

Распаковщики сообщений

Распаковщики сообщений

- 1) специальные **макросы**, которые позволяют упростить большой **switch** в оконной процедуре
- 2) «распаковывают» параметры сообщений, содержащиеся в **WPARAM** и **LPARAM**
- 3) содержатся в заголовочном файле **WindowsX.h**
- 4) предназначены для того, чтобы сделать код более легким для чтения и уменьшить число явных преобразований типов

Макрос HANDLE_MSG

```
#define HANDLE_MSG(hwnd, message, fn) \
    case (message):  
    return HANDLE_##message((hwnd), (wParam), (lParam), (fn))
```

- 1) содержится в заголовочном файле **WindowsX.h**
- 2) позволяет распаковывать разные сообщения

Пример 1 (а)

сообщение WM_CLOSE

```
#define HANDLE_MSG(hwnd, message, fn) \
    case (message):  
    return HANDLE_##message((hwnd), (wParam), (lParam), (fn))
```



```
case (WM_CLOSE):  
    return HANDLE_WM_CLOSE((hwnd), (wParam),  
        (lParam), (Cls_OnClose));
```

Пример 1 (б)

```
case (WM_CLOSE):  
    return HANDLE_WM_CLOSE((hwnd), (wParam),  
        (LPARAM), (Cls_OnClose));
```



```
/* void Cls_OnClose(HWND hwnd) */  
  
#define HANDLE_WM_CLOSE(hwnd, wParam, lParam, fn) \  
    ((fn)(hwnd), 0L)  
  
#define FORWARD_WM_CLOSE(hwnd, fn) \  
    (void)(fn)((hwnd), WM_CLOSE, 0L, 0L)
```

Пример 2 (а)

сообщение WM_COMMAND

```
#define HANDLE_MSG(hwnd, message, fn) \
    case (message):  
    return HANDLE_##message((hwnd), (wParam), (lParam), (fn))
```



```
case (WM_COMMAND):  
    return HANDLE_WM_COMMAND((hwnd), (wParam),  
        (lParam), (Cls_OnCommand));
```

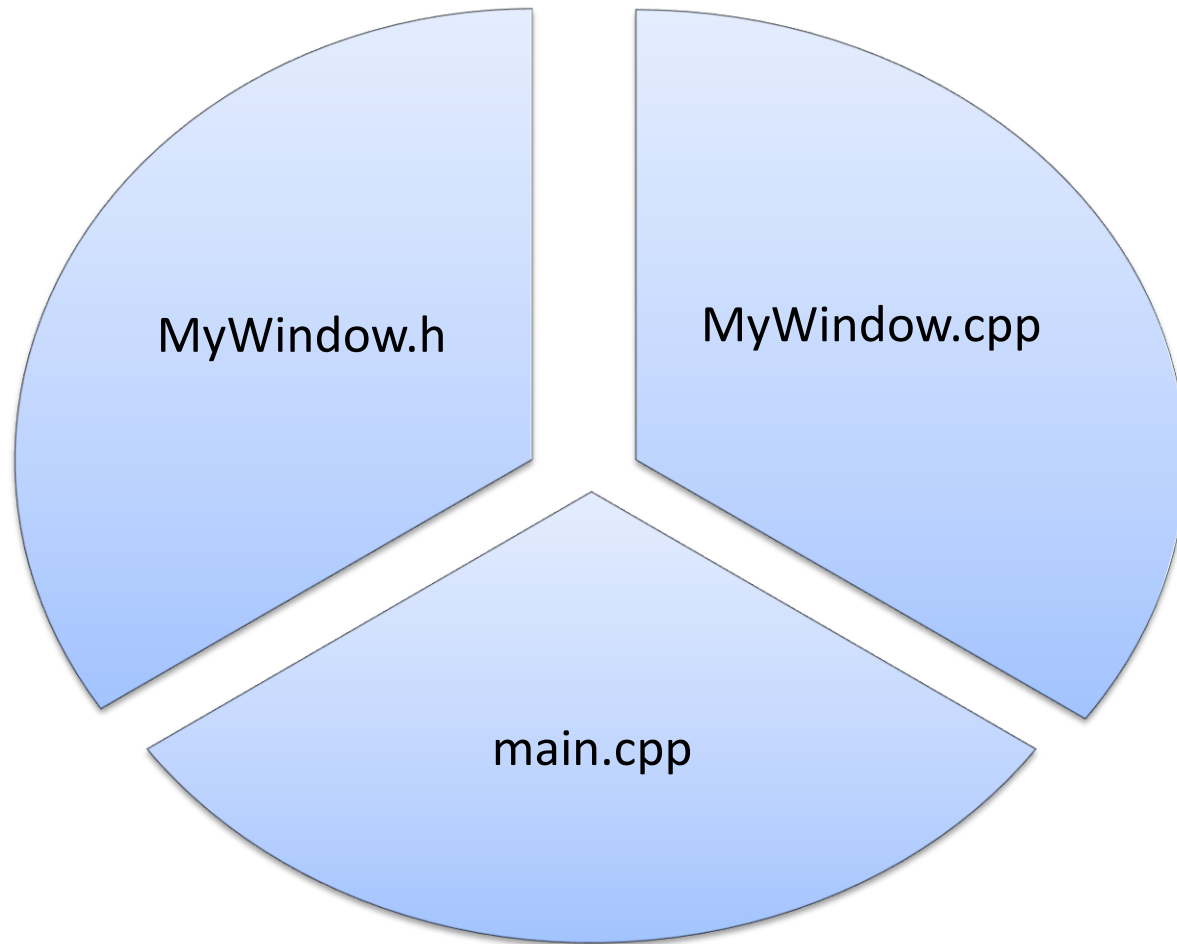
Пример 2 (б)

```
/* void Cls_OnCommand(HWND hwnd, int id, HWND hwndCtl,  
UINT codeNotify) */
```

```
#define HANDLE_WM_COMMAND(hwnd, wParam, lParam, fn) \  
((fn)((hwnd), (int)(LOWORD(wParam)), (HWND)(lParam),  
                (UINT)HIWORD(wParam)), 0L)
```

```
#define FORWARD_WM_COMMAND(hwnd, id, hwndCtl,  
                                codeNotify, fn) \  
(void)(fn)((hwnd), WM_COMMAND,  
            MAKEWPARAM((UINT)(id), (UINT)(codeNotify)),  
            (LPARAM)(HWND)(hwndCtl))
```

