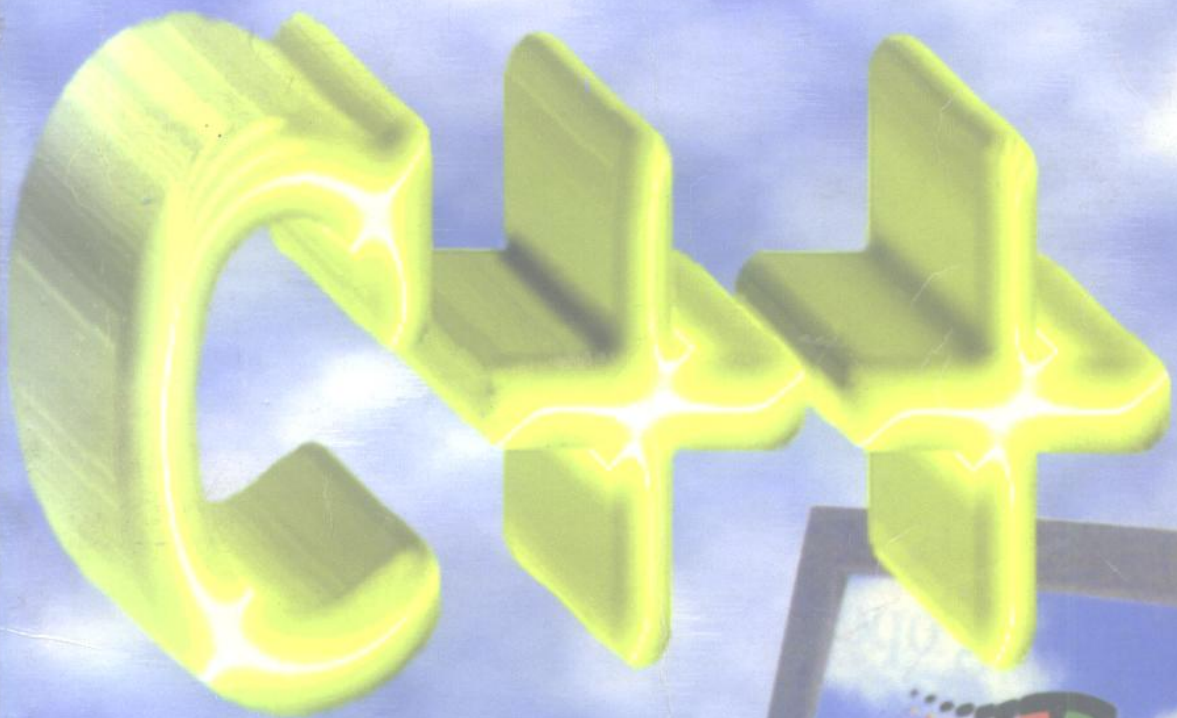
Visual C++ Windows

实用编程技术

周升锋　编著



北京航空航天大学出版社

Visual C++
Windows 实用编程技术

周升锋　编著

北京航空航天大学出版社

内 容 简 介

本书将 Windows 的编程进行分类，由浅入深，以学习实用编程技术为目的，逐步介绍各个部分，并列举大量程序实例和图例，讲解力求通俗易懂。

Windows 编程的学习重点在于掌握基本的代码结构和程序设计方法。本书向读者介绍了如何打破传统的程序设计方法，掌握 Windows 编程的技术技巧。

全书分为 2 个部分，共 12 章。第 1 章至第 8 章为基本编程部分，介绍了输出设计、输入设计、菜单和加速键设计、对话框、图形设计、文件存取；第 9 章到第 12 章介绍更高一级的编程技术，主要包括剪贴板、动态数据交换、动态链接库和多文档界面技术。书中例程全部调试通过。

本书可作为大专院校及各种学习班的教材，也适合广大软件开发者参考阅读。

图书在版编目(CIP)数据

Visual C++ Windows 实用编程技术/周升锋编著. —
北京：北京航空航天大学出版社，1996.12
ISBN 7-81012-667-9

Ⅰ.V… Ⅱ.周… Ⅲ.C 语言-程序设计 Ⅳ.TP312C

中国版本图书馆 CIP 数据核字(96)第 19251 号

书　　名：**Visual C++ Windows 实用编程技术**
　　Visual C++ Windows SHIYONG BIANCHENG JISHU
编 著 者：周升锋
责任编辑：章　华
责任校对：李宝田
出 版 者：北京航空航天大学出版社
地　　址：北京市海淀区学院路 37 号(邮编 100083，发行部电话 62015720)
印 刷 者：北京朝阳科普印刷厂印刷
发　　行：新华书店总店北京发行所
经　　售：全国各地书店
开　　本：787×1092　1/16
印　　张：25
字　　数：640 千字
印　　数：4000 册
版　　次：1996 年 12 月第 1 版
印　　次：1996 年 12 月第 1 次印刷
书　　号：ISBN 7-81012-667-9/TP・219
定　　价：34.50 元

