

## Обмен сообщениями

### Цели:

- Получить навыки создания сообщений пользователя
- Закрепить навыки обработки сообщений
- Получить навыки отправки сообщений

# Обработчик сообщений

Создадим новый проект, на главной форме разместим компонент `TMemo (Memo1)` и `TButton (Button1)`.

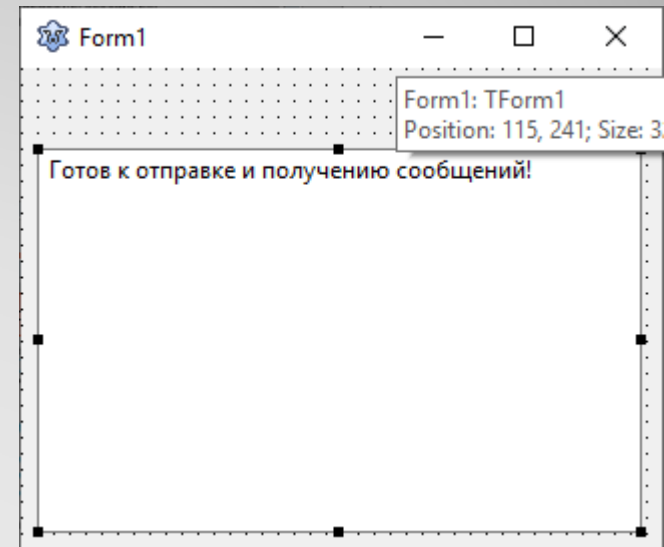
Добавим библиотеку `Windows` в раздел `uses`.

В раздел `var` добавим две переменные:

```
PrevWndProc : WNDPROC;
```

```
MY_MSG      : DWORD;
```

Первая будет хранить диспетчер обработки сообщений по умолчанию, а вторая номер созданного нами сообщения.



Создадим диспетчер обработки сообщений:

```
function WndCallback(ahwnd: HWND; uMsg: UINT; wParam:
    WPARAM; lParam: LPARAM) : LRESULT; stdcall;
begin
    if uMsg=MY_MSG then
    begin
        Form1.Memo1.Lines.Add('Got message');
    end
    else
        result:=CallWindowProc(PrevWndProc,ahwnd,uMsg,wParam
        ,lParam);
    end;
```

Данный диспетчер проверяет, является ли сообщение “нашим” и если нет, вызывает диспетчер по умолчанию.

## Создание сообщения

Создадим новое сообщение и сменим диспетчер обработки события. Данные действия необходимо произвести при инициализации приложения, поэтому разместим их в событии `OnCreate`.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    PrevWndProc:=Windows.WNDPROC(

        SetWindowLongPtr(Self.Handle,GWL_WNDPROC,
                        PtrInt(@WndCallback)));
    MY_MSG:=RegisterWindowMessage('My own message');
end;
```

## Отправка сообщения

Создадим обработчик события нажатия на кнопку, в котором будет осуществляться отправка сообщения.

`HWND_BROADCAST` показывает, что сообщение будет отправлено всем приложениям и процессам.

```
SendMessage(HWND_BROADCAST, MY_MSG, 0, 0);
```

## Дополнительное задание

Добавить на форму компонент **TEdit** и реализовать отправку введенного в него числа.

