Поволжский Государственный Университет Телекоммуникаций и Информатики

Факультет ИСТ

Кафедра ИВТ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам

"Программирование в системе MATLAB"

Использование GUI

Автор-составитель:	Акчурин Э.А.	д.т.н., профессор

Рецензент

Тарасов В.Н.

д.т.н., профессор

Самара

2012





Факультет информационных систем и технологий Кафедра «Информатика и вычислительная техника»



Автор - д.т.н., профессор Акчурин Э.А.

Другие материалы по дисциплине Вы найдете на сайте www.ivt.psati.ru

Содержание

1. Введение	3
2. Изучение GUIDE	4
3. Статический текст - Text	12
4. Редактор текста - Edit	14
5. Независимые переключатели – CheckBox	17
6. Зависимые переключатели – RadioButton	21
7. Выпадающее меню - Рорир Мепи	
8. Ползунок - Slider	

1. Введение

Лабораторный цикл содержит работы по изучению программирования графического интерфейса пользователя GUI с использованием математической системы MATLAB/

Средства графического пользовательского интерфейса (GUI – Graphic User Interface) предназначены для создания в МАТLAВ приложений с пользовательским интерфейсом. В этих приложениях присутствуют управляющие элементы, при изменениях пользователем свойств которых вызываются подпрограммы, выполняющие некоторые действия.

Для упрощения разработки GUI MATLAB включает инструмент GUIDE (Graphic User Interface Development Environment – среда разработки GUI).

Рекомендуемая литература:

- 1. Дьяконов В. МАТLAB 7.*/R2006/R2007: Самоучитель. М.: ДМК Пресс, 2008. 768 с.
- 2. Hunt. MATLAB R2007 с нуля! М.: Лучшие книги, 2008, 352 с.
- 3. Дьяконов В. Simulink 5/6/7: Самоучитель. М.: ДМК Пресс, 2008. 784 с.:

Содержание отчета по каждой работе:

- Название работы.
- Задание в соответствии с вариантом.
- Программа.
- Результаты выполнения программы на ПК.

2. Изучение GUIDE

Предметная область

Для упрощения разработки GUI MATLAB включает инструмент GUIDE (Graphic User Interface Development Environment – среда разработки GUI).

Пример выполнения

Для выбора инструмента используется команда <u>File=>New=>GUI</u>. Окно инструмента содержит поле, в котором размещаются компоненты. Доступные компоненты представлены в палитре, расположенной в правом поле.

🛃 u	ntitle	d.fig		C		•			1	-	-	1	 -	and the second			X
<u>F</u> ile	<u>E</u> di	t <u>V</u> ie	w	<u>L</u> ayo	ut	<u>T</u> oo	s j	<u>H</u> elp									
D																	
k																	<u>^</u>
OK	8008																
۲	Ø																
EDĮT	TXT																
-	E																
100 (T=	***		_		_							_				_	
	ŏ																
			-		_					 _		+					

Для удобства можно изменить вид панели компонент, показывая и их имена в.окне GUI Для этого в окне предпочтений нужно установить флаг Show names...

A Preferences		
-General	*	GUIDE Preferences
		Show names in component palette
Java Heap Memory		Show file extension in window title
⊞-Keyboard ⊟-Fonts	111	Show file path in window title
Custom ⊞Colors		Add comments for newly generated callback functions
Code Analyzer Toolbars		
-Command Window Command History		
⊞-Editor/Debugger Help		
Web Current Folder		
Variable Editor Workspace		
<mark>GUIDE</mark> Time Series Tools	Ŧ	
		OK Cancel Apply Help

Теперь окно редактора будет с именами еомпонент. При этом размер рабочего поля уменьшиться.

