

高等学校计算机基础教育教材精选



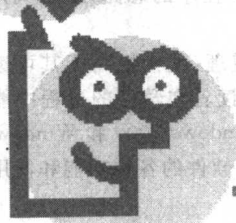
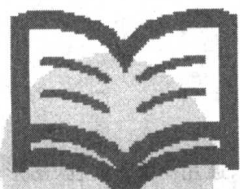
杨永国 主编

Visual C++ 6.0 实用教程



清华大学出版社

高等学校计算机基础教育教材精选



Visual C++ 6.0

实用教程

杨永国 主编

张冬明
王继尧
徐艳杰 副主编

清华大学出版社
北京

MT590/14

内 容 简 介

本书主要介绍了 Visual C++ 6.0 的集成开发环境、MFC 应用程序的结构、通用 MFC 编程、图形应用程序编程、多线程编程、ActiveX 控件、数据库编程等内容。书中由浅入深地讲解了 Visual C++ 6.0 的体系结构和它的实际使用方法,并且帮助读者形成正确编写 Windows 程序的思路,了解利用 Visual C++ 编写程序的奥妙。

本书的特色在于借助大量编程实例来阐述问题,避免讲述深奥的道理,大大提高了其可读性。此外,本书对实例程序编制过程的描述非常完整,便于读者上机实习,引导读者在学习的过程中能够举一反三,达到事半功倍的效果。本书所有示例的完整源程序代码均已在 Windows 2000 和 Windows XP 下调试成功,通过本书的介绍,读者可以比较容易地掌握 Visual C++ 开发软件的各种功能和使用方法及技巧。

本书在内容安排上循序渐进,论述深入浅出,同时附有大量的程序、屏幕图形和应用实例,是一本非常实用的入门教程,其中列举的实例对程序设计人员也有很好的辅助和提高作用。

本书可作为高等院校非计算机专业本科及研究生的教材或程序设计人员的参考书。

版权所有,翻印必究。

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。

图书在版编目(CIP)数据

Visual C++ 6.0 实用教程/杨永国主编. —北京:清华大学出版社,2004.1

(高等学校计算机基础教育教材精选)

ISBN 7-302-07743-6

I. V… II. 杨… III. C 语言—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 112309 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客 户 服 务: 010-62776969

责任编辑: 王听讲

印 装 者: 清华大学印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 19.5 字数: 445 千字

版 次: 2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

书 号: ISBN 7-302-07743-6/TP·5660

印 数: 1~5000

定 价: 25.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

出版说明

高等学校计算机基础教育教材精选

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,面向各高校开设的计算机必修课、选修课以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本、并保持不断更新),坚持宁缺勿滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是:jiaoh@tup.tsinghua.edu.cn;联系人:焦虹。

清华大学出版社

前言

Visual C++ 6.0 实用教程

1998 年底,微软推出了其开发工具企业版套件 Visual Studio 6.0, Visual C++ 是其中之一。Visual C++ 的功能十分强大,支持面向对象编程技术,支持组件共享,不仅可以提高软件系统开发的速度,而且可以大大提供软件的质量。同时,和其他可视化集成开发环境(比如 Visual Basic, Delphi, C++ Builder)一样, Visual C++ 集程序的代码编辑、编译、连接和调试等功能于一体,给程序员提供了一个完整方便的开发界面和许多的辅助开发工具, Visual C++ 的应用程序向导可以生成应用程序框架很大一部分的代码,程序员几乎不必书写程序代码,就可以生成一些完整的可以运行的程序。因此,学习和掌握 Visual C++ 的编程方法和技巧,是开发高质量应用程序必不可少的条件之一。

全书分为 14 章。主要内容为: Visual C++ 6.0 概述; MFC 的结构和向导工具的使用; Visual C++ 程序调试; 常用控件; 对话框; 菜单、工具条; 集成鼠标和键盘消息; 绘图; 单文档应用程序; 多文档界面应用程序; 高级控件; 组件和 ActiveX 技术; 多线程的调度和处理; 数据库编程。

本书内容新颖,力求用简明的语言、丰富的图形、典型的范例和循序渐进、论述深入浅出方式引导读者学习,只要读者理解了书中每一个重要环节上的说明,并针对书中的例子上机操作,在较短的时间内掌握 Visual C++ 6.0 的使用和编程技巧是完全可能的。

