

Меню

Виды меню

Главное меню (относится ко всему приложению)

Системное меню (вызывается щелчком по иконке)

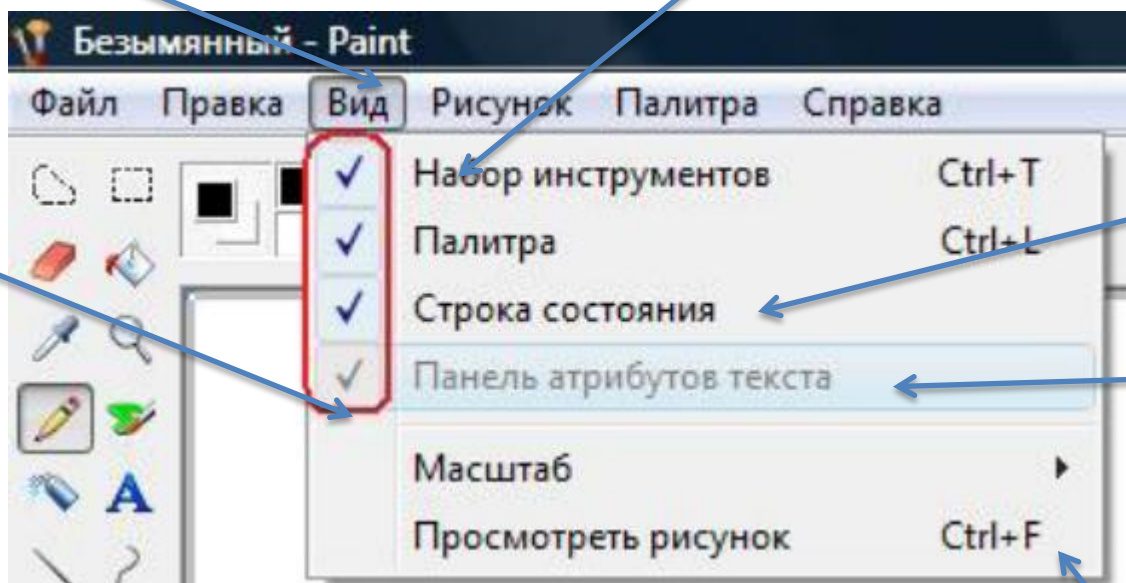
Контекстное меню (вызывается щелчком правой кнопкой)

Главное меню

выпадающее меню
(popup menu; пункт-подменю)

пункт-команда
(при выборе выполняется
некоторое действие)

разделитель
(сепаратор)



разрешенный
пункт меню

недоступный
пункт меню

акселератор

Создание меню с помощью IDE

- 1) добавить меню в ресурсы
(Add/Resource/Menu/New)
- 2) в открывшемся редакторе ресурсов **создать необходимые пункты** меню и **установить их свойства**
 - ID (идентификатор пункта-команды меню)
 - Caption (текст пункта; с помощью & перед буквой можно задать горячую клавишу)
 - Grayed (недоступность пункта меню)
 - Checked (пункт отмечен флажком)
 - Separator (устанавливает разделитель)

Создание меню с помощью IDE

- 3) загрузить меню с помощью функции
LoadMenu

```
HMENU LoadMenu(HINSTANCE hInst, // дескриптор
                // приложения
                LPCTSTR lpszMenuID) // указатель на строку
                                   // с ID меню из ресурсов
```

- 4) установить меню с помощью функции
SetMenu

```
BOOL SetMenu(HWND hWnd, // дескриптор родит. окна
              HMENU hMenu) // дескриптор меню
```

Создание главного меню (пример)

```
static HMENU hMenu;
```

```
// ...
```

```
case WM_INITDIALOG:
```

```
    hMenu = LoadMenu(GetModuleHandle(NULL),  
                      MAKEINTRESOURCE(IDR_MENU1));
```

```
    SetMenu(hWnd, hMenu);
```

```
return true;
```

Специальные функции (1)

- получить дескриптор главного меню

```
HMENU GetMenu(HWND hWnd) // дескриптор главного окна
```

- получить дескриптор подменю

```
HMENU GetSubMenu(HMENU hMenu, // дескриптор меню  
int iPos) // позиция пункта-подменю
```

Специальные функции (2)

- установить доступность / недоступность пункта

```
BOOL EnableMenuItem(HMENU hMenu, // дескриптор меню
                    UINT ultemID, // ID | pos пункта меню
                    UINT uFlags) // флаги
```

- MF_BYCOMMAND (ID) или MF_BYPOSITION (позиция)
- MF_ENABLED | MF_DISABLED | MF_GRAYED

```
static HMENU hMenu;
// ...
case WM_INITDIALOG:
    EnableMenuItem(hMenu, IDM_SAVE, MF_BYCOMMAND |
                  MF_GRAYED);
return true;
```


Специальные функции (3)

- выставить / снять флажок у пункта меню

```
DWORD CheckMenuItem(HMENU hMenu, // дескриптор меню
                    UINT ultemID, // ID | pos пункта меню
                    UINT uFlags) // флаги
```

- MF_BYCOMMAND (ID) или MF_BYPOSITION (позиция)
- MF_CHECKED (установить) | MF_UNCHECKED (снять)

```
static HMENU hMenu;
// ...
case WM_INITDIALOG:
    CheckMenuItem(hMenu, IDM_OPEN, MF_BYCOMMAND |
                  MF_CHECKED);
return true;
```

Специальные функции (4)

- отметить переключателем пункт меню

```
DWORD CheckMenuRadioItem(HMENU hMenu, // дескриптор
UINT uIDFirst, // ID или позиция первого пункта в группе
UINT uIDLast, // ID или позиция последнего пункта в группе
UINT uIDCHECKED, // ID или позиция выбранного пункта
UINT uFlags) // флаги
```

Идентификаторы пунктов меню в группе должны иметь сплошную нумерацию!