Структура кода



MyWindow.h

- статическое поле-указатель на объект класса
- поля класса (например, дескрипторы окна и его элементов управления)
- объявление статической функции `WndProc / DlgProc`
- конструктор
- методы для обработки отдельных сообщений (имена начинаются с `Cls_On...`)

MyWindow.h

```
class MyWindow {  
    static MyWindow* ptr;  
    HWND hDialog;  
    HWND hButOK, hButCancel;  
  
    void Cls_OnClose(HWND);  
    BOOL Cls_OnInitDialog(HWND, HWND, LPARAM);  
    void Cls_OnCommand(HWND hwnd, int id, HWND hwndCtl,  
                        UINT codeNotify);  
  
public:  
    MyWindow();  
    static BOOL CALLBACKDlgProc(HWND, UINT, WPARAM,  
                                LPARAM);  
};
```

MyWindow.cpp

- инициализация статического поля-указателя на объект класса значением **NULL**
- конструктор (инициализация поля-указателя **this**'ом)
- реализация статической функции WndProc / DlgProc (**switch** с набором макросов **HANDLE_MSG**)
- реализация методов **Cls_On...** для обработки отдельных сообщений

MyWindow.cpp (1)

```
#include "MyWindow.h"
```

```
MyWindow* MyWindow::ptr = NULL;
```

```
MyWindow::MyWindow() {  
    ptr = this;  
}
```

MyWindow.cpp (2)

```
BOOL CALLBACK MyWindow::DlgProc(HWND hWnd,  
    UINT uMsg, WPARAM wParam, LPARAM lParam) {  
  
    switch (uMsg){  
        HANDLE_MSG(hWnd, WM_CLOSE, ptr->Cls_OnClose);  
        HANDLE_MSG(hWnd, WM_INITDIALOG, ptr->  
            Cls_OnInitDialog);  
        HANDLE_MSG(hWnd, WM_COMMAND, ptr->  
            Cls_OnCommand);  
    }  
  
    return false;  
}
```

MyWindow.cpp (3)

```
BOOL MyWindow::Cls_OnInitDialog(HWND hWnd,  
                                HWND hFocus, LPARAM lParam){  
    hDialog = hWnd;  
    hButOK = GetDlgItem(hDialog, IDOK);  
    hButCancel = GetDlgItem(hDialog, IDCANCEL);  
    return true;  
}  
  
void MyWindow::Cls_OnClose(HWND hWnd){  
    EndDialog(hWnd, 0);  
}
```

MyWindow.cpp (4)

```
void MyWindow::Cls_OnCommand(HWND hwnd, int id,
                               HWND hwndCtl, UINT codeNotify) {
    if (id == IDOK){
        MessageBox(hDialog, L"OK!", L"Info", MB_OK);
    }
    else if (id == IDCANCEL){
        MessageBox(hDialog, L"CANCEL!", L"Info", MB_OK);
        SendMessage(hDialog, WM_CLOSE, 0, 0);
    }
}
```

main.cpp

- Функция WinMain
 - создание объекта класса MyWindow
 - вызов функции для создания окна

