



Пользовательский графический интерфейс в MATLAB

Лекция #7

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММИКН ЮФУ

Содержание

 Два метода создания GUI

 Пример 1. Привет, МИР!

 Пример 2. Логотип MatLab в GUI

 Пример 3. Сумматор

 Программирование GUI

Uicontrol Elements

Базовые элементы графического интерфейса

- Check boxes
- Editable text fields
- Frames
- List boxes
- Pop-up menus
- Push buttons
- Radio buttons
- Sliders
- Static text labels

Два способа создания GUI

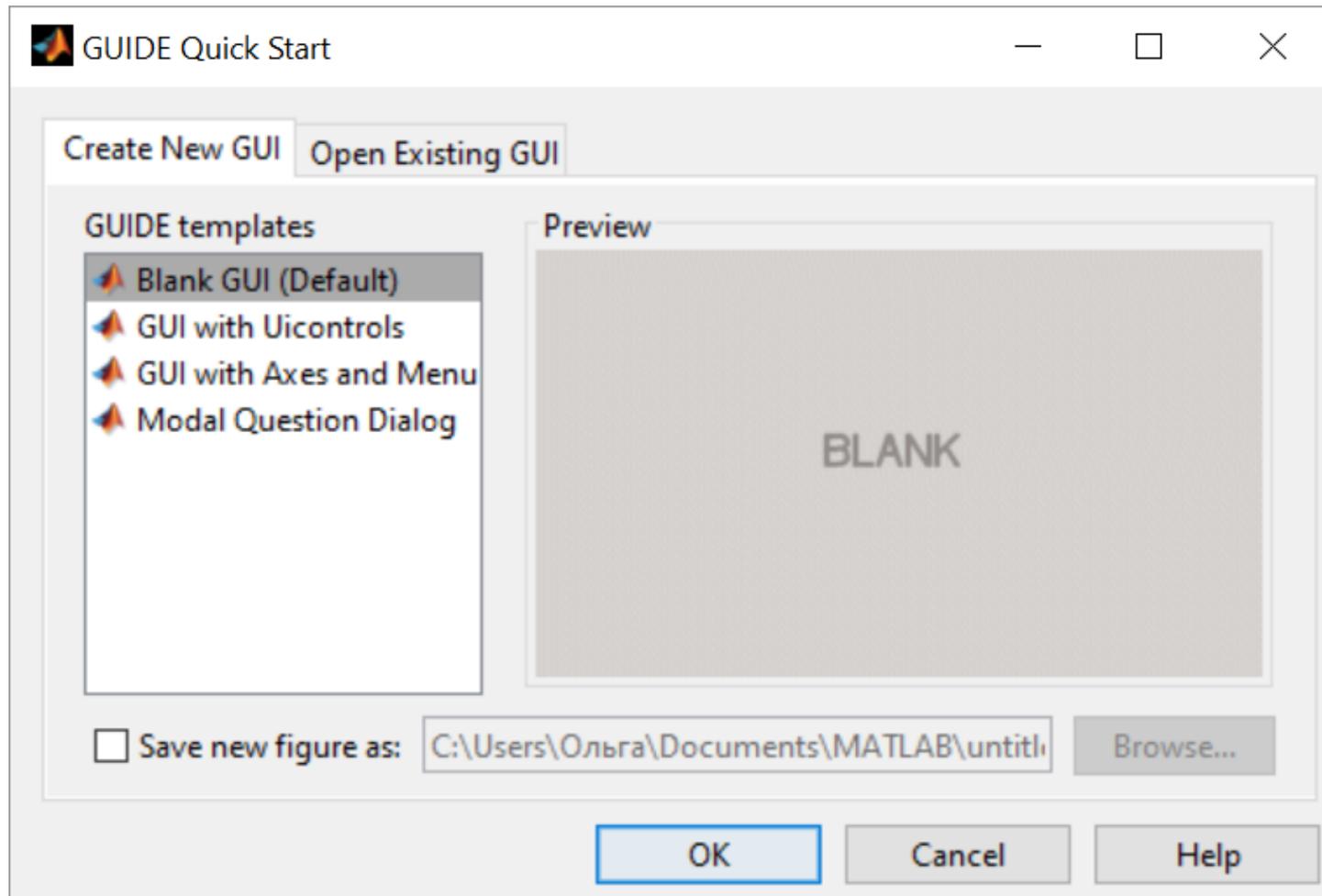
Первый способ заключается в программировании в m-файле отдельно каждого элемента. Т.е. описание окон, осей, кнопок и т.д. То есть задания свойств и функций обратного вызова (Callback), которые вызываются при обращении к элементам.

Второй способ - с помощью интерактивного инструмента GUIDE.
Для вызова GUIDE нужно в командной строке выполнить команду:

```
>> guide
```

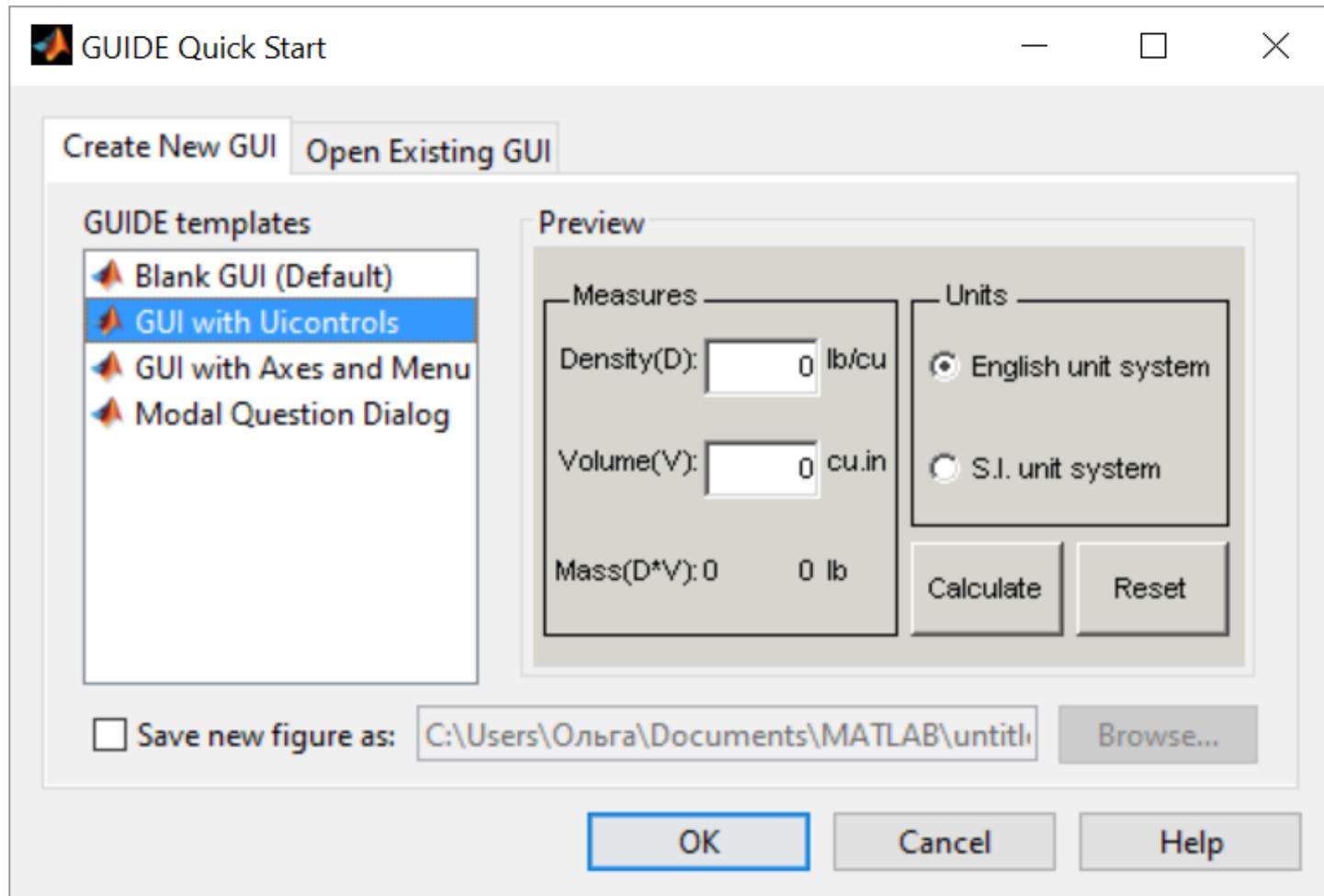
GUIDE

```
>> guide
```



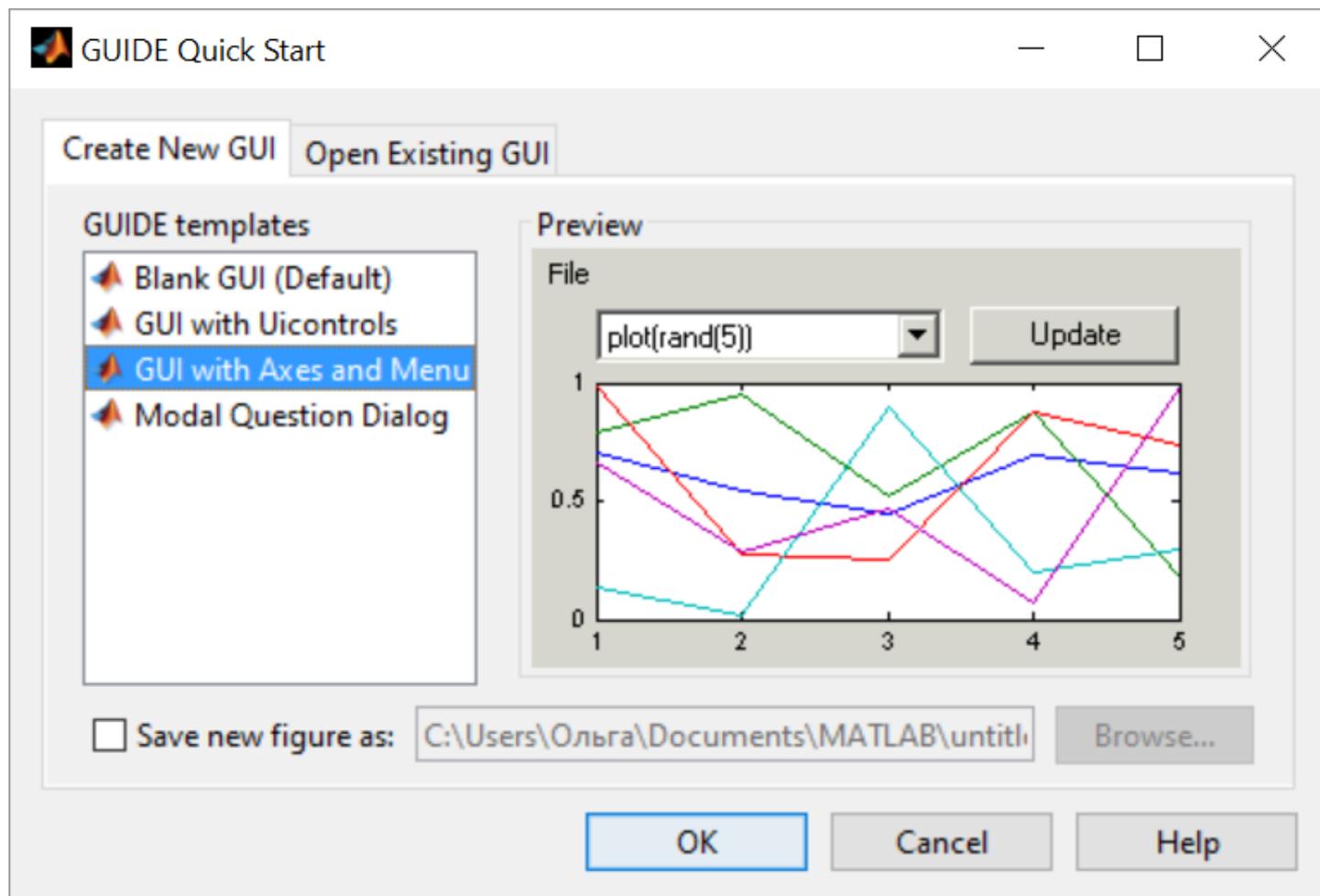
GUIDE

GUI with Uicontrols



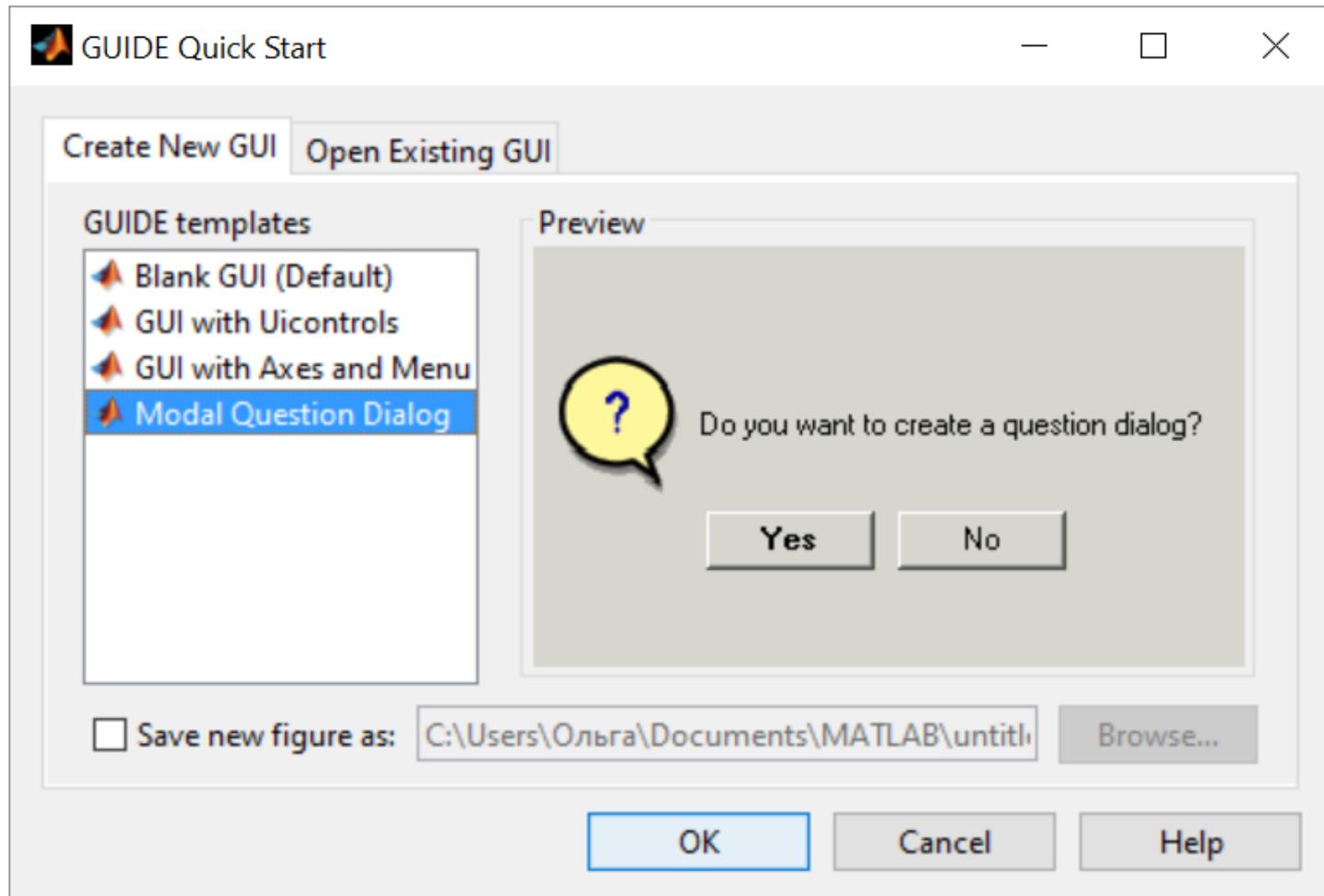
GUIDE

GUI with Axes and Menu



GUIDE

Modal Question Dialog



ПРИМЕР 1. Привет, МИР!

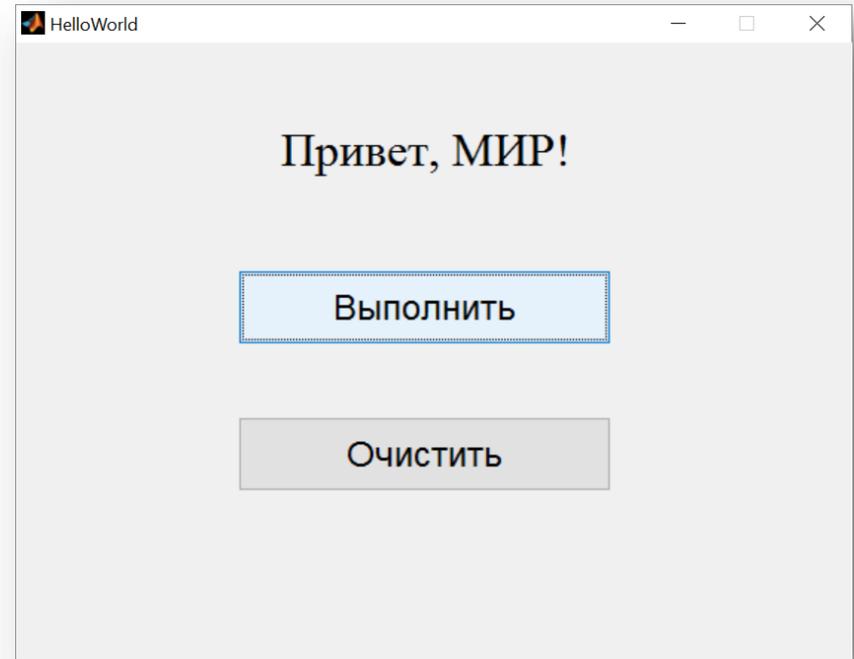
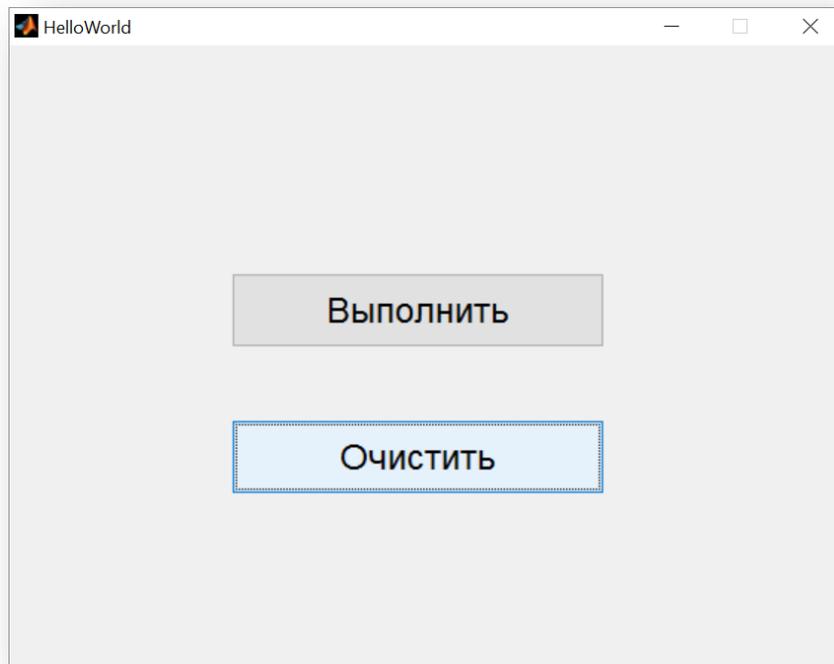
GUIDE. Пример. Привет, МИР! Цель

Создать GUI с двумя кнопками:

Выполнить и Очистить.

