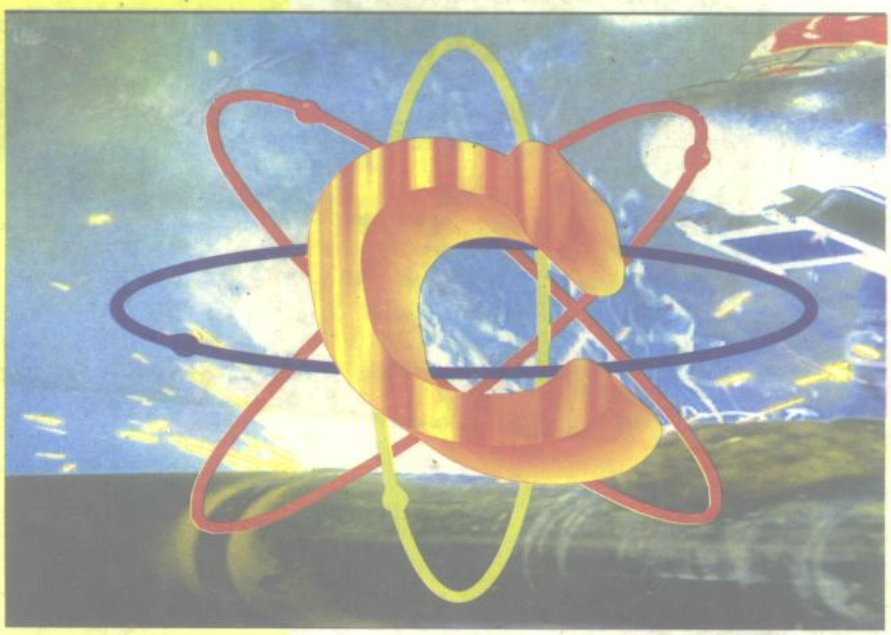
Visual C++ for Windows

——面向对象程序设计

刘培奇 席一凡 严西社



西安电子科技大学出版社

Visual C++ for Windows
—面向对象程序设计

刘培奇　席一凡　严西社

西安电子科技大学出版社
1996 年

(陕)新登字 010 号

内 容 简 介

本书是以 Visual C++ 1.5 for Windows 为基础，全面系统地介绍 Visual C++ 语言程序设计的基础教程。在内容安排上采取循序渐进的方式，通俗易懂的讲解方法，并辅以大量的说明性例题，以帮助读者快速掌握该语言的基本内容，达到应用它编写程序的目的。本书图文并茂、结构清晰、理论与实例并举。书中所用实例均为作者在 386 和 486 机器上开发设计出来的 VC++典型程序。

本书既适合于学习 Windows 应用程序设计的初学者，也适用于有一定 VC++ 编程经验的程序设计人员，还可作为大专院校计算机专业和非计算机专业的 Visual C++ for Windows 程序设计课程教材。

Visual C++ for Windows

—面向对象程序设计

刘培奇　席一凡　严西社

责任编辑　谭玉瓦

西安电子科技大学出版社出版发行

西安电子科技大学印刷厂印刷

新华书店经销

开本 787×1092　1/16　印张 26 8/16 字数 632 千字

1996 年 4 月第 1 版　1996 年 4 月第 1 次印刷　印数 1－6 000

ISBN 7－5606－0422－6/TP・0174　定价：27.50 元

前　言

自从 Microsoft 公司推出 Windows 操作系统之后，特别是 Windows 3.1 操作系统的问世，立刻受到广大计算机用户的青睐。因为它具有优秀的图形用户界面(Graphic User Interface，GUI)，可支持多媒体和多介质的用户接口，卓有成效的多任务操作管理，支持面向对象的程序设计(Object Oriented Programming，OOP)方法及提供的高级软件开发工作平台(workbench)等重要特性，故为世人所瞩目。因此，Windows 操作系统已成为 90 年代个人计算机操作系统发展的主流。