🗹 untitled5.	fig								X
Eile Edit View Layout Tools Help									
🗅 🖻 🖩 🕺 🖣	6 🖪 10	∝ ₽	i	🛐 🛃	₩ <u>4</u> ►				
🕨 Select									4
📧 Push Button									
🚥 Slider									
Radio Button									
Check Box									
💷 Edit Text									
The static Text									-
📼 Pop-up Menu									
El Listbox									-
Toggle Button									
Axes									 _
Pariel									
ActiveX Control									 _
A MEMOR CONTROL									
	<				1111			 	2

При работе с GUIDE компоненты GUI выбираются из палитры компонент инструмента GUIDE мышью и переносятся в рабочую область GUI. GUIDE сохраняет GUI в двух файлах:

- Рисунок, fig-файл.
- Програмный код (m-файл), управляющий работой GUI, генерируется автоматически при сохранении проекта.

После каждого редактирования оба файла изменяются. В листинге m-файла автоматически создаются строки исполняемого кода и комментарии (на английском языке). Пользователь должен добавить в код свою функциональность. Комментарии можно удалить, либо перевести на русский язык.

Внимание. В m-файле имеется фрагмент кода, который редактировать нельзя.

Для создания GUI в MATLAB используются следующие компоненты:

Компонент	Назначение
Push Button	Кнопка. При нажатии отображается нажатой, при отпускании генерирует действие и отображается отжатой.
Toggle Button	Переключаемая кнопка. При нажатии отображается нажатой, при отпускании генерирует действие, но остается нажатой.
Check Box	Независимый переключатель. Генерирует действие, когда нажат, и отображает включенное состояние птичкой в прямо- угольнике.
Radio Button	Зависимый переключатель (Радио кнопка). Генерирует дейст- вие, когда нажат, и отображает включенное состояние точкой в кружке. Похож на Check Box, но при использовании в группе при включении выключает остальные.
Edit Text	Редактор текста. Действие генерируется при нажатии клави- ши Enter.
Static Text	Статический текст.
Slider	Ползунок. Отображает численные значения в выбранном
	диапазоне с выбор значения перемещением ползунка.
ListBox	Список тем для выбора. Позволяет выбрать одну или не- сколько тем.
Pop-Up Menu	Выпадающее меню (иначе ComboBox). Подобен ListBox, но список открывается только при выборе.
Axes	Координатные оси. Используются для рисования графиков.
Panel	Панель. Это контейнер для группы компонент GUI. Панель используется для облегчения понимания назначения элемен- тов управления путем их визуального объединения.
Button Group	Группа кнопок. Подобна панели, но используется только для радиокнопок и переключаемых кнопок.
ActiveX Con-	Это элементы управления, разработанные Microsoft для опе-
trol	рационной системы Windows. При переносе этого компонента в рабочую область вызывается список ActiveX элементов для выбора.

Для создания GUI используются два метода:

- Команда guide в командном окне.
- Выбор в меню MATLAB команды File=>New=>GUI.

В обоих случаях генерируется диалоговое окно создания GUI, которое содержит две вкладки:

- Creating New GUI создание нового GUI.
- Open Existing GUI открыть существующий GUI.

На вкладке создания нового GUI есть две панели: слева список шаблонов для выбора, справа предварительный просмотр. Можно выбрать один из предлагаемых шаблонов:

- Blank GUI (Default). Пустой GUI (по умолчанию).
- GUI with Uicontrols. GUI с элементами управления пользовательским интерфейсом.
- GUI with Axes and Menu. GUI с объектом Axes (координатные оси) и меню.
- Modal Question Dialog. Модальное окно диалога с обязательным ответом на вопрос.

GUIDE Quick Start									
Create New GUI Open Existing GUI									
GUIDE templates	Preview								
 Blank GUI (Default) GUI with Uicontrols GUI with Axes and Menu Modal Question Dialog 	BLANK								
Save new figure as: C:\Pro	gram Files (x86)\MATLAB\R2010b\bin Browse								
	OK Cancel Help								



GUIDE Quick Start	
GUIDE templates Blank GUI (Default) GUI with Uicontrols GUI with Axes and Menu Modal Question Dialog	Preview ? Do you want to create a question dialog? Yes No
Save new figure as: C:\Pro	gram Files (x86)\MATLAB\R2010b\bin Browse OK Cancel Help

Вкладка открытия существующего GUI включает список имеющихся объектов GUI, если они есть. Кнопка Browse (Просмотреть) открывает стандартный диалог открытия файла.