本书的特色在于借助大量编程实例来阐述问题,避免讲述深奥的道理,大大提高了其可读性,对于初级入门者,本书由浅入深地讲解 Visual C++ 6.0 的体系结构和它的实际使用方法,对于中、高级读者,本书可以帮助读者形成正确的编写 Windows 程序的思路,了解利用 Visual C++ 编写程序的奥妙。此外,本书也可作为参考资料,便于读者随时查询。本书对实例程序编制过程描述完整,便于读者上机实习,希望读者在学习的过程中能够举一反三,相信会使广大读者受益匪浅。

本书所有示例的完整源程序代码均已在 Windows 2000 和 Windows XP 下调试成功,通过本书的介绍,读者可以比较容易地掌握 Visual C++ 开发软件的各种功能和使用方法及技巧。

本书由杨永国任主编,第 1 章由王继尧编写,第 2 章由张冬明和王继尧编写,第 3~13 章由杨永国、张冬明联合编写,徐艳杰编写了第 14 章。书中示例程序由张冬明和徐艳杰调试完成,全书由杨永国统稿并审校。

由于编者水平有限,加之编写时间仓促,书中可能有错误之处,敬请读者和同行批评指正。

作 者

2004 年 1 月

目录

第 1 章 Visual C++ 6.0 概述	1
1.1 可视化技术	1
1.2 Visual C++ 6.0 的安装	2
1.3 Visual C++ 6.0 的用户界面	3
1.3.1 项目工作区	3
1.3.2 编辑区	5
1.3.3 Visual C++ 6.0 的菜单栏	5
1.3.4 工具条	11
1.3.5 WizardBar(向导栏)	13
1.3.6 Visual C++ 6.0 中的快捷特性	14
1.4 Visual C++ 6.0 的新特性	14
1.5 面向对象技术和 Windows 编程	16
1.5.1 面向对象和面向过程	16
1.5.2 类和对象	21
1.5.3 类的继承和类的层次	22
1.5.4 虚函数与多态性	25
1.5.5 Windows 编程	27
1.6 小结	28
1.7 习题	28
第 2 章 MFC 的结构和向导工具的使用	30
2.1 Visual C++ 的文件组织	30
2.2 MFC 编程	31
2.3 MFC 的应用程序框架	31
2.4 使用 AppWizard 创建应用程序框架	34
2.4.1 AppWizard 概述	34

2.4.2	创建应用程序框架	34
2.4.3	应用程序的框架结构	41
2.5	ClassWizard	55
2.6	Document/View 结构	61
2.6.1	Document/View 结构的优点	61
2.6.2	文档、视图和框架的关系	62
2.7	MFC 的消息管理	64
2.7.1	消息映射机制	64
2.7.2	消息的管理	66
2.7.3	自定义消息	67
2.7.4	消息的发送	68
2.8	MFC 的类库构成	70
2.9	使用 MFC 的好处	73
2.10	学习 MFC 的方法	73
2.11	小结	74
2.12	习题	74
第 3 章	Visual C++ 程序调试	75
3.1	Visual C++ 调试工具	75
3.1.1	设置工程的调试属性	75
3.1.2	设置断点	77
3.2	控制程序的运行	81
3.3	使用查看工具	81
3.3.1	弹出式调试信息条	81
3.3.2	观察窗口	82
3.3.3	内存窗口	82
3.3.4	寄存器窗口	83
3.3.5	快速查看变量对话框	83
3.4	高级调试技术	84
3.4.1	TRACE 宏的利用	84
3.4.2	ASSERT 宏的利用	85
3.4.3	ASSERT_VALID 宏的利用	85
3.4.4	对象的 Dump 函数的利用	85
3.4.5	检查内存漏洞	86

3.5	小结	87
3.6	习题	87
第4章	常用控件	88
4.1	控件的基本概念	88
4.2	改变控件的颜色	90
4.3	用来管理对话框控件的 CWnd 成员函数	92
4.4	在程序中使用控件	92
4.5	小结	103
4.6	习题	103
第5章	对话框	104
5.1	对话框的基本概念	104
5.2	最简单的对话框——消息框	105
5.2.1	消息框的基本概念	105
5.2.2	创建一个消息框实例	105
5.3	创建一个基于对话框的应用程序	107
5.4	进一步学习对话框的知识	118
5.5	无模式对话框	120
5.5.1	无模式对话框的特点	120
5.5.2	无模式对话框实例	121
5.6	关于窗口的删除	127
5.7	公用对话框的使用	128
5.8	小结	130
5.9	习题	131
第6章	菜单、工具条	132
6.1	菜单概述	132
6.2	创建应用程序实例	133
6.3	创建多语言菜单	137
6.4	快捷菜单	141
6.5	工具条	146
6.6	更新命令用户接口(UI)消息	151
6.7	小结	153

