

## Лабораторная работа № 14

**Тема:** Разработка программ с меню, статусной строкой и диалоговыми окнами на Win32API.

**Цель:** Освоить создание приложений содержащих главное и контекстное меню, научиться описывать ресурсы, создавать и работать с диалоговыми окнами.

### Краткая теория

Меню используют для выбора команд и изменения режимов работы приложения. Если строка меню связана с командой, при ее выборе приложение получает сообщение WM\_COMMAND и идентификатор этой команды. Основным элементом меню являются строки или графические объекты.

Виды меню:

- Главное меню – меню, находящееся ниже заголовка окна, и его строки расположены в одну или несколько линий. При выборе строки главного меню, как правило, активизируется раздел меню. Раздел меню представляет собой временное меню.
- Контекстное (плавающее) меню – временное меню, которое может появиться в любой точке рабочей области (как правило, при нажатии правой кнопки мыши).
- Системное меню – меню, расположенное слева от текста заголовка окна. По способу создания различают статическое и динамическое меню.
- Статическое меню создают до запуска и не изменяют в процесса работы приложения.
- Динамическое меню создают в процессе работы приложения и после создания оно может быть изменено.

Любой элемент меню может быть описан с помощью структуры MENUITEMINFO:

```
typedef struct tagMENUITEMINFO{  
    UINT cbSize;           //Размер структуры  
    UINT fMask;            //Маска параметров меню  
    UINT fType;            //Тип элемента меню  
    UINT fState;           //Состояние элемента меню  
    UINT wID;              //Идентификатор команды, этого элемента  
    HMENU hSubMenu;       //Дескриптор подменю  
    HBITMAP hbmpChecked;   //Дескриптор изображения метки для отмеч. элемента  
    HBITMAP hbmpUnchecked; //Дескриптор изображения метки для неотмеч. элемента  
    ULONG_PTR dwItemData;  //Определяемое приложением значение, связанное с  
                          //элементом  
    LPCTSTR dwTypeData;    //Содержание отображения меню  
    UINT cch;              //Длина текста элемента меню или 0  
    HBITMAP hbmpItem;      //Дескриптор изображения элемента  
} MENUITEMINFO, *LPMENUITEMINFO;
```

Значения поля fMask определяют с какими из полей описания элементов меню работать:

- MIIM\_CHECKMARKS – hbmpChecked и hbmpUnchecked,
- MIIM\_DATA – dwItemData,
- MIIM\_ID – wID,
- MIM\_STATE – fState,
- MIIM\_SUBMENU – hSubMenu,
- MIIM\_TYPE – fType и dwTypeData.

Значения поля fType:

- MFT\_BITMAP – элемент меню отображать графическим объектом, дескриптор которого находится в младшем слове dwTypeData, а поле cch игнорировать.
- MFT\_MENUBARBREAK – элемент меню отображать в начале новой строки или нового столбца. Между столбцами появится вертикальная линия.
- MFT\_MENUBREAK – тоже самое, но между столбцами не появляется вертикальная линия.
- MFT\_RADIOCHECK – если hbmpChecked==NULL, то элемент помечать как радиокнопку.
- MFT\_RIGHTJUSTIFY – выравнивать этот и последующие элементы по правому краю.
- MFT\_SEPARATOR – в качестве элемента отобразить разделитель в виде горизонтальной линии.
- MFT\_STRING – элемент меню отображать в виде строки, содержащейся в dwTypeData. Значения поля fState:
  - MFS\_CHECKED – отмечен,
  - MFS\_DEFAULT – элемент по умолчанию,
  - MFS\_DISABLED – заблокирован,
  - MFS\_ENABLED – в активном состоянии,
  - MFS\_GRAYED – заблокирован,
  - MFS\_HILITE – подсвечен,
  - MFS\_UNCHECKED – не отмечен,
  - MFS\_UNHILITE – не подсвечен.