目前，尽管 Windows 已经成为个人计算机的主要操作系统，但是，还没有几个完全建立在 Windows 操作系统之下的面向对象程序设计语言环境。值得庆幸的是，最近 Microsoft 公司新开发的 Visual C++ for Windows 语言环境已经面世，它为 Windows 操作系统提供了一套面向对象程序设计的软件开发平台，使 Windows 应用程序的开发和设计变得非常简便、灵活，程序的运行效率更高。

本书所介绍的 Visual C++程序设计语言具有以下显著特点：

• Visual C++是一个完全建立在 Windows 操作系统之下的软件开发平台。利用 Visual C++既可开发 Windows 应用程序，也可开发出 Quickwin 应用程序。

• Visual C++是一个面向对象的程序设计语言。在 80 年代末，面向对象的程序设计方法(Object - Oriented Programing，简称 OOP)受到了程序设计人员的普遍重视，它从根本上改变了近半个世纪以来的程序设计模式，并且在今后几年中，Visual C++将成为程序设计的规范。

• Visual C++具有“可视性”(Visuality)。所谓可视性就是指用户可以直接使用 Visual C++语言提供的 App Studio 工具和 Windows 环境中的各种对话框、列表框、菜单及控件(如 Botton)等资源来设计用户界面。这使得设计用户界面就好像在白纸上画图一样简单、直观、灵活和高效。如果 Visual C++和 Windows 所提供的工具资源不能满足设计需要的话，可以通过使用 Windows 的 SDK(Software Development Kits)和 CCDK(Custom Control Development Kit)软件来设计所需要的工具。

• Visual C++具有高效性。高效性既反映在开发程序的高效性上，也反映在用户程序运行的高效性上。

本书在编写中，以 Microsoft 公司最新的版本——Visual C++ 1.5 for Windows 为基础，重点突出 Visual C++的特点。在内容安排上，采用教学方式，由浅入深、循序渐进，全面系统地介绍了 Visual C++的基本概念、基本语法规则和基本编程技巧。书中所有的例题都是根据每章的具体内容而精心安排的，通过实例可以加深读者对所学的内容进一步的理解，并且有助于掌握程序设计的方法。

如果你没有 C 语言编程经验，却又想学习和掌握一种新的程序设计语言，但不知应该从众多的计算机语言中选择哪一种语言来学习的话，我们建议你应选择 Visual C++程序设计语言。因为它反映了当前程序设计的发展方向，而且它具有其它高级语言所没有的特性。本书就是基于这一点考虑的，因此它既可作为那些没有 C 语言程序设计知识的读者学

习 Visual C++程序设计语言的自学教材，又可作为大专院校计算机和非计算机专业学习 Visual C++ for Windows 的教课书。

本书由西安建筑科技大学的刘培奇、席一凡、严西社同志共同编写。全书共 13 章，其中，从第 1 章到第 3 章、第 5 章由席一凡同志编写；第 6 章、第 7 章、第 9 章、第 10 章由刘培奇同志编写；第 4 章、第 8 章、第 11 章到第 13 章由严西社同志编写。在本书编写过程中得到了许多热心同志的支持和帮助，特别是西安电子科技大学出版社同志们所给予我们的大力支持，没有他们的支持，本书不可能在如此短的时间内正式出版。在此，作者向那些在本书的拟稿、编著、审阅、编排、校对中给予我们热情支持和辛勤工作的同志们致以衷心的感谢！

最后，由于作者水平有限，加之编写的时间仓促，因此，在编写中虽然经过了认真审校，但书中难免出现错误和缺点，希望广大读者给予批评和指正。

作　者

西安建筑科技大学

1995 年 8 月 31 日

目　录

第 4 章 输 入 输 出

第 5 章 函 数

第 6 章 构造数据类型

第 7 章 指 针

第 8 章 文 件

第 9 章 工 程 (project)