6.8	习题	153
第 7 章	集成鼠标和键盘消息	154
7.1	键盘消息	154
7.2	鼠标消息	155
7.3	创建一个应用程序示例	156
7.4	小结	162
7.5	习题	162
第 8 章	绘图	163
8.1	绘图概述	163
8.2	图形设备接口	164
8.2.1	MFC 中与 GDI 有关的类	164
8.2.2	设备场景类 CDC	164
8.2.3	绘图对象类	166
8.3	基本绘图函数	169
8.4	屏幕映射方式	172
8.5	创建一个绘图应用程序	174
8.6	字体与文本	184
8.6.1	字体与文本的基本知识	184
8.6.2	字体与文本使用实例	187
8.7	关于 OpenGL	192
8.7.1	翻译描述表	192
8.7.2	创建一个使用 OpenGL 的实例	195
8.8	小结	199
8.9	习题	199
第 9 章	单文档应用程序	200
9.1	建立应用程序框架	200
9.2	类的设计	201
9.3	文档序列化	207
9.4	增加画线功能	208
9.5	加深对文档/视图结构的理解	213
9.6	小结	217



9.7 习题	217
第 10 章 多文档界面应用程序	218
10.1 多文档界面应用程序的特点	218
10.2 创建一个 MDI 的应用程序框架	218
10.3 MDI 应用程序框架结构	220
10.4 模板、文档和视图	220
10.5 了解多文档应用程序的内幕	223
10.5.1 文件的打开和新建	223
10.5.2 新建窗口	224
10.5.3 菜单的更新	225
10.6 完善程序的功能	225
10.6.1 设置文档类	225
10.6.2 创建第一个视图	226
10.6.3 创建第二个视图	228
10.7 另一种流行的视图结构——切分窗口	230
10.8 小结	232
10.9 习题	232
第 11 章 高级控件	233
11.1 标签式对话框	233
11.2 树控件	237
11.3 使用进度控件、数字滚选按钮控件、滑竿控件	245
11.4 小结	249
11.5 习题	250
第 12 章 组件和 ActiveX 技术	251
12.1 Visual C++ 组件	251
12.2 ActiveX 控件	256
12.2.1 ActiveX 控件简介	256
12.2.2 ActiveX 控件体系结构	257
12.2.3 创建 ActiveX 控件	259
12.2.4 ActiveX 控件的安装	265
12.2.5 在程序中使用 ActiveX 控件	266

12.3	小结	267
12.4	习题	267
第 13 章	多线程的调度和处理	268
13.1	多任务、多进程和多线程	268
13.2	MFC 对多线程的支持	269
13.3	线程的创建和终止	269
13.3.1	基本知识	269
13.3.2	创建一个多线程实例	270
13.3.3	线程间通信	273
13.4	进程和线程的优先级问题	275
13.5	线程同步问题	276
13.6	用户界面线程	278
13.7	小结	279
13.8	习题	279
第 14 章	数据库编程	280
14.1	ODBC 的基本概念	280
14.2	MFC 的 ODBC 类	281
14.2.1	CDatabase 类	281
14.2.2	CRecordView 类	282
14.3	创建一个 ODBC 数据源的数据库管理程序实例	283
14.4	深入理解 CRecordset 类	287
14.5	几个有用的函数	289
14.6	增加程序的功能	289
14.7	DAO 简介	291
14.8	小结	296
14.9	习题	296
参考文献	297

面向对象的程序设计技术是当今全球程序员普遍采用的一种程序设计方法,是软件开发的最新潮流。在众多的开发工具中,Microsoft 公司的 Visual C++ 6.0 独树一帜,将面向对象的程序设计方法和可视化的软件开发环境完美地结合起来,使得开发 Windows 平台的应用程序更加方便、深入。Visual C++ 程序的执行速度以及对操作系统访问的权限之高,是其他许多语言无法比拟的,加上 Windows 操作系统的支持,就使得 Visual C++ 的高级程序员对整个计算机的硬件系统和软件系统在各方面的访问和控制更加游刃有余。因此,有人说:“用 Visual C++ 的眼光看,整个计算机都是透明的,或者说是完全裸露的”。