J GUIDE Quick Start
Create New GUI Open Existing GUI
Recently opened files:
 ↓ <u>IVVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\ListBoxGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\EditTextGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\StaticTextGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\PopupMenuGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\RadioButtonGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\RadioButtonGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\SiderGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\SiderGUL.fig</u> ↓ <u>IVMK Акчурин\ЧМВ\ЛПЗ_ЧМВ\LR_MATLAB_GUL_Examples\CheckBox_GUL.fig</u>
< •
Browse
Open Cancel Help

Команда <u>File=>Preferences</u> вызывает окно предпочтений MATLAB, в котором будут отображены параметры GUI по умолчанию. В нем можно, например, скорректировать размер шрифта.

A Preferences	1.0	A REAL PROPERTY AND A REAL PROPERTY.		
General MAT-Files Confirmation Dialogs Source Central	*	Fonts Preferences Desktop code font Construction Construct Vision Construct Vision Filter		
Java Heap Memory Java Heap Memory Seyboard Custom Custom Colors Colors		Monospaced Sample The quick brown fox jumps over the lazy dog. 1234567	90	
-Coolbars Command Window Command History @-Editor/Debugger Help Meb	Е	Desktop text font Currently used by: Help Navigator, Current Folder, Workspace, Variable Editor, Fu Use system font Senoe UI	nction Browser	9 🗸
		Sample The quick brown fox jumps over the lazy dog. 1234567890		
Figure Copy Template Compiler Report Generator SystemTest Bioinformatics Tools Database Toolbox Image Acquisition		Custom fonts Currently used by: HTML Proportional Text To set a custom font for any desktop tool, go to the <u>Custom Fonts</u> preferences.		
Instrument Control	Ŧ	OK Cancel	Apply	Help

3. Статический текст - Text

Предметная область

Статический текст. Используется для меток и для отображения значений компонента Slider.

Задание

Создать GUI с компонентами статического текста, в которые командой кнопки PushButton вывести фамилию, имя, отчество студента и желаемую оценку работы.

Пример выполнения

Окно интерфейса:

🛒 StaticTe	xtGUI.fig		-		x	
<u>F</u> ile <u>E</u> dit	<u>V</u> iew <u>L</u> a	yout <u>T</u> o	ols <u>H</u> elp			
1 🖆 🛃	X 🖻 🕯	90	₿ 🛃	P 🔬 🛛	🛃 👋	
					Â	
		Static Text		Static Text	=	
		Static Text				
		Static Text		Push Button		
	•					
Current Point: [300, 175] Position: [520, 556, 300, 245]						

Листинг кода

```
function varargout = StaticTextGUI(varargin)
% Начало инициализации, не редактировать
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
```

```
'gui_OpeningFcn', @StaticTextGUI_OpeningFcn, ...
'gui_OutputFcn', @StaticTextGUI_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% Конец инициализации, не редактировать
```

function StaticTextGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = StaticTextGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```
function pushbutton1_Callback(hObject, eventdata, handles)
set(handles.text1,'string','Akchurin');
set(handles.text2,'string','Eduard');
set(handles.text3,'string','Aleksandrovich');
set(handles.text4,'string',5);
```

StaticTextGUI	StaticTextGUI
Static Text Static Text	Akchurin 5
Static Text	Eduard
Push Button Static Text	Push Button Aleksandrovich

4. Редактор текста - Edit

Предметная область

Редактор текста. Используется для ввода и редактирования строк текста. Действие генерируется при нажатии клавиши Enter для однострочного текста и Ctl+Enter для многострочного текста.

Задание

Создать GUI с компонентом редактора текста, в который при прогоне ввести фамилию и имя студента. Введенные данные перемещать в компонент статического текста, для которого предусмотреть заголовок с использованием дополнительного компонента статического текста.

Пример выполнения

Окно интерфейса:

🛃 EditText	GUI.fig					X	
<u>F</u> ile <u>E</u> dit	<u>V</u> iew	<u>L</u> ayout	<u>T</u> ools	<u>H</u> elp			
1 🖆 📕	X 🖣	i 🛍 🤊	(で) 串	🎽 🛓	b 💰 🖻] 🛃	»
							Â
		Family +	Name ->	Sta	itic Text		
							=
		Edit	Text				
li 🕅	4					4	Ŧ
. Current Point: [258, 197] Position: [520, 580, 299, 226]							

