICQ——Visual C++ 网络编程	256
8.1 程序效果说明	256
8.2 生成动态链接库（DLL）	257
8.3 创建基于TCP协议的Socket类	259
8.3.1 WinSock介绍	259
8.3.2 在DLL中添加CTCPSocket类	264
8.3.3 成员变量及其说明	264

8.3.4 成员函数及其说明	266
8.3.5 建立连接	267
8.3.6 连接方连接函数	274
8.4 两人聊天的OICQ	277
8.4.1 用AppWizard建立工程.....	277
8.4.2 生成用户界面	279
8.4.3 加入所需变量	280
8.4.4 编写初始化函数	281
8.4.5 进行函数映射	281
8.5 本章知识点回顾	284

第1章 初识 Visual C++

本章我们将简单介绍Visual C++，让读者对它有一个初步认识，为以后进一步学习和使用Visual C++打下良好的基础。本章是全书的基础，所介绍的内容比较多，但都是学习Visual C++编程所必备的知识。

1.1 什么是Visual C++

Visual C++是指由Microsoft公司开发的可视化集成编程软件Microsoft Visual Studio成员之一，目前最高版本是6.0。Microsoft Visual C++以C++语言为基础，并结合MFC进行编程。

长期以来，Microsoft Windows操作系统一直占据着个人计算机操作系统的主导地位，因此，Microsoft Visual C++受到越来越多的编程爱好者的青睐。

1.2 C++的新特性

Visual C++是以C++语言为基础的，很多读者可能都学过C语言，但是对C++并不是很熟悉。下面我们简单介绍C++的新特性。

(1) 注释语句：除了可以用/*和*/外，行注释还可以用//。

(2) 声明语句：在C中变量的声明只能在程序块开头，但是在C++中，局部变量的声明可以放在程序中的任何位置，只要是变量的首次声明即可。

(3) 作用域操作符 (::)：在C中作用域内的变量将覆盖同名的作用域外的变量，但是在C++中，也可以访问同名的作用域外的变量，只要加上作用域操作符 (::) 即可。例如：

```
double a;//全局变量a
void main()
{
    int a ; //局部变量a
    a=5;    //局部变量赋值
    ::a=10 //全局变量赋值
}
```

(4) 默认参数值：C++在定义函数时可以定义一些参数的默认值来简化编程。例如下

面代码行

```
void ShowMessage (char *Text,int Length = -1,int Color = 0)
```

中就定义了参数Length, Color的默认值。

(5) 引用类型: 声明为引用的变量是另一变量的别名。可用&操作符声明引用, 例如:

```
int count = 0;
int &rencount = count;
```

在这段代码中rencount声明为int型引用, 并初始化为int型变量count, 这个定义使rencount成为count的别名, 即rencount和count指向同一内存地址。

(6) 函数和引用: 引用类型也可以用于函数, 例如如下代码:

```
FuncA(int &parm)
{
    ++parm;
}

FuncB(int parm)
{
    ++parm;
}

void main()
{
    int N = 0;
    FuncA(N);
    //N equals 1;
    FuncB(N);
    //N still equals 1
}
```

函数A中的变量parm是int型引用, 所以函数A中的语句++parm将修改实际变量N的值, 因为其实parm只是N的一个别名, 而在函数B中参数parm只是一个新创建的内部变量, 由函数将变量N的值传给它, 所以++parm语句并不修改实际变量N的值。

(7) 常量: C++中可以用const定义常量, 例如:

```
const int a = 100;
```

(8) new和delete操作符: 在C++中可以用new来为一个变量分配内存空间, 用delete来释放一个不再使用的变量的内存空间。

(9) 面向对象机制: C++是既面向过程又面向对象的编程语言, 所以C++具有所有面向对象语言的特性。

1.3 面向对象简介

1.3.1 基本概念

下面我们介绍类、对象、构造函数、析构函数等基本概念。

1. 类和对象

在介绍C++中的类之前，先介绍C语言中的结构（struct），因为C++中的类是从C语言中的结构演化而来的。在C语言中，可以如下所示定义一个结构：

```
struct TPoint
{
    int x;
    int y;
}
```

