高等学校计算机专业规划教材

基于MFC的可视化数据结构

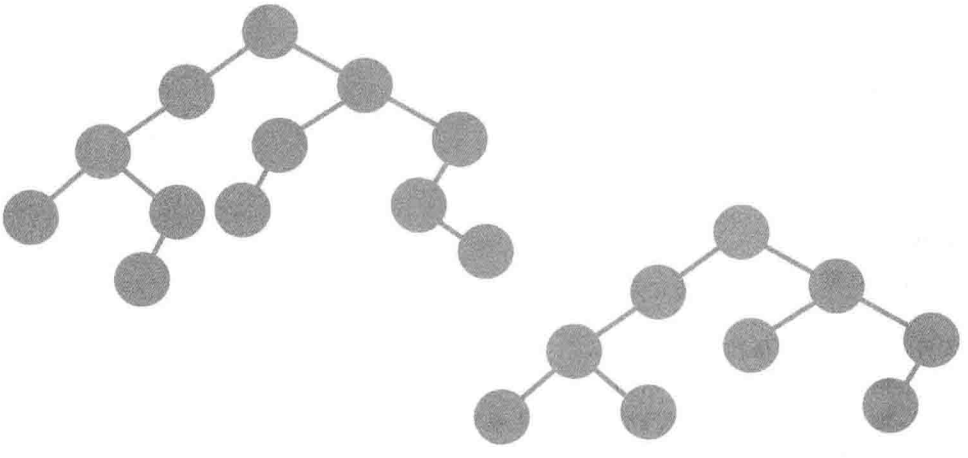
连远锋 李国和 张秀美 赵旭霞 吴双元 编著

清华大学出版社

高等学校计算机专业规划教材

基于MFC的可视化数据结构

连远锋 李国和 张秀美 赵旭霞 吴双元 编著



清华大学出版社
北京

内 容 简 介

本书首先简要介绍 MFC 程序设计的基础知识，然后系统地介绍线性表、栈与队列、树与二叉树以及图等数据结构，介绍了各种数据结构的逻辑关系、存储表示，基于面向对象语言 Visual C++ 的 MFC 平台，详细阐述可视化线性结构、可视化树结构、可视化图结构和可视化排序的设计思想和实现方法，使读者循序渐进地理解数据抽象、面向对象思想和可视化程序设计等现代化软件设计风格，理解并掌握数据结构知识体系，提高应用 Visual C++ 解决实际问题的能力。

本书内容丰富，重点突出，概念讲解清楚，表达严谨，逻辑性强，文字通俗易懂。书中插图结合简练的叙述，代码配合详尽而简洁的注释，使得深奥抽象的概念和过程具体化并便于理解和记忆。本书示例源代码等可以从 www.tup.com.cn 下载。

图书在版编目(CIP)数据

基于 MFC 的可视化数据结构/连远锋等编著. —北京：清华大学出版社，2014
高等学校计算机专业规划教材
ISBN 978-7-302-36947-9

Ⅰ. ①基… Ⅱ. ①连… Ⅲ. ①C 语言－程序设计－高等学校－教材 ②数据结构－高等学校－教材 Ⅳ. ①TP312 ②TP311.12

中国版本图书馆 CIP 数据核字(2014)第 136407 号

责任编辑：龙启铭
封面设计：常雪影
责任校对：梁　毅
责任印制：李红英

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，010-62795954
印 刷 者：清华大学印刷厂
装 订 者：三河市溧源装订厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：29.25　　**字　　数**：727 千字
版　　次：2014 年 9 月第 1 版　　**印　　次**：2014 年 9 月第1次印刷
印　　数：1～2000
定　　价：49.00 元

产品编号：053516-01

前言

“数据结构”是计算机专业的重要专业基础课，是软件开发和维护的基础。它所涉及的知识内容及研究的技术方法，不仅为程序设计提供方法论的指导，并且在软件工程的开发周期中发挥着不可替代的作用。

随着计算机技术的发展，社会对计算机程序开发水平的要求也日益提高，为适应此形势，高校的数据结构教学内容也在不断地改革。现实生活中的问题经过抽象，设计相应的存储结构及对应的求解问题算法，才可以被计算机处理。

