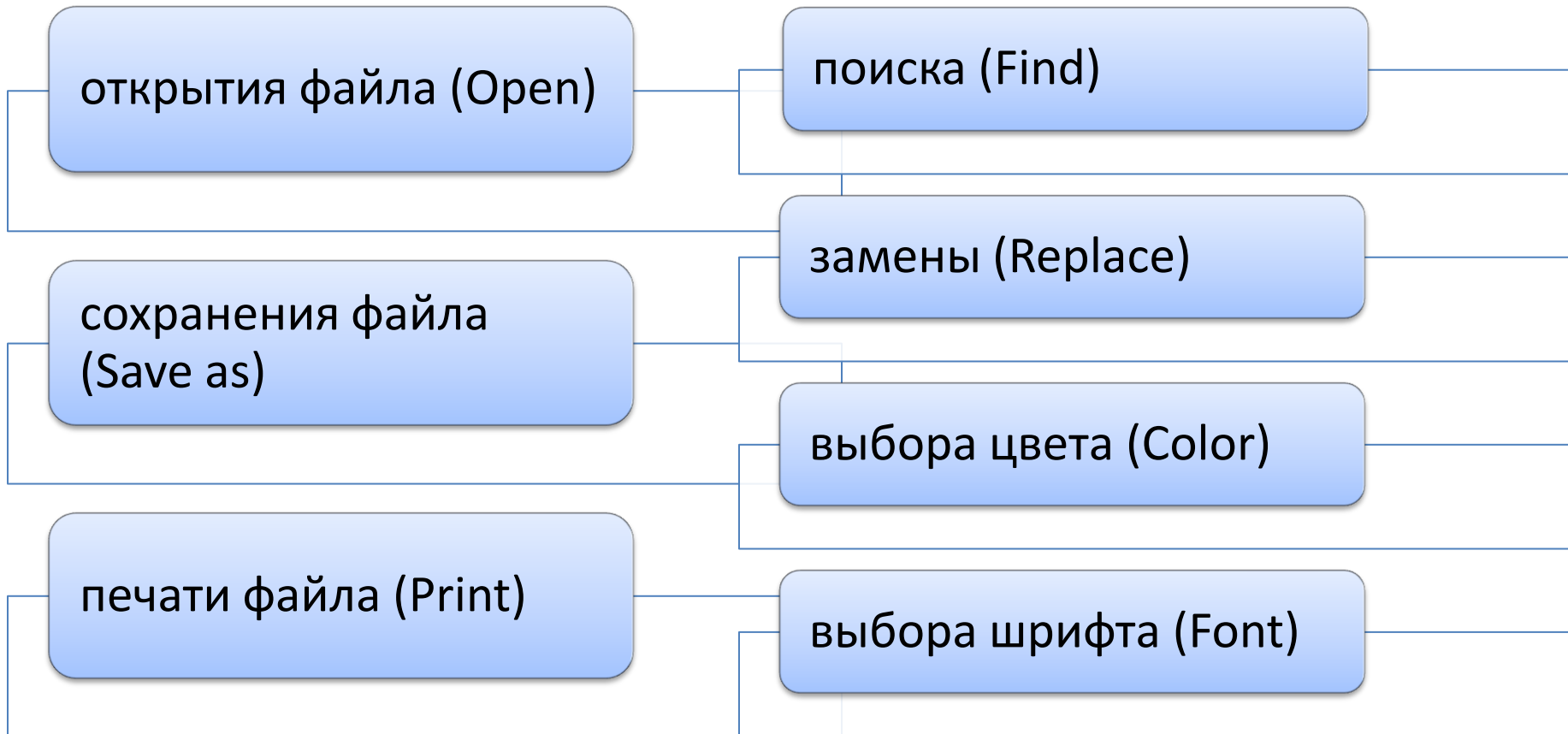


Стандартные диалоговые окна

Библиотека диалоговых окон

- содержит функции, которые вызывают стандартные окна диалога для:



Использование библиотеки

- подключить `commdlg.h`
- проинициализировать поля
предопределенной структуры
- вызвать специальную функцию с адресом
структуры

Общие диалоговые окна

Диалог	Структура	Функция
Открыть (Open)	OPENFILENAME	GetOpenFileName
Сохранить как (Save As)	OPENFILENAME	GetSaveFileName
Печать (Print)	PRINTDLG	PrintDlg
Найти (Find)	FINDREPLACE	FindText
Заменить (Replace)	FINDREPLACE	ReplaceText
Шрифт (Font)	CHOOSEFONT	ChooseFont
Цвет (Color)	CHOOSECOLOR	ChooseColor

Структура OPENFILENAME (1)

```
struct OPENFILENAME {  
    DWORD    lStructSize;           // размер структуры  
    HWND      hwndOwner;            // владелец окна диалога  
    HINSTANCE hInstance;           // дескриптор приложения  
    LPCWSTR   lpstrFilter;          // фильтр  
    LPWSTR    lpstrCustFilter;      // буфер для пользоват.  
    DWORD     nMaxCustFilter;       //    фильтра и размер буфера  
    DWORD     nFilterIndex;         // индекс шаблона в фильтре  
    LPWSTR    lpstrFile;           // буфер для пути к файлу  
    DWORD     nMaxFile;            // размер буфера для пути  
    LPWSTR    lpstrFileTitle;      // буфер для имени файла  
    /* ... */  
};
```

Структура OPENFILENAME (2)

```
struct OPENFILENAME {  
    /* ... */  
    DWORD      nMaxFileTitle;    // размер буфера для  
                                   имени файла  
    LPCWSTR    lpstrInitialDir;  // начало обхода директорий  
    LPCWSTR    lpstrTitle;       // заголовок окна диалога  
    DWORD      Flags;            // опции окна диалога  
    WORD       nFileOffset;      // смещение (много файлов)  
    WORD       nFileExtension;  // не используется  
    LPCWSTR    lpstrDefExt;      // расширение (по умолч.)  
    LPARAM     lCustData;  
    LPOFNHOOKPROC lpfnHook;      // фильтр для обработки  
                                   сообщений  
    LPCWSTR    lpTemplateName;  
};
```

Флаги

Флаги	Назначение
OFN_NOCHANGEDIR	Запретить изменять стартовый каталог для поиска файлов
OFN_ALLOWMULTISELECT	Разрешить выбирать несколько файлов
OFN_FILEMUSTEXIST	Проверять, существует ли файл с заданным именем
OFN_PATHMUSTEXIST	Проверять, существует ли заданный путь
OFN_CREATEPROMPT	Спрашивать у пользователя, нужно ли создать файл, которого еще не было
OFN_OVERWRITEPROMPT	Спрашивать, нужно ли перезаписать уже существующий файл
OFN_HIDEREADONLY	Не показывать флажок «Только для чтения»

Open file (пример)

```
#include <CommDlg.h>
```

```
static OPENFILENAME openFile;    // объявляем переменные  
wchar_t szFileName[256];
```

```
wchar_t szFileTitle[100];
```

```
/* ... */
```

```
ZeroMemory(&openFile, sizeof(OPENFILENAME)); // заполняем
```

```
openFile.lStructSize = sizeof(OPENFILENAME);    // структуру
```

```
openFile.hInstance = hInstance;
```

```
openFile.hwndOwner = hWnd;
```

```
openFile.lpstrFilter = L"All Files(*.*)\0*.*\0\0";
```

```
openFile.nMaxFile = 256;
```

```
openFile.nMaxFileTitle = 30;
```

```
openFile.lpstrFileTitle = szFileTitle;
```

```
openFile.lpstrFile = szFileName;
```

```
/* ... */
```

```
GetOpenFileName(&openFile);    // показываем диалог
```