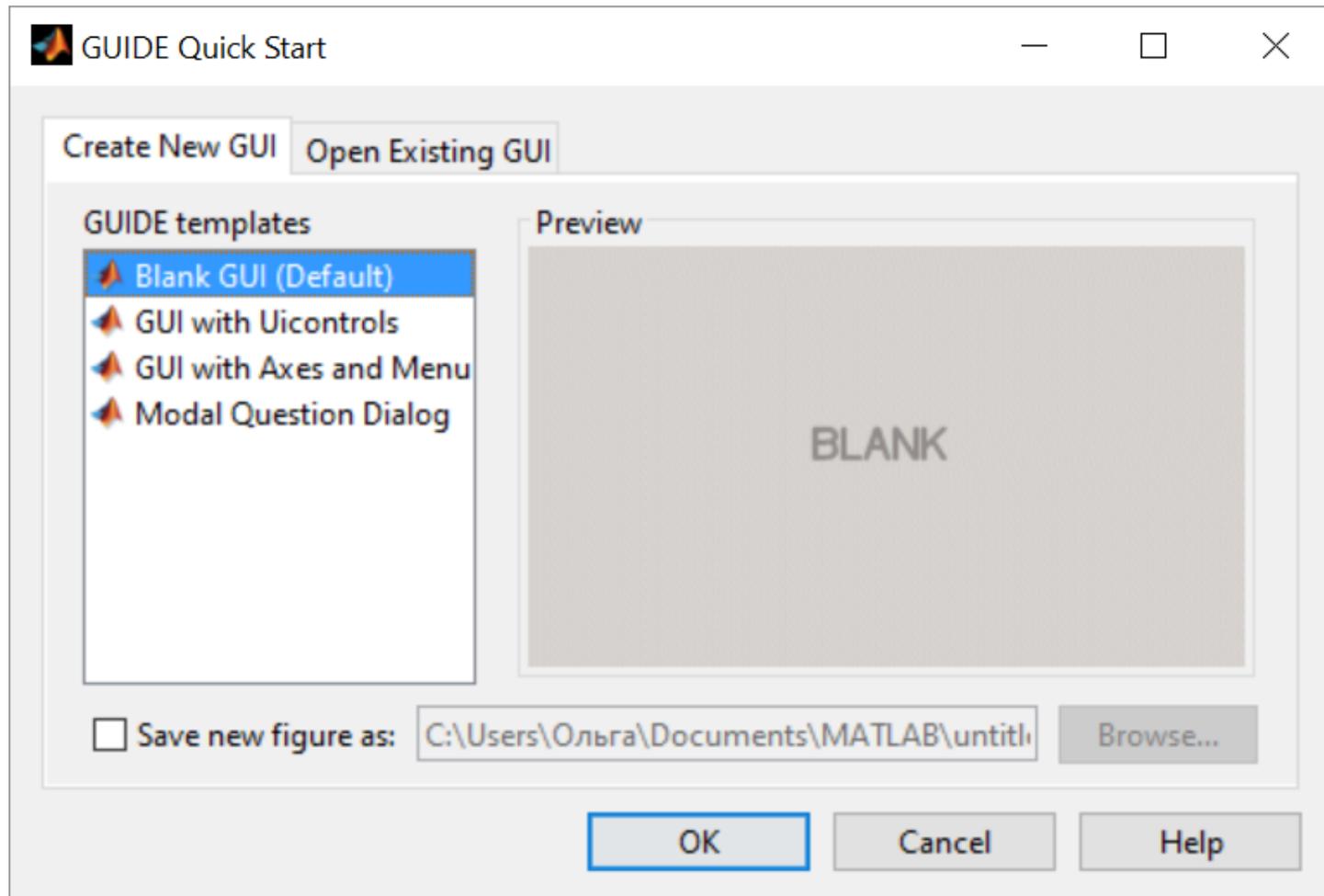
При нажатии на Выполнить должно появляться сообщение «Привет, МИР!»

При нажатии на кнопку Очистить экран должен очищаться.



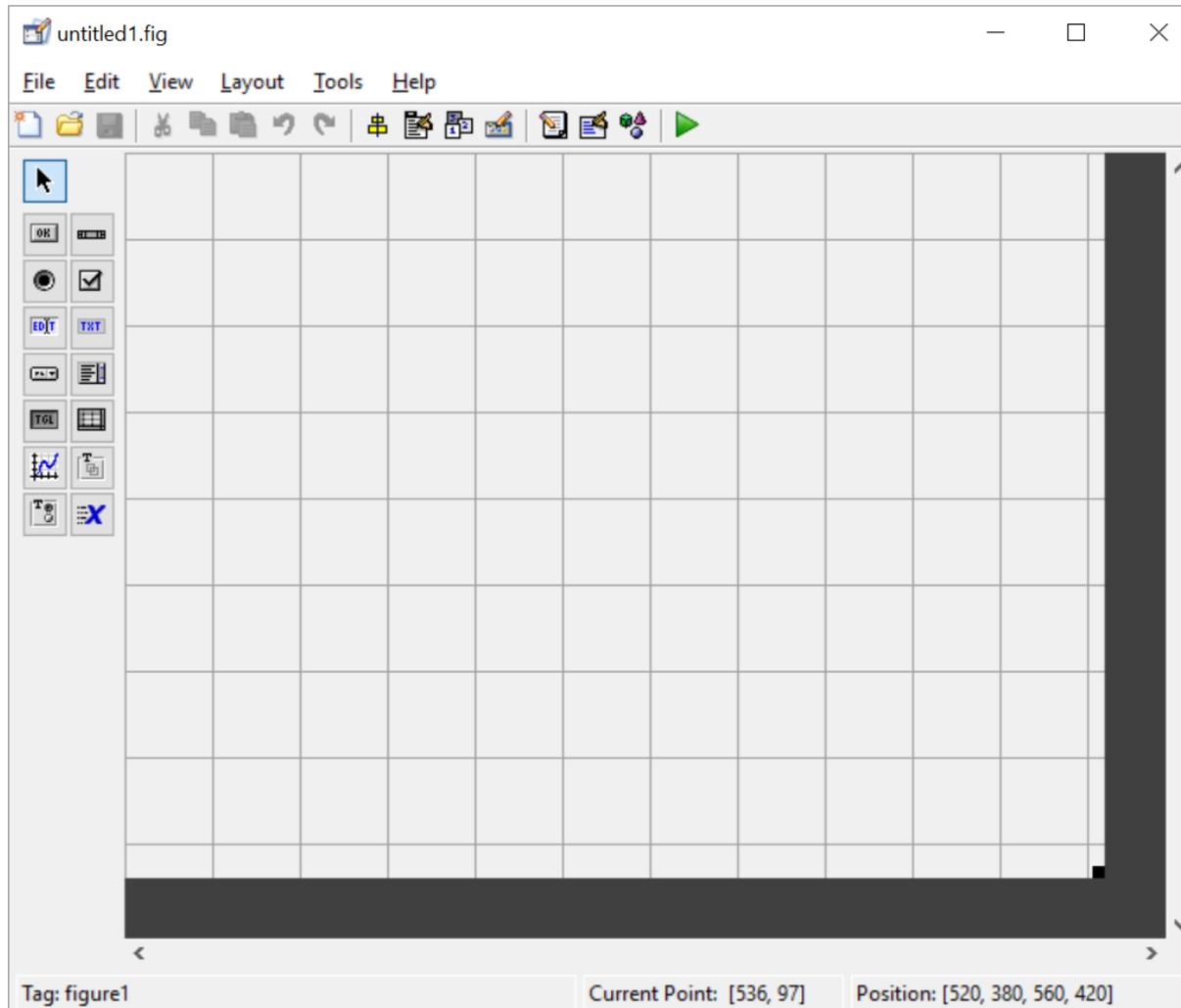
GUIDE. Пример. Привет, МИР! Шаг 1

Blank GUI (Default)



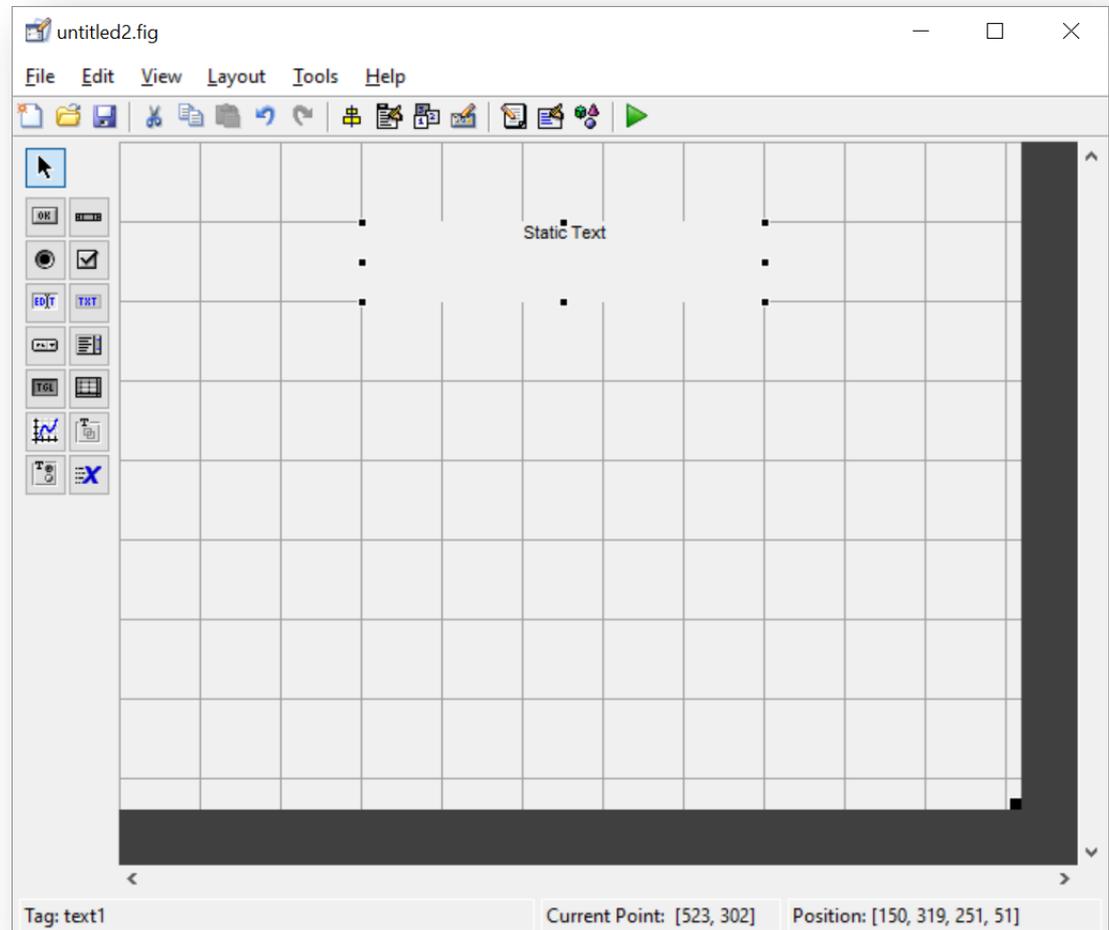
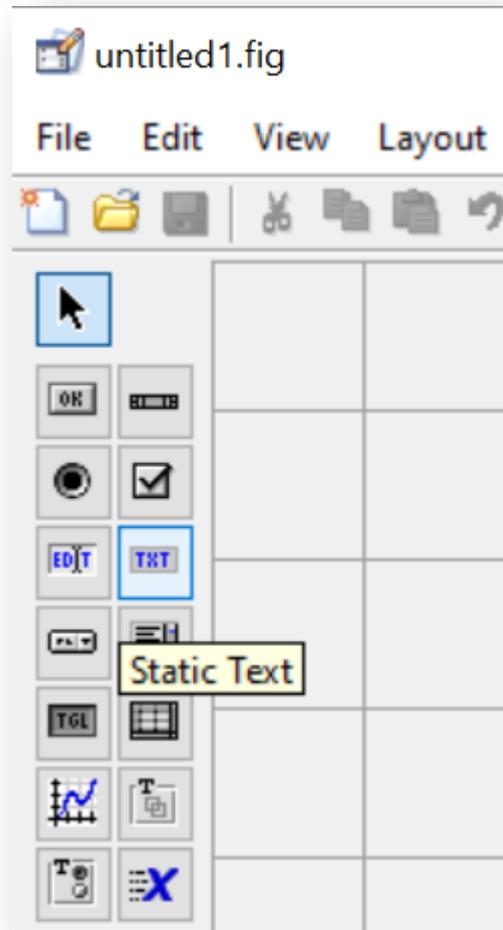
GUIDE. Пример. Привет, МИР! Шаг 2

Blank GUI (Default)



GUIDE. Пример. Привет, МИР! Шаг 3

Static Text



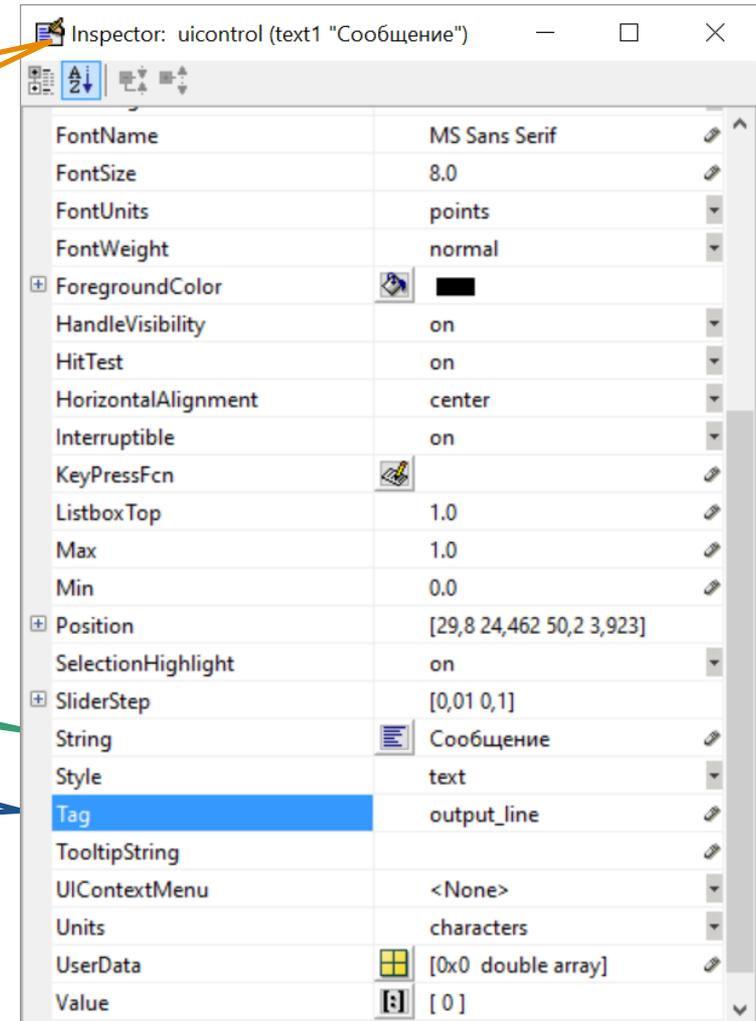
GUIDE. Пример. Привет, МИР! Шаг 4

Static Text / Property Inspector / Tag = output_line

Property Inspector МОЖНО открыть двойным щелчком мыши на объекте или правой кнопкой мыши.

Выбрать свойство **String** и назначить значение для ЭТОГО СВОЙСТВА – Сообщение.

Выбрать свойство **Tag** и назначить значение для ЭТОГО СВОЙСТВА – **output_line**.



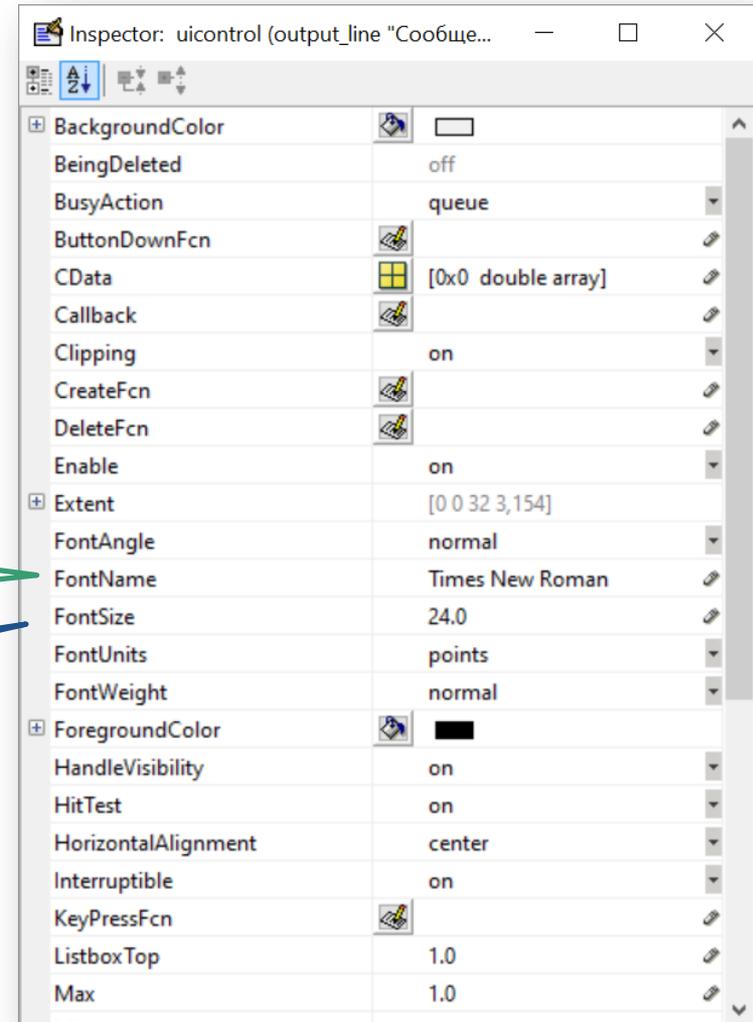
GUIDE. Пример. Привет, МИР! Шаг 4

Static Text Property Inspector / **FontSize = 24**

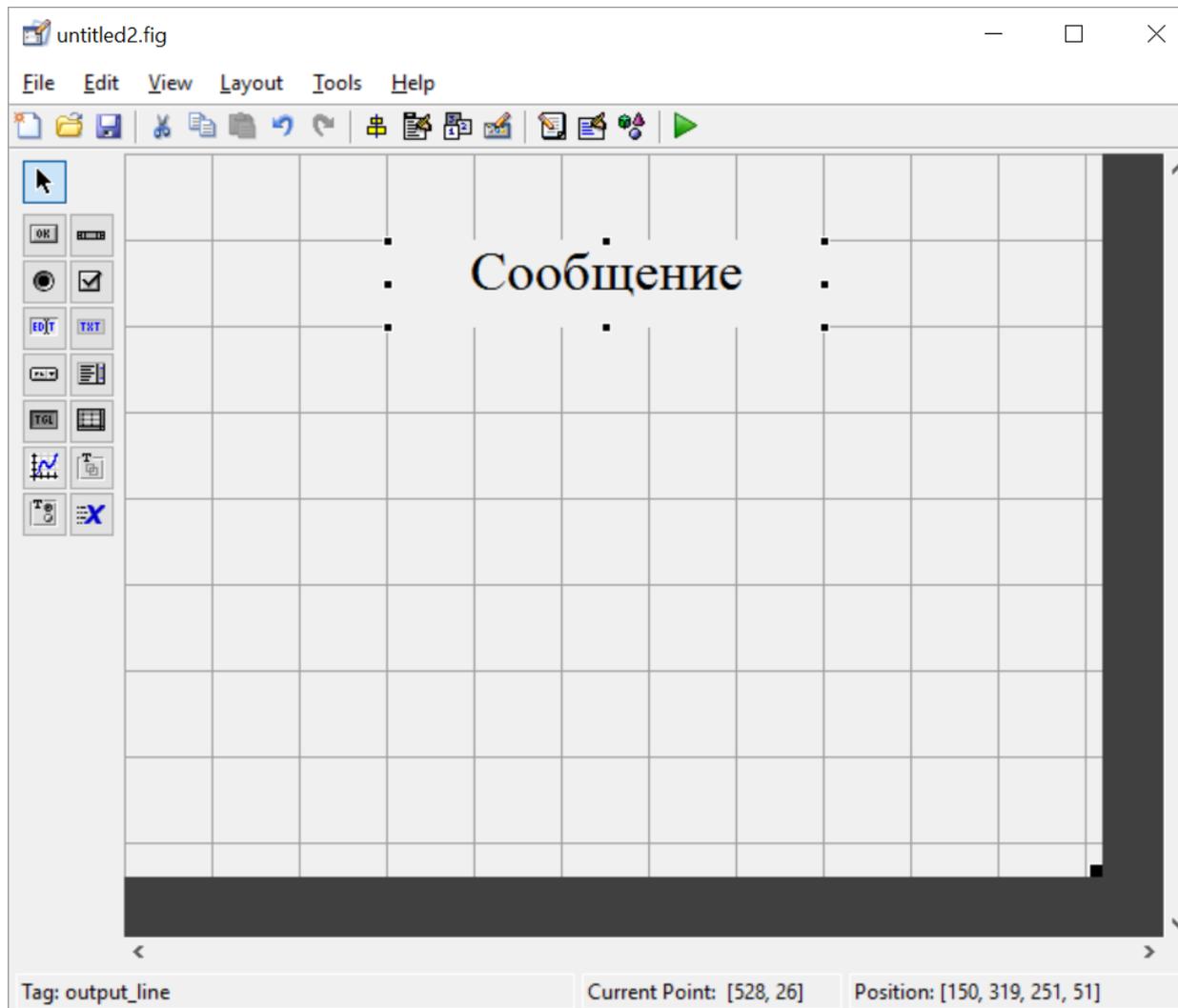
Property Inspector МОЖНО
открыть двойным щелчком
мыши на объекте или
правой кнопкой.