许多读者在学习数据结构课程时感觉概念较为抽象，算法理解困难，这在某种程度上源自于对数据结构特点认识上的偏差。同时，本课程的学习过程也是复杂程序设计的训练过程，要求学生编写的程序结构清楚和正确易读，符合软件工程的规范。随着软件工程技术的不断发展，面向对象的可视化编程技术已经成为当今软件开发的重要手段之一，为此，将可视化编程技术引入到数据结构课程学习，有助于读者逐步理解并有效地使用数据结构，还可以体会这些数据结构的多种实现技巧。

目前，C++ 语言已经成为高校理工科学生的必修或选修课程，该课程使学生初步理解和掌握了面向对象程序设计的思想和方法。MFC 作为首先使用 C++ 编写的 Windows 应用程序框架，深刻体现了设计者的计算思维和程序设计技巧，而这些思维方式和技巧反过来又影响了当今软件工程技术的思想潮流和发展方向。为了帮助读者更好地学习数据结构课程，理解和掌握算法设计所需的技术，作者通过多年的教学实践，以 Visual C++ 为开发环境，系统地讲解基于 MFC 可视化数据结构所涉及的各种技术。考虑到该课程的教学特点，合理地对数据结构内容进行了取舍。书中穿插了线性结构、树结构、图结构及排序等开发案例，有助于读者掌握数据结构知识并身临其境地体验实际的项目案例开发过程，从而提高读者的实际项目经验。

全书由 7 章构成，各章内容如下：

第 1 章　认识 Visual C++，主要介绍 Visual C++ 简介及发展历程、代码编辑器使用技巧、Visual C++ 调试技术及高级调试技术。

第 2 章　MFC 编程基础，主要介绍 MFC 简介、MFC 的常用类、设备环境、基本绘图工具、文本字体及多线程等内容。

第 3 章　框架与窗口，围绕可视化数据结构分割窗口实例，主要介绍 MFC 框架结构、窗口分割、MFC 控件等内容。

第 4 章　可视化线性结构，介绍线性表的定义、线性表的两种主要的存储结构、栈的定义、栈的存储结构、队列的定义及队列的存储结构。重点阐述可视化单链表、可视化循环队列和可视化双端队列的实现方法。

第 5 章　可视化树结构，介绍树的定义、树的逻辑表示方法、树的性质、树的遍历和树的存储结构二叉树、二叉树的定义、二叉树的性质、树/森林和二叉树的转换与还原、二叉树存储结构、二叉树基本运算算法设计、二叉树的递归和非递归遍历算法、二叉树的构造、线索二叉树和哈夫曼树、二叉排序树、平衡的二叉排序树、B-树和 B+树、哈夫曼树及堆排序。重点阐述了可视化二叉排序树、可视化平衡二叉树和可视化 B-树的实现方法。

第 6 章　可视化图结构，介绍图的定义、图的存储结构、图的基本运算算法设计、图的两种遍历算法以及图的应用(包括图的最小生成树、最短路径、拓扑排序和关键路径等)。重点阐述了无向图深度优先遍历和有向图 Dijkstra 可视化实现方法。

第 7 章　可视化排序，介绍排序的定义、插入排序方法、交换排序方法、选择排序方法、归并排序方法和基数排序方法。重点阐述了简单插入排序、冒泡排序和选择排序的可视化实现方法。

本书兼顾读者对数据结构知识点的学习和 MFC 编程能力的提高，在讲述可视化数据结构程序设计方法的同时，力求将繁多的数据结构知识点与 MFC 实际开发技术进行关联。对数据结构的知识点、概念、难点，都力求以较精练的语言讲解，并对关键的知识点配以必要的实例，实例中配以较为详细的步骤说明、代码说明，力求让读者较好地掌握可视化数据结构程序设计的思路、开发技巧与体系。

本书可作为计算机专业本科学生“数据结构”课程的教材，也可以作为“数据结构课程设计”的教学用书，还可以作为从事计算机应用等工作的信息技术人员的参考用书。

基于 Visual C++ 的程序设计涉及的技术内容复杂。在本书编写过程中，作者参考了国内外的相关出版物及互联网上的电子资料，在此向各文献的作者表示衷心的感谢。书中不当之处在所难免，敬请广大读者批评指正。