Алгоритм создания меню в общем случае содержит следующие шаги:

1. Создание пустого меню.
2. Вставка элемента в меню.
3. Шаг 2 повторяется столько раз, сколько элементов должно содержать меню.

Алгоритм подключения главного меню к окну и отображения полосы меню:

1. Создание главного меню (аналогично предыдущему алгоритму).
2. Подключение главного меню к окну.
3. Перерисовка полосы меню окна.

Функции для работы с меню:

Создание пустого главного меню:

HMENU CreateMenu (void) ;

Создание пустого временного меню:

HMENU CreatePopupMenu (void) ;

Функция подключения главного меню к окну:

BOOL SetMenu (HWND hwnd, HMENU hmenu) ;

Функция перерисовки строки меню (независимо от состояния окна): void WINAPI DrawMenuBar (HWND hwnd) ;

Функция вставки элементов в меню:

BOOL InsertMenuItem (HMENU hMenu, UINT uItem, BOOL fByPosition, LPCMENUITEMINFO lpmi) ;

- hMenu – идентификатор меню,
- uItem содержит номер позиции перед которой необходимо вставить новый элемент (если fByPosition=TRUE, иначе – идентификатор команды вставляемого элемента).
- lpmi – указатель на заполненную структуру MENUITEMINFO.

Функция удаления элемента из меню. Если элемент содержит временное меню, то функция

удаляет и это меню.

```
BOOL DeleteMenu(HMENU hMenu, UINT uPosition, UINT uFlags);
```

- hMenu – дескриптор изменяемого меню,
- uPosition – определяет удаляемый элемент:
- Если uFlags = MF\_BYCOMMAND, то значение uPosition – идентификатор удаляемого элемента.
- Если uFlags = MF\_BYPOSITION, то значение uPosition – номер позиции удаляемого элемента.

Функция удаления элемента из меню, не разрушая связанные с ним элементы.

```
BOOL RemoveMenu(HMENU hMenu, UINT uPosition, UINT uFlags);
```

Функция удаления всего меню:

```
BOOL DestroyMenu(HMENU hMenu);
```

Используя эту функцию можно удалять те меню, которые были созданы, но не подключены. Подключенные меню удаляются при разрушении окна.

Функция изменения состояния элемента меню:

```
BOOL SetMenuItemInfo(HMENU hMenu, UINT uItem, BOOL  
fByPosition, LPMENUITEMINFO lpmi);
```

Функция чтения состояния элемента меню:

```
BOOL GetMenuItemInfo(HMENU hMenu, UINT uItem, BOOL  
fByPosition, LPMENUITEMINFO lpmi);
```

В этих функциях, если fByPosition=FALSE, то uItem – идентификатор команды элемента, иначе uItem – номер позиции элемента в меню.

Функция определения главного меню окна:

```
HMENU GetMenu(HWND hwnd);
```

Функция определения временного меню по его позиции:

```
HMENU GetSubMenu(HMENU hMenu, int nPos);
```

Функция проверки существования меню:

```
BOOL IsMenu(HMENU hMenu);
```

Функция определения количества элементов в меню: **int** GetMenuItemCount(HMENU hMenu);

От элементов меню могут поступать следующие сообщения:

- WM\_INITMENU - поступает перед отображением главного меню. Параметр wParam равен дескриптору меню.
- WM\_INITMENUPOPUP – поступает перед отображением временного меню. Параметр wParam равен дескриптору меню. Младшее слово параметра lParam равно позиции этого меню в меню верхнего уровня, старшее слово – равно 1 для системного меню и 0 – для обычного.
- WM\_COMMAND – поступает после выбора строки меню. Младшее слово wParam равно идентификатору выбранной команды.
- WM\_MENUSELECT – поступает в процессе курсора меню по строкам меню. Младшее слово wParam содержит идентификатор команды или позиции строки (если при выборе строки отображается временное меню), а старшее слово содержит флажки состояния элементов меню:
  - MF\_CHECKED – отмечен,
  - MF\_DISABLED – заблокирован,
  - MF\_GRAYED – недоступен,

- MF\_HILITE – высвечен,
- MF\_MOUSESELECT – выбран мышью,
- MF\_POPUP – открывает временное окно,
- MF\_SYSMENU – принадлежит системному меню окна.