第 10 章 面向对象的程序设计

第 11 章 程序调试与排错

第 12 章 图 形 设 计

第 13 章 Visual C++实用工具

附录 键盘编辑命令

第 1 章　Visual C++集成开发环境

Visual C++是一个在 Windows 环境下运行的面向对象程序设计语言和环境。因此，在设计 VC++(Visual C++的简称)程序之前，应该了解一些 Windows 的有关知识，并熟悉 Visual C++程序设计环境，即在 VC++的工作平台上如何编辑、编译和调试程序。

本章作为学习 Visual C++语言程序设计的入门，将按照由浅入深，循序渐近的原则来介绍 Windows、Visual C++集成开发环境以及如何建立一个 VC++程序的全过程。通过本章的学习，应熟练地掌握 Visual C++工作平台(Visual C++ Workbench，VWB)的使用方法，为学习后面各章打下良好的基础。

1.1　Windows 基础知识

Microsoft Windows 是一个多任务的操作系统，它是集多用户、多任务、多功能和多媒体为一体的软件集成环境。它提供了图形用户界面(Graphic user Interface，简称 GUI)、高级应用程序编程接口(API)函数和软件开发工具箱(SDK)，并实现了动态数据交换、模块动态链接、内存自动管理等功能。它的设计思想与 DOS 有很大差别，具有 DOS 环境不具有的许多特征，如高级应用程序编程接口(API)函数的设备无关性，保证了应用程序与各种外部设备的兼容性，使得程序设计人员能够集中精力进行软件开发，而不用过多考虑相应的设备驱动程序。另一个特点就是它采用了面向对象程序设计(即 Object Oriented Programming，简称 OOP)思想，支持各种 OOP 语言，并提供有对象类库。

1.1.1　鼠标器(Mouse)

鼠标是一种被 Windows 普遍采用的输入设备，它比使用键盘更方便更灵活。下面将介绍鼠标的功能和使用方法。

鼠标器一般有三个键，按其位置分为左、中、右键。鼠标与系统连通之后，当用户移动鼠标器时，其指针便在屏幕或窗口视区内移动。当按下鼠标键或放开鼠标键时，系统将向有关应用程序发送消息，并激活由鼠标选定的对象。对于鼠标器上的按键而言，只有两个状态：

- 按下(Down)
- 放开(Up)

正是这几个键的两种状态不同组合，完成了所有菜单、窗口等的选择。

鼠标的操作方法是：

①选择操作(Click)：将鼠标指针(mouse pointer)移动到要选择的任选项上，然后单击鼠标左键，单击的意思是按一下键后立即放开此键。

②激活操作(Double_click)：先把鼠标指针移动到待激活的选项上，然后双击鼠标左

键，即连续快速按两下鼠标键。

③拖拉(Drag)操作：首先选择任选项，即使鼠标指向待选项，然后按住左键(不要放开)移动鼠标器，直到光标位置，这时放开鼠标左键。

1.1.2 窗口(Window)

窗口是指系统用于和用户界面各个部分进行信息交互的一个屏幕工作区域，它是Windows应用程序基本的I/O设备，它是应用程序访问屏幕的唯一途径。窗口由标题栏、菜单栏、滚动条、边界及用户区等部分组成。

窗口是由应用程序创建的，但必须由应用程序和Windows共同管理，Windows管理标准窗口的元素，如标题栏、滚动条等，并完成对窗口的直接操作，而应用程序主要是负责维护窗口的用户区。

每一窗口必须有一窗口事件(如位置移动、大小改变等)的处理函数，在Windows中称为窗口函数或事件函数。由Windows系统实时响应任何改变窗口的行为，并以消息方式发送给窗口函数。窗口函数接收此消息并作出相应的操作或处理。这些消息或者指定函数应执行的动作，或者要求函数提供辅助信息。

1.1.3 菜单(Menu)