编著者

2014 年 7 月

目 录

第1章 认识 Visual C++

1.1 Visual C++ 概述

1.1.1 Visual C++ 简介及发展历程

Visual C++ 是基于 Windows 平台的 C/C++ 语言的集成开发工具，集代码编辑、编译、链接、调试等功能于一体，并提供了大量类库的辅助开发工具，为程序开发人员提供了快捷、完整的开发环境。

Visual C++ 拥有两种编程方式：一种是基于 Windows API 的 C 编程方式，代码效率较高，但开发难度和工作量也大；另一种是基于 MFC 的 C++ 编程方式，代码运行效率相对较低，但开发难度小，源代码效率高。

1998 年，Visual Studio 6.0 发布，增加了支持 IE 4 的控件和类，扩展了对 OLE DB 的支持并提供了独立的 MSDN 帮助系统。此后数年内微软公司一直潜心于 .Net 的开发，所以 Visual Studio 没有发布新的版本，只在其间发布了一些补丁，这使得 Visual Studio 6.0 成为支持标准 C/C++ 规范的最后版本。此后，微软公司陆续发布了 Visual Studio .Net 2002、Visual Studio .Net 2003、Visual Studio 2005。但时至今日，就 Visual C++ 而言，大量的开发者依然在使用 Visual Studio 6.0，因此，本书中所有案例也使用 Visual Studio 6.0 开发。

1.1.2 Visual C++ 优势

Windows API 函数本身是用 C 语言编写的，所以 Windows 环境下最佳的编程语言是 C 或 C++ 语言。C 和 C++ 可以直接调用 Windows API 函数，而其他的语言需要转换。C 或 C++ 语言可以直接利用系统底层的资源，其程序运行速度远高于其他语言。

Visual C++ 6.0 作为可视化数据结构的开发工具具有如下优势：

- Visual C++ 具有丰富的向导窗口。它不仅是 C++ 语言的集成开发环境，而且与 Win32 紧密相连，方便开发人员操作。
- Visual C++ 是一种编译语言，其运行速度比 Java 等解释性语言的速度快。
- Visual C++ 能够访问系统底层，可以灵活地开发从底层软件直到上层直接面向用户的软件。
- Windows 应用程序接口是由微软提供的 API。微软提供的接口就是 MFC 类库，而 Visual C++ 对 MFC 类库提供了最全面的支持。

- Visual C++ 的调试工具为大型复杂软件的开发提供了有效的调试、排错手段。

1.2 代码编辑器使用技巧

Visual C++ 的功能强大、内容丰富。但在编写程序的过程中，错误是难免的。因此，程序员在编写过程中很重要的一部分就是调试程序。如果能熟练地使用它所特有的方法和技巧将有效地提高程序员的工作效率，提高开发速度。本节主要介绍 Visual C++ 的使用技巧和处理方法。

1.2.1 检测代码中的括号是否匹配

当某个函数内部的程序代码层次结构较为复杂时，括号之间的匹配关系就难于理清了，这时可以通过集成开发环境中的辅助功能来检测括号是否匹配。将光标移动到所需要检测的括号(大括号{}，中括号[]，小括号()和尖括号<>)前面，按 Ctrl+]或 Ctrl+E 键。如果当前括号匹配，光标会自动跳转到匹配括号处；否则，光标不移动并发出警报。

1.2.2 代码对齐

在编写程序时，有时仅考虑程序的算法，忘记了代码的缩进格式。一个规范化的程序段可以通过代码的缩进关系展示函数的逻辑结构。在编程代码时，每行一般只有一条语句，否则会导致代码凌乱不齐，降低程序的可读性。Visual C++ 提供了默认的对齐方式，每编写一条语句后按 Enter 键，下行代码会根据上面的结构自动对其。如果程序段有多行未对齐的代码，则可按 Alt+F8 键来对齐代码。

1.2.3 显示函数参数

用户在调用某个函数时，当光标在函数的括号中时，Visual C++ 会显示函数参数信息，但是不久该信息就会消失。如果用户想要查看函数的参数信息，可以将光标放在函数的括号中，按 Ctrl+Shift+Space 键，此时会再次显示函数的参数信息。