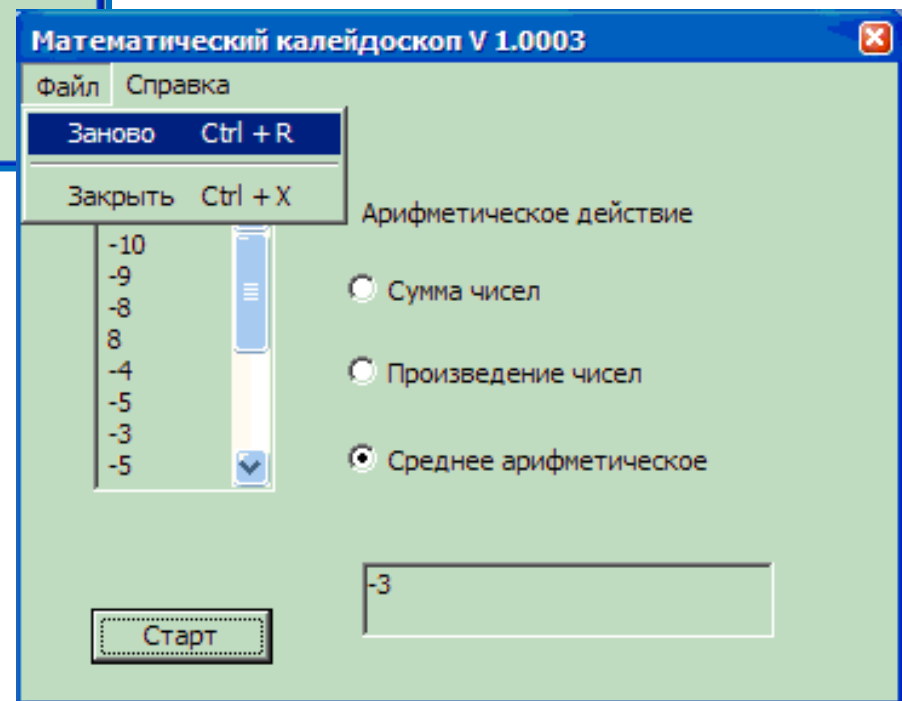
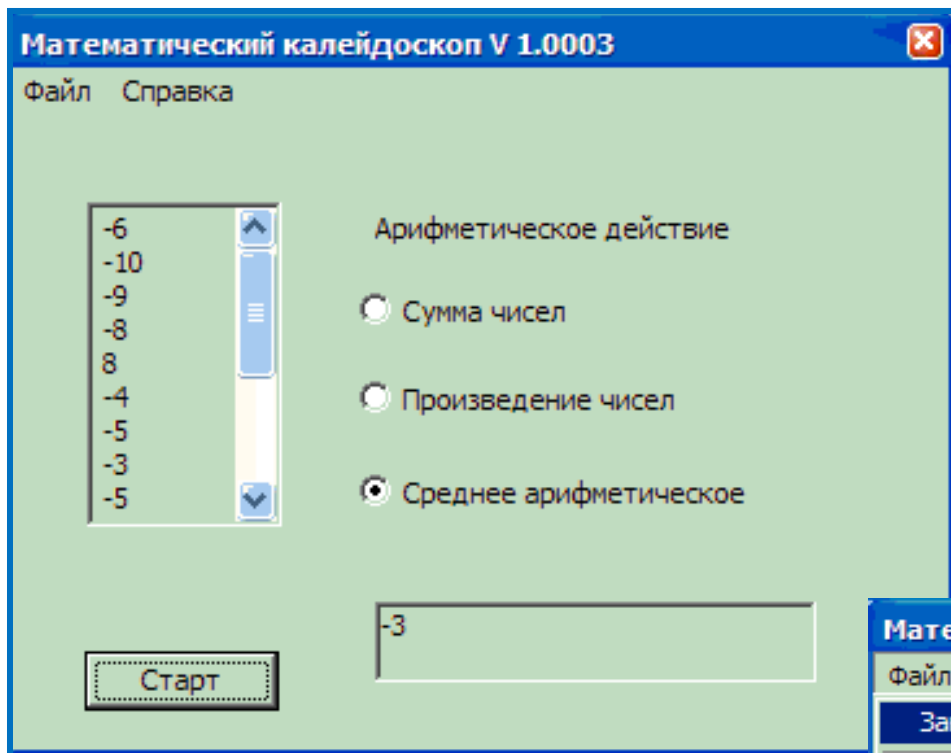
```
int WINAPI WinMain(HINSTANCE hInst, HINSTANCE, LPSTR, int){  
    MyWindow wnd;  
    return DialogBox(hInst, MAKEINTRESOURCE(IDD_DIALOG1),  
                     NULL, MyWindow::DlgProc);  
}
```


Задание

Дан исходный код программы «Математический калейдоскоп» (файл LastTask.cpp) со следующей функциональностью:

На форме находится список, кнопка, текстовое поле и три переключателя. По кнопке «Старт» в список заносится от 10 до 20 случайных чисел в диапазоне от -10 до 10 (0 не включается). Переключатели позволяют увидеть сумму чисел, их произведение или среднее арифметическое.

Создайте утерянный интерфейс для программы. Исправьте в ошибки, допущенные в коде, и перепишите его с использованием распаковщиков сообщений.



Вопросы?