1.1 可视化技术

可视化技术是当前发展迅速并引人注目的技术之一,它的特点是把原来抽象的数字、表格、功能逻辑等用直观的图形、图像的形式表现出来。可视化编程是它的重要应用之一。所谓可视化编程,就是指在软件开发过程中,用直观的具有一定含义的图标按钮、图形化的对象取代原来手工的抽象的编辑、运行、浏览操作,软件开发过程表现为鼠标操作和拖放图形化的对象以及指定对象的属性、行为的过程。这种可视化的编程方法易学易用,而且大大提高了工作效率。

作为一种 C/C++ 语言的集成开发环境 (IDE),产生 Visual C++ 的最早的根源其实要追溯到 DOS 时代的 Borland 公司,当初 Borland 公司开发的 Turbo Pascal 和 Turbo C 让程序员们深刻感受到了把编辑器和编译器集成在一起的 IDE 是多么的方便,微软也看到了这一优点,于是相继开发了 Quick C 和 Microsoft C/C++ 等多个 DOS 版本的 C/C++ 集成开发工具。随着 Windows 的不断成熟,微软决定放弃 DOS 下的开发工作,正式推出了 Windows 下的 Visual C++ 1.0,早期的 Visual C++ 功能并不怎么强大,使用起来也不方便,1.0 版和 1.5 版都是 16 位编程工具。Visual C++ 的革命性改变得益于 Windows 95 的推出,从 Visual C++ 2.0 开始,微软又放弃了 16 位编程,以后的 Visual C++ 都只用于 32 位编程开发,为了与 MFC 类库的版本号保持一致,微软跳过了版本 3,直接推出 Visual C++ 4.0,这个版本及修订版 Visual C++ 4.2 的部分界面风格一直被

保留到最新的 Visual C++ 6.0 中。从 Visual C++ 4.0 到 Visual C++ 6.0, Visual C++ 的各种功能不断增强, MFC 类库的内容也越来越丰富, 现在利用 Visual C++ 开发应用程序已经是一件相当轻松的事情了。

Visual C++ 6.0 是微软公司新近推出的 Visual C++ 5.0 的改进版本。它在以前的版本上又增加了一些新特性, 使得 Visual C++ 的开发环境更易于使用, 更便于用户迅速掌握它的使用方法。下面, 将逐步介绍 Visual C++ 6.0 的集成开发环境和它的新特征。

1.2 Visual C++ 6.0 的安装

Visual C++ 是 Visual Studio 这一通用的应用程序集成开发环境的一个组件。除了支持 Visual C++, Visual Studio 还支持 Visual Basic、Visual J++、Visual InterDev 等 Microsoft 系列开发工具。因此, Visual Studio 6.0 企业版是一个套装产品, 共有六张光盘, 其中后两张是 MSDN 库, 如果只安装 Visual C++ 6.0, 那么需要用到第一张光盘和 MSDN 库(中文版与此略有区别)。Visual C++ 6.0 可以安装到所有运行 Windows 98/2000/XP/NT 的机器上。

下面介绍安装步骤。以 Windows 98 为例, 把 Visual Studio 6.0 的第一张光盘插入光驱中, 运行光盘根目录下的 Setup.exe, 这是主安装程序, 它首先将对系统进行检查, 根据需要更新系统中的 Microsoft Java 虚拟机, 并安装 DCOM 98, 为了完成这个过程可能需要重新启动系统两次。接着主安装程序将询问 Visual Studio 6.0 通用文件的安装路径, 然后启动可视化工具的安装程序。

在指定了可视化工具的安装路径后, 建议至少应安装如图 1-1 所示的四个项目, 即 Visual C++ 6.0、附加的 ActiveX 控件、数据存取部件 Data Access 和一些辅助工具, 这些内容可用于一般应用程序的开发工作, 大致要占用 350MB 左右的硬盘空间。