菜单是Windows应用程序实现交互式用户界面和接收输入的主要手段之一。菜单是应用程序提供的一系列命令选项，对用户来说，这些选项是一组命令，可以看见并执行它们。

和窗口一样，菜单也是由应用程序在程序设计或执行时建立的，但必须由Windows和应用程序协调管理，Windows负责显示和管理菜单，应用程序则实现菜单项功能的处理。当用户选取菜单时，由Windows系统将相应的消息发送给应用程序的菜单命令处理函数，从而实现菜单的操作。

1.1.4 对话框(Dialog box)

对话框是一种用于应用程序和用户之间对话的临时窗口，对话框可以包含多个控制，一种控制是一种简单的窗口，完成简单的输入输出。例如，编辑控制对话框允许用户输入和编辑文字。对话框中的控制帮助用户输入各种信息(如文件名)，选择有关的选项或激活一个命令。

多数对话框是模式对话框(Modal dialog box)，意思是当模式对话框出现在屏幕上，使用者可以在对话框内完成它所提供的各种控制操作。对话框中包括以下几种控制(如图1.1所示)：

1. 输入域(Input fields)

在输入域中，用户可以输入任意的文本信息，如文件名、路径等。

2. 单选按键(Option button)

单选按键只允许从一组任选项中选择其中的一个任选项。

3. 检测框(Check box)

检测框具有从一组任选项中同时选择或释放多个任选项的功能。

4. 列表框(List box)

在列表框内显示出一系列选项，如果显示不完，可单击滚动条以显示更多的项。如果要选择列表框中的一个选项，可将鼠标指针指向该选项，然后单击鼠标左键。

5. 组合框(Combo boxes)

组合框把输入域和列表框联合在一起，如图 1.1 所示。

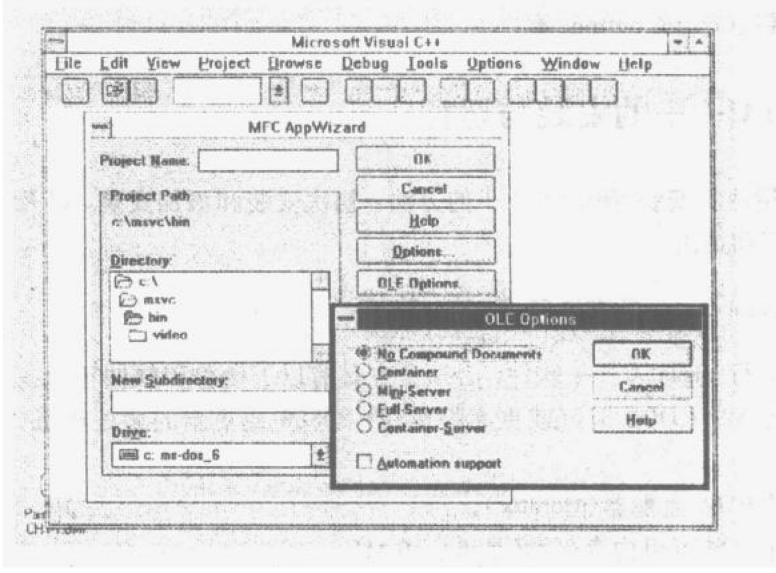


图 1.1　对话框

1.2　Visual C++语言特色

Visual C++是在 Windows 操作系统环境下运行的一种面向对象程序设计语言。它与其它程序设计语言相比较，具有以下特色：

(1) 继承了 C 语言的一切特征，C 语言仅仅是 Visual C++语言的一个子集。

(2) Visual C++是基于 Windows 操作系统环境下的一个编程工具。因此，Visual C++可以充分利用 Windows 系统环境中的一切资源，如对话框、应用程序设计接口函数(API 函数)、动态数据交换、软件开发工具箱(SDK)、对象类库、对象的链接与嵌入等，来设计自己的应用程序。