Можно изменить свойство
FontName и назначить
значение для этого свойства
– **Times New Roman**.

Выбрать свойство **FontSize**
и назначить значение для
этого свойства – **24**.

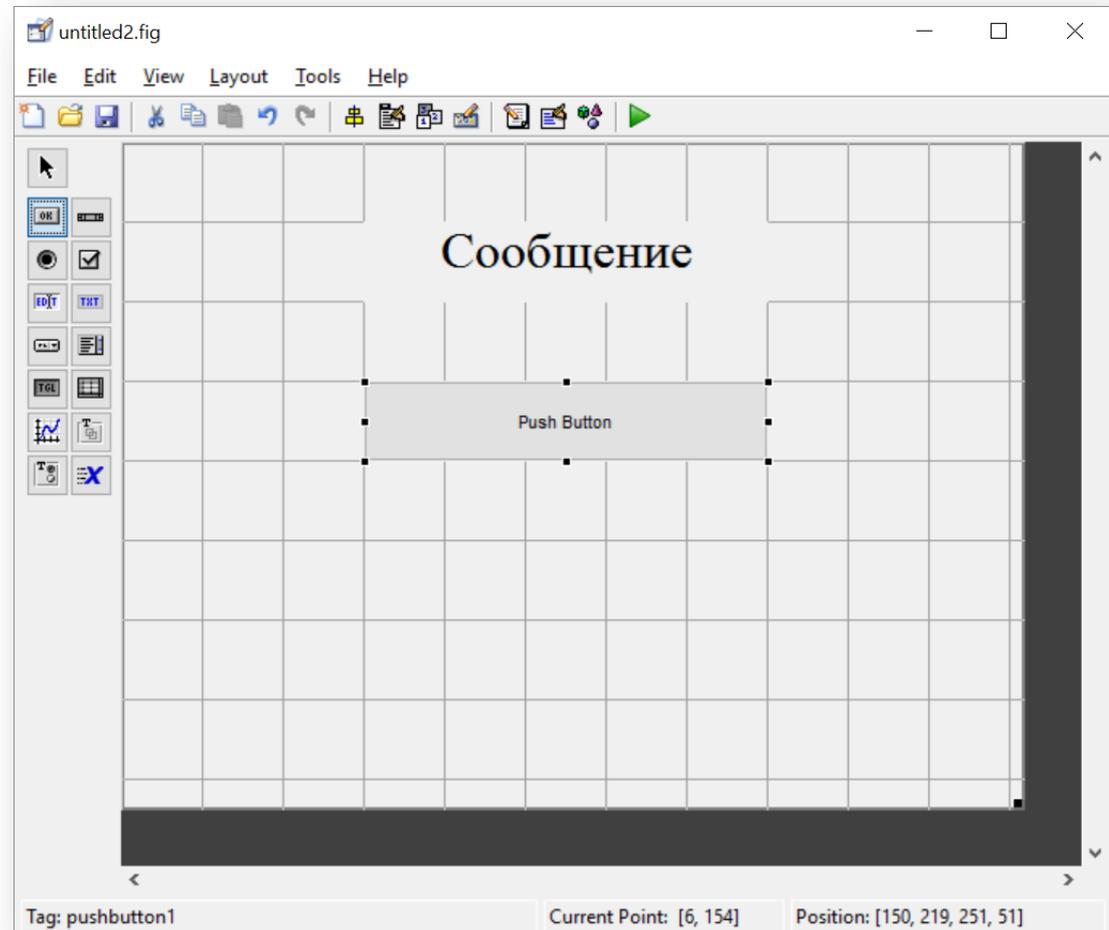
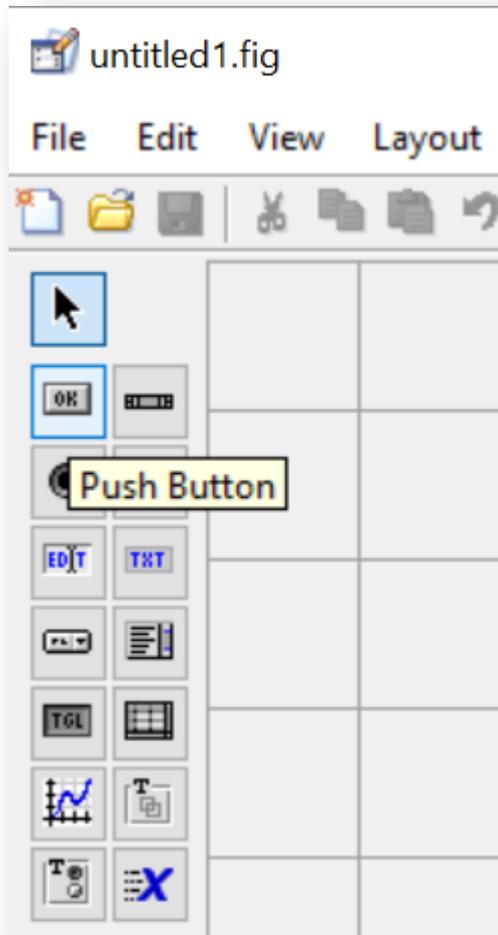


GUIDE. Пример. Привет, МИР! Шаг 4. Результат



GUIDE. Пример. Привет, МИР! Шаг 5

Push Button

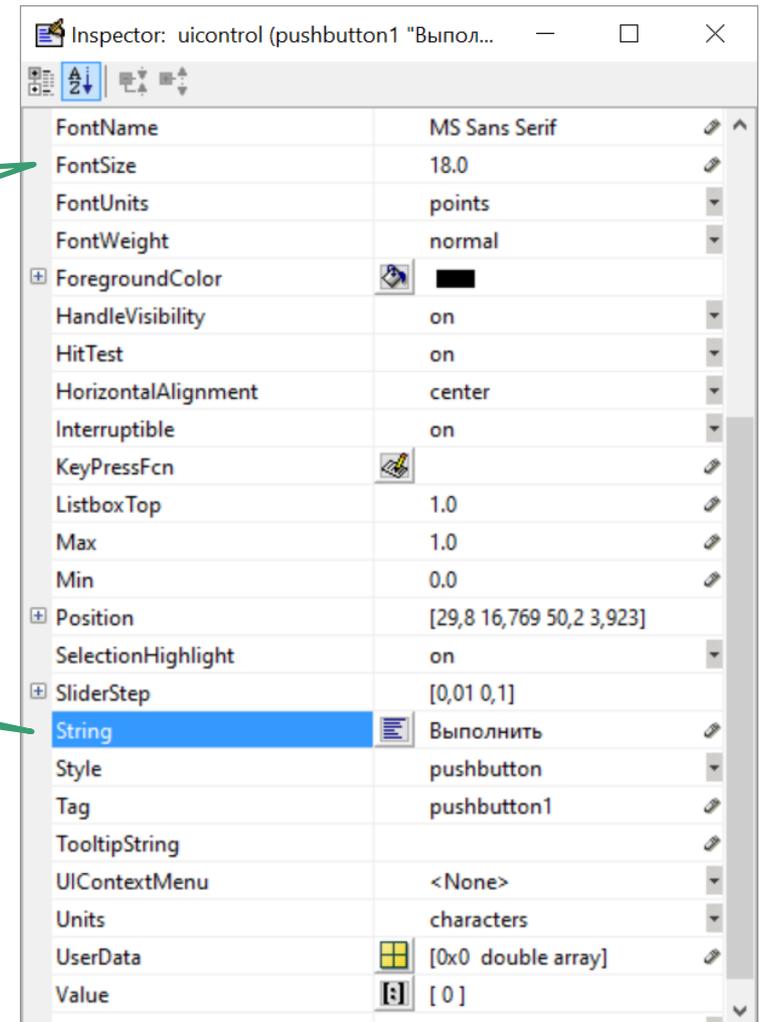


GUIDE. Пример. Привет, МИР! Шаг 5

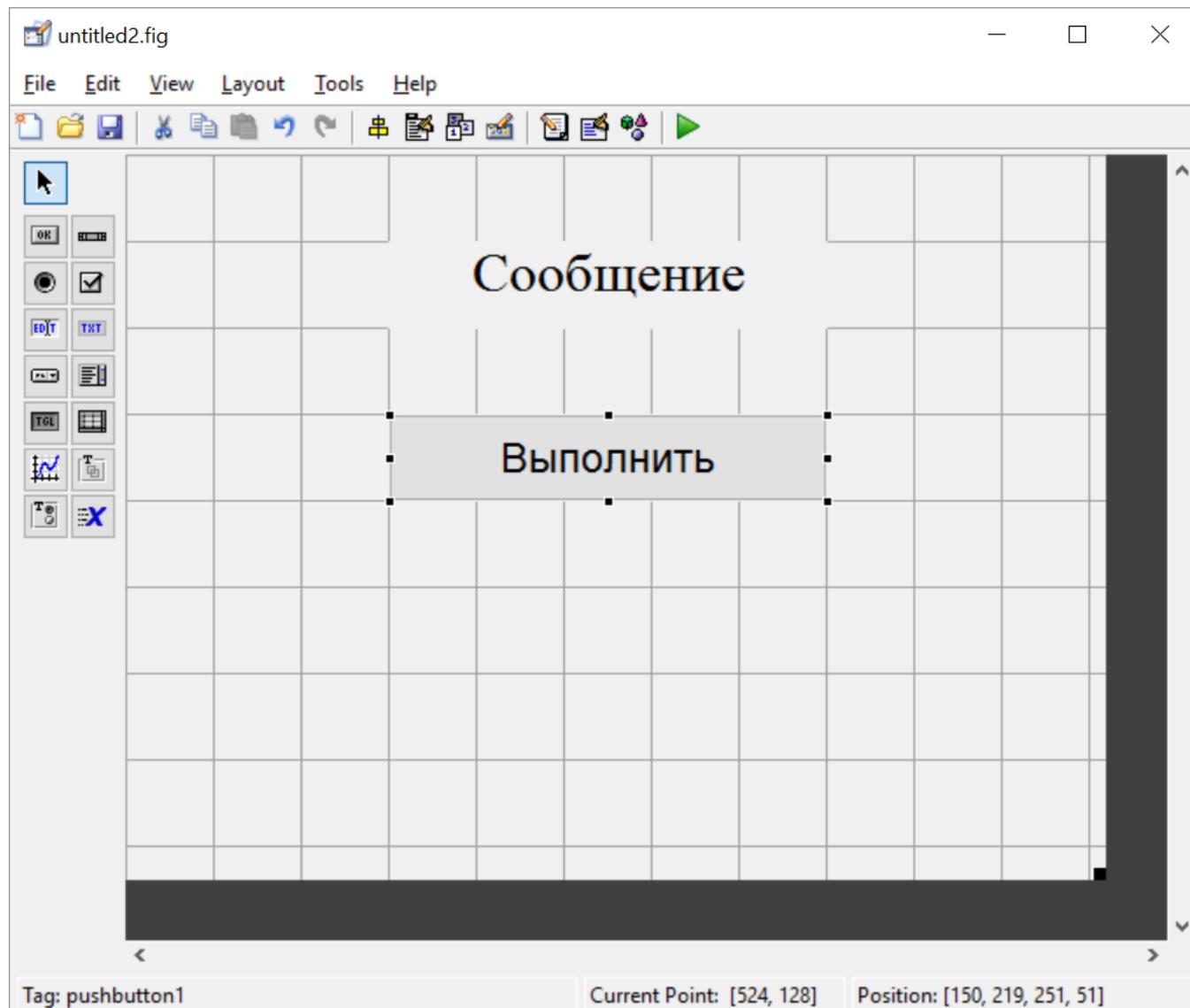
Push Button / Property Inspector / String = Выполнить

Выбрать свойство **FontSize**
и назначить значение для
этого свойства – 18.

Выбрать свойство **String** и
назначить значение для
этого свойства – **Выполнить**.

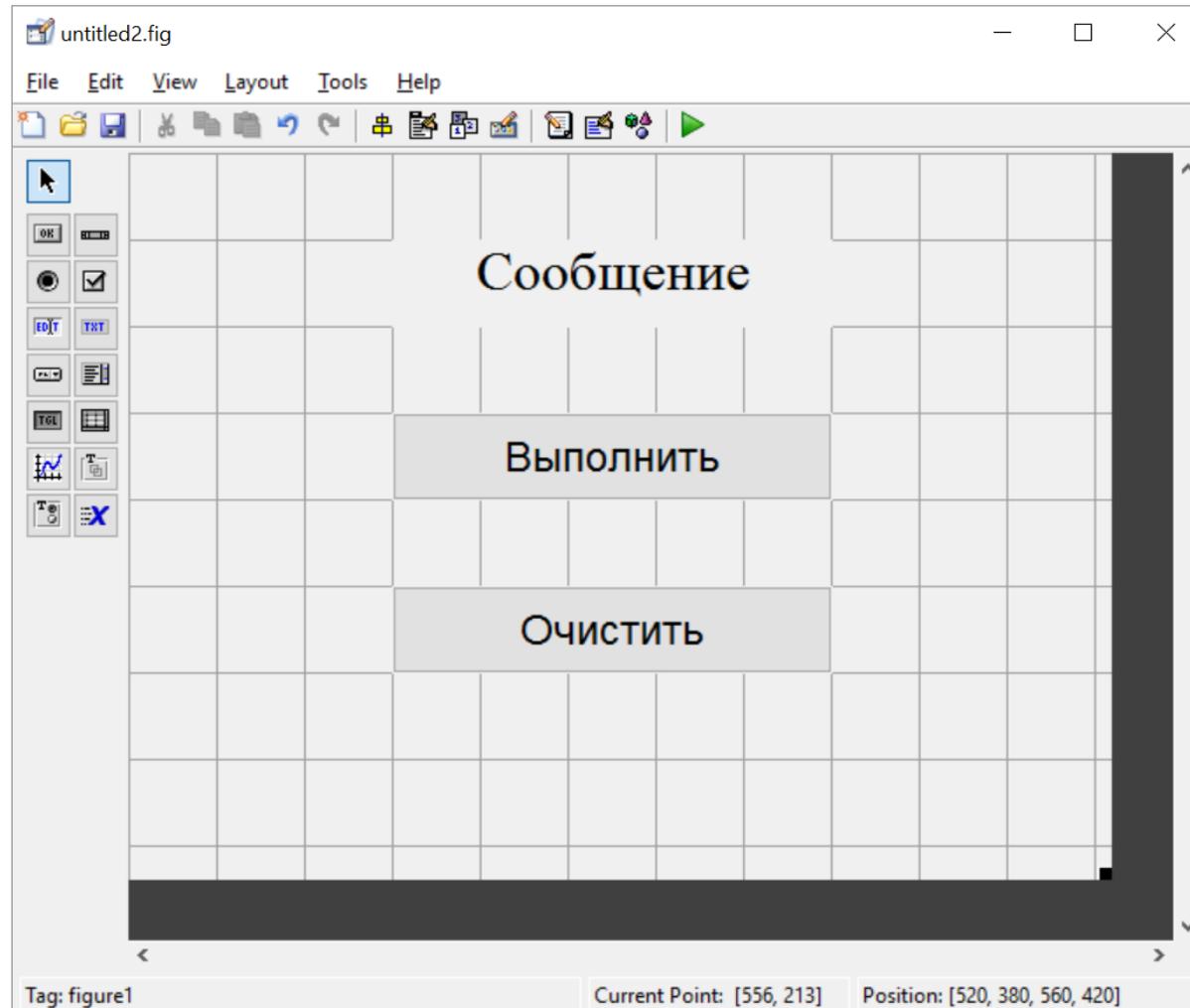


GUIDE. Пример. Привет, МИР! Шаг 5. Результат



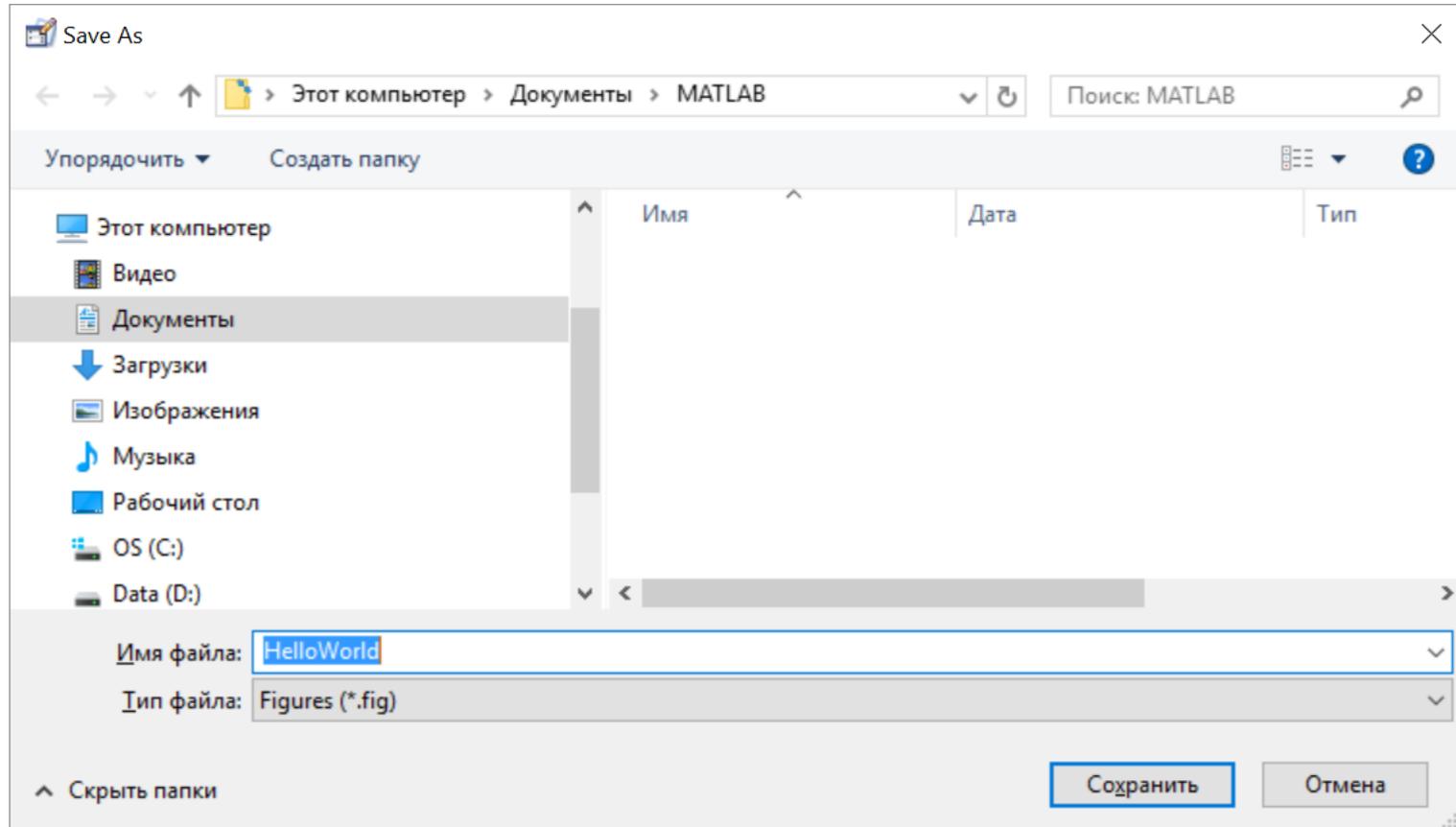
GUIDE. Пример. Привет, МИР! Шаг 6

Push Button / Property Inspector / String = Очистить / Результат



GUIDE. Пример. Привет, МИР! Шаг 7

Сохранить файл / HelloWorld.fig



GUIDE. Пример. Привет, МИР! Шаг 7. Результат

```
FILE EDIT NAVIGATE BREAKPOINTS RUN
HelloWorld.m x +
1 function varargout = HelloWorld(varargin)
2 % HELLOWORLD MATLAB code for HelloWorld.fig
3 %     HELLOWORLD, by itself, creates a new HELLOWORLD or raises the existing
4 %     singleton*.
5 %
6 %     H = HELLOWORLD returns the handle to a new HELLOWORLD or the handle to
7 %     the existing singleton*.
8 %
```

...