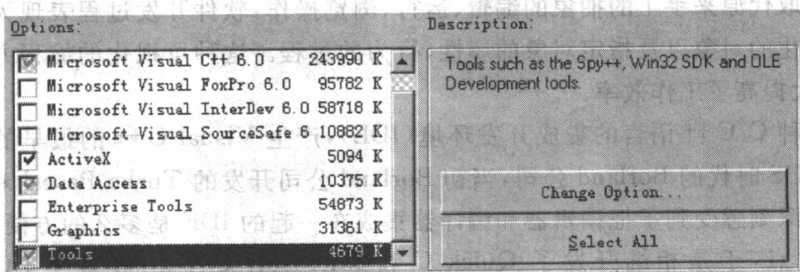


图 1-1 选择安装 Visual Studio 6.0 的组件

安装完 Visual C++ 6.0 之后, 需要重新启动一次系统, 然后安装程序将启动 MSDN 库的安装程序。如果硬盘空间允许的话, 建议安装“全文搜索索引”、“Visual C++ 6.0 帮助文档”、“Platform SDK 文档”和“全部其他文档”共四项, 这些内容基本上可以满足一般编程工作的需要, 大致要占用 550MB 左右的硬盘空间。至于 VB、VFP 等可视化开发工

具的帮助文档可以仍然放在光盘上,需要查阅时,MSDN 库的帮助浏览程序会自动提示插入所需的 MSDN 光盘。另外,Visual C++ 6.0 还带有大量的样例程序,它们都被放在了 MSDN 库的第一张光盘上,需要时可从上面拷贝。

MSDN 库安装完后,主安装程序会继续询问是否安装其他一些客户端和服务器的工具,大家可以根据自己的需要来选择,例如将来要制作安装程序的话,可安装一份 InstallShield,如果是在 NT 上安装 Visual Studio 6.0,还可安装一份 NT Option Pack。

Visual Studio 6.0 安装完毕后,开始菜单的程序中应生成了如图 1-2 所示的 Microsoft Visual Studio 6.0 和 Microsoft Developer Network 两个程序组。

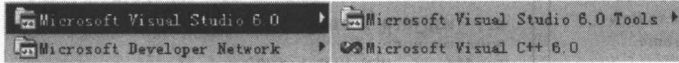


图 1-2 开始菜单中的 Visual Studio 6.0 和 MSDN

Visual Studio 6.0 是 1998 年下半年发布的,由于其中还存在着很多 Bugs,以及一些其他原因,截止到 1999 年 5 月底,微软已经专门为 Visual Studio 6.0 推出了三次 Service Pack(软件更新包),其中 SP1 解决了与老版本 Visual Studio 工具存在着的二进制兼容性问题,SP2 又更换了 Java 虚拟机,新近发布的 SP3 则再次修订了大量的 Bugs,因此建议在安装完 Visual C++ 6.0 后,再找一份 SP2 或 SP3 来打个补丁,SP2 可在某些报纸的配套光盘上找到,SP3 则暂时需要到微软的 MSDN 站点上去下载。

1.3 Visual C++ 6.0 的用户界面

Visual C++ 的集成开发环境 Visual Studio 提供了大量的实用工具以支持可视化编程特性,它们包括:项目工作区、ClassWizard、AppWizard、WizardBar、Component Gallery 等。下面对它们作逐一介绍。

Visual C++ 的用户界面如图 1-3 所示。

与 Visual C++ 5.0 不同的是,Visual C++ 6.0 系统的联机帮助目录并不显示在工作区窗口中,它的内容是显示在一个独立的窗口中的(如果已安装了 MSDN 的话)。

1.3.1 项目工作区

Visual C++ 使用项目工作区来组织项目、元素以及项目信息在屏幕上出现的方式。一个工作区可包含由不同的开发工具包生成的工程,如 Visual C++ 和 Visual J++。在桌面上,工作区以窗口方式组织项目、文件和项目设置。

每个项目视图都有一个相应的文件夹,包含了关于该项目的各种元素。展开该文件夹可以显示该视图方式下工作区的详细信息。项目视图区包含三个选项卡,单击各选项卡,可以在其间切换。这三个选项卡从左到右依次是:ClassView、ResourceView、FileView。