Контекстное меню создают обычным способом, но не вставляют в другое меню. Для отображения и выбора строк контекстного меню используют функцию:

```
BOOL TrackPopupMenu(HMENU hMenu, UINT uFlags, int x, int y,
                    int nReserved, HWND hwnd, CONST RECT
                    *prcRect);
```

- hMenu – дескриптор отображаемого плавающего меню. Он может быть создан функцией CreatePopupMenu или получен с помощью GetSubMenu.
- X – координата по горизонтали от левого края экрана,
- Y – координата по вертикали от правого края экрана,
- hwnd – дескриптор окна-владельца.
- prcRect – указатель на прямоугольную область, находясь в пределах которой можно работать с меню.

Значения uFlags	
TPM_CENTERALIGN	Центр меню по горизонтали совпадает с X
TPM_LEFTALIGN	Левый край меню по горизонтали совпадает с X
TPM_RIGHTALIGN	Правый край меню по горизонтали совпадает с X
TPM_BOTTOMALIGN	Нижний край меню по вертикали совпадает с Y
TPM_TOPALIGN	Верхний край меню по вертикали совпадает с Y
TPM_VCENTERALIGN	Центр меню по вертикали совпадает с Y
TPM_NONOTIFY	Не посылать сообщения о выборе строки
TPM_RETURNCMD	Возвращать идентификатор вызванной команды
TPM_LEFTBUTTON	Прослеживает левую кнопку мыши
TPM_RIGHTBUTTON	Прослеживает правую кнопку мыши

Создание строки состояния осуществляется функцией:

```
HWND CreateStatusWindow(LONG style, LPCTSTR lpszText, HWND hwndParent, UINT
wID);
```

- style – стиль окна, должен быть WS\_CHILD | WS\_VISIBLE.
- lpszText – строка, содержащая первоначальный текст строки состояния.
- hwndParent – дескриптор родительского окна.
- wID – идентификатор строки состояния.

Сообщения, отсылаемые строке состояния:

- SB\_SETPARTS – установить количество секций строки состояния.
  - wParam – количество секций,
  - lParam – указатель на массив длин секций.
- SB\_SETTEXT – установить текст секции строки состояния.

- wParam – номер секции,
- lParam – указатель на текст.

При создании приложений с использованием Win32API очень удобно использовать ресурсные файлы – ASCII файлы с расширением .rc, в которых в специальном формате описываются различные элементы приложения. Файл описания ресурсов транслируется компилятором ресурсов (rc.exe – Visual Studio, rorc.exe – PellesC) в бинарный файл с расширением .res. Затем компоновщик подключает этот файл вместе с объектными файлами (.obj) и файлами статических библиотек (.lib) в исполняемый файл программы.

Как правило, интегрированные среды разработки приложений содержат специальные редакторы ресурсов, которые визуализируют процесс описания ресурсов приложения, делая его более удобным.

Редактор ресурсов позволяет создавать и редактировать следующие элементы: пиктограммы, курсоры, растровые образы, меню, диалоговые окна, таблицы строк и горячих клавиш, редактор информации о версии и др. Ресурсный файл может создаваться мастером автоматически или может быть добавлен в проект после его создания.

Диалоговые окна являются подвидом обычных окон, имеющих класс «всплывающих» окон, и содержат элементы управления, которые рассматриваются как дочерние окна диалогового окна.

Диалоговые окна подразделяются на два типа:

- модальные – окна, которые при открытии блокируют все остальные окна приложения, а разблокируют после закрытия.
- немодальные – окна, которые работают в «параллельном» по отношению к другим окнам режиме.