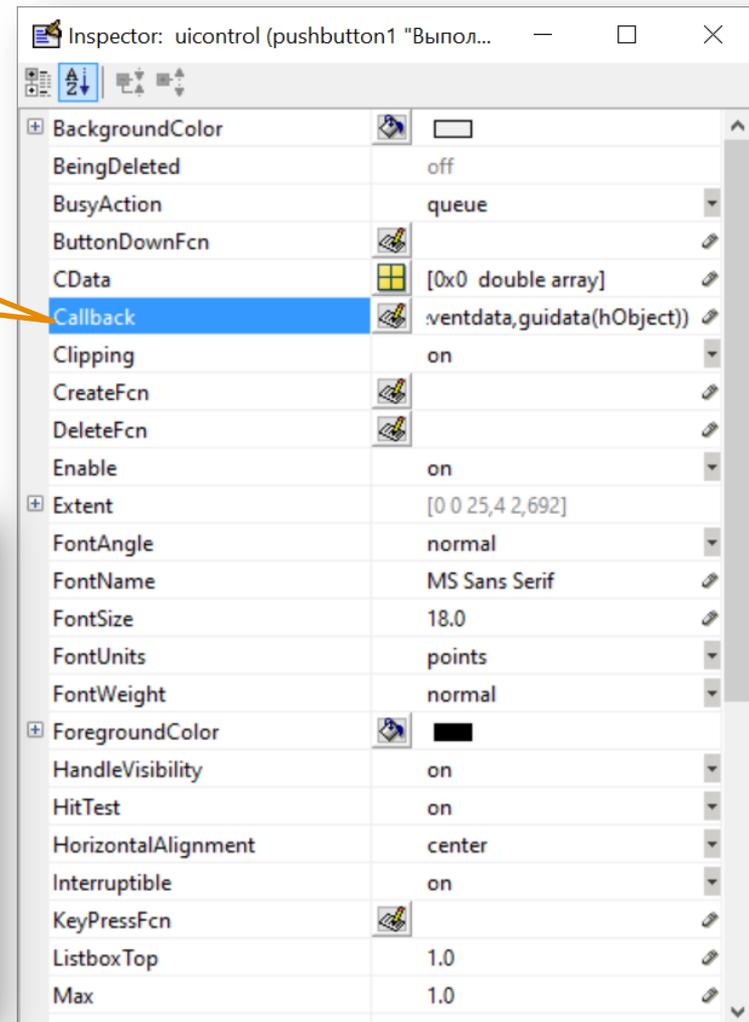
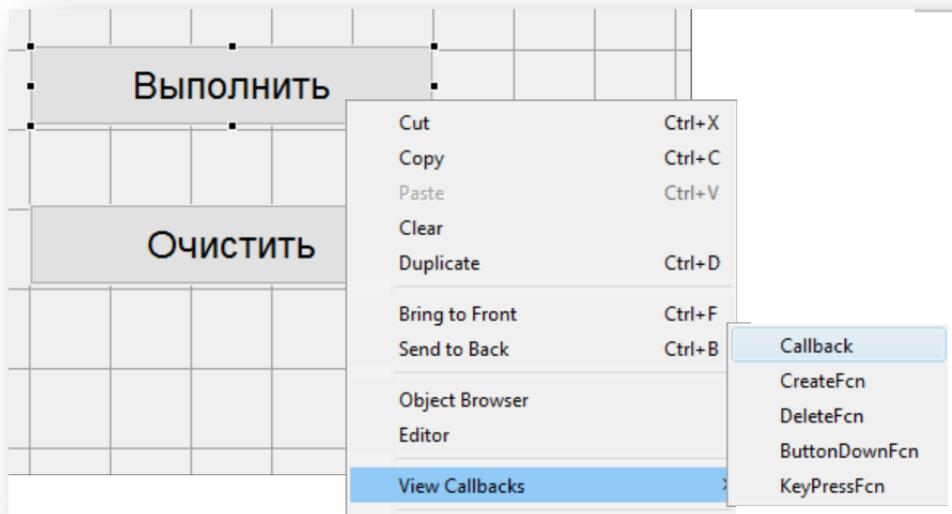
```
83 % --- Executes on button press in pushbutton2.
84 function pushbutton2_Callback(hObject, eventdata, handles)
85 % hObject     handle to pushbutton3 (see GCBO)
86 % eventdata   reserved - to be defined in a future version of MATLAB
87 % handles     structure with handles and user data (see GUIDATA)
```

`>> guide HelloWorld` - чтобы открыть GUI в редакторе

GUIDE. Пример. Привет, МИР! Шаг 8

Создание функции обратного вызова

Зайти в редактор Callback



GUIDE. Пример. Привет, МИР! Шаг 8

Создание функции обратного вызова

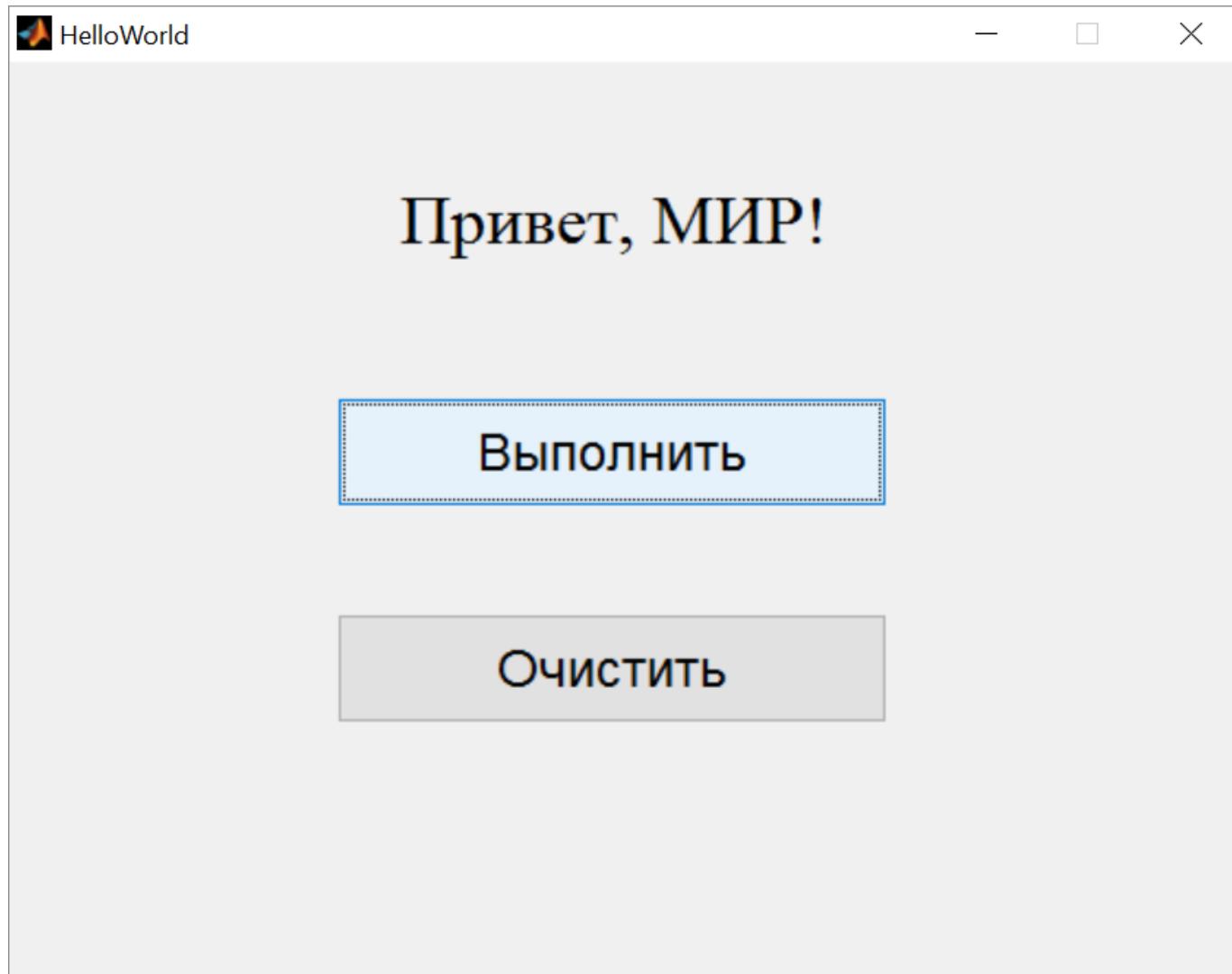
Функции «обратного вызова» - это инструкции, которые будут выполняться, когда пользователь нажимает кнопки или что-то делает с компонентами, которые включены в графический интерфейс Matlab.

```
76 % --- Executes on button press in pushbutton1.
77 function pushbutton1_Callback(hObject, eventdata, handles)
78 % hObject    handle to pushbutton1 (see GCBO)
79 % eventdata  reserved - to be defined in a future version of MATLAB
80 % handles    structure with handles and user data (see GUIDATA)
81 set(handles.output_line,'String','Hello World!!!')
```

```
set(handles.output_line,'String','Привет, МИР!')
```

Добавляем эту строку в функцию для pushbutton1

GUIDE. Пример. Привет, МИР! Шаг 8. Результат



GUIDE. Пример. Привет, МИР! Шаг 9

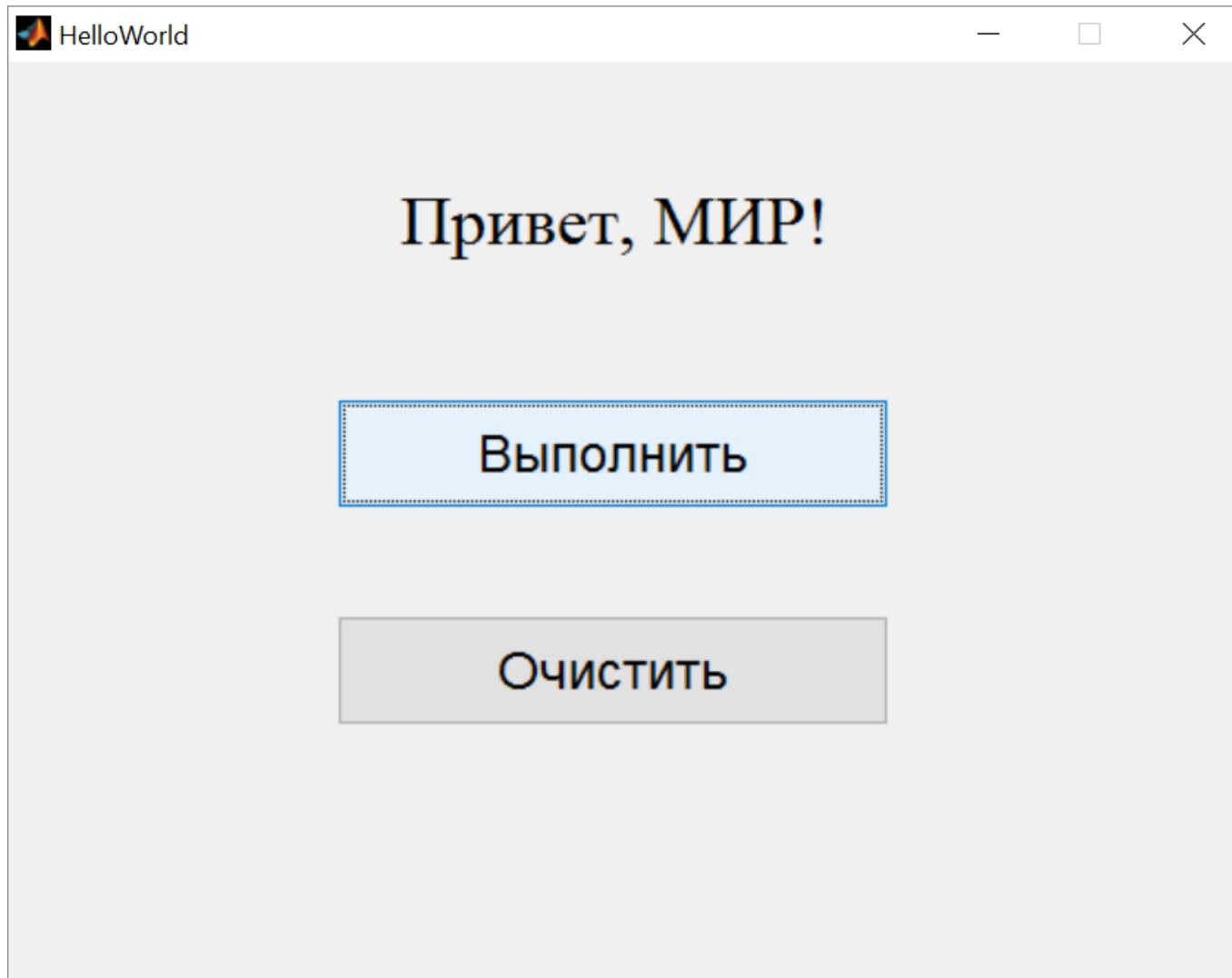
Создание функции обратного вызова для кнопки Очистить

```
% --- Executes on button press in pushbutton2.  
function pushbutton2_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton3 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
set(handles.output_line, 'String', ' ')
```

```
set(handles.output_line, 'String', ' ')
```

Добавляем эту строку в функцию для pushbutton2

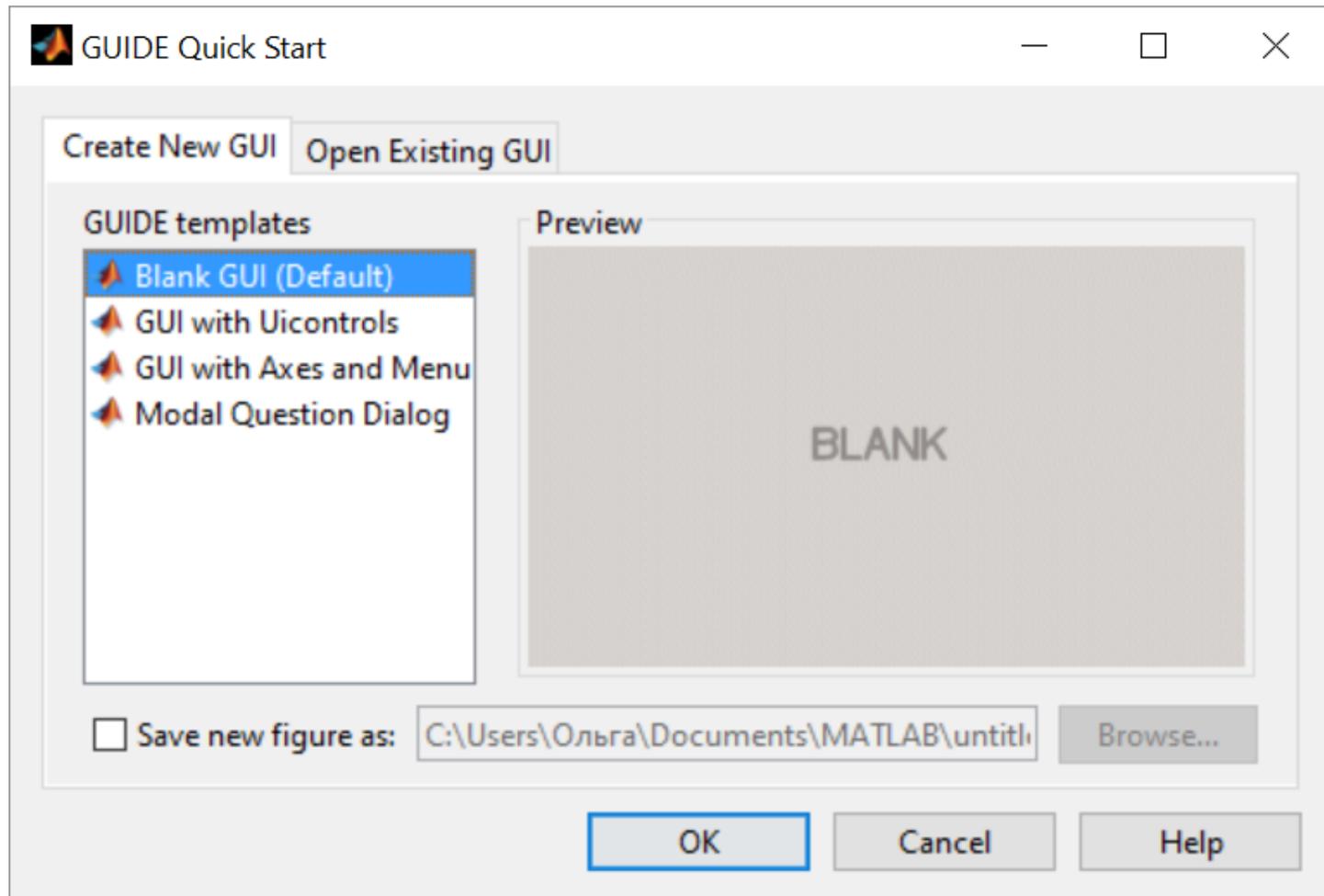
GUIDE. Пример. Привет, МИР! Итог



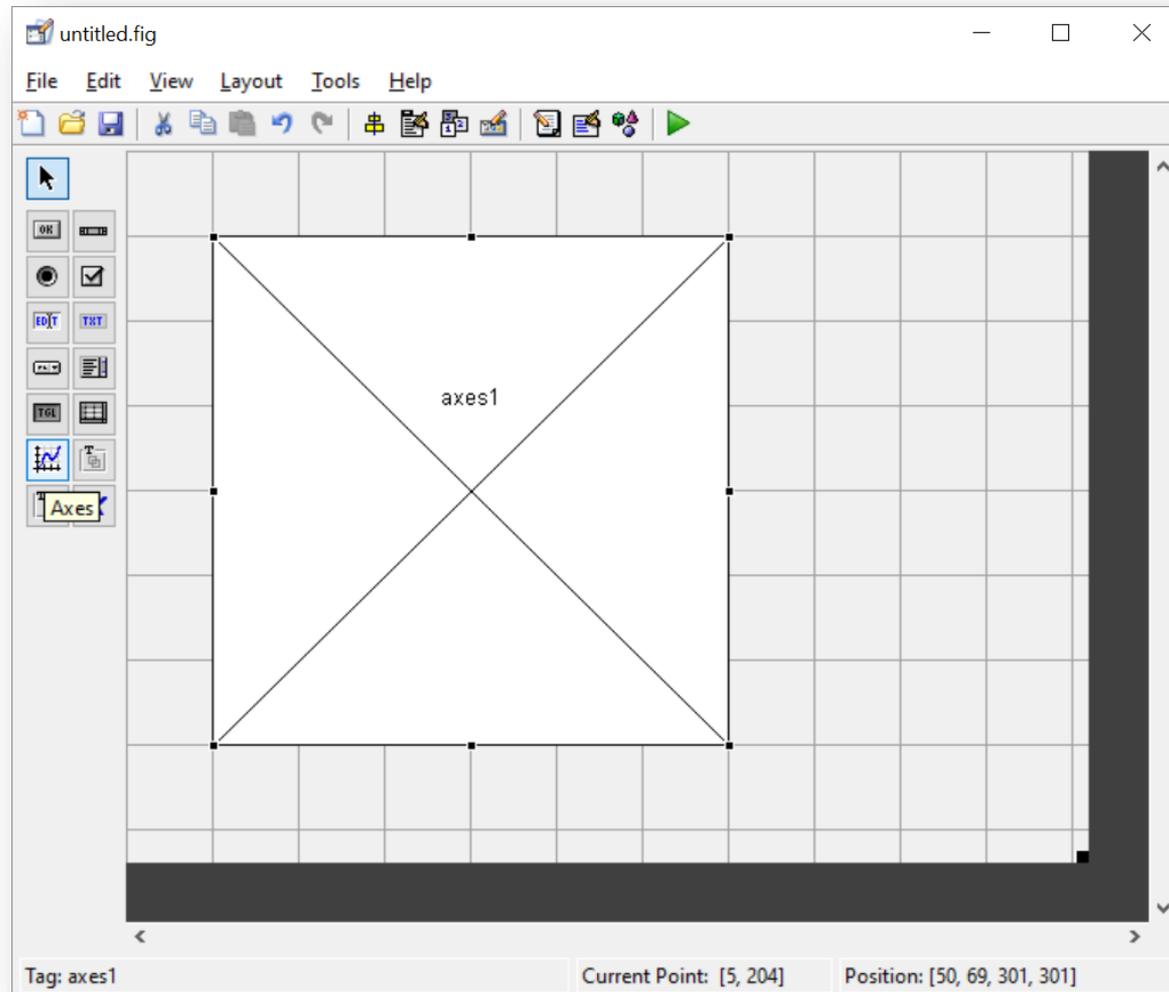
ПРИМЕР 2. Логотип Matlab в GUI

GUIDE. Пример 2. Шаг 1

Blank GUI (Default)