1.2.4 完全取词功能不可用

在使用 Visual C++ 时，通常情况下，在代码编辑器中输入一个对象后，输入“.”后会显示该对象的成员变量和方法，供用户选择。但有时该功能不可用。如果出现该情况，可以关闭当前的工程，然后在工程目录下删除扩展名为.ncb 的文件，然后重新打开工程，Visual C++ 会重建一个.ncb 文件，此时完全取词功能将恢复正常。

1.2.5 快速删除项目下 Debug 文件夹中的临时文件

在开发应用程序时，有时需要删除 Debug 文件中的临时文件，可以按如下方法进行。在工作区的 FileView 视图中选中根结点，然后右击，在弹出的快捷键菜单中选择 Clean 菜单项即可。

1.2.6 如何彻底地从工程中删除一个类

用户在开发程序中，当不需要某个类时，为了减小应用程序的大小，通常会将该类从工程中删除。方法是在工作区的 File View 选项卡中逐一删除该类的头文件和源文件，然后在该项目的工程文件夹下删除这些文件。将工程文件夹下的.clw 文件删除，重新打开工程，在 Visual C++ 开发环境中单击菜单栏中 View→Class Wizard 菜单项，重新生成.clw 文件。

1.2.7 从其他文件中抓取资源

许多应用软件具有漂亮的界面，用户可以获取其中的某些位图、图标、对话框等资源。利用 Visual C++，用户可以方便的做到这一点。在 Visual C++ 开发环境中单击菜单栏中 File→Open 菜单命令，在 Open as 组合框中选择某目标应用程序的 Resource，将其打开。这样，就可以浏览该文件的资源，如图标、光标、位图、对话框等，将所需的资源复制到自己的工程中即可。

1.2.8 Visual C++ 中命名规则

程序员在系统开发过程中，不可避免的要定义变量、类，而培养规范的代码书写习惯是十分必要的。规范的代码能够增强程序的可读性，便于将来的二次开发和维护升级。

1. 函数命名规则

Windows 函数的命名通常采用动词加名词的组合方式。若由多个单词组合，则每个单词的开头字母一律大写，其余字母要小写。例如

SetCursor()：设置光标函数。

ShowWindow()：设置窗口的显示状态。

PtInRect()：判断点是否在某个举行范围内(Pt 代表 Point，In 表示在某一个范围内，Rect 表示 rectangle)。

2. 变量命名规则

在 Windows 程序变量命名过程中，通常采用匈牙利命名法则。该标记方法是由微软公司的一个匈牙利籍工程师设计发明的。匈牙利命名法用一个或则多个小写字符来定义变量的名称，如表 1.1 所示。

通常给成员变量加前缀“m_”表示该变量是一个类或结构体的成员变量。

1.2.9 Workspace 和 Project 之间的关系

一个 Workspace 中可以包括若干个 Project，但只有一个 Project 处于 Active 状态，各个 Project 之间可以有依赖关系，在 Project 的 Setting…中可以设定，处于 Active 状态的 Project 可以依赖于其他的提供其函数调用的静态库。

1.2.10 在编辑状态下发现成员变量或函数不能显示提示时如何打开显示功能

可按照以下步骤操作，恢复编辑状态下成员变量或函数显示提示功能：

表 1.1 匈牙利命名法则

前缀	数据类型	前缀	数据类型
b	BOOL	I	int
by	BYTE	l	LONG
c 或 ch	char/WCHAR/TCHAR	n	short
clr	COLORREF,24 位颜色值	p	pointer
cx、cy	水平或垂直距离	s	string
C	类	S	结构
dw	DWORD	sz	以 0 结尾的字符串
fn	function(函数)	x、y	x、y 坐标
h	Handle(句柄)	w	WORD

(1) 关闭当前工程 Project;

(2) 打开当前工程所在的工作目录,定位到“工程名.ncb”文件,并将其删除;

(3) 重新打开工程 Project 后,可以发现编辑状态下成员变量或函数显示提示功能恢复。

1.2.11 如何清楚所有的断点

单击菜单 Edit→Breakpoints…,打开 Breakpoints 对话框,单击 Remove All 按钮即可,或者选择快捷键 Ctrl+Shift+F8。