(3) Visual C++是一种面向对象的程序设计语言。它具有面向对象程序设计语言的一切特征，如：

- 对象；
- 可编程性；
- 访问、控制对象；
- 继承性；
- 多态性；
- 可预见性。

(4) Visual C++提供几种图形用户界面设计工具，它们是：

- App studio;
- Class wizard;
- App wizard。

这些工具都是一个应用程序，用户可以利用它们来创建和编辑各种资源，如程序肖像(Icons)、各种对话框(Dialogs)、命令菜单(Menu)、组合框(Combo box)、列表框(List box)以及各种控制按钮(Control button)等。

1.3 Visual C++的安装与运行

本节将介绍两种安装 Visual C++的方法：初次安装和添加安装，以及 Visual C++的支撑环境、运行和退出。

1.3.1 Visual C++的支撑环境

要成功地运行 Visual C++软件，必须至少具有以下硬件和软件环境：

(1) 能运行 MS - DOS 5.0(或更高版本)的 80386 或更新的微处理机(如 80486，Pentium 等)。

(2) 至少是 VGA 监视器(Monitor)。

(3) 至少有 4 M 可用内存(最好是 8～16 M)。

(4) 需要一个用来安装 Visual C++的硬盘，并且至少具有 75 M 空闲磁盘空间。

(5) 至少有一个 1.2 M(5.25 英寸)的软盘驱动器，或者一个 1.44 M(3.5 英寸)的软盘驱动器。

(6) 可在增强模式下运行的 Windows 3.1 操作系统。

1.3.2 初次安装

安装 Visual C++必须在 Windows 环境下进行，如果你的计算机上还没有 Windows 系统，应首先安装 Windows 系统，然后再安装 Visual C++。整个安装过程如下：

(1) 把 Visual C++软件的第一张软盘插到 A 驱动器里。

(2) 如果 Windows 操作系统已经被装入到计算机中，启动 Windows 便进入 Windows 工作环境，如图 1.2 所示。

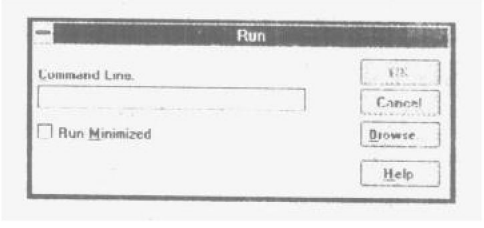


图 1.2 Run 对话框

(3) 打开(即单击 Click)File 菜单，从中选择 Run 命令(即用鼠标单击该项)，这时在屏幕上显示出一个 Run 对话框，如图 1.2 所示。

(4) 在 Run 对话框的文本输入框中，输入安装命令。即在 Command Line 框中，键入如下命令：

A:\SETUP

(5) 按 Run 框中的 OK 键。当按 OK 键后，可开始安装 Visual C++了。在安装的过程中，会出现选择安装项目的对话框，用户可根据自己的需要进行有选择的安装。我们建议用户在初次安装 Visual C++时，采用系统的缺省选项，这时将把系统默认的 Visual C++

默认的部件都装到硬盘上并实现其功能。

(6) 安装完第 1 张盘之后，直接选择 Continue 键，直到安装完最后一张盘。

1.3.3　添加安装

添加安装是在安装了 Visual C++之后，再装入以前没有装入的文件和应用程序等。其安装过程如下：

(1) 和初次安装一样，先启动 Windows，打开 File 菜单，从中选择 Run 命令，然后在 Run 对话框的 Command Line 框中输入 A:\setup，按 OK 键。

(2) 在安装选择项对话框中，选择所需要的部件，同时把不需要安装的部件其前面的检查框标记清除(由☐变成☐)。

(3) 选择 OK 键。

(4) 选择 Continue 键。

1.3.4　运行 Visual C++

当安装 Visual C++成功之后，在 Windows 的程序管理器(Program manager)窗口中出现一个 Visual C++的肖像(或称图标，Icons)，只需双击(Double_click)该肖像，就可启动 Visual C++并进入 Visual C++的集成开发环境中。

1.3.5　退出 Visual C++环境

当需要从 Visual C++中退出时，打开 File 菜单，选择 Exit 命令，或按 Alt+F4 键或双击 Workbench 集成环境左上角的控制按键即可从 Visual C++集成环境中退出。