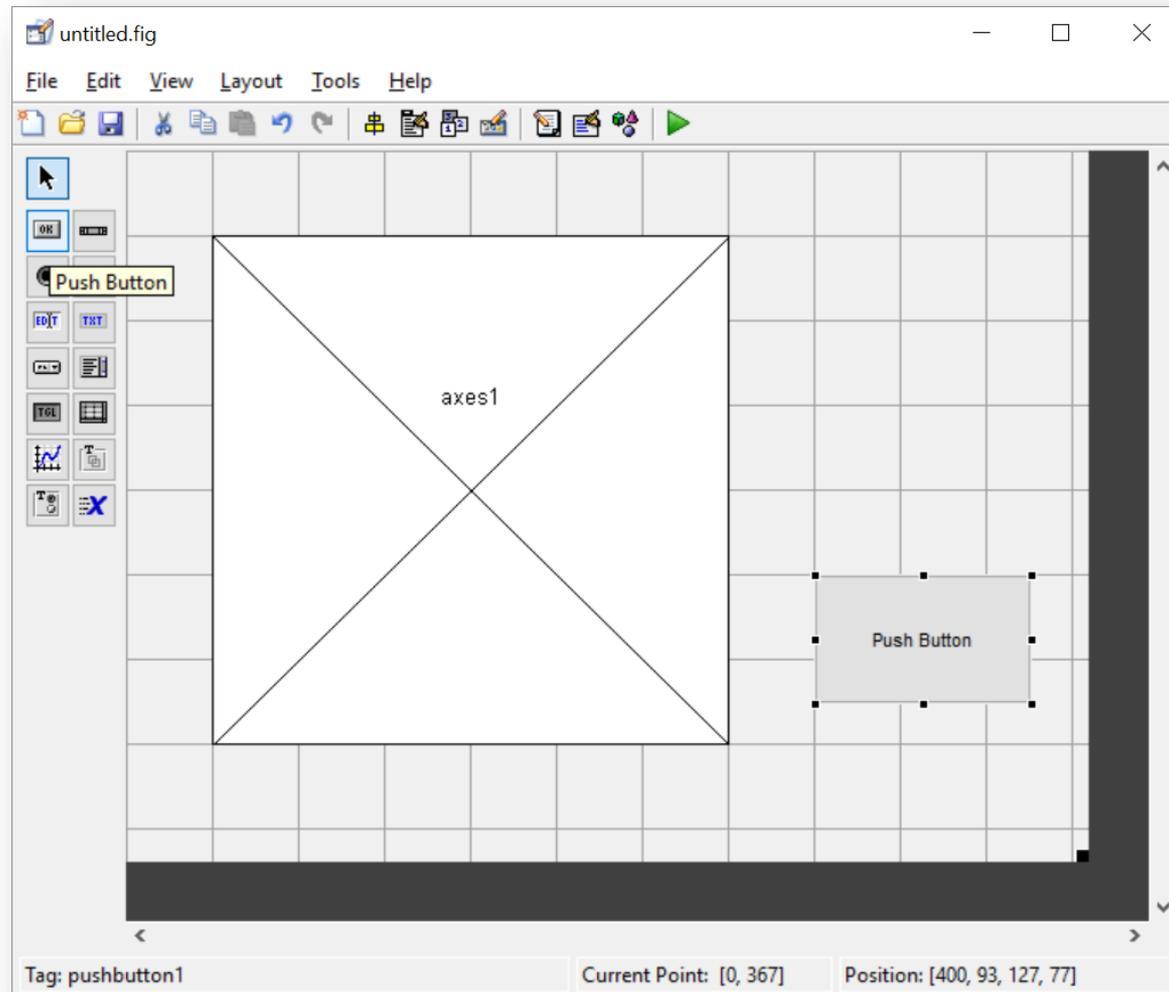


GUIDE. Пример 2. Шаг 2



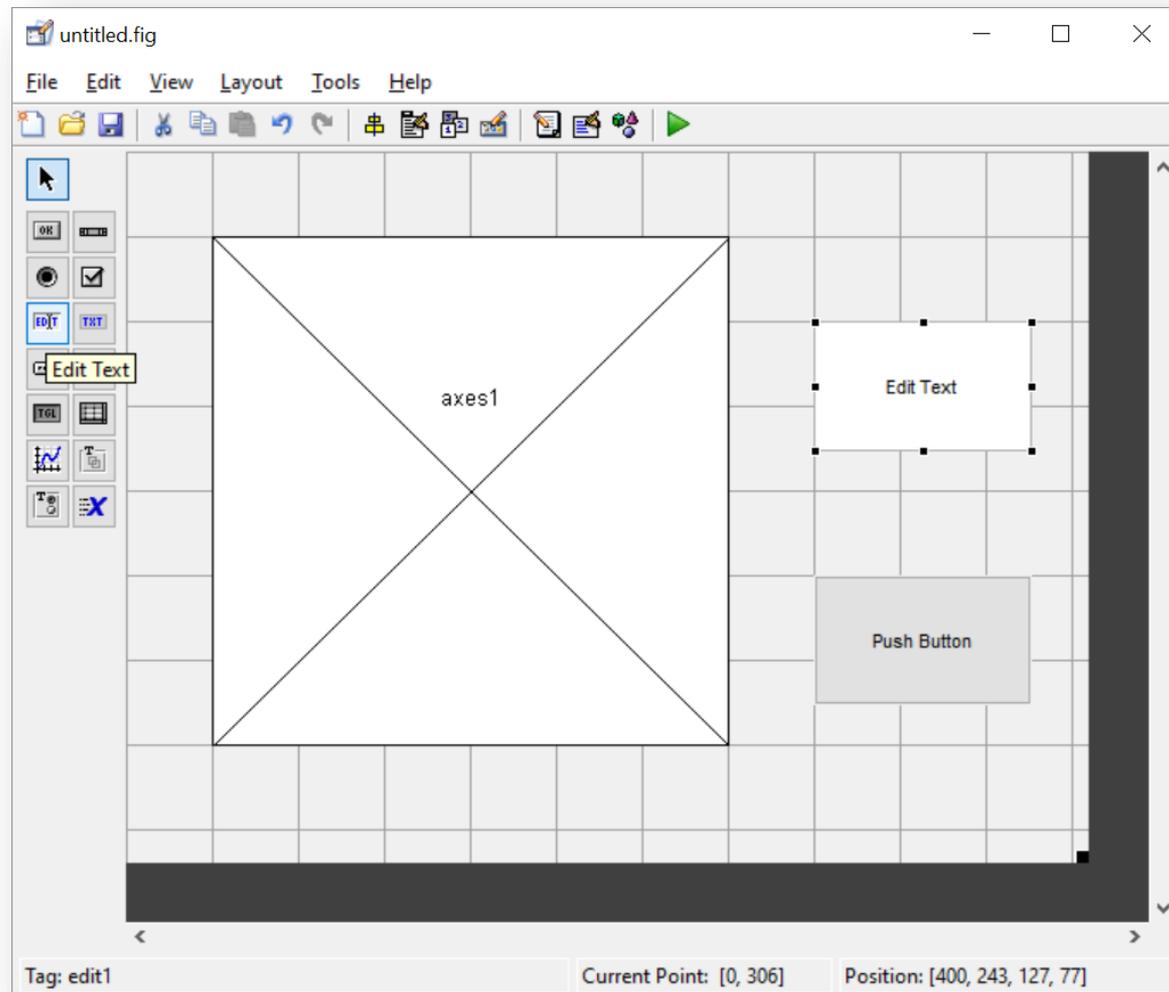
Создание Осей axes

GUIDE. Пример 2. Шаг 3



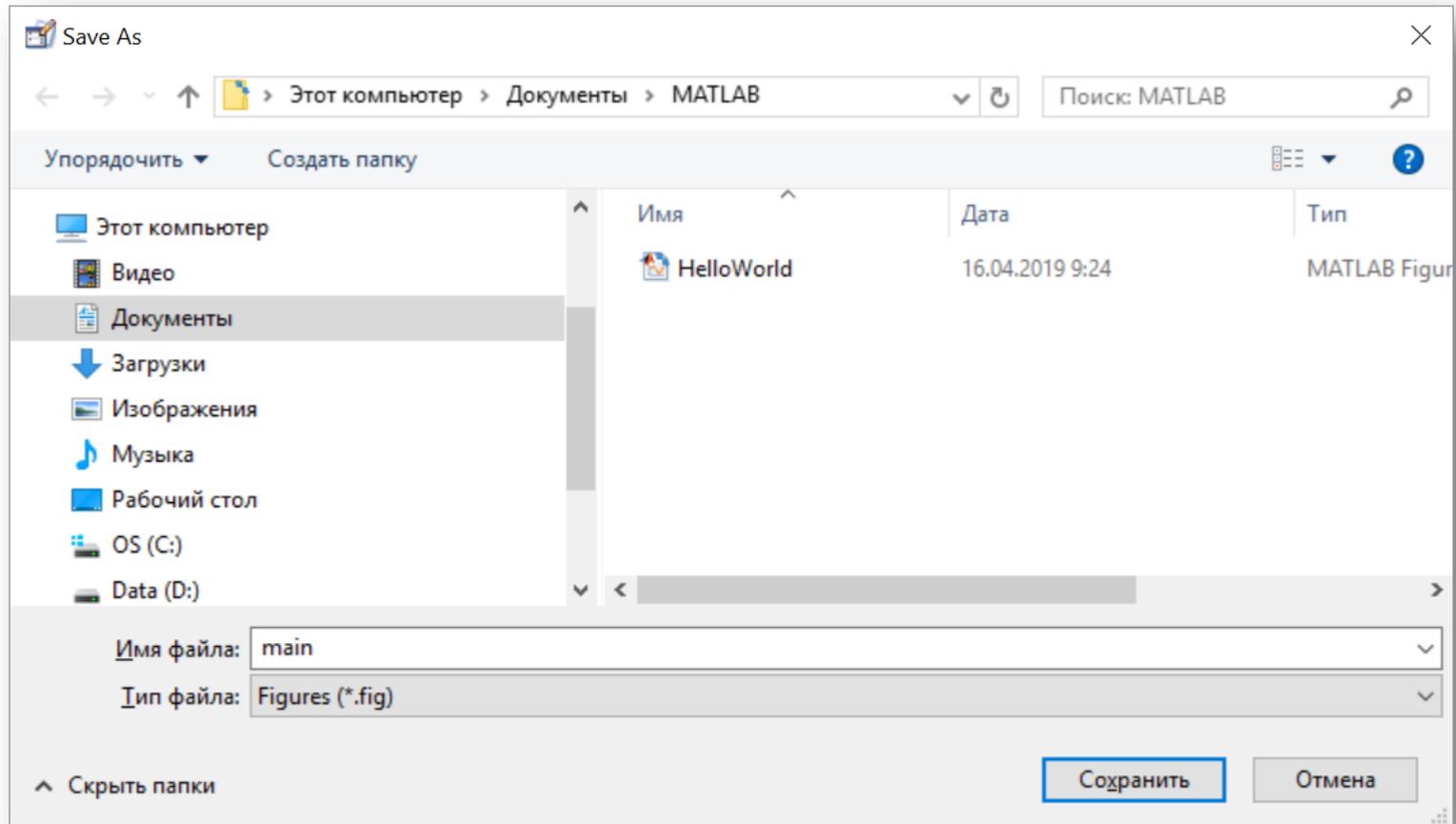
Создание кнопки Push Button

GUIDE. Пример 2. Шаг 4



Создание редактируемого текстового поля Edit Text

GUIDE. Пример 2. Шаг 5



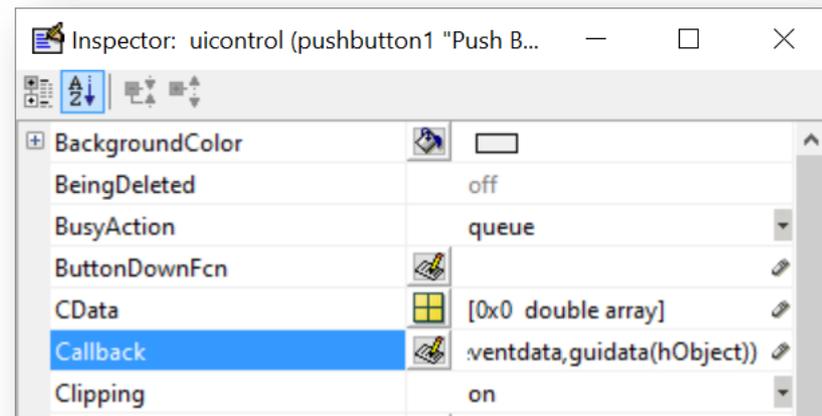
Сохраняем в файле с расширением fig

GUIDE. Пример 2. Шаг 6

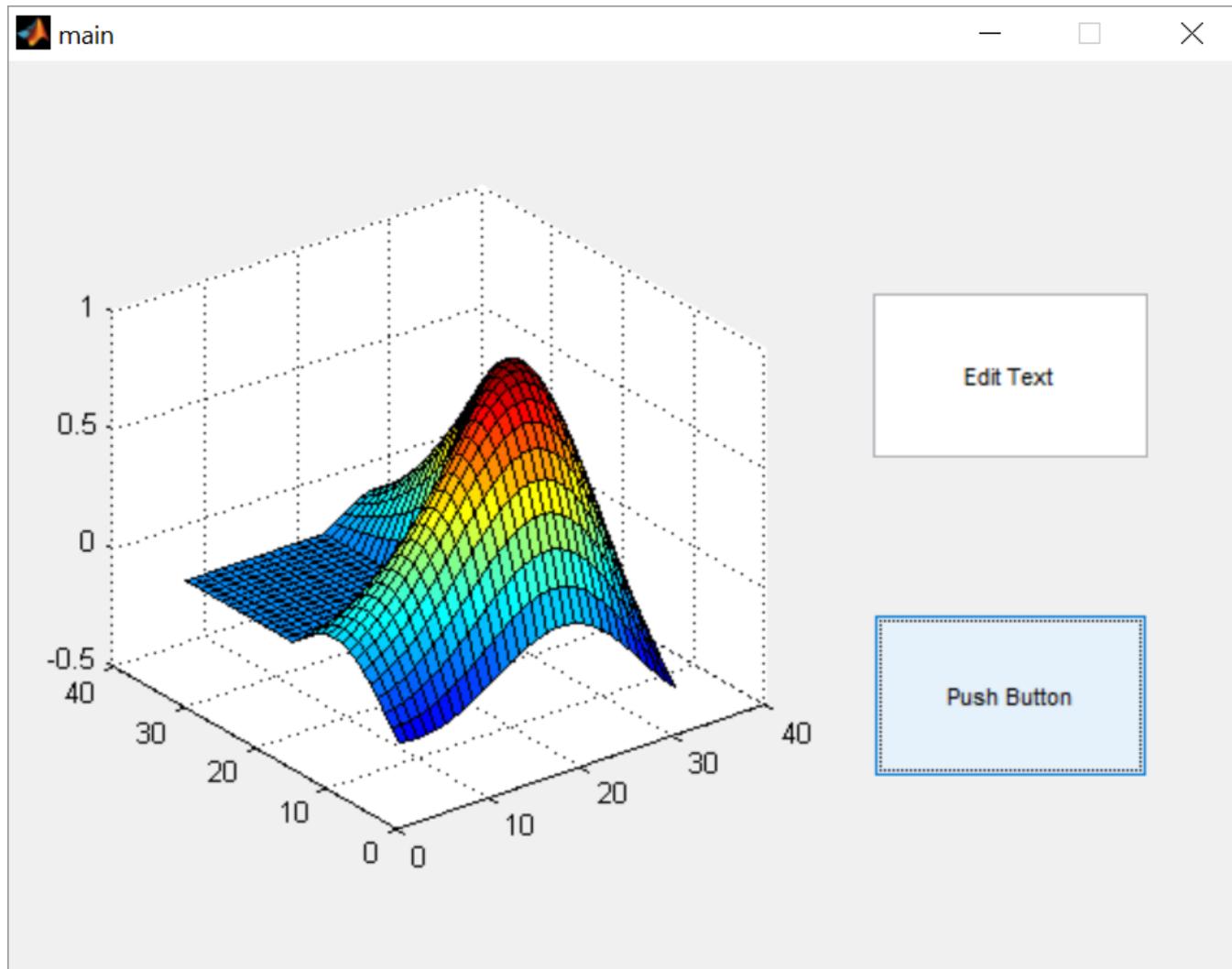
Редактирование функции обратного вызова для pushbutton1