目　录

上篇　基本编程技术

第 1 章　Visual C++ Windows 编程概述

第 2 章　基本输出设计

第 3 章　输入设计

第 6 章　对话框设计

第 7 章　与设备无关的图形接口

第 8 章　文件存取

下篇　高级编程技术

第 9 章　剪贴板

第 10 章　动态数据交换 DDE

第 11 章　动态链接库 DLL

第 12 章　多文档界面

上　篇

基本编程技术

Visual C++ Windows 编程概述
基本输出设计
输入设计
Windows/Visual C++系统资源
菜单和加速键设计
对话框设计
与设备无关的图形接口
文件存取

第 1 章 Visual C++ Windows 编程概述

1.1 Windows 程序特点

Windows 是 Microsoft 公司开发的一个图形窗口环境软件，最初是在 1983 年 11 月宣布的。1985 年 11 月推出了第一个公开版本。此后，Windows 经过了不断修改与更新，先后推出了 Windows95 和 Windows NT。Windows 的推出，立即引起了用户的强烈反响，使得操作计算机的方法和软件开发过程发生了根本性的变化。

Windows 运行于 MS-DOS 之上，是 MS-DOS 的扩展。它与 MS-DOS 共同管理计算机的硬件资源，MS-DOS 继续管理文件系统，而 Windows 管理其他一切，包括显示器、键盘、鼠标器、打印机和串行口，并负责内存管理和程序的执行、调度。

Windows 丰富多彩的界面改变了存储管理能力；能同时运行多个任务；可以通过多种方式进行应用程序间的数据交换，实现信息共享；能支持网络系统；在多媒体技术方面更是大显身手。

与 DOS 环境相比，Windows 无论是对用户还是对应用程序开发人员都提供了更加强大的功能，特别是其先进的程序设计思想和方法，更是程序设计人员必须熟悉和掌握的。

1.1.1 漂亮、统一的用户界面

用户界面友好是衡量应用程序质量的一个重要方面。在 DOS 环境下，若程序员采用 C 语言，为了设计出友好的用户界面，往往要花费很多的精力和时间，而且设计出的界面还是因人而异，没有统一的风格，这给用户使用和掌握应用程序带来了困难。

Windows 为设计漂亮、统一和友好的用户界面提供了一种全新的方法。Windows 是一个图形用户界面(GUI)，各种操作对象都是以图形形式显示在屏幕上。通过鼠标器或键盘，用户就可以在屏幕上直接操作这些对象。这样用户与程序之间的交互不再是单一的用键盘输入信息到屏幕上去显示，而是直接与屏幕上的对象进行交互操作。

所有 Windows 应用程序的基本外观均差不多，每个程序占据一个窗口。窗口是屏幕上的一个矩形区域，它是由标题栏来标识的。程序中绝大多数功能均可通过菜单来执行。某些菜单会触发出对话框，通过对话框，用户可以输入较多的附加信息。图 1-1 是一个典型的 Windows 程序界面。

下面对菜单的一些术语作一介绍：

系统菜单　系统菜单由 Windows 系统预先定义，并始终出现在窗口左上角。当用户用鼠标对这个区域进行选择时，就出现一个下拉菜单，其中有多个功能选项，包括移动窗口、改变窗口大小、关闭窗口、切换到其他窗口等等；