1.4　Visual C++工作平台(Workbench)

系统安装完毕后，Setup 程序生成了一个 Visual C++分组，其中包括 Visual C++和各种支撑文件，并均由肖像表示。Visual C++工作平台(Visual Workbench)本身就是一个集成化的程序设计环境，这个集成环境包括：正文编辑器(Editor)、浏览器(Browser)、编译器(Compiler)、链接器(Linker)、调试器(Debugger)、Make 实用程序和联机帮助系统(Help)。所有这些都为程序开发提供了一个方便、高效的程序设计环境。

进入 Windows 环境之后，会出现一个“Microsoft Visual C++”分组肖像，双击该分组肖像，即可打开该分组窗口，它列出了该窗口里的所有工具和应用程序，如图 1.3 所示。

为了进入 Visual C++环境，只须双击该窗口中的 Visual C++肖像即可进入 Visual Workbench 的集成环境，如图 1.4 所示。

1.4.1　Visual C++工作平台的组成

由图 1.4 可以看出，Visual Workbench 集成环境由 5 个可见部分组成：标题栏(Title)、菜单条(Meun bar)、工具条(Tool bar)、正文编辑窗口区和状态条(Status bar)。其中许多菜单选择项还提供有自己的对话框(Dialog box)。下面将对每个部分作简单的介绍。

Visual Workbench 环境普遍以鼠标器作为输入设备，尽管所有的命令也能用相应的键

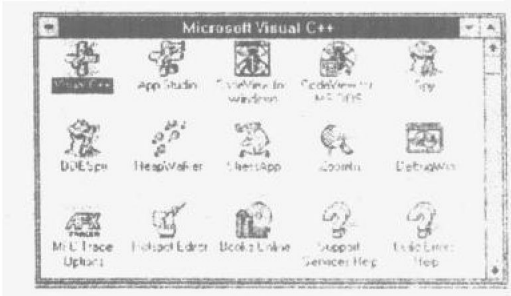


图 1.3 Visual C++分组窗口

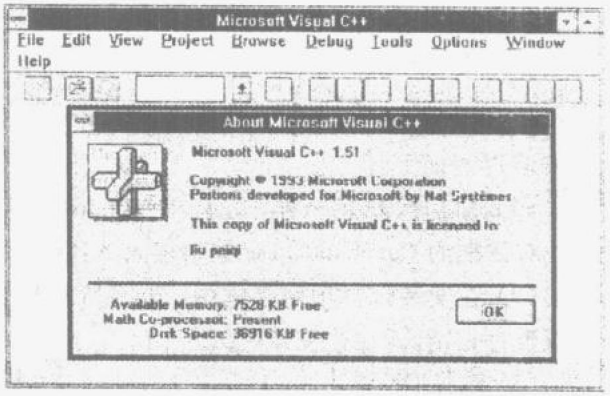


图 1.4 Visual C++工作平台

盘命令完成，但我们建议用户最好还是用鼠标器来完成命令的选择输入。

1. 标题栏(Title bar)

每一个窗口或对话框都有一个标题栏，它用来显示该窗口的名称或对话框的名称。有了它，我们很容易知道当前所使用的窗口或对话框。

2. 菜单条(Menu bar)

在 Visual C++的菜单条上共有 10 个菜单命令项，它们的名称和功能如下：

①File 菜单：它又包含许多命令项。通过选择 File 中的命令项，可以在多个窗口中打开并创建一个程序文件，还可以保存修改后的文件、设置打印机参数、完成文件打印等文件操作命令以及退出 Visual Workbench 环境。

②Edit 菜单：它也包含许多命令选择项，每个命令选择项都完成一个独立的功能。使用 Edit 菜单中的命令选择项，用户可在编辑窗口里编辑、裁剪、拷贝、粘贴正文；如果操作失误，可以通过选择恢复(Undo)命令，使当前的操作失效而恢复操作前的状态；还能完成一系列文本编辑功能，如搜索指定的正文、替换正文、设置文件属性等。