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton1 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
surf(membrane(1));
```

Логотип MATLAB



GUIDE. Пример 2. Шаг 6. Результат

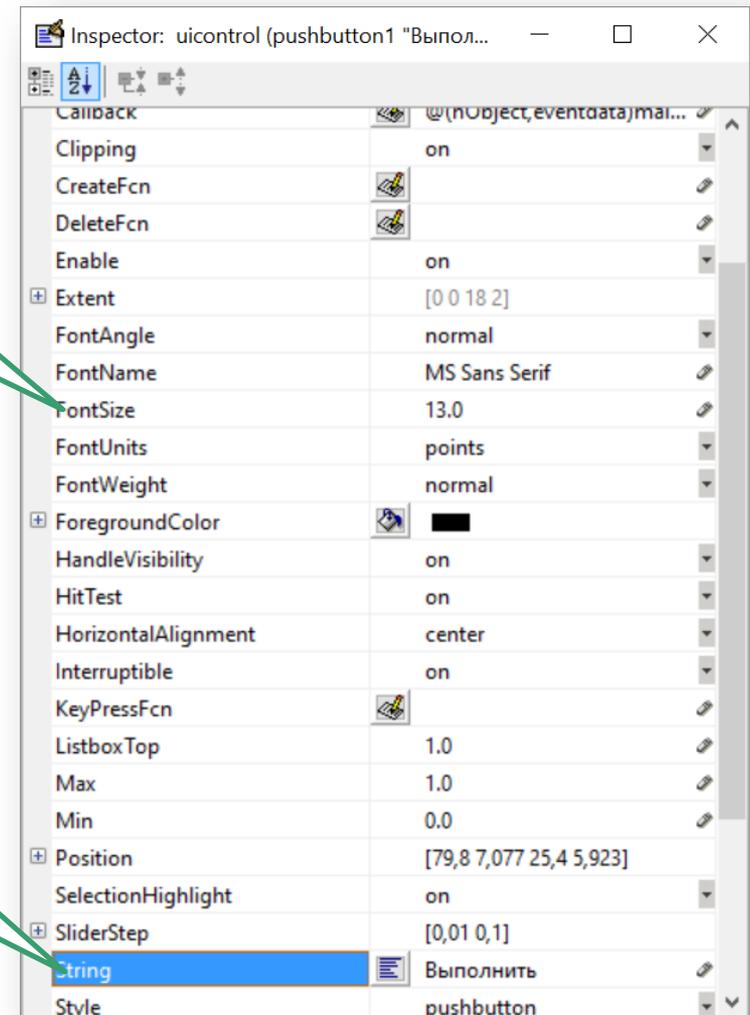


GUIDE. Пример 2. Шаг 7

Редактирование свойств кнопки pushbutton1

FontSize = 13

String = Выполнить



GUIDE. Пример 2. Шаг 7

Изменение свойств редактируемого поля Edit Text

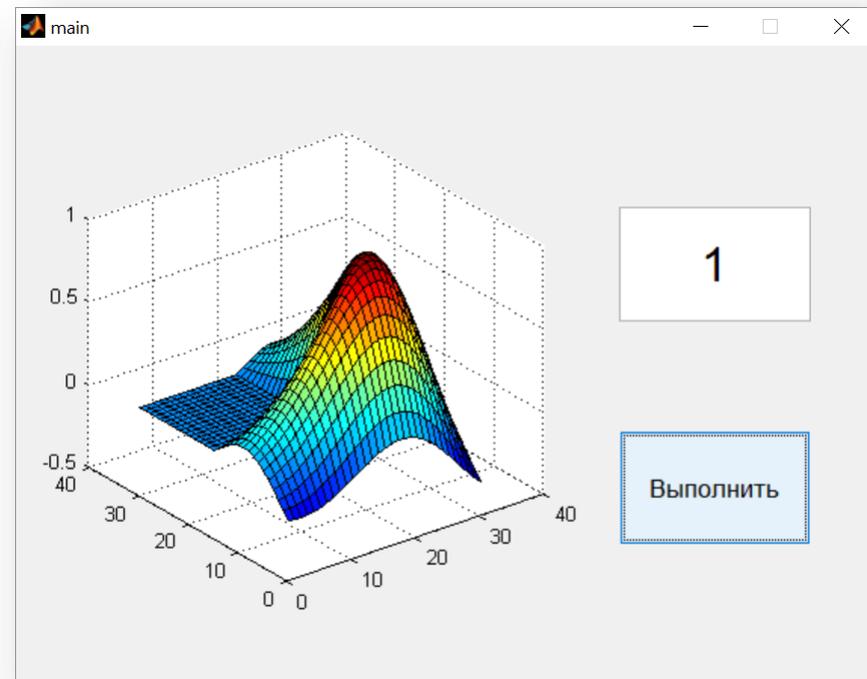
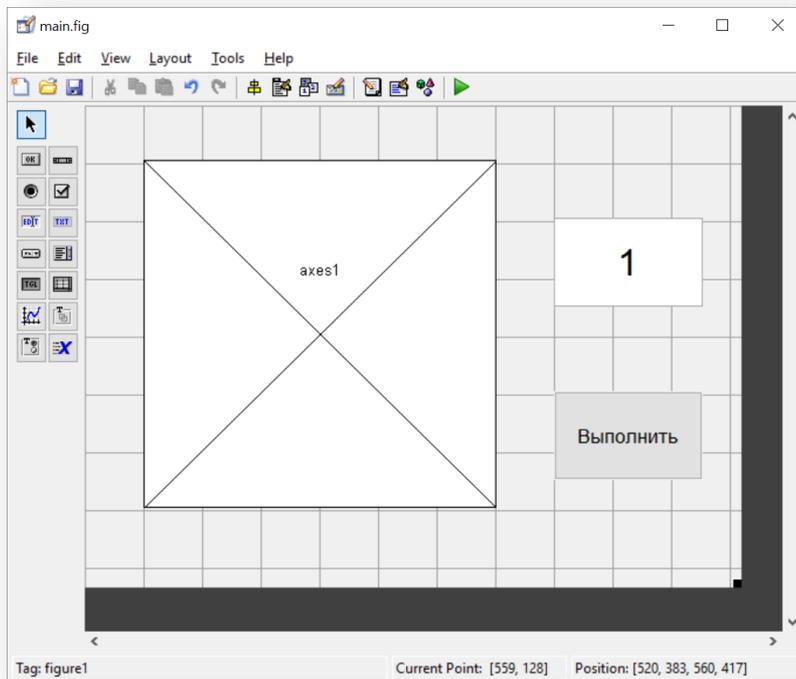
`FontSize = 24`

`String = 1`

Inspector: uicontrol (edit1 "1")

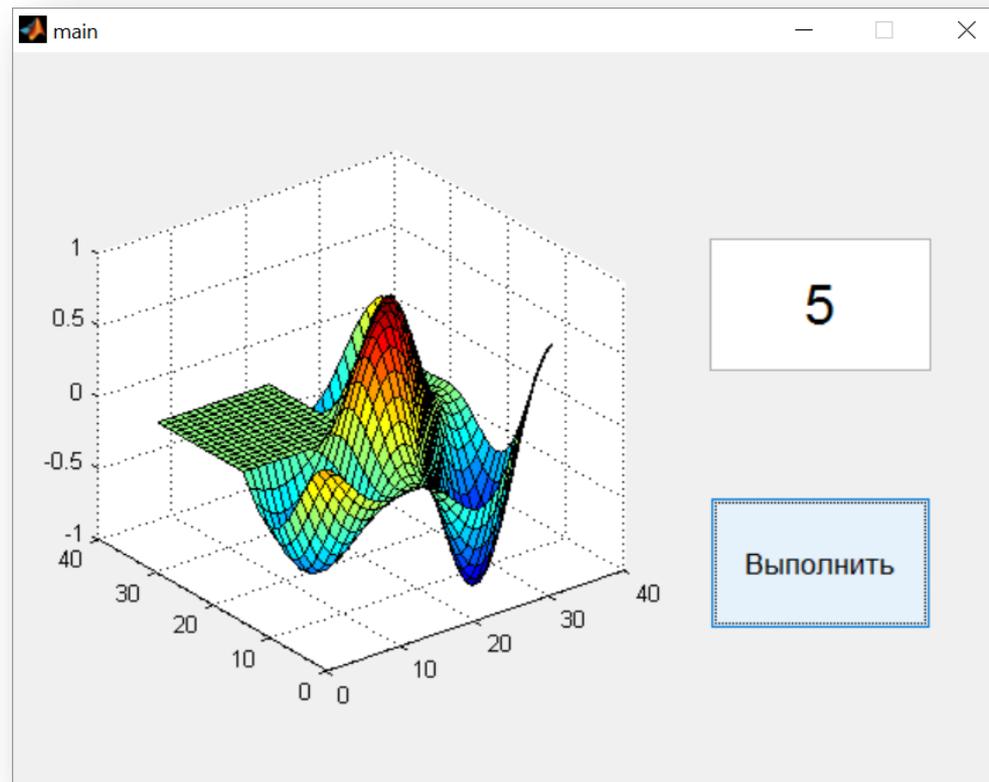
FontSize	24.0	
FontUnits	points	
FontWeight	normal	
ForegroundColor		
HandleVisibility	on	
HitTest	on	
HorizontalAlignment	center	
Interruptible	on	
KeyPressFcn		
ListboxTop	1.0	
Max	1.0	
Min	0.0	
Position	[79,8 18,615 25,4 5,923]	
SelectionHighlight	on	
SliderStep	[0,01 0,1]	
String	1	
Style	edit	
Tag	edit1	
TooltipString		
UIContextMenu	<None>	
Units	characters	
UserData	[0x0 double array]	
Value	[0]	
Visible	on	

GUIDE. Пример 2. Шаг 7. Результат



GUIDE. Пример 2. Шаг 8

Отредактируем функцию обратного вызова для `pushbutton1` таким образом, чтобы при вводе в текстовое поле целого числа от 1 до 5 выводилась соответствующая мода поверхности `membrane`:



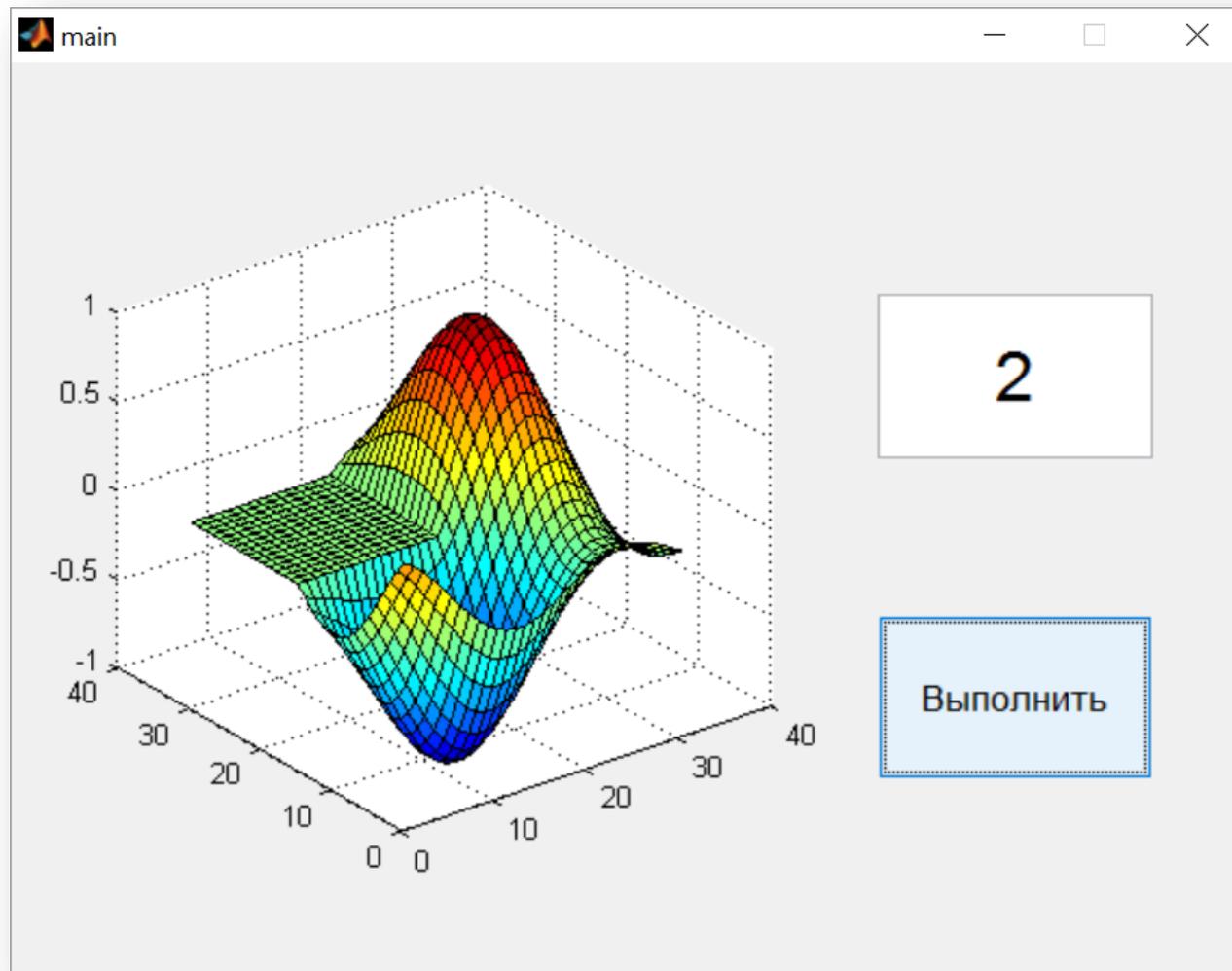
GUIDE. Пример 2. Шаг 8

Редактирование функции pushbutton1

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton1 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
  
% создали переменную var,  
% в которую записали значение поля edit1 в виде строки  
var=get(handles.edit1,'string')  
  
% строку преобразовали в число  
var=str2double(var)  
  
% будет строиться мода, соответствующая введенному числу  
surf(membrane(var));
```

GUIDE. Пример 2. Шаг 8

Сохранили файл и запустили на выполнение для разных мод



GUIDE. Пример 2. Шаг 9

Изменим код таким образом, чтобы перерисовка производилась в том числе в момент ввода числа в текстовое поле.

Для этого сначала создадим функцию обновления осей:

```
function updateAxes(handles)
var=get(handles.edit1,'string')
var=str2double(var)
surf(membrane(var));
```

Этот код добавим в конец файла main.fig

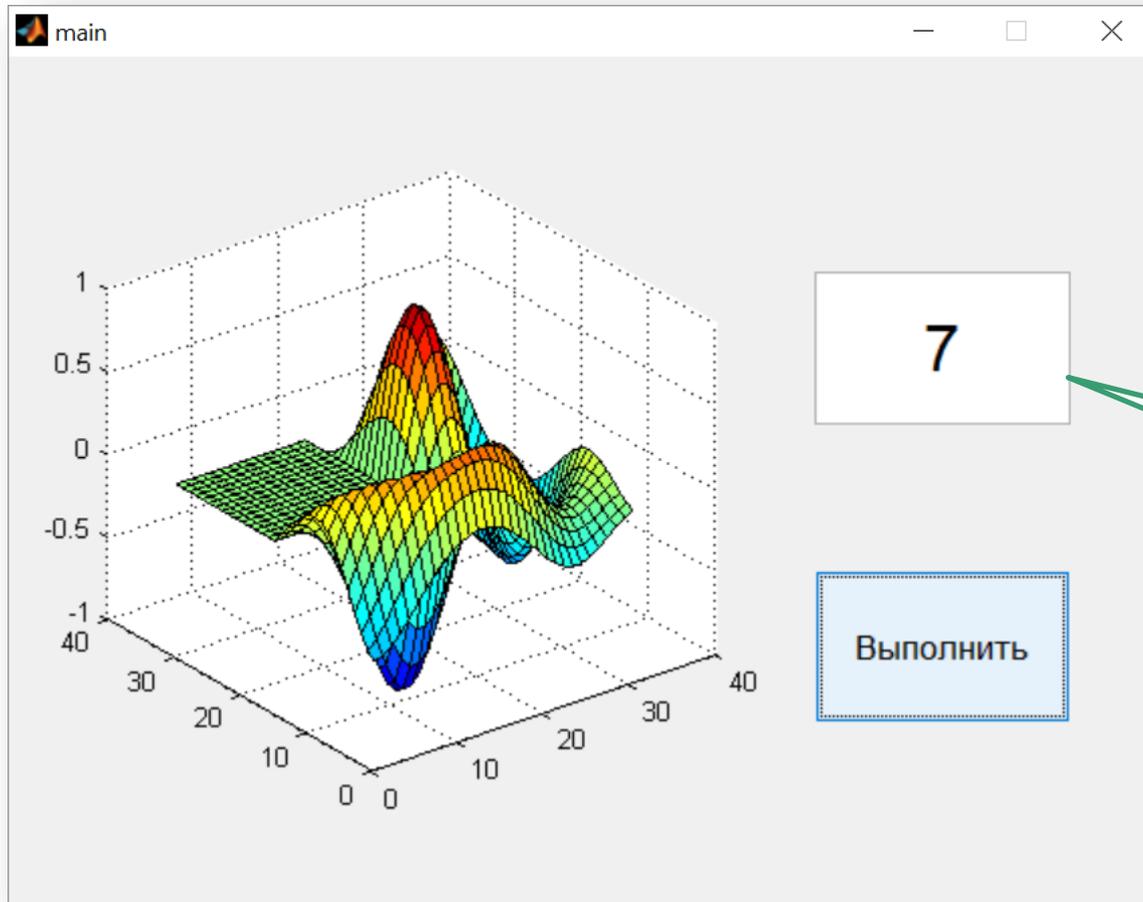
GUIDE. Пример 2. Шаг 9

Затем эту функцию будем использовать в качестве реакции при нажатии на кнопку, так и при заполнении текстового поля.

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton1 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
updateAxes(handles)
```

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject      handle to pushbutton1 (see GCBO)  
% eventdata    reserved - to be defined in a future version of MATLAB  
% handles      structure with handles and user data (see GUIDATA)  
updateAxes(handles)
```

GUIDE. Пример 2. Шаг 9. Результат



**Перестроение
происходит при вводе
числа в текстовое
поле.**

GUIDE. Пример 2. Шаг 10

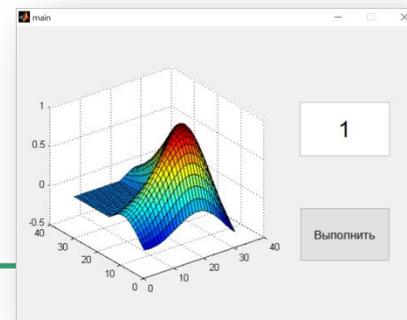
Добавим вызов функции `updateAxes(handles)` в конец функции

```
function main_OpeningFcn(hObject, eventdata, handles, varargin)
```

для того, чтобы мембрана прорисовывалась при загрузке GUI

```
function main_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject      handle to figure
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
% varargin     command line arguments to main (see VARARGIN)
% Choose default command line output for main
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

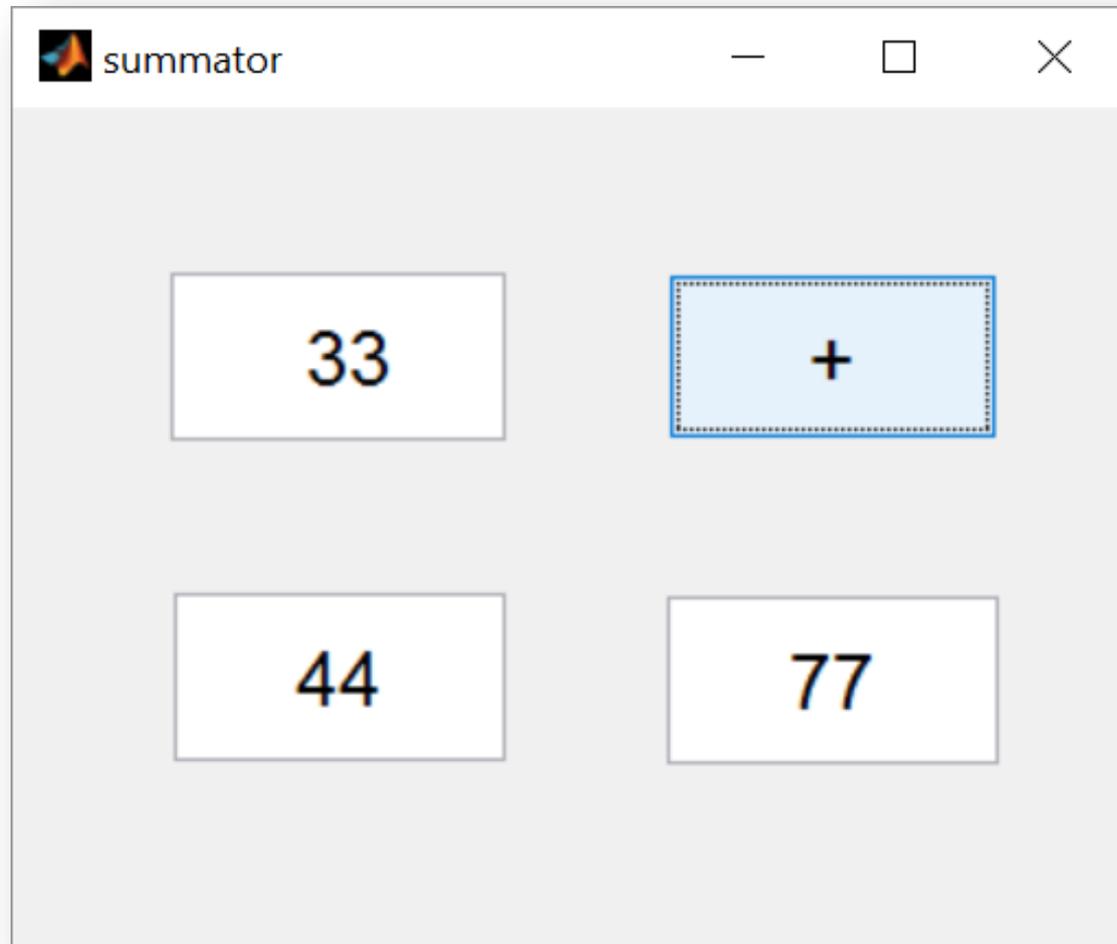
updateAxes(handles)
```