```
Листинг кода
```

```
function varargout = EditTextGUI(varargin)
% Начало инициализации, не редактировать
qui Singleton = 1;
gui_State = struct('gui_Name',
                                 mfilename....
            'gui_Singleton', gui_Singleton, ...
           'gui_OpeningFcn', @EditTextGUI_OpeningFcn, ...
           'gui_OutputFcn', @EditTextGUI_OutputFcn, ...
           'gui LayoutFcn', [], ...
           'qui Callback', []);
if nargin && ischar(varargin{1})
  gui State.gui Callback = str2func(varargin{1});
end
if nargout
  [varargout{1:nargout}] = gui mainfcn(gui State, varargin{:});
else
  gui_mainfcn(gui_State, varargin{:});
end
% Конец инициализации, не редактировать
```

```
function EditTextGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
```

```
function varargout = EditTextGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

```
function edit1_Callback(hObject, eventdata, handles)
s=get(handles.edit1,'string');
set(handles.text2,'string',s);
```

```
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

ditTextGUI		EditTextGUI	
Family + Name ->	Static Text	Family + Name ->	Static Text
Edit Text		Akchurin Eduard	
	EditToutCLIT		



5. Независимые переключатели – CheckBox

Предметная область

Независимый переключатель. Генерирует действие, когда нажат, и отображает включенное состояние птичкой в прямоугольнике. Используется в группах независимых переключателей для установки свойств объектов.

Задание

Создать GUI с двумя независимыми переключателями, которые вызывают рисование в графическом поле axes графиков функций с разным числом периодов.

N⁰	Число периодов	Функция 1	Функция 2	
0	4	sin(x)	cos(x)	
1.	3	abs(sin(x))	abs(cos(x))	
2.	4	sin(x)*sin(x)	$\cos(x)^*\cos(x)$	
3.	2	sin(x)*cos(x)	$\cos(x)^*\cos(x)$	
4.	3	sin(x)	abs(cos(x))	
5.	4	abs(sin(x))	cos(x)	
6.	2	sin(x)*sin(x)	$\cos(x)^*\cos(x)$	
7.	3	sin(x)*cos(x)	$\cos(x)^*\cos(x)$	
8.	4	sin(x)	$\cos(x)^*\cos(x)$	
9.	3	abs(sin(x))	sin(x)*sin(x)	

Варианты заданий

Пример выполнения

Создать GUI с двумя независимыми переключателями, которые вызывают рисование в графическом поле axes графиков функций sin(x) и cos(x) с двумя периодами.

Окно интерфейса:



Листинг кода

function varargout = CheckBox_GUI(varargin) % Начало инициализации, не редактировать gui_Singleton = 1; gui_State = struct('gui_Name', mfilename, ... 'gui_Singleton', gui_Singleton, ... 'gui_OpeningFcn', @CheckBox_GUI_OpeningFcn, ... 'gui_OutputFcn', @CheckBox_GUI_OutputFcn, ... 'gui_LayoutFcn', [], ... 'gui_Callback', []); if nargin && ischar(varargin{1})

```
gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% Конец инициализации, не редактировать
```

function CheckBox_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
guidata(hObject, handles);

function varargout = CheckBox_GUI_OutputFcn(hObject, eventdata, handles)

function checkbox1_Callback(hObject, eventdata, handles)

function checkbox2_Callback(hObject, eventdata, handles)