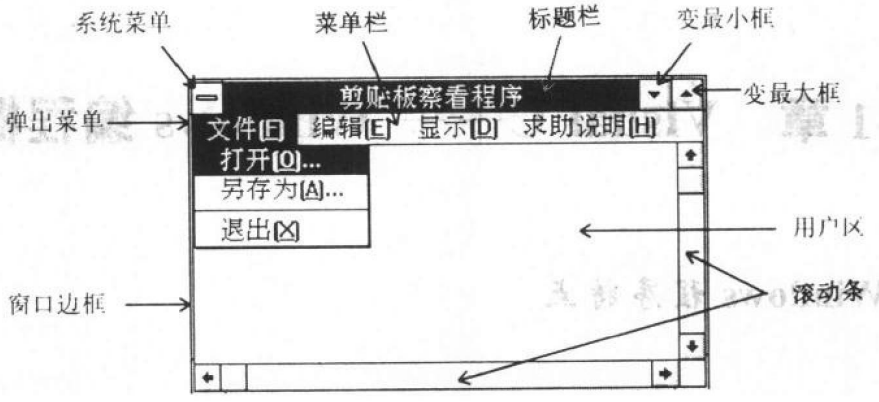


图 1-1　典型的 Windows 程序界面

标题栏　能显示一条说明信息，用来表示本窗口的功能特点。当这个区域的背景色为深颜色时，表示这个窗口当前是活动窗口；

菜单栏　显示用户菜单的第一级菜单选项。当用户用鼠标选择某一个选项时，可以出现这个选择项的二级菜单，从而用鼠标可以选择菜单中的任一选项。当一个选项名字中某一字母有下划线时，表示可以用 Alt+<字母键>来选择这个选项；

用户区　是主要的信息输入/输出的区域；

变最小框　当用户用鼠标选择这个区域，窗口将直接缩小成一个图标；

变最大框　当用户用鼠标选择这个区域，窗口将增大到最大可能的尺寸。

Windows 还提供了一种多文档界面技术，在一个应用程序中利用它可以创建多个窗口，分别用来显示和处理不同的文档。

由于用户界面的一致性，用户一旦学会了如何使用一种 Windows 应用程序，就会很快掌握其他 Windows 应用程序的使用方法。在 Windows 下设计统一的用户界面是很容易的事情。

1.1.2　面向对象的程序设计

Windows 程序设计方法实际上是面向对象的程序设计，这是目前最先进的程序设计方法之一。Windows 编程所涉及的对象很多，主要有窗口、菜单、对话框等。

如前所述，窗口是屏幕上的一个矩形区域，窗口通过键盘或鼠标接收用户的输入，并将图形输出到显示屏幕上。

菜单是 Windows 应用程序接收用户输入的主要手段。一个菜单是一组可供用户选择和查看的命令列表，用户只要选择菜单项就能执行相应的功能。

对话框是一种特殊的窗口，它为用户显示更多的有关命令的信息，并能接收用户的输入信息。一个对话框可以包含一个或多个控制窗口。控制窗口也是一种特殊的窗口，又称子窗口控制。控制窗口的形式包括各种按钮、文本编辑框、列表框、组合框和滚动条，可以供用户作出选择和输入。

用户在屏幕上看到这些对象，可直接与它们进行交互。交互的方法是按一下按钮、滚动一个滚动条或选择一个菜单项等。其实，用户所执行的这些操作是通过发送消息来告诉程序的，消息是应用程序执行命令的信号。

在面向对象的程序设计中，对象表示代码和数据的集合。以窗口为例，窗口是一个对象，其代码是窗口函数(该函数处理窗口的各种消息)，而数据就是与窗口相关的消息。

1.1.3 消息驱动的程序结构

Windows 消息提供了应用程序与应用程序之间和应用程序与 Windows 系统之间进行通讯的手段。几乎每个 Windows 程序所做的事情都是为了响应发送到窗口函数的某条消息。在大多数应用程序中，很大一部分代码内容是用来处理这些消息的。

Windows 的消息由三部分组成：消息号、字参数和长参数。消息号由事先定义的消息名标识；字参数和长参数包含与消息号相关的值。例如，当光标在窗口用户区时，按下鼠标器的左按钮，Windows 就向该窗口发送一条 WM _LBUTTONDOWN 消息(宏定义是在 windows.h 中定义的)，其长参数含有光标的 X 和 Y 坐标(分别在其低位字和高位字中)，而其字参数则包含指示各种虚拟键盘是否被按下的值。窗口函数接收到此消息后，就可以作出相应的处理。Windows 应用程序创建的每一个窗口都有一个窗口函数，用以处理发送到该窗口的消息。窗口函数由 Windows 采用消息驱动的形式直接调用，而不是由应用程序显式地调用的。窗口函数处理完消息后又将控制权返回给 Windows。