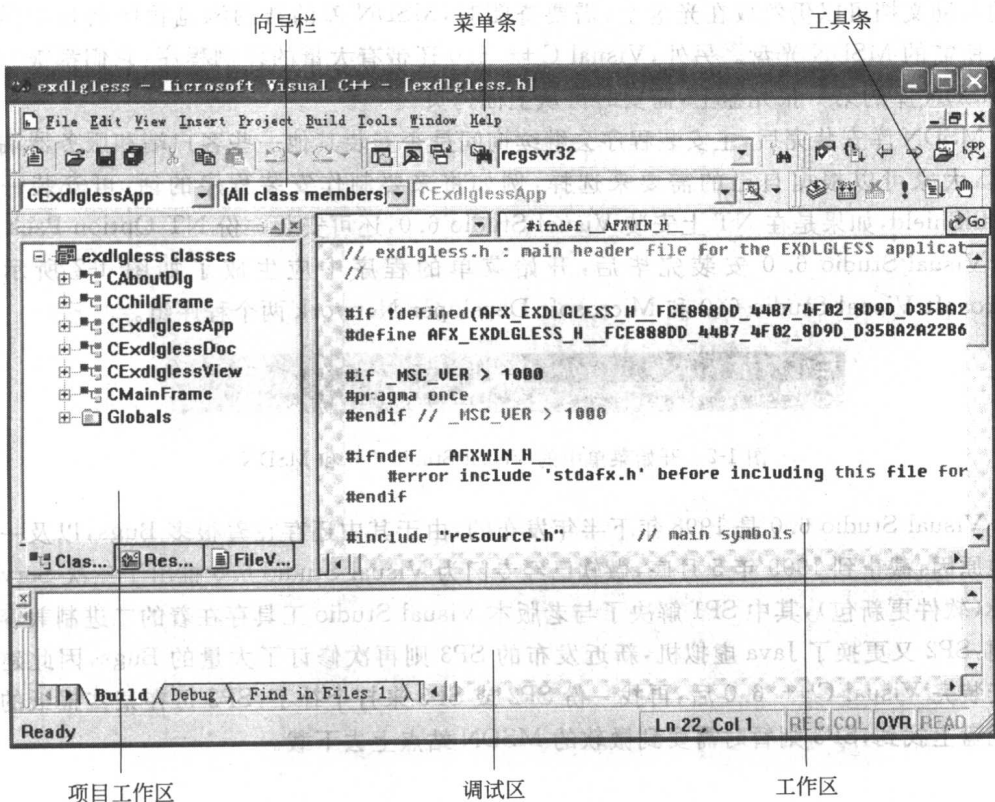


图 1-3 Visual C++ 的 IDE 的用户界面

(1) ClassView(类视图)

显示项目中定义的C++类,展开文件夹显示工程中所定义的所有类,展开类可查看类的数据成员和成员函数以及全局变量、函数和类型定义。ClassView 显示所有已定义的类以及这些类中的数据成员、成员变量。Visual C++ 自动从项目工作区中所包含的源程序文件中分离出类。在 ClassView 中,文件夹代表工程文件名。展开 ClassView 顶层的文件夹后,显示工程中所包含的所有的类。双击一个类的图标(或单击图标旁的“+”号)时,ClassView 展开该类并显示其类成员及成员函数。利用 ClassView 不仅可以浏览应用程序所包含的类以及类中的成员,还可以快速跳到一个类或成员的定义,方法如下:选择要查找的定义或声明的符号并双击之。要打开关于某一个类声明的头文件,只需双击类名即可。要想查找某一个变量的定义,可将光标定位在该符号处并单击鼠标右键,在弹出菜单中选择 Reference。

(2) ResourceView(资源视图)

显示项目中所包含的资源文件。展开文件夹可显示所有的资源类型。

(3) FileView(文件视图)

显示所创建的工程。展开文件夹可以查看工程中所包含的文件。FileView 窗格显示了工程文件和项目工作区中所包含的文件的逻辑关系。一个工作区可以包含多个工程,其中活动工程以黑体显示。活动配置决定了编译活动工程时的编译选项。活动工程

是使用 Build 或 Rebuild All 时要编译的那一个工程。可以用 Build 菜单上的 Set Active Configuration 选择不同的活动配置；也可以在 Project 菜单上用 Set Active Project 选择不同的活动工程。使用 FileView 可以查看文件、管理文件(包括增加、删除、移动、重命名、拷贝文件等)。要增加一个文件到过程中,可以选择 Project→Add to Project→Files 菜单,弹出文件对话框,选择相应文件即可;要从工程中删除一个文件,可打开工程文件夹,选择相应文件,然后按 Del 键。

每个视图都是按层次方式组织的。可以展开文件夹和其中的项查看其内容。在项目视图中,如果一项不可以再展开,那么它是可编辑的,双击这一项便可以打开相应的文档编辑器进行编辑:对类和源程序文件来说,是打开文本编辑器;对于对话框来说是打开对话框编辑器等。每个视图还支持右键快捷菜单。