上述代码中struct TPoint是一种自定义数据类型。事实上，它和其他的数据类型一样，现在编译器并没有给它分配任何空间，它仅仅是一种与int、long等类似的数据类型，可以用它来定义数据实例。例如：

```
struct TPoint pLeftTop={0, 0};
struct TPoint pBottomRight={800, 600};
```

如果使用typedef，那么代码就会简单明了：

```
typedef struct TPoint
{
    int x;
    int y;
}POINT;
```

这时可以使用POINT来代替struct TPoint。例如：

```
POINT pLeftTop={0, 0};
POINT pBottomRight={800, 600};
```

pLeftTop, pBottomRight称为实例（instance），它们相当于C++中的对象，编译时编译器会给它们分配一定的内存空间。

在C++中用类和对象（object）来代替上面的结构和实例。请看下面的例子：

```
class TPoint
{
    public;
    int x;
    int y;
};
```

可以用下面的代码来创建和使用对象：

```
TPoint p1; //没有初始化
p1.x=0;
p1.y=0;
cout<<'x*y='<<p1.x*p1.y;
```

上面的x和y称为类的元素或成员，类的成员可以是其他的对象甚至它本身的对象（这时必须是指针），也可以是函数。事实上，类是对象和用来建立、操作、撤消这些对象的函数集合，这种函数常常称为方法，这些方法可以访问所有的数据成员。也就是说C++中的类可以包含函数这样的数据成员。不论是函数还是数据成员，默认都是公有的（public），即可以被外部直接访问。

2. 构造函数

从上面可以看出，通过赋值语句来初始化对象的数据成员使很繁琐的，实际上，对于简单的类，可以用同结构实例相同的初始化方法来初始化类的对象。例如：

```
TPoint p1={0,0};
```

但是大多数情况下类不是这么简单的，它可能需要根据不同的情况来进行初始化。C++中的类可以包含成员函数和数据成员，成员函数可以访问类的所有数据成员。这些函数中包含一个或多个特定成员函数——构造函数，它们被用作初始化对象，构造函数定义前没有返回值（其实构造函数是有返回值的，这个返回值就是构造完整后的对象的引用），构造函数可以重载，即类的构造函数可以有多个，他们通过不同的参数表来区分。如果类的定义中不带任何构造函数，那么编译器将产生一个不带任何参数的默认构造函数，对于任何一个作为类的数据成员的对象来说，编译器会调用默认的构造函数。当编译器使用默认构造函数时，会为其对应的类对象分配空间，但并不对其内部类型的值进行初始化。例如上面的例子，可以按照下面的方法加入构造函数：

```
//-----
#include<iostream.h>
#include<conio.h>
//-----
class TPoint
{
public:
    int X;
    int Y;
    TPoint(int x,int y)
    {
        X=x;
        Y=y;
    }
};
```

```
//-----  
void main()  
{  
    TPoint p1(10,10);  
    cout<<"p1: "<<p1.X<<"*"<<p1.Y<<"\n";  
    getch();  
}  
//-----
```

运行结果如下:

p1: 10*10

如果把构造函数去掉, 让编译器调用默认的构造函数, 看看能得到什么结果:

```
//-----  
#include<iostream.h>  
#include<conio.h>  
//-----  
class TPoint  
{  
    public:  
    int X;  
    int Y;  
    /*  
    TPoint(int x, int y)  
    {  
        X=x;  
        Y=y;  
    }  
    */  
};  
//-----  
void main()  
{  
    TPoint p1;  
    cout<<"p1: "<<p1.X<<"*"<<p1.Y<<"\n";  
    getch();  
}  
//-----
```

运行结果为:

p1:256*1

显然这个结果是随机的, 这说明编译器并没有给类赋初值。上例中, 把构造函数声明