```
function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
cla;
x=[0:0.01:4*pi];y1=sin(x);y2=cos(x);
a1=get(handles.checkbox1,'value');
a2=get(handles.checkbox2,'value');
if a1==1 plot(x,y1);grid on;
end
if a2==1 plot(x,y2);grid on;
end
if a1*a2==1 plot(x,y1,x,y2);grid on;
end
```



6. Зависимые переключатели – RadioButton

Предметная область

Зависимый переключатель. Генерирует действие, когда нажат, и отображает включенное состояние точкой в кружке. Похож на Check Box, но при использовании в группе при включении выключает остальные.

Задание

Создать GUI с двумя радиокнопками, которые вызывают рисование в графическом поле axes графиков функций с разным числом периодов.

N⁰	Число периодов	Функция 1	Функция 2
0	4	sin(x)	cos(x)
1.	3	abs(sin(x))	abs(cos(x))
2.	4	sin(x)*sin(x)	$\cos(x)^*\cos(x)$
3.	2	sin(x)*cos(x)	$\cos(x)^*\cos(x)$
4.	3	sin(x)	abs(cos(x))
5.	4	abs(sin(x))	cos(x)
6.	2	sin(x)*sin(x)	$\cos(x)^*\cos(x)$
7.	3	sin(x)*cos(x)	$\cos(x)^*\cos(x)$
8.	4	sin(x)	$\cos(x)^*\cos(x)$
9.	3	abs(sin(x))	sin(x)*sin(x)

Варианты заданий

Пример выполнения

Создать GUI с двумя радиокнопками, которые вызывают рисование в графическом поле axes графиков функций sin(x) и cos(x) с двумя периодами. Окно интерфейса:



Листинг кода

function varargout = RadioButtonGUI(varargin) % Начало инициализации, не редактировать gui_Singleton = 1; gui_Singleton = 1; gui_Singleton', gui_Singleton, ... 'gui_OpeningFcn', @RadioButtonGUI_OpeningFcn, ... 'gui_OutputFcn', @RadioButtonGUI_OutputFcn, ... 'gui_LayoutFcn', [], ... 'gui_Callback', []); if nargin && ischar(varargin{1}) gui_State.gui_Callback = str2func(varargin{1});

end

```
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% Конец инициализации, не редактировать
```

function RadioButtonGUI_OpeningFcn(hObject, eventdata, handles, varargin) guidata(hObject, handles);

```
function varargout = RadioButtonGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1);
x=[0:0.01:4*pi];
if get(handles.radiobutton1,'value')==1 y=sin(x);
elseif get(handles.radiobutton2,'value')==1 y=cos(x);
end;
plot(x,y);
grid on;
```



7. Выпадающее меню - Popup Menu

Предметная область

Компоненты со списками:

- <u>ListBox</u> список тем для выбора. Позволяет выбрать одну или несколько тем.
- <u>РорирМепи</u> выпадающее меню (иначе ComboBox). Подобен ListBox, но список открывается только при выборе. Используется для экономии места.

Задание

Создать GUI с выпадающим меню, которое содержит список выбора функции с разным числом периодов, которая затем отображается в графическом поле axes графиков функций кнопкой обновления.

N⁰	Число периодов	Функция 1	Функция 2	
0	4	sin(x)	cos(x)	
1.	3	abs(sin(x))	abs(cos(x))	
2.	4	sin(x)*sin(x)	$\cos(x)^*\cos(x)$	
3.	2	sin(x)*cos(x)	$\cos(x)^*\cos(x)$	
4.	3	sin(x)	abs(cos(x))	
5.	4	abs(sin(x))	cos(x)	
6.	2	sin(x)*sin(x)	cos(x)*cos(x)	
7.	3	sin(x)*cos(x)	cos(x)*cos(x)	
8.	4	sin(x)	$\cos(x)^*\cos(x)$	
9.	3	abs(sin(x))	sin(x)*sin(x)	

Варианты заданий

Пример выполнения

Создать GUI с выпадающим меню, которое содержит список выбора функции с разным числом периодов, которая затем отображается в графическом поле ахеs графиков функций кнопкой обновления. Список содержит 4 функции: sin(x), cos(x), abs(sin(x)), - abs(sin(x)).

Окно интерфейса:



Листинг кода