③View 菜单：View 菜单也提供有许多选择项命令，它可以帮助用户在自己的文件中查找并显示文本、函数说明或定位到出错位置；还可以设置、取消正文标识(Bookmark)块。

④Project 菜单：它包含了所有的工程命令选择项。如创建一个工程，从工程中删除或添加文件，设置工程文件任选项等。

⑤Browse 菜单：它包含了所有的浏览文本文件的命令任选项，如打开浏览窗口观察某个程序等。

⑥Debug 菜单：它包括了有关调试程序、监视程序运行的命令项。

⑦Tools 菜单：在 Tools 菜单中，列出了在 Visual Workbench 环境下所能调用的所有工具。

⑧Window 菜单：它包含了所有的窗口管理命令，如并排安排多个窗口、瀑布式排列多个窗口、打开一个附加窗口、关闭所有的窗口等。

⑨Option 菜单：该菜单中的命令项允许用户显示、改变和设置 Visual Workbench 集成环境内的各种缺省项设置，如改变 Tab 键的缺省设置(原缺省设置为 4)。Option 中的命令大多数都有自己的对话框。

⑩Help 菜单：它允许用户访问并显示在特殊窗口中的联机帮助信息，其中包括 Visual Workbench、Visual C++语言等各方面的帮助信息。

说明：

可以用多种方式打开菜单并从其中选择命令项，其中最简单的方式就是直接用鼠标器单击(Click)菜单或菜单中的命令选项即可。

如果没有鼠标器，可用键盘键来选择菜单或命令项，例如，要打开 File 菜单，先按住 Alt 键，再按 F 键(用 Alt+F 表示)。若还要从该菜单中选择命令项，还用 Alt 键再按命令项的首字母。

注意：当菜单或菜单中某个命令选项是虚线显示的，这表示该菜单或该命令项没有被激活，是不可选取的，该命令是失效的。

3. 工具条(Tools bar)

在 Visual Workbench 的菜单条下面有一个工具条，它上面有 15 个工具肖像，如图 1.5 所示。

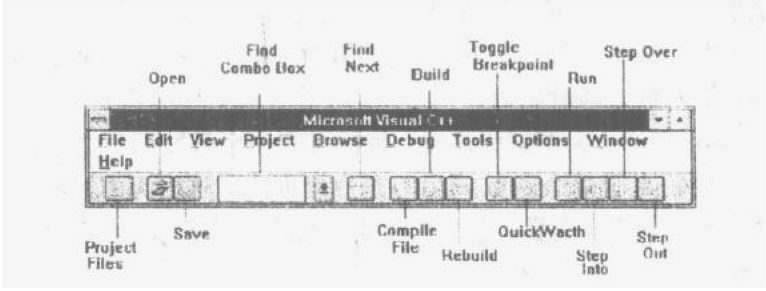


图 1.5　Visual Workbench 的工具条

在这里需要指出的是：要想使用工具条上面所列出的工具，必须先打开 View 菜单，从中选择(即单击)Toolbar 命令项，此时该命令前有一个激活标记(√)，表示 Workbench 环境中的工具条已被激活，它上面的工具是有效的。注意，Toolbar 命令的缺省值就是(√)。如果要关闭工具条，即使工具条失效，只要再选择一次 View 中的 Toolbar 命令。下面将分别介绍工具条上的各个工具的功能：

①Project Files 工具：当用户用鼠标单击这个工具后，将在编辑窗口上显示出一个下拉式列表框，并列出当前工程(Project)包括的所有源程序文件(Source File)，用户可在其中选择所需要的文件，同时将它显示在一个刚打开的编辑窗口内。

②Open 工具：单击该工具后，立即显示出一个 Open File 对话框，允许用户打开工程中的一个文件。具体功能和用法类似于 File 菜单中的 Open 命令。