ПРИМЕР 3. СУММАТОР

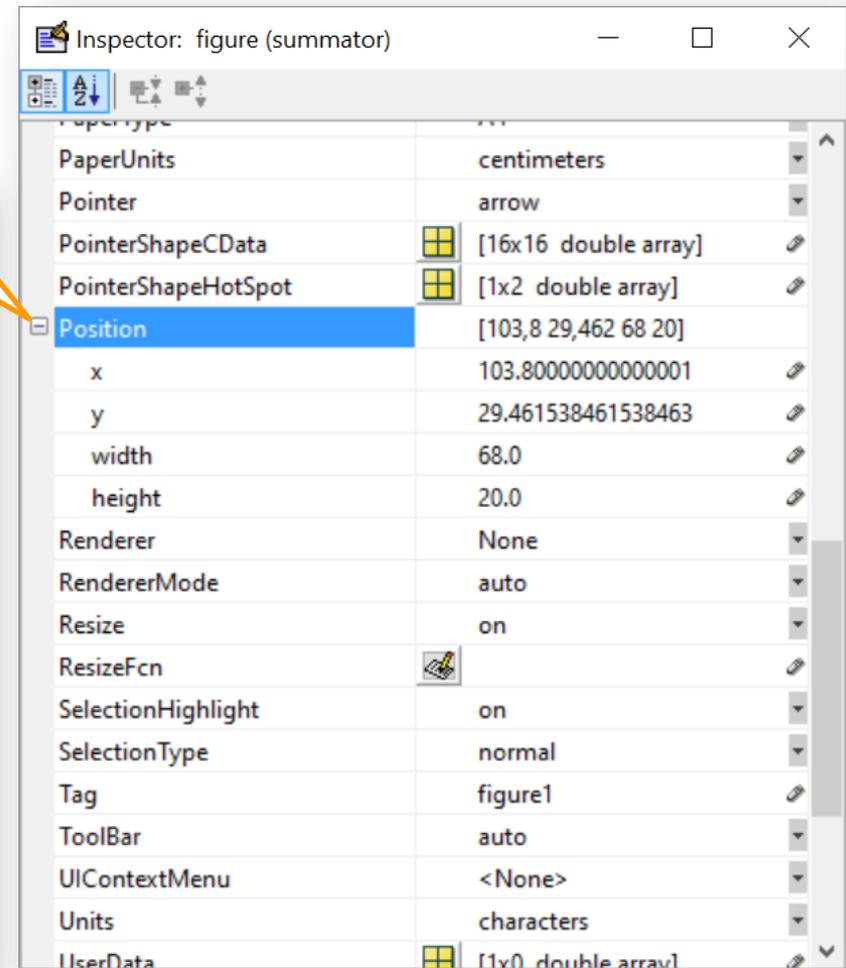
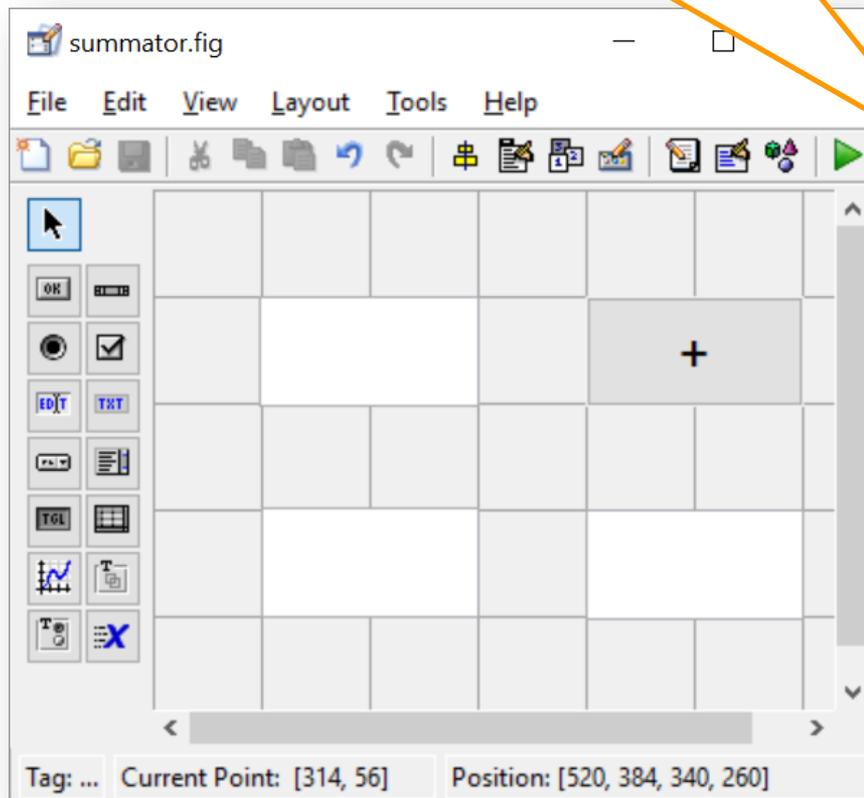
GUIDE. Пример 3

Создать GUI, который позволяет складывать числа



GUIDE. Пример 3. Шаг 1

В свойстве Position можно задать ширину и высоту объекта



GUIDE. Пример 3. Шаг 2

```
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)
sum_(handles)
```

```
function sum_(handles)
n1=str2double(get(handles.edit1, 'String'))
n2=str2double(get(handles.edit2, 'String'))
n3=n1+n2
set(handles.edit3, 'String', n3);
```

Программирование GUI

Элементы GUI

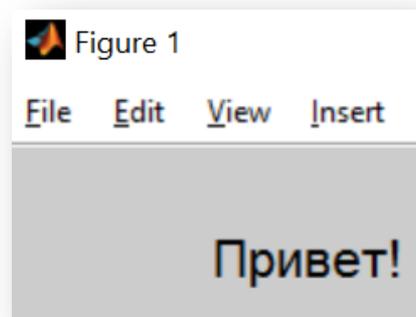
`figure` – графическое окно, в котором находятся дочерние элементы. `figure` – родитель или общий предок.

`uicontrol` – конструктор для всех элементов управления GUI

Элементы GUI. Text

```
clear
f=figure
cf=get(f, 'color')
myText1=uicontrol( gcf, 'style', 'text', ...
                  'position', [20, 360, 200, 30], ...
                  'BackgroundColor', cf, ...
                  'String', 'Привет!', ...
                  'FontSize', 16);
```

В свойстве position указываются: координаты нижнего левого угла, ширина и высота объекта.



Как получить информацию из поля:

```
s=get(myText1, 'string')
```

Элементы GUI. Edit Text

```
clear  
f=figure  
myEditText1=uicontrol( gcf, 'style','edit',...  
                        'position',[20,360,200,30],...  
                        'FontSize',16,...  
                        'string', 123);
```

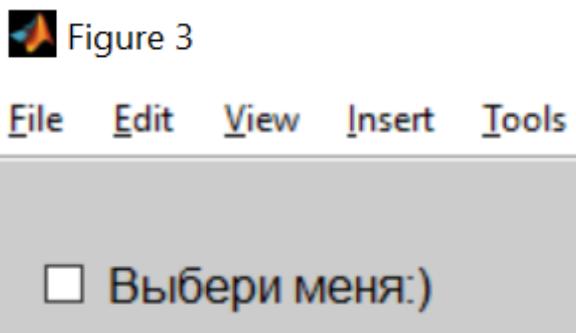


Как получить информацию из поля:

```
% num2str - convert number to string  
n=num2str(get(myEditText1,'string'))  
  
%str2double - convert string to double  
f=str2double(get(myEditText1,'string'))
```

Элементы GUI. Check Box

```
myCheckBox1=uicontrol(gcf, 'style', 'checkbox', ...  
    'position', [20,360, 240,40], ...  
    'backgroundcolor', get(gcf, 'color'), ...  
    'string', ' Выбери меня:)', ...  
    'FontSize', 12)
```

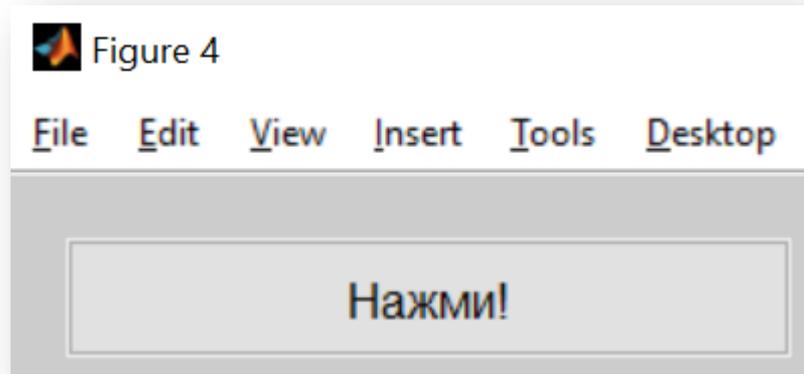


Как получить информацию из поля:

```
check=get(myCheckBox1, 'value')
```

Элементы GUI. Pushbutton

```
myPushButton1 = uicontrol( gcf, ...  
    'style', 'pushbutton', ...  
    'position', [20, 360, 240, 40], ...  
    'String', 'Нажми!', ...  
    'FontSize', 12, ...  
    'Callback', 'NameFile');
```



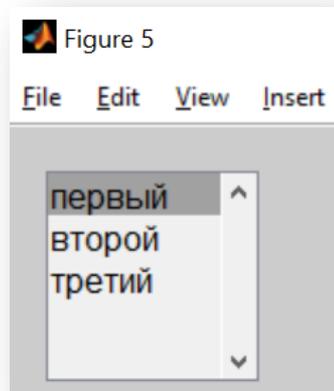
Callback – свойство, определяющее какое действие нужно выполнить при нажатии на кнопку.

Значениями свойства могут быть:

- ✓ Имя функции
- ✓ Имя файла

Элементы GUI. ListBox

```
myListBox1=uicontrol(gcf,...
```



```
'style','listbox',...
```

```
'position',[20,300, 100,100],...
```

```
'string',{'первый','второй','третий'},...
```

```
'FontSize',12,...
```

```
'HorizontalAlignment','left');
```

Как получить информацию:

```
k=get(myListBox1,'value')
```

```
List=get(myListBox1,'string')
```

```
choice= List{k}
```

```
k = 1
```

```
List =
```

```
    'первый'
```

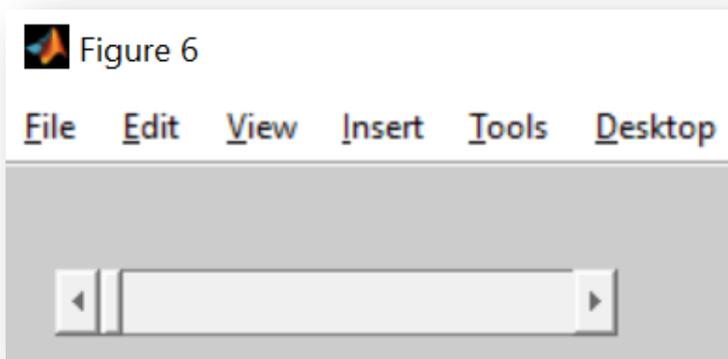
```
    'второй'
```

```
    'третий'
```

```
choice = первый
```

Элементы GUI. Slider

```
mySlider1=uicontrol(gcf,...  
                    'style','slider',...  
                    'position',[20,360, 200,24],...  
                    'HorizontalAlignment','center',...  
                    'FontSize',12,...  
                    'sliderstep',[1,0])
```

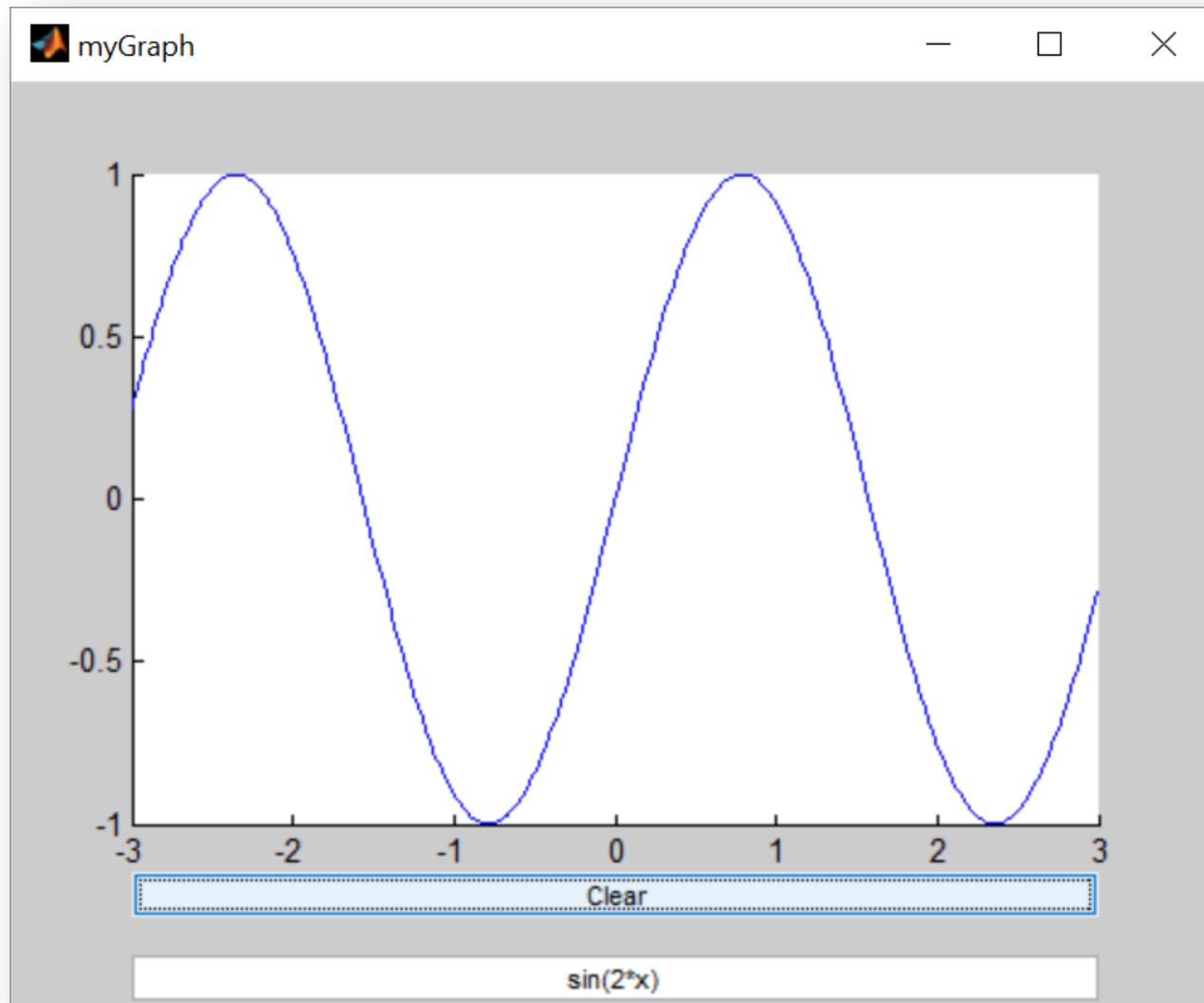


Как получить информацию:

```
k=get(mySlider1,'value')
```

ПРИМЕР 4. Программирование GUI

GUI. Пример 4



GUI. Пример 4. Шаг 1

```
function myGraph
% создаем графическое окно с тегом win
hF = figure('Name', 'myGraph',...
            'NumberTitle', 'off',...
            'MenuBar', 'none',...
            'Units', 'characters',...
            'Position', [10 10 100 30],...
            'Tag', 'win');
```

GUI. Пример 4. Шаг 2

```
% создаем оси с тегом ax  
axes('Position', [0.1 0.2 0.8 0.7], ...  
     'Tag', 'ax', ...  
     'NextPlot', 'replacechildren');
```

GUI. Пример 4. Шаг 3

```
% создаем кнопку с тегом btn
uicontrol('Style', 'pushbutton',...
          'Units', 'normalized',...
          'Position', [0.1 0.1 0.8 0.05],...
          'String', 'Plot', ...
          'Callback', @BtnCallback,...
          'Tag', 'btn');
```

GUI. Пример 4. Шаг 4

```
% создаем область ввода текста с тегом edtFun
uicontrol('Style', 'edit',...
          'Units', 'normalized',...
          'Position', [0.1 0.01 0.8 0.05],...
          'BackgroundColor', 'w',...
          'Tag', 'edtFun');
```

GUI. Пример 4. Шаг 5

```
function BtnCallback(src, evt)
% подфункция обработки события нажатия на кнопку
% записываем в структуру handles указатели на объекты приложения
handles = guihandles(src);

% сейчас в поле ax структуры handles находится указатель на оси
% делаем их текущими
axes(handles.ax)

% Проверяем, совпадает ли надпись на кнопке с Plot
if isequal(get(src, 'String'), 'Plot')
    % надпись на кнопке Plot

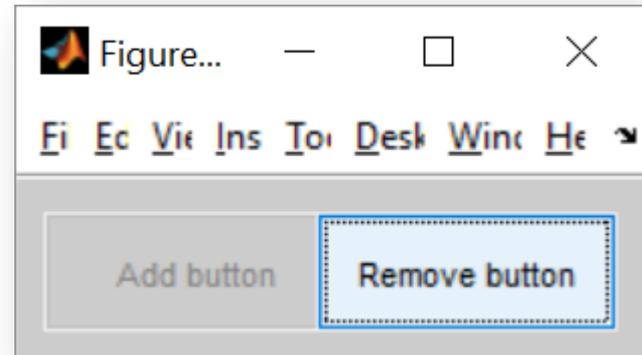
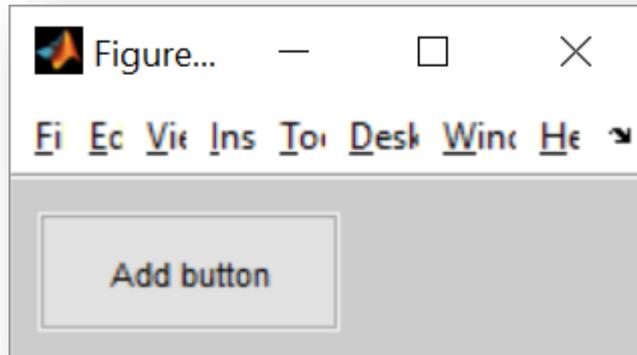
    % берем текст из строки ввода (выражение для функции)
    str = get(handles.edtFun, 'String');

    % строим график в осях
    fplot(str, [-3 3])

    % изменяем надпись на кнопке на Clear
    set(src, 'String', 'Clear')
else
    % очищаем оси
    cla
    % изменяем надпись на кнопке на Plot
    set(src, 'String', 'Plot')
end
```

ПРИМЕР 5. Программирование GUI

GUI. Пример 5



При нажатии на кнопку Add button должна появляться в окне приложения кнопка Remove button.

При нажатии на кнопку Remove button эта кнопка должна исчезать.

