

Работа в среде GUIDE. Стандартные диалоговые окна

В системе MATLAB предусмотрены "стандартные" диалоговые графические окна, предназначенные для вывода различных сообщений, чтения и записи файлов, выбора цвета и др. Перечень стандартных диалоговых окон приведен в табл. 4.1.

Табл. 4.1.

Наименование диалогового окна	Назначение диалогового окна
dialog	Создание и отображение окна общего назначения
errordlg	Создание и отображение диалогового окна с сообщением об ошибке
helpdlg	Создание и отображение диалогового окна помощи
msgbox	Создание и отображение диалогового окна с сообщением
inputdlg	Создание и отображение диалогового окна ввода
listdlg	Создание и отображение диалогового окна со списком для выбора
pagedlg	Отображение диалогового окна для настройки параметров страницы
printdlg	Отображение диалогового окна для настройки параметров печати
questdlg	Создание и отображение диалогового окна с запросом
uigetfile	Отображение диалогового окна для извлечения имени файла для чтения
uiputfile	Отображение диалогового окна для извлечения имени файла для записи
uigetcolor	Отображение диалогового окна для установки цвета
uigetfont	Отображение диалогового окна для установки параметров шрифта
warndlg	Создание и отображение диалогового окна с предупреждением

Окно с сообщением об ошибке вызывается командой **errordlg**. Варианты синтаксиса:

```
errordlg;  
errordlg('Information','Name');  
errordlg('Information','Name','on').
```

При выполнении команды **errordlg('Information','Name')** будет выведено окно с именем Name, текстом Information и кнопкой Ok (рис. 4.1).

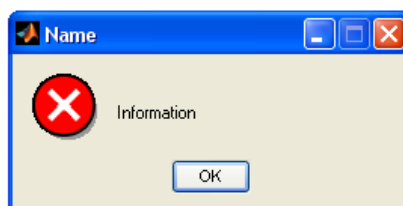


Рис. 4.1.

При наличии в команде аргумента **'on'** вывод сообщения производится на старое окно, если оно есть. Закрытие окна осуществляется обычным способом. Пауза после вывода окна не предусмотрена.

Окно со справочной информацией вызывается командой **helpdlg**. Варианты синтаксиса:

```
helpdlg;  
helpdlg('Information');  
helpdlg('Information','Name').
```

После выполнения последней команды будет выведено окно, изображенное на рис. 4.2.

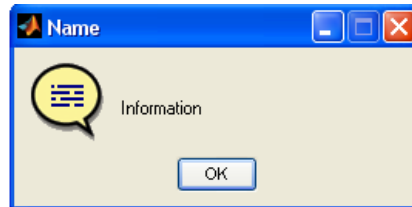


Рис. 4.2.

Закрытие окна осуществляется обычным способом. Пауза после вывода окна не предусмотрена. Это окно отличается от окна с сообщением об ошибке лишь дизайном.

Окно с предупреждением. Это окно вызывается командой **warndlg** и отличается от окна со справочной информацией и от окна с сообщением об ошибке лишь иконкой. После выполнения команды

```
warndlg('Information','Name')
```

будет выведено окно, изображенное на рис. 4.3.

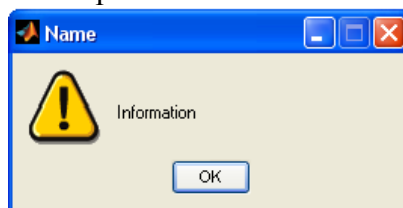


Рис. 4.3.

Диалоговое окно с сообщением. В этом диалоговом окне текст может сопровождаться иконкой (рис. 4.4). Вызов окна осуществляется функцией **msgbox**:

```
msgbox(Text, Name, Icon),
```

где строковая переменная **Text** содержит текст сообщения, **Name** – наименование сообщения, **Icon** – имя выводимого значка. **Icon** может принимать значения **'none'** (иконка отсутствует), **'error'**, **'help'**, или **'warn'** (выводятся иконки окна сообщения об ошибке, окна помощи или предупреждения). Предусмотрен и вариант вывода иконки пользователя.