Save file (пример)

```
#include <CommDlg.h>
```

```
static OPENFILENAME saveFile;    // объявляем переменные
```

```
wchar_t szFileName[256];
```

```
wchar_t szFileTitle[100];
```

```
/* ... */
```

```
ZeroMemory(&saveFile, sizeof(OPENFILENAME));
```

```
saveFile.lStructSize = sizeof(OPENFILENAME);
```

```
saveFile.hInstance = hInstance;
```

```
saveFile.hwndOwner = hWnd;
```

```
saveFile.lpstrFilter = L"All Files(*.*)\0*.*\0\0";
```

```
saveFile.nMaxFile = 256;
```

```
saveFile.nMaxFileTitle = 30;
```

```
saveFile.lpstrFileTitle = szFileTitle;
```

```
saveFile.lpstrFile = szFileName;
```

```
saveFile.lpstrTitle = L"Сохранить файл?";
```

```
/* ... */
```

```
GetSaveFileName(&saveFile);    // показываем диалог
```

Структура CHOOSECOLOR

```
struct CHOOSECOLOR
{
    DWORD      IStructSize;           // размер структуры
    HWND       hwndOwner;             // владелец
    HWND       hInstance;
    COLORREF    rgbResult;             // выбранный цвет
    COLORREF*   lpCustColors;          // массив для хранения
                                     // дополнительных цветов
    DWORD      Flags;                 // флаги
    LPARAM      lCustData;
    LPCCHOOKPROC lpfnHook;
    LPCWSTR     lpTemplateName;
};
```

Выбор цвета (пример)

```
#include <CommDlg.h>

CHOOSECOLOR chooseColor;
static COLORREF colorRef[16];
/* ... */
ZeroMemory(&chooseColor, sizeof(CHOOSECOLOR));
chooseColor.lStructSize = sizeof(CHOOSECOLOR);
chooseColor.hInstance = NULL;
chooseColor.hwndOwner = hwnd;
chooseColor.lpCustColors = colorRef;
chooseColor.rgbResult = RGB(100, 100, 255);
chooseColor.Flags = CC_RGBINIT | CC_FULLOPEN;
/* ... */
ChooseColor(&chooseColor);           // показываем диалог
GetClientRect(hwnd, &rect);
SetClassLong(hwnd, -10,
    (LONG)CreateSolidBrush(chooseColor.rgbResult));
InvalidateRect(hwnd, &rect, true);
```

Структура CHOOSEFONT

```
struct CHOOSEFONT
{
    DWORD      IStructSize;
    HWND       hwndOwner;
    HDC         hDC;
    LPLOGFONTW  lpLogFont;
    INT         iPointSize;
    DWORD      Flags;
    COLORREF    rgbColors;
    LPARAM      lCustData;
    LPCFHOOKPROC lpfnHook;
    LPCWSTR     lpTemplateName;
    HINSTANCE    hInstance;
    LPWSTR      lpszStyle;
    WORD        nFontType;
    WORD        __MISSING_ALIGNMENT__;
    INT         nSizeMin;
    INT         nSizeMax;
};
```

Выбор шрифта (пример)

```
#include <CommDlg.h>
```

```
LOGFONT logFont;
```

```
CHOOSEFONT chFont;
```

```
/* ... */
```

```
ZeroMemory(&chFont, sizeof(CHOOSEFONT));
```

```
chFont.lStructSize = sizeof(CHOOSEFONT);
```

```
chFont.hwndOwner = hWnd;
```

```
chFont.lpLogFont = &logFont;
```

```
chFont.Flags = CF_INITTOLOGFONTSTRUCT | CF_SCREENFONTS |  
CF_EFFECTS;
```

```
/* ... */
```

```
ChooseFont(&chFont);
```

```
hdc = GetDC(hWnd);
```

```
GetClientRect(hWnd, &rect);
```

```
hFont = CreateFontIndirect(chFont.lpLogFont);
```

```
SendMessage(hWnd, WM_SETFONT, (WPARAM)hFont, 0);
```

```
InvalidateRect(hWnd, &rect, TRUE);
```

```
ReleaseDC(hWnd, hdc);
```

Вопросы?