```
function varargout = PopupMenuGUI(varargin)
% Начало инициализации, не редактировать
gui_Singleton = 1;
gui_Singleton = 1;
gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @PopupMenuGUI_OpeningFcn, ...
'gui_OutputFcn', @PopupMenuGUI_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
[varargout{1:nargout}] = gui_mainfcn(gui_State.varargin{:});
```

```
else
gui_mainfcn(gui_State, varargin{:});
end
% Конец инициализации, не редактировать
```

```
function PopupMenuGUI_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
if strcmp(get(hObject,'Visible'),'off')plot(sin([0:0.01:4*pi]));grid on;
end
```

```
function varargout = PopupMenuGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
axes(handles.axes1); cla;
popup_sel_index = get(handles.popupmenu1, 'Value');
switch popup_sel_index
    case 1
    plot(sin([0:0.01:4*pi]));grid on;
    case 2
    plot(cos([0:0.01:4*pi]));grid on;
    case 3
    plot(abs(sin([0:0.01:4*pi])));grid on;
    case 4
```

```
plot(-abs(sin([0:0.01:4*pi])));grid on;
```

```
end
```

```
function FileMenu_Callback(hObject, eventdata, handles)
```

```
function OpenMenuItem_Callback(hObject, eventdata, handles)
file = uigetfile('*.fig');
if ~isequal(file, 0)open(file);
end
```

```
function PrintMenuItem_Callback(hObject, eventdata, handles)
printdlg(handles.figure1)
```

```
function CloseMenuItem_Callback(hObject, eventdata, handles) selection = questdlg(['Close ' get(handles.figure1,'Name') '?'],...
```

```
['Close ' get(handles.figure1,'Name') '...'],'Yes','No','Yes');
if strcmp(selection,'No')return;
end
delete(handles.figure1)
```

function popupmenu1_Callback(hObject, eventdata, handles)

```
function popupmenu1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```



8. Ползунок - Slider

Предметная область

Ползунок. Отображает численные значения в выбранном диапазоне с выбором значения перемещением ползунка. Выбранное значение отображается в компоненте Static Text.

Задание

Создать GUI с ползунком, имеющим заданные значения минимума и максимума. После перемещения ползунка командой от кнопки вывести значения минимума, максимума и текущего в блоки статического текста.

Варианты заданий

N⁰	Минимум	Максимум
0	0	50
1.	5	70
2.	10	120
3.	15	130
4.	20	200
5.	-5	60
6.	-10	80
7.	-15	110
8.	-20	135
9.	-25	190

Пример выполнения

Создать GUI с ползунком, имеющим значения минимума 0 и максимума 100. После перемещения ползунка командой от кнопки PushButton вывести значения минимума, максимума и текущего положения ползунка в блоки статического текста.

Окно интерфейса:



Листинг кода

```
function varargout = SliderGUI(varargin)
% Начало инициализации, не редактировать
gui_Singleton = 1;
gui_Singleton = 1;
gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @SliderGUI_OpeningFcn, ...
'gui_OutputFcn', @SliderGUI_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
```

% Конец инициализации, не редактировать

function SliderGUI_OpeningFcn(hObject, eventdata, handles, varargin)
guidata(hObject, handles);

function varargout = SliderGUI_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```
function slider1_Callback(hObject, eventdata, handles)
```

```
function slider1_CreateFcn(hObject, eventdata, handles)
if isequal(get(hObject, 'BackgroundColor'), ...
get(0,'defaultUicontrolBackgroundColor'))
set(hObject, 'BackgroundColor',[.9 .9 .9]);
end
```

```
function pushbutton1_Callback(hObject, eventdata, handles)
min=get(handles.slider1,'Min');
set(handles.text1,'String',min);
max=get(handles.slider1,'Max');
set(handles.text3,'String',max);
val=get(handles.slider1,'Value');
set(handles.text2,'String',val);
```



🛃 SliderGUI			🚺 SliderGUI		
	100			100	
46.789	• •	Push Button	100	0	Push Button