Создание и обработка диалогового окна требует выполнения следующей последовательности действий:

1. Определение шаблона диалогового окна.
2. Определение диалоговой процедуры.
3. Вызов функции создания диалогового окна.
4. Обмен данными между диалоговой процедурой и вызывающей функцией окна.

Определение шаблона диалогового окна осуществляется в файле ресурсов, как правило с использованием визуального редактора.

Диалоговая функция реализуется аналогично функции обработки сообщений обычного окна.

```

BOOL CALLBACK DlgProc(HWND hDlg, UINT msg, WPARAM wParam, LPARAM lParam)
{
    switch(msg) {
        case WM_INITDIALOG:
            ...
            return
            TRUE; case
            WM_CLOSE:
                EndDialog(hDlg,
                0); return TRUE;
        case WM_COMMAND:
            ...
            return TRUE;
    }
    return FALSE;
}

```

Функция открытия диалогового окна в модальном режиме:

```

INT_PTR DialogBox(HINSTANCE hInst, LPDLGITEMTEMPLATE lpTemp, HWND
                hWndParent, DLGPROC lpDialogFunc)

```

Функция открытия диалогового окна в не модальном режиме:

```

HWND CreateDialog(HINSTANCE hInst, LPDLGITEMTEMPLATE lpTemp, HWND

```

hWndParent, DLGPROC lpDialogFunc)

- hInst – дескриптор приложения.
- lpTemp – указатель на ресурс, содержащий описание диалогового окна. Обычно используется макрос MAKEINTRESOURCE, в который передается идентификатор ресурса диалогового окна.
- hWndParent – дескриптор родительского окна.
- lpDialogFunc – указатель на функцию-обработчик диалогового окна.

Функция завершения диалогового окна:

BOOL EndDialog(HWND hDlg, INT\_PTR nResult)

- hDlg – дескриптор диалогового окна,
- nResult – значение, возвращаемое из функции DialogBox.

Функция определения дескриптора элемента управления:

HWND GetDlgItem(HWND hDlg, int nIDDlgItem)

Функция определения идентификатора элемента управления:

int GetDlgItem(HWND hCtrl)

Функция чтения текста элемента управления:

UINT GetDlgItemText(HWND hDlg, int ctrlID, LPTSTR lpString, int MaxLen)

Функция записи текста элемента управления:

UINT SetDlgItemText(HWND hDlg, int ctrlID, LPTSTR lpString, int MaxLen)

Функция чтения текста в форме целочисленного значения элемента управления:

UINT GetDlgItemInt(HWND hDlg, int ctrlID, BOOL \*lpTranslated, BOOL bSigned)

- lpTranslated – признак успешности преобразования текста в число,
- bSigned – признак ввода знакового числа.

### Ход работы

В рамках данной лабораторной работы необходимо, используя Win32API, разработать программу, в главном окне которого отображается список записей определенного типа. В главном меню окна содержится два пункта: «Файл» и «Сортировка». В пункте «Файл» содержатся пункты: «Открыть», «Сохранить» и «Выход». При выборе пункта «Открыть» осуществляется загрузка списка из бинарного файла (имя кодируется жестко в программе). При выборе пункта «Сохранить» осуществляется сохранение списка в бинарный файл с тем же именем, что и при открытии файла. При выборе пункта «Выход» осуществляется завершение программы. Пункт «Выход» должен быть отделен от остальных пунктов разделителем. В пункте «Сортировать» содержатся пункты для выполнения сортировки списка, набор пунктов задается в индивидуальном варианте задания.

При вызове контекстного меню на списке отображается меню, содержащее пункты «Добавить», «Изменить», «Удалить». При выборе пунктов «Новый» или «Изменить» открывается диалоговое окно в модальном режиме, для создания нового элемента списка или изменения выбранного элемента списка. При выборе пункта «Удалить» у пользователя запрашивается подтверждение удаления и, в случае утвердительного ответа, выбранный в списке элемент удаляется.