1.2.12 如何添加 Lib 文件到当前工程

单击菜单 Project→Settings 弹出 Project Setting 对话框,切换到 Link 标签页,在“Object/library modules”处输入 Lib 文件名称。如果需要添加多个 Lib 文件到当前工程,需要将不同的 Lib 文件名用空格分隔。

1.3 Visual C++ 调试技术

在开发程序的过程中,出现错误是难免的。要查找出程序中的错误,就需要利用调试工具来帮助开发人员对程序进行调试。所谓程序调试,就是将编写的程序投入实际运行前,用手工或编译程序等方法进行测试,修正语法错误和逻辑错误的过程,是诊断、消除错误以保证计算机信息系统正确性的必不可少的步骤。目前有许多调试工具,而集成在 Visual C++ 中的调试工具以其强大的功能,能够帮助开发人员有效地找出问题所在。下面先来介绍 Visual C++ 中的调试工具的使用。

调试工具(Debugger),是指一种用于调试其他程序的程序集合,它能够让代码在指令组模拟器(ISS)中选择性地运行,以便查看代码中变量的内容并查找及消除错误。

典型的除错器通常能够在程序运行时拥有以下这些功能,例如单步运行(single-

stepping)、利用中断点(breakpoint)使程序遇到各种事件(event)时停止(breaking)(一般用于使程序停止在想要检查的状态),以及追踪某些变量的变化。有些除错器还能在想要除错的程序在运行状态时,去改变它的状态,而不仅仅只是用来观察。

1.3.1 调试环境的建立

在 Visual C++ 中每当建立一个工程(Project)时,都会自动建立两个版本:Release 版本和 Debug 版本。在这两种版本下产生的可执行文件代码差异较大。

Release 版本是当程序完成后,准备发行时用来编译的版本,它对可执行程序的二进制代码进行了优化,不能调试运行引用程序,当然也不能提供任何调试信息。

Debug 版本是用在开发过程中进行调试时所用的版本。开发人员可以在程序内部设置断点并观察变量的运行状态。可以调试程序中的错误,此时,应用程序包含微软的格式的调试信息。由于不进行任何代码优化,在这种模式下编译、链接后的可执行文件比较大。

在新建立的工程中,默认的是 Debug 版本,若要选择 Release 版本,可以选择菜单 Project Settings 中的 Setting 命令,这时屏幕上面弹出 Project Settings 对话框,如图 1.1 所示。在 Settings For 下拉列表中选择 Win32 Debug,单击 OK 按钮退出。

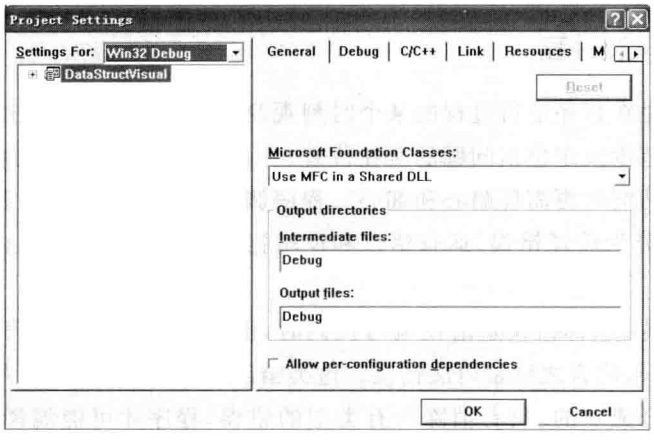


图 1.1 Visual C++ 调试环境窗口

在调试程序的时候必须使用 Debug 版本,可以在 Project Settings 对话框的 C/C++ 页中设置调试选项,如图 1.2 所示。

各个选项的含义如下:

- Line Numbers Only:表示程序经过编译和链接产生的(.OBJ)或(.EXE)文件仅仅包含全局和外部符号以及行号信息;
- C7 Compatible:表示产生一个(.OBJ)或(.EXE)文件行号信息以及符号化的调试信息;
- Program Database:表示产生一个存储程序信息的数据文件(.PDB),它包含类型信息和符号化的调试信息。
- None:表示不产生任何调试信息。