Windows 中有一个系统消息队列，简称系统队列。当开始执行一个 Windows 应用程序时，Windows 为该程序建立一个“消息队列”，称为应用程序队列。这个队列存放该程序可能创建的各种窗口的所有消息。程序中含有一段“消息循环”代码，用来从消息队列中检索这些消息并把它们分发到相应的窗口函数中。有些消息不必放到消息队列中，而直接送给窗口函数。系统队列、应用程序队列、消息循环、窗口函数之间的关系如图 1-2 所示。

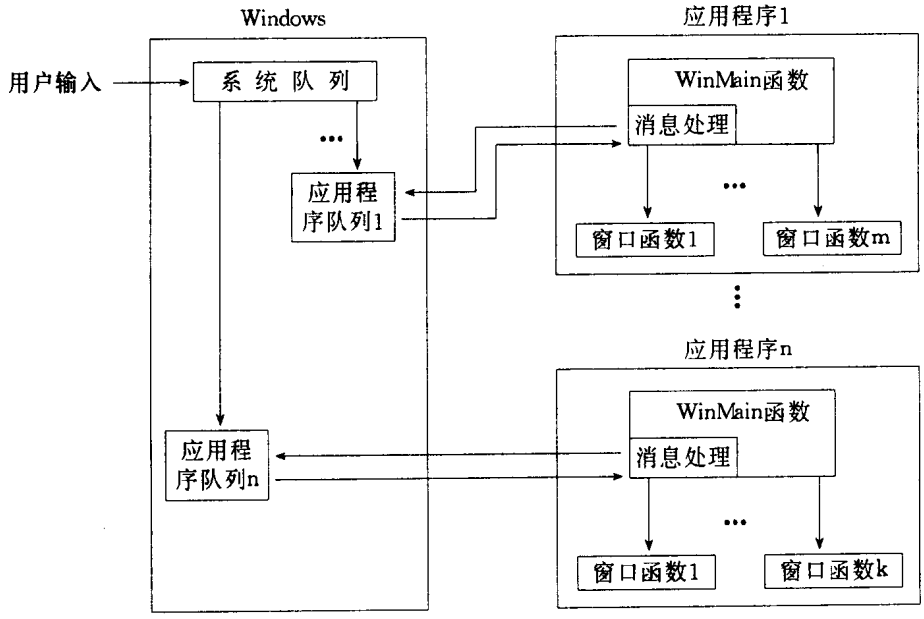


图 1-2 Windows 程序中的消息处理

Windows 应用程序接收用户输入的方式也不同于 DOS 环境下的应用程序。在 DOS 环境下，程序通过显式地调用一个函数(如 getch())来读取键盘的输入。该函数一般要等待用户按下一个键后，再把相应的字符返回给程序。在 Windows 环境下，Windows 接收所有来自键盘、

鼠标器等外设的输入，并把它们放在相应的应用程序的消息队列中。在应用程序需要获取输入时，只不过是简单地从消息队列中读取下一个输入的消息。

1.1.4 多任务

Windows 是一种多任务系统，可以同时显示和运行多个应用程序，每个应用程序都有各自的窗口。用户可以在屏幕上移动这些窗口，改变它们的尺寸，在不同的程序间进行切换，并可以在应用程序之间进行数据交换。

在标准 MS-DOS 环境下编写应用程序时，一般都假定它独占计算机的所有资源，这些资源包括输入输出设备、内存、系统显示器、CPU 等；而在 Windows 环境下，应用程序必须与所有当前正在运行的其他应用程序共享这些资源。因此，程序设计时要考虑多任务的特点，协调并共享资源，以避免耗尽资源。

Windows 的多任务是通过在程序之间进行切换来实现的。只有当正在运行的应用程序消息队列中没有消息(或只有 WM _ PAINT 和 WM _ TIMER 消息)时，才能转去执行其他等待消息的程序，因此 Windows 是一种非抢先的多任务系统。

1.1.5 高效的内存管理

Windows 为保证多任务的运行，采取了一系列措施来提高内存的利用率，从而保证能以较少的内存资源运行较大的程序。

Windows 内存管理的特点是：

(1) 当 Windows 运行同一应用程序的多个拷贝(每个拷贝称为一个实例)时，每个实例使用相同的代码段和相同的资源；

(2) 在 Windows 环境中分配的内存块多数是可移动的，从而便于 Windows 对内存块的管理和提高内存资源利用率；

(3) 代码段和程序资源通常都是视需要而被装入内存的，而且在多数情况下分配为可丢弃的内存块。在内存资源紧张时，这种内存块可以被丢弃而释放所占内存空间。当再次需要时，Windows 能自动地从磁盘上复制相应的内容到内存中。