Диалоговое окно для создания или редактирования записей должно позволять вводить или редактировать все атрибуты типа данных. А также содержать две кнопки: «ОК» - подтверждения создания или редактирования элемента, «Отмена» - отказ от создания или редактирования элемента.

Для разработки программы необходимо создать проект типа Win32 Application.

Все текстовые редакторы, используемые для ввода значений, должны быть ограничены по длине. Если предполагается ввод целочисленного значения, то в текстовом редакторе

должны быть запрещен ввод не числовых данных. При реализации полей ввода дат и времени желательно использовать специализированные компоненты.

При реализации использовать структуры данных из лабораторной работы №7.

Варианты заданий:

№	Структура	Сортировка
1	СТУДЕНТ	По ФИО в порядке возрастания, по ФИО в порядке убывания, по курсу в порядке возрастания, по курсу в порядке убывания
2	КНИГА	По названию в порядке убывания, по названию в порядке возрастания, по году издания в порядке возрастания, по году издания в порядке убывания
3	МАШИНА	По марке в порядке убывания, по марке в порядке возрастания, по году выпуска в порядке возрастания, по году выпуска в порядке убывания.
4	ФАЙЛ	По имени файла в порядке убывания, по имени файла в порядке возрастания, по времени создания в порядке возрастания, по времени создания в порядке убывания.
5	ПРОЦЕССОР	По производителю в порядке убывания, по производителю в порядке возрастания, по тактовой частоте в порядке возрастания, по тактовой частоте в порядке убывания.
6	САМОЛЕТ	По марке-модели в порядке убывания, по марке-модели в порядке возрастания, по году выпуска в порядке возрастания, по году выпуска в порядке убывания.
7	БИЛЕТ	По названию сеанса в порядке убывания, по названию сеанса в порядке возрастания, по номеру ряда в порядке возрастания, по номеру ряда в порядке убывания.
8	СООБЩЕНИЕ	По номеру устройства в порядке убывания, по номеру устройства в порядке возрастания, по тексту в порядке возрастания, по тексту в порядке убывания.
9	НАКЛАДНАЯ	По номеру накладной в порядке убывания, по номеру накладной в порядке возрастания, по ФИО ответственного лица в порядке возрастания, по ФИО ответственного лица в порядке убывания.
10	ФИРМА	По названию в порядке убывания, по названию в порядке возрастания, по УНН в порядке возрастания, по УНН в порядке убывания.
11	МАРШРУТ	По номеру в порядке убывания, по номеру в порядке возрастания, по станции отправления в порядке возрастания, по станции отправления в порядке убывания.
12	ТЕЛЕФОН	По номеру в порядке убывания, по номеру в порядке возрастания, по тарифному плану в порядке возрастания, по тарифному плану в порядке убывания.
13	ВИДЕОДИСК	По названию в порядке убывания, по названию в