③Save 工具：允许存储当前所打开的文件，具体功能和用法与 File 菜单中的 Save 命令相类似。

④Find combo box 工具：该命令允许用户输入要检索的文本并在当前文件中检索它。

⑤Find next 工具：单击该命令工具后，光标将定位到当前位置的下一次正文出现的位置处。该命令一般与 Find combo box 命令一起使用，一般先使用 Find combo box 命令，这时出现 Find combo Box 对话框，用户输入要查找的正文，系统将把光标定位到该正文第一次出现的位置处，这时若要继续查找正文，则用 Find next 命令工具，系统将把光标再定位到当前位置的下一次正文出现的位置处。

⑥Compile file 工具：编译当前文件并形成目标模块(.obj)，但不链接用户程序形成可执行的.EXE 文件。它与 Project 中的 Compile File 类似。

⑦Build 工具：该命令工具只编译当前工程中自上次编译以来修改过的文件并将其链接成可执行文件(.EXE)。它与 Project 中的 Build 命令类似。

⑧Rebuild All 工具：该命令工具编译当前工程中的所有文件并链接成可执行文件(.EXE)。它与 Project 中的 Rebuild All 命令一样。

⑨Toggle Breakpoint 工具：使用该工具命令可以在源程序文件的当前行上设置断点或取消断点。它的功能和用法与 Debug 菜单中的 Toggle Breakpoint 命令相同。

⑩Quick Watch 工具：使用该工具可以通过 Quick Watch 对话框立即观察(显示)断点处变量的值，并且允许用户修改这些值。

⑪Run 工具：激活当前程序，并使程序从断点处继续运行，它的功能和用法类似 Debug 菜单中的 Go 命令。

⑫Step into 工具：该命令对程序进行单步(single step)跟踪调试(执行)，碰到函数调用时跟踪进入到函数体内。它的用法和功能与 Debug 中的 Step Into 命令类似。

⑬Step over 工具：该命令与 Step Into 命令不同的是，这种调试方法将函数调用语句当作一条指令看待，而不进入函数体内。

⑭Step out 工具：用该命令调试程序时，每执行完一个函数调用之后，光标就停止在它的下一条语句处，这相当于每个函数调用都是一个断点，这时可以通过 Register、Local 和 Watch 窗口来观察程序中某些变量、表达式的值，以调试程序。它与 Debug 中的 Step out 命令功能及用法类似。

4. 窗口(Window)

VWB 集成环境中的大多数操作都发生在窗口中。窗口是屏幕上的一块区域，可以打开、关闭、移动、缩放、排列(Tile)多窗口、重叠(cascade)多窗口等。这里所介绍的窗口区相当于一个功能强大的编辑器，具有编辑器的所有操作功能。

5. 状态条(Status bar)

状态条位于屏幕的底部，它显示与当前屏幕有关的各种资源信息和系统信息。使用时应将 View 菜单中的 Status Bar 命令设置为 on，只要单击该命令即可。此时该命令前有一个标记(√)表示该命令生效，处于 on 设置，否则 Status bas 不显示出来。

状态条(Status bar)上的信息说明如下(如图 1.6 所示)：

①OVR：指出编辑器是处在 Overtype(覆盖)状态，还是处在 Insert(插入)状态。

②READ：指出当前活动文件是否是只读文件(Read_only)。

③CAPS：显示 CAPS LOCK 键的当前状态(on/off)。

④NUM：显示 NUM LOCK 键的当前状态(on/off)。

⑤LINE：显示当前光标所在的行号。

⑥COLUMN：显示当前光标所在的列号。

以上对 Visual Workbench 环境作了一个简单介绍，其中菜单条是最重要的部分，而工具条上的每个工具都可在菜单条上某个菜单中找到相应的命令项。因此，下一节将对菜单条上的每个菜单中的命令功能和用法作详细介绍，而不再对工具条等作详细说明，请参阅菜单条上的有关命令。