Windows 突破了 MS-DOS 对内存 640KB 的限制，提供了实模式、标准模式和 386 增强模式等运行模式。标准模式和 386 增强模式统称为保护模式。

在基于 Intel 8086CPU(或 80286、80386 及以上的 CPU，内存少于 1MB)的机器上，Windows3.0 将以实模式运行。在实模式下，Windows 系统以及 Windows 应用程序占用 640KB 常规内存的上部，而 MS-DOS、设备驱动程序以及内存驻留程序占用 640KB 常规内存的下部。Windows 3.1 不支持实模式。

在基于 80286 CPU，内存至少 1MB，或 80386 及以上 CPU，内存在 1MB～2MB 之间的机器上，Windows3.X 将以标准模式运行。标准模式又称为 286 兼容的保护模式。在标准模式下 Windows 可使用多达 16MB 的常规内存和扩展内存。

在基于 80386 CPU，内存至少 2MB 的机器上，Windows 3.X 将以 386 增强模式运行。这种模式的主要特点是使用 CPU 的页寄存器实现虚拟内存管理。CPU 的内存页面长度规定为 4KB，Windows 可将内存页面交换到磁盘上，需要时再从磁盘上重新装入内存。页交换和重新装入的过程均由 Windows 系统自动进行。在编写应用程序时，不必考虑页交换的过程。

当 Windows 系统启动时，它根据机器的硬件配置自动选择适当的运行模式运行。用户也可以从 DOS 命令中使用/R(实模式)、/2(标准模式)、/3(386 增强模式)参数来启动 Windows 系统，以改变 Windows 的缺省模式。

1.1.6 数据交换与共享

Windows 提供了多种手段来实现应用程序之间的数据交换与共享，包括剪贴板(Clipboard)、动态数据交换(DDE)和动态链接库(DLL)。

剪贴板是共享内存块的管理器，应用程序既可以向其中写入数据，也可以从中读出数据。利用剪贴板，能实现应用程序内部或应用程序之间的数据交换。

动态数据交换是一种更高级的数据交换手段，它是一种数据交换的消息协议。通过建立应用程序之间的数据链接，使得应用程序之间可以进行自动的数据传送，而不需要用户干预。这种功能对于开发需要链接实时数据的应用程序具有十分重要的意义。

动态链接库是应用程序之间实现代码和资源共享的一种手段。在 Windows 环境下，由于可以同时执行多个任务，使用 DLL 比使用静态链接库具有更多的优点。如果两个应用程序同时运行，且它们使用了某一静态库中的同一函数，那么系统中就会出现该函数的两个副本；而 DLL 却能允许若干个应用程序共享某个函数的单个副本，从而节省了内存空间。此外，DLL 还可以用来实现其他资源的共享，如数据和硬件资源的共享。

Windows 所有的运行库均是 DLL，如 GDI.EXE，USER.EXE 等，它们是 Windows 的主要构成部分之一。标准的 Windows 设备驱动程序也是用 DLL 实现的。程序员也可以根据需要，开发出自己的 DLL。

1.1.7 与设备无关的图形接口

Windows 提供了丰富的、与设备无关的图形处理功能。应用程序能很方便地画出各种图形，而不需要直接与具体的输出设备打交道。由于 Windows 提供了设备无关性，因此在应用程序中可以使用同一函数在打印机上或显示器上输出同一图形。

Windows 应用程序并不直接访问图形显示设备，而是通过其图形程序设计语言(图形设备接口，即 GDI)来实现图形输出。Windows 应用程序的输出可以适用于任何显示或打印设备，只要在 Windows 环境下安装相应的设备驱动程序即可。由设备驱动程序将 GDI 图形输出请求转换为显示器、打印机等输出设备上的图形输出。