		порядке возрастания, по продолжительности в порядке возрастания, по продолжительности в порядке убывания.
14	ПОЕЗД	По названию в порядке убывания, по названию в порядке возрастания, по количеству вагонов в порядке возрастания, по количеству вагонов в порядке убывания.
15	УСТРОЙСТВО	По модели в порядке убывания, по модели в порядке возрастания, по серийному номеру в порядке возрастания, по серийному номеру в порядке убывания.
16	СТУДЕНТ	По специальности в порядке возрастания, по специальности в порядке убывания, по дате рождения в порядке возрастания, по дате рождения в порядке убывания.
17	КНИГА	По ФИО автора в порядке убывания, по ФИО автора в порядке возрастания, по количеству страниц в порядке возрастания, по количеству страниц в порядке убывания.
18	МАШИНА	По серийному номеру в порядке убывания, по серийному номеру в порядке возрастания, по цвету в порядке возрастания, по цвету в порядке убывания.
19	ФАЙЛ	По размеру файла в порядке убывания, по размеру файла в порядке возрастания, по дате создания в порядке возрастания, по дате создания в порядке убывания.
20	ПРОЦЕССОР	По модели в порядке убывания, по модели в порядке возрастания, по суммарному размеру КЭШ-памяти в порядке возрастания, по суммарному размеру КЭШ-памяти в порядке убывания.
21	САМОЛЕТ	По признаку в порядке убывания, по признаку в порядке возрастания, по времени налета в порядке возрастания, по времени налета в порядке убывания.
22	БИЛЕТ	По дате и времени в порядке убывания, по дате и времени в порядке возрастания, по номеру места в порядке возрастания, по номеру места в порядке убывания.
23	СООБЩЕНИЕ	По ID сообщения в порядке убывания, по ID сообщения в порядке возрастания, по дате и времени в порядке возрастания, по дате и времени в порядке убывания.
24	НАКЛАДНАЯ	По дате накладной в порядке убывания, по дате накладной в порядке возрастания, по общей сумме в порядке возрастания, по общей сумме в порядке убывания.
25	ФИРМА	По ФИО владельца в порядке убывания, по ФИО владельца в порядке возрастания, по дате основания в порядке возрастания, по дате



		основания в порядке убывания.
26	МАРШРУТ	По конечной станции в порядке убывания, по конечной станции в порядке возрастания, по времени отправления в порядке возрастания, по времени отправления в порядке убывания.
27	ТЕЛЕФОН	По ФИО владельца в порядке убывания, по ФИО владельца в порядке возрастания, по дате подключения в порядке возрастания, по дате подключения в порядке убывания.
28	ВИДЕОДИСК	По кинокомпании в порядке убывания, по кинокомпании в порядке возрастания, по серийному номеру в порядке возрастания, по серийному номеру в порядке убывания.
29	ПОЕЗД	По номеру в порядке убывания, по номеру в порядке возрастания, по станции отправления в порядке возрастания, по станции отправления в порядке убывания.
30	УСТРОЙСТВО	По ФИО в порядке убывания, по ФИО владельца в порядке возрастания, по дате основания в порядке возрастания, по дате основания в порядке убывания.
31	ЛЕКАРСТВО	По наименованию в порядке убывания, по наименованию в порядке возрастания, по производителю в порядке возрастания, по производителю в порядке убывания.
32	ФИЛЬМ	По названию в порядке убывания, по названию в порядке возрастания, по году выхода в порядке возрастания, по году выхода в порядке убывания.
33	ДЕТАЛЬ	По артикулу в порядке убывания, по артикулу в порядке возрастания, по материалу в порядке возрастания, по материалу в порядке убывания.
34	ПАССАЖИР	По серии паспорта в порядке убывания, по серии паспорта в порядке возрастания, по номеру рейса в порядке возрастания, по номеру рейса в порядке убывания.
35	АБОНЕНТ	По номеру телефона в порядке убывания, по номеру телефона в порядке возрастания, по улице в порядке возрастания, по улице в порядке убывания.
36	ТОВАР	По артикулу в порядке убывания, по артикулу в порядке возрастания, по производителю в порядке возрастания, по производителю в порядке убывания.
37	ЧЕМПИОНАТ	По названию в порядке убывания, по названию в порядке возрастания, по количеству участников в порядке возрастания, по количеству участников в порядке убывания.
38	ДОМАШНЕЕ ЖИВОТНОЕ	По виду в порядке убывания, по виду в порядке возрастания, по породе в порядке возрастания, по породе в порядке убывания.
39	ФУТБОЛИСТ	По специализации в порядке убывания, по специализации в порядке возрастания, по названию

		команды в порядке возрастания, по названию команды в порядке убывания.
40	СОТРУДНИК	По отделу в порядке убывания, по отделу в порядке возрастания, по должности в порядке возрастания, по должности в порядке убывания.