GUI. Пример 5. Шаг 1

```
function simpleGuiNested
h.fig = figure('position', [800 30 210 60]);
h.buttonOne = uicontrol('style', 'pushbutton', ...
    'position', [10 10 100 40], ...
    'string', 'Add button', ...
    'callback', {@addButton});
```

GUI. Пример 5. Шаг 2

```
function addButton(hObject, eventdata)
h.buttonTwo = uicontrol('style', 'pushbutton', ...
    'position', [100 10 100 40], ...
    'string', 'Remove button', ...
    'callback', {@removeButton});
set(h.buttonOne, 'enable', 'off');
end
```

GUI. Пример 5. Шаг 3

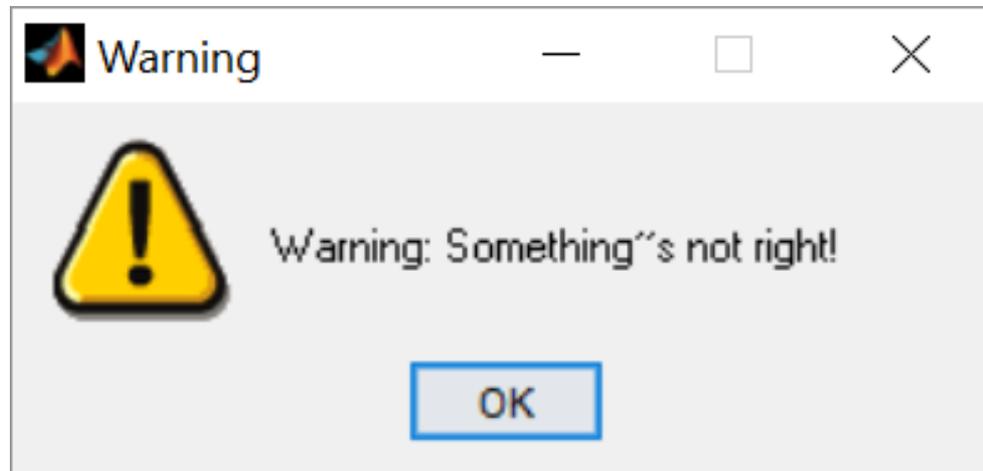
```
function removeButton(hObject, eventdata)
delete(h.buttonTwo)
h = rmfield(h, 'buttonTwo'); % удаляет ряд данных
set(h.buttonOne, 'enable', 'on')
end
```

ДИАЛОГОВЫЕ ОКНА

Пример **warndlg**

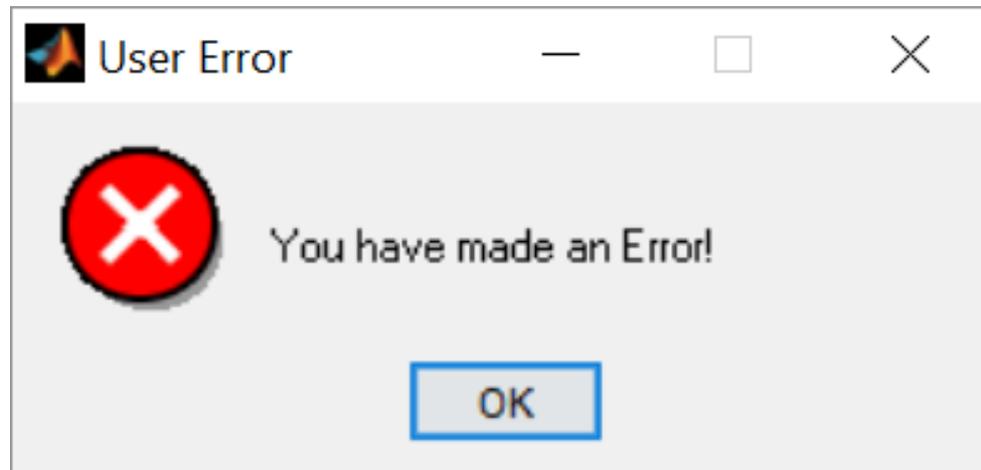
```
% Open warning dialog box
```

```
warnfig = warndlg('Warning: Something''s not  
right!', 'Warning')
```



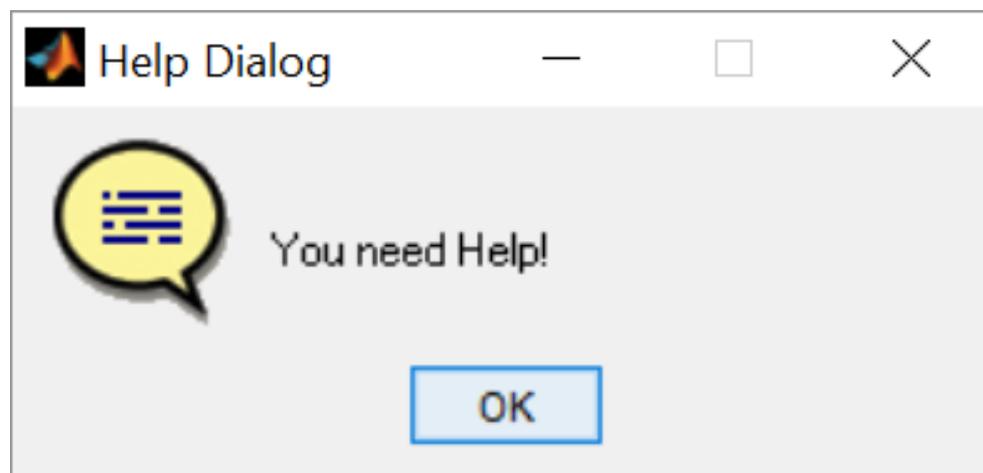
Пример `errordlg`

```
% Create and open error dialog box  
  
errfig = errordlg('You have made an  
Error!', 'User Error', 'replace')
```



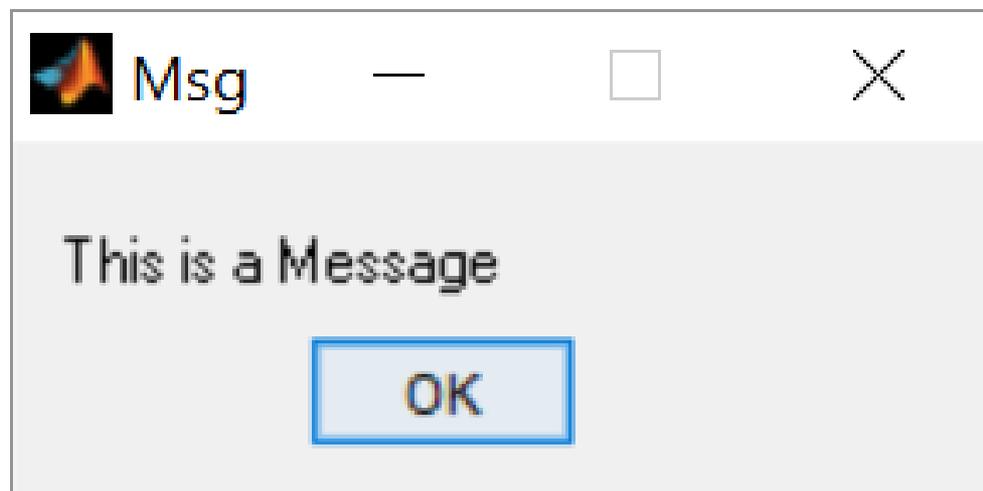
Пример `helpdlg`

```
helpfig = helpdlg('You need Help!');
```



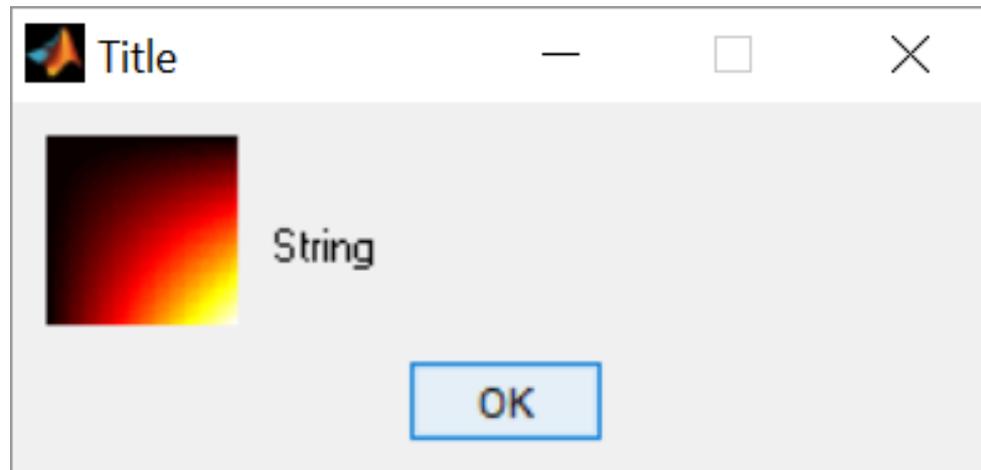
Пример `msgbox`

```
msgfig = msgbox('This is a Message', 'Msg');
```



Пример `msgbox`

```
Data=1:64;  
Data=(Data'*Data)/64;  
  
msgfig =msgbox('String', 'Title', ...  
              'custom',Data,hot(64))
```



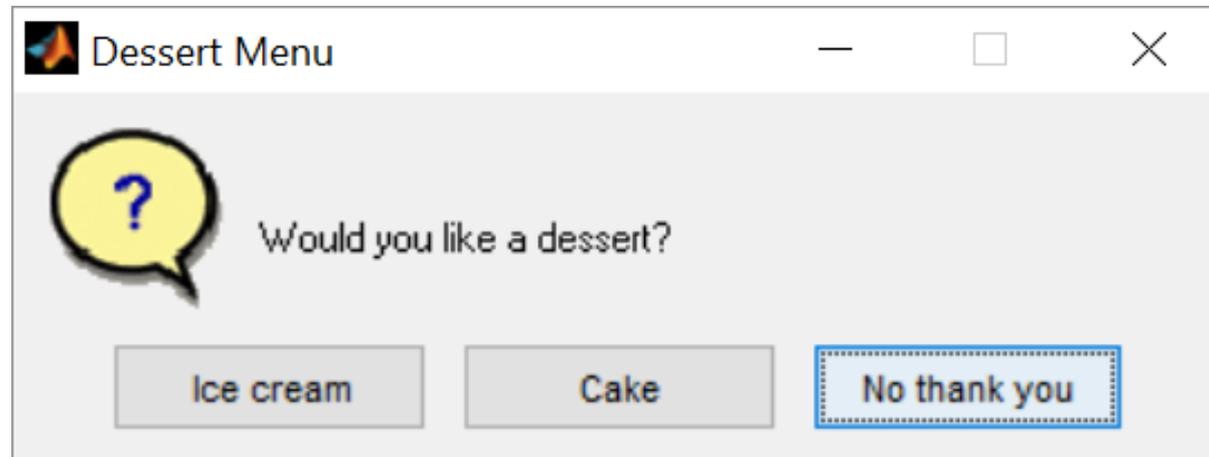
Пример `questdlg`

```
% Construct a questdlg with three options
choice = questdlg('Would you like a dessert?', ...
    'Dessert Menu', ...
    'Ice cream', 'Cake', 'No thank you', 'No thank you');

% Handle response
switch choice
    case 'Ice cream'
        disp([choice ' coming right up.'])
        dessert = 1;
    case 'Cake'
        disp([choice ' coming right up.'])
        dessert = 2;
    case 'No thank you'
        disp('I'll bring you your check.')
        dessert = 0;
end
```

I'll bring you your check.

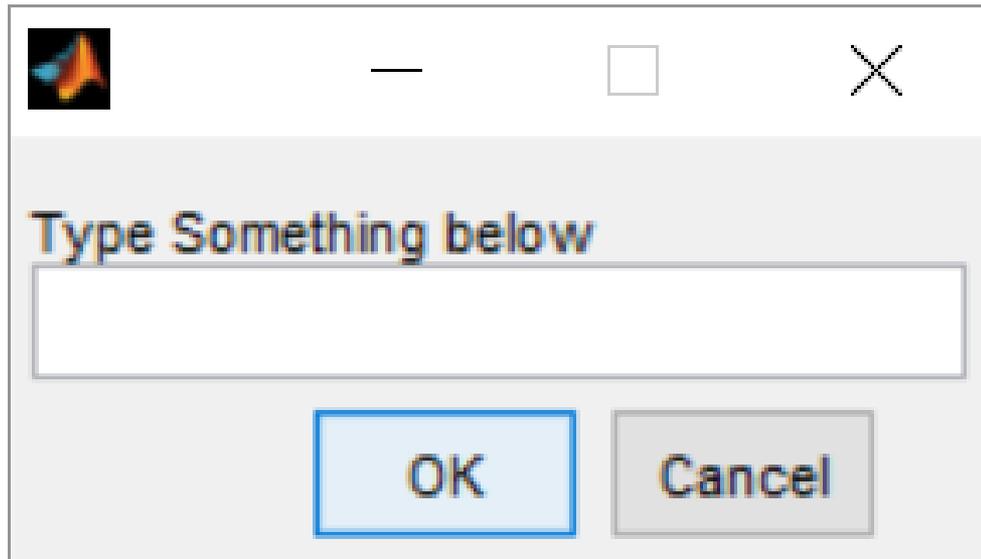
Пример `questdlg`



`I'll bring you your check.`

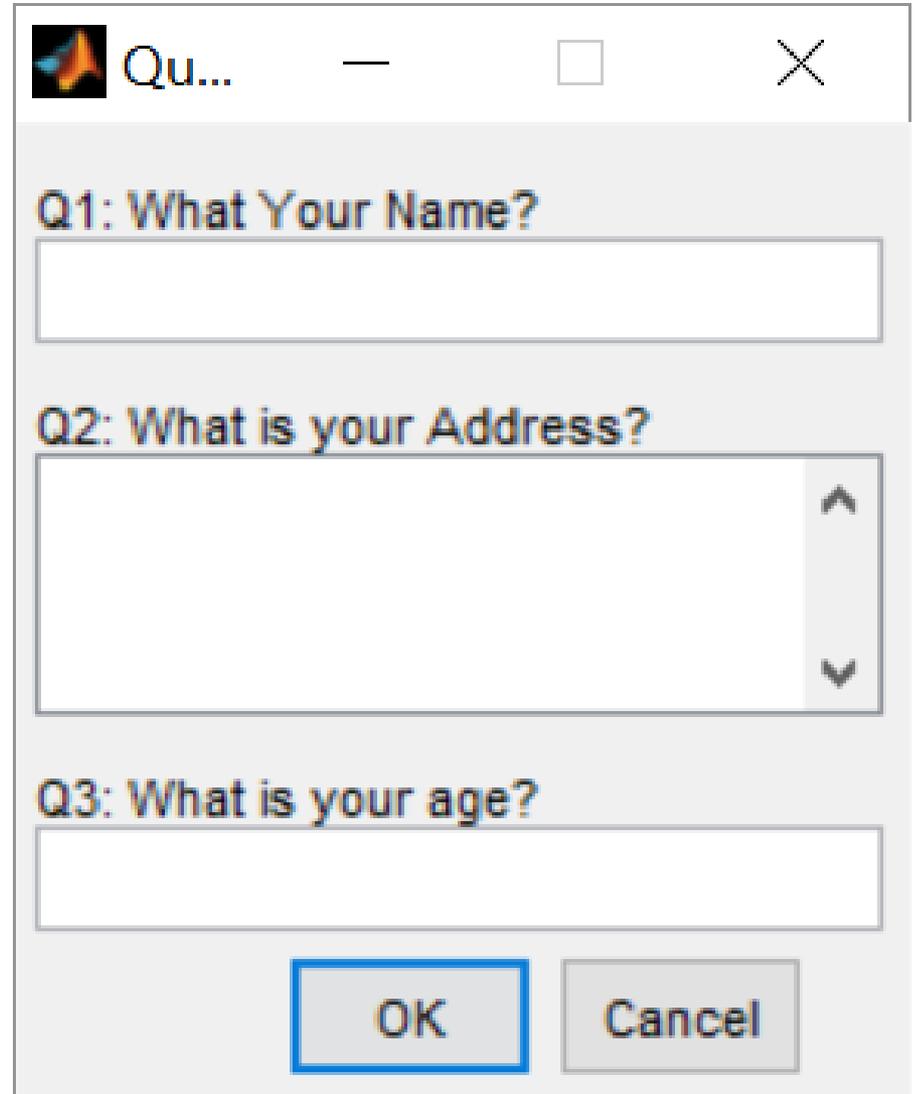
Пример `inputdlg`

```
Answers = inputdlg('Type Something below')
```



Пример `inputdlg`

```
Answers = inputdlg(  
{  
'Q1: What Your  
Name? ',  
  
'Q2: What is your  
Address? ',  
  
'Q3: What is your  
age? ' },  
  
'Questionnaire',  
  
[1 3 1]);
```

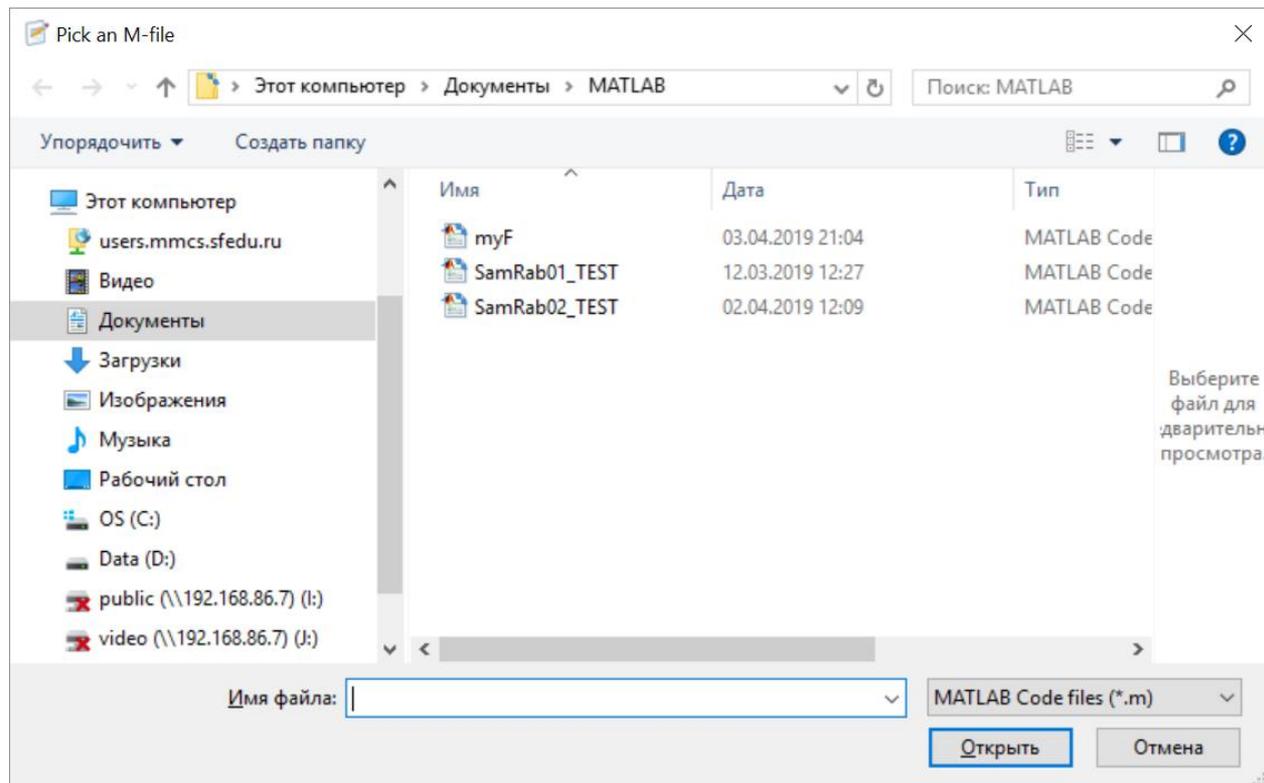


The screenshot shows a MATLAB dialog box titled "Questionnaire" with three input fields and two buttons at the bottom. The first field is labeled "Q1: What Your Name?" and is a standard text input box. The second field is labeled "Q2: What is your Address?" and is a text area with a vertical scrollbar on the right side. The third field is labeled "Q3: What is your age?" and is a standard text input box. The "OK" button is highlighted with a blue border, and the "Cancel" button is also visible.

Пример File/Directory Selection Dialog Boxes

```
[filename, pathname] =
```

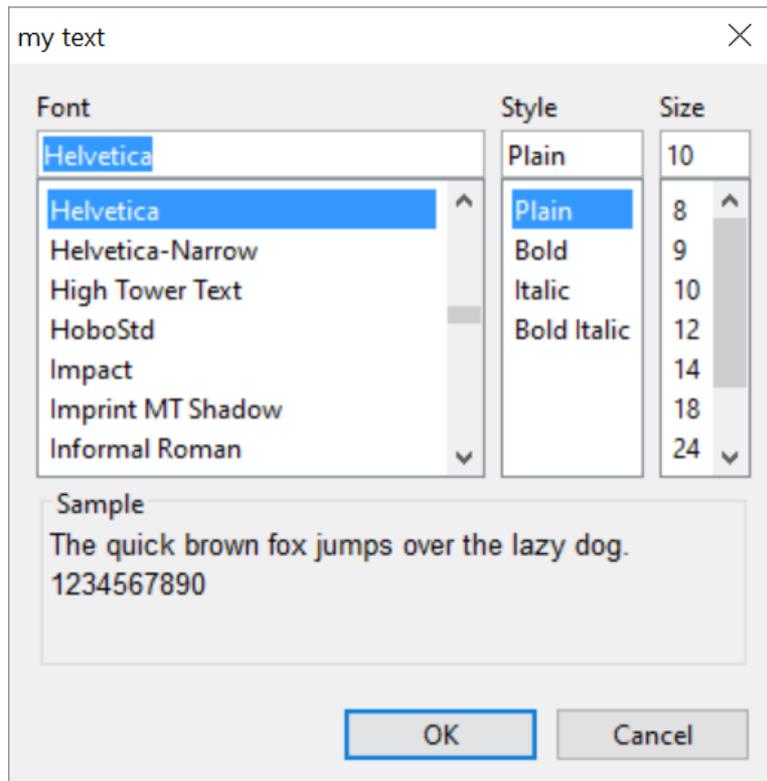
```
uigetfile('*.*', 'Pick an M-file');
```



Пример `uifont` / `uicolor`

```
uifont(xlbl, 'my text');
```

```
uicolor(myfig);
```





Спасибо за внимание!