1.2 Windows/Visual C++编程环境

1.2.1 开发环境

从开发应用程序的角度看，我们建议最好使用如下的硬件环境：

(1) Intel 80386 或以上的处理器；

(2) 8MB 以上的内存空间，200MB 以上的硬盘；

(3) VGA 以上的显示卡和相应的彩色显示器；

(4)一个 Microsoft 或兼容鼠标器。

建议用如下软件环境：

(1) MS-DOS 6.22；

(2) Microsoft Visual C++ 1.51 以上；

(3) Windows 3.2。

1.2.2 Visual C++ Workbench 编程环境

Visual C++编译器所支持的最重要的工具就是程序员工作平台，Microsoft 习惯上把它称作 Visual C++ Workbench。Visual C++ Workbench 是一个集成开发环境，共有 10 个菜单选项，每个菜单包含一个选择组，这些选择执行某项任务或设置一个选项。

在此环境下运行应用程序非常方便。用户首先进入 Windows 环境，如图 1-3 所示。

图 1-3　包含 Visual C++程序组的 Windows 界面

用鼠标双击 Visual C++图标(或光标在 Visual C++上时按 ENTER 键)，将看到 Visual C++的 Workbench 编程环境，如图 1-4 所示。

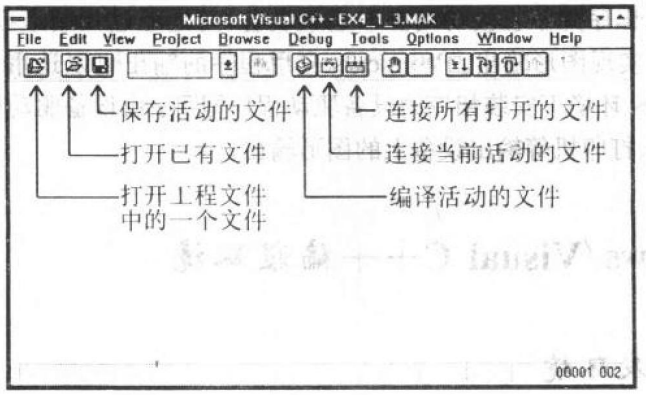


图 1-4　Visual C++ Workbench 编程环境

1.2.3 Visual C++ 编辑器的用法

Visual C++ Workbench 编辑器是用户花费时间最多的地方，可输入新的源程序代码，编辑已有的程序或改正程序中的错误。用户通过 File 菜单选择 New 或 Open 命令打开文件进入编辑。Open 命令对已存在的文件适用，New 命令用于建立新文件。

大量的键盘命令、鼠标移动和菜单选择为编辑系统提供了丰富的功能。当用户在文本编辑窗口中输入文本时，光标从左至右在屏幕上移动。当结束一个代码行时，按回车键 Enter。

1. 编辑控制键

在编辑窗口中，屏幕底部的水平滚动条显示光标的当前列位置，它相对于光标所在行的开始和末尾。垂直滚动条显示光标当前行的位置，它相对于文件中文本第一行和最后一行。每个滚动条都由一个滚动块标示，并且可由鼠标快速地拖动到文本的其他位置。Visual C++编辑控制键的命令见表 1-1。

表 1-1

功　　能	按　　键
左移一个字符	左箭头键
右移一个字符	右箭头键
左移一个词	Ctrl+左箭头
右移一个词	Ctrl+右箭头
上移一行	上箭头键
下移一行	下箭头键
移到当前行的第一个缩进位置	Home 键
移到当前行的开始位置	Home 键，Home 键
移至行末尾	End 键
移至文件开始处	Ctrl+Home 键
移至文件末尾处	Ctrl+End 键
删除光标左边字符	Backspace 键
删除光标右边字符	Del 键
删除选择好的文本块并复制到剪贴板	Ctrl+X，Shift+Del 键
键盘插入/覆盖方式切换	Ins 键
把选择好的文本块复制到剪贴板并保存	Ctrl+C，Ctrl−Ins 键
插入剪贴板的内容	Ctrl+V，Shift+Ins 键
当前行复制到剪贴板后删除	Ctrl+Y
取消最后一次的编辑操作	Ctrl+Z，Alt+Backspace 键
对于选择好的行右移一个 Tab 位	Tab 键
对于选择好的行左移一个 Tab 位	Shift+Tab 键
上滚一页	PgUp 键
下滚一页	PgDn 键
打开搜索对话框	Alt+F3 键
搜索指定字符串	Ctrl+F3 键
搜索下一个指定字符串，向前找	F3 键
搜索下一个指定字符串，向后找	Shift+F3 键
查找配对的括号	Ctrl+]
设置一个标签	Ctrl+F2 键
跳到下一个标签	F2 键