Сообщения от меню

- при выборе разрешенного пункта-команды посылается сообщение **WM_COMMAND**:
 - LOWORD(wParam) содержит **идентификатор пункта**
 - HIWORD(wParam) и LPARAM содержат **0**
- при выборе любого пункта-команды (разрешенного, недоступного, запрещенного) также посылается сообщение **WM_MENUSELECT**:
 - LOWORD(wParam) содержит **идентификатор пункта**
- перед отображением меню посылается сообщение **WM_INITMENU** или **INITMENUPOPUP**

Обработка сообщений (пример)

```
static HMENU hMenu;
```

```
// ...
```

```
case WM_INITDIALOG:
```

```
    EnableMenuItem(hMenu, IDM_SAVE, MF_BYCOMMAND |  
                                                            MF_GRAYED);
```

```
return true;
```

```
case WM_COMMAND:
```

```
    if (LOWORD(wParam) == IDM_OPEN){  
        EnableMenuItem(hMenu, IDM_SAVE, MF_BYCOMMAND |  
                                                                MF_ENABLED);  
        CheckMenuItem(hMenu, IDM_OPEN, MF_BYCOMMAND |  
                                                                MF_CHECKED);
```

```
}
```

```
return true;
```

Функции для программного создания меню (1)

1) HMENU `CreateMenu()` – создать меню

2) HMENU `CreatePopupMenu()` – создать подменю

3) BOOL `AppendMenu`(HMENU hMenu, UINT uFlags, UINT_PTR uldNewItem, LPCTSTR lpNewItem) –
добавить пункт в конец меню

MF_POPUP MF_SEPARATOR MF_STRING

MF_CHECKED MF_UNCHECKED

MF_ENABLED MF_DISABLED MF_GRAYED

Функции для программного создания меню (2)

4) BOOL **InsertMenu**(HMENU hMenu,
 UINT uPos, // перед каким пунктом вставлять
 UINT uFlags, // MF_BYCOMMAND или
 MF_BYPOSITION и другие флаги
 UINT_PTR uldNewItem, // ID нового пункта
 LPCTSTR lpNewItem // содержимое нового
 пункта меню
)

Вставка пункта в меню перед пунктом UINT uPos

Создание меню (пример)

```
static HMENU hSubM1, hSubM2, hMenu;  
// ...  
case WM_INITDIALOG:  
    hSubM1 = CreatePopupMenu();  
    AppendMenu(hSubM1, MF_STRING, 11, L"Open");  
    AppendMenu(hSubM1, MF_STRING, 12, L"Save");  
    AppendMenu(hSubM1, MF_SEPARATOR, 0, 0);  
    AppendMenu(hSubM1, MF_STRING, 13, L"Exit");  
    hSubM2 = CreatePopupMenu();  
    AppendMenu(hSubM2, MF_STRING, 12, L>About...");  
    hMenu = CreateMenu();  
    AppendMenu(hMenu, MF_POPUP, (UINT_PTR)hSubM1, L"1");  
    AppendMenu(hMenu, MF_POPUP, (UINT_PTR)hSubM2, L"2");  
    SetMenu(hWnd, hMenu);  
return true;
```

Функции для программного создания меню (3)

5) BOOL **ModifyMenu**(HMENU hMenu, UINT uPos, UINT uFlags, PTR uIDNewItem, LPCTSTR lpNewItem)

– изменить пункт меню

6) BOOL **DeleteMenu**(HMENU hMenu, UINT uPos, UINT uFlags) – удалить пункт меню

7) BOOL **DrawMenuBar**(HMENU hMenu) – перерисовать меню

8) int **GetMenuString**(HMENU hMenu, UINT uIDItem, LPTSTR lpStrBuf, int iMaxStrLen, UINT uFlags) – получить строку из пункта меню

Создание акселераторов

- 1) изменить заголовки пунктов меню (добавить «\tCtrl+клавиша»)
- 2) добавить таблицу акселераторов в ресурсы ([Add/Resource/Accelerator/New](#))
- 3) заполнить таблицу акселераторов
 - **ID** (должен совпадать с ID нужного пункта меню)
 - **Modifier** (None | Ctrl | Alt и др.)
 - **Key** (код клавиши)
 - **Type** (VIRTKEY | ASCII (для алфавитно-цифровых и пунктуации))

Создание акселераторов

4) загрузить таблицу с помощью функции
`LoadAccelerators`

```
HACCEL LoadAccelerators(HINSTANCE hInstance,  
                        LPCTSTR lpAccelID) // ID таблицы акселераторов
```

5) изменить цикл обработки сообщений

```
while(GetMessage(&msg, 0, 0, 0)){  
    if (!TranslateAccelerator(hDialog, hAccel, &msg)){  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

Создание акселераторов (пример)

```
HWND hDialog = CreateDialog(hInstance,  
    MAKEINTRESOURCE(IDD_DIALOG1), NULL, DlgProc);  
ShowWindow(hDialog, 1);
```

```
HACCEL hAccel = LoadAccelerators(hInstance,  
    MAKEINTRESOURCE(IDR_ACCELERATOR1));
```

```
MSG msg;  
while(GetMessage(&msg, 0, 0, 0)){  
    if (!TranslateAccelerator(hDialog, hAccel, &msg)){  
        TranslateMessage(&msg);  
        DispatchMessage(&msg);  
    }  
}
```

Вопросы?