1.3.2 编辑区

编辑区是 Visual C++ 的一个最重要的组成部分,实际上,大部分工作都要在该区进行。当然,该区中的文档的切换要依赖其他的工具,尤其要依赖项目视图区。

完成编辑区的任务,几乎已经完成了整个应用程序的开发,在该区可进行代码的编辑,窗口的布局设计,资源的编辑。

1.3.3 Visual C++ 6.0 的菜单栏

该菜单栏包含九个菜单,即 File、Edit、View、Insert、Project、Build、Tools、Window 和 Help,如图 1-4 所示。下面对它们进行分别介绍。

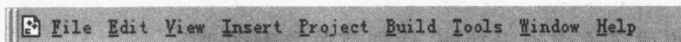


图 1-4 Visual C++ 6.0 的顶层菜单

1. File 菜单

该菜单包含如图 1-5 所示的子菜单。

New 菜单可创建新的文件、工程、工作区或其他文档,如图 1-6 所示。

Open 菜单用于打开一个已存在的文件。

Close 关闭当前被打开的文件。

Open Workspace 打开一个已存在的 Workspace。

Save Workspace 保存当前被打开的 Workspace。

Close Workspace 关闭当前被打开的 Workspace。

Save 保存当前文件。

Save As 将当前文件另存为一个新文件。

Save All 保存所有打开的文件。

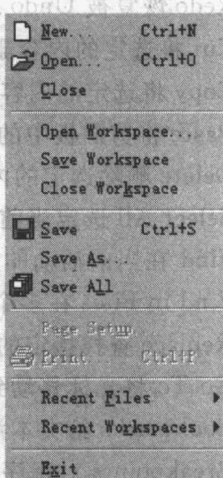


图 1-5 File 菜单的子菜单项

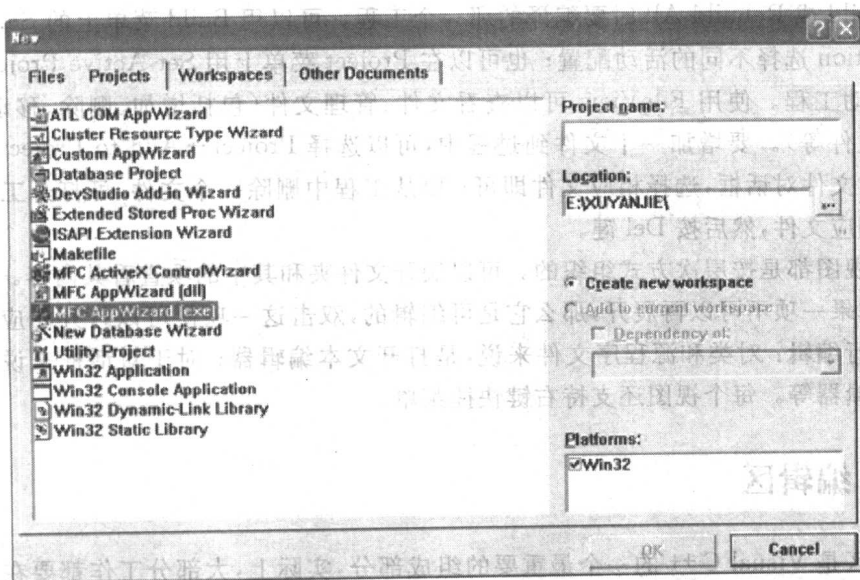


图 1-6 New 菜单的弹出对话框

Page Setup 设置文件的页面。

Print 打印文件。

Recent Files 最近的文件列表。

Recent Workspaces 最近的 Workspace 列表。

Exit 退出集成开发环境。

2. Edit 菜单

Edit 菜单中集成了许多编辑功能,详细菜单参看图 1-7。

Undo 撤销上一次的编辑操作。

Redo 恢复被 Undo 取消的编辑操作。

Cut 将选定的内容删除,同时将之移到剪贴板。

Copy 将选定的内容复制到剪贴板。

Paste 将剪贴板中的内容复制到光标的当前位置处。

Delete 删除选定的内容或光标所在位置的后一个字符。

Select All 选定当前活动窗口中所有内容。

Find 在当前活动的文件中查找指定的字符或字符串。

Find in Files 在多个文件中查找指定的字符或字符串。

Replace 查找指定的字符或字符串,并将之替换为指定的字符或字符串。

Go To 将光标移动到指定位置。

Bookmarks 给文本添加书签。

Breakpoints 对程序中的断点进行编辑。

List Members 列出全部关键字。