'error'



'help'



'warn'

Рис. 4.4.

Пример: дополнение примера с функциями кнопкой с сообщением

```
uicontrol('Style','pushbutton',...  
         'String','Помощь',...
```

```

'Units','pixels',...
'Position',[260 10 50 20],...
'Callback',@btnHelp_Callback);

function btnHelp_Callback (hObject, eventdata)
% функция обработки события Callback кнопки Помощь (через массив
ячеек)
с = {'Использование функции Fplot'; 'Переменная x: [0 10]'};
h = msgbox(с, 'Сообщение', 'warn')

```

Окно для ввода строк. Для ввода текстовой информации предусмотрено стандартное диалоговое окно, вызываемое функцией **inputdlg**. Функция **inputdlg** создает диалоговое окно с заданным числом строк ввода. Число строк ввода и заголовки к ним определяются при помощи входных аргументов. Окно содержит кнопки *OK* и *Cancel*. Если пользователь нажал *OK*, то выходной аргумент функции **inputdlg** является массивом ячеек с информацией, введенной пользователем в строки ввода. Если окно было закрыто нажатием на кнопку *Cancel* или любым другим способом (кроме кнопки *OK*), то выходной аргумент - пустой массив ячеек.

Синтаксис:

```
Num=inputdlg({'N1',...,'Nk'}, 'Name', lines, def_res, 'flag').
```

Выполнение такой команды приводит к появлению окна с именем *Name* с *k* полями для ввода (рис. 4.5), над каждым из которых будет подпись *Ni* (*i*=1,...,*k*). Каждое поле имеет *lines* строк для ввода данных, в которые при обращении к **inputdlg** выводится содержимое вектора *def_res* (по умолчанию пустая строка). Флаг '*flag*' может принимать значения '*on*' или '*off*'. В первом случае можно изменять размеры окна, а при флаге в положении '*off*' такая возможность отсутствует.

После ввода данных в поля ввода и нажатия *Ok* окно исчезнет, и в вектор *Num* будут переданы набранные данные.

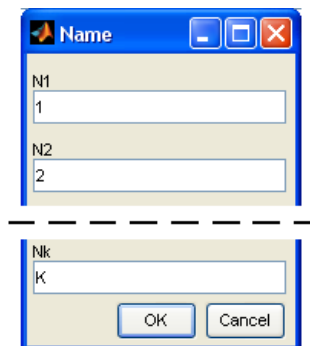


Рис. 4.5.

Для занесения в поле ввода значения числа, содержащегося в переменной *Num_Old*, необходимо это число перевести в строковый формат с помощью функции **num2str**:

```
def_res = num2str(Num_Old).
```

В этом случае при выполнении команды

```
New = inputdlg({'N1'}, 'Name', lines, def_res, 'on')
```

в поле ввода будет прописано числовое содержимое *Num_Old*. После его корректировки, нажатия на *Ok* и выполнения команды

```
Num_Old = str2num(char(New))
```

Num_Old примет новое значение, введенное в поле ввода *N1* окна функции **inputdlg**.

Аналогичная процедура для текстовой переменной реализуется с помощью функции **char**:

```
def_res = char(Num_Old);
```

```
New = inputdlg({'N1'}, 'Name', lines, def_res, 'on');
Num_Old = char(New).
```

Пример:

```
Lines = 1;
A = pi;
def_res = {'B-1', 'B-2', num2str(A)};
New = inputdlg({'Bar_1', 'Bar_2', 'Bar_3'}, 'Input_Data', lines, def_res, 'on');
```

В результате действия этих команд появится окно Input_Data с заполненными тремя полями для ввода данных (рис. 4.6). После изменения этих данных и нажатия Ok вектор New будет содержать данные, введенные в поля для ввода. После выполнения команды

```
New_var = [char(New(1));char(New(2));str2num(char(New(3)))]
```

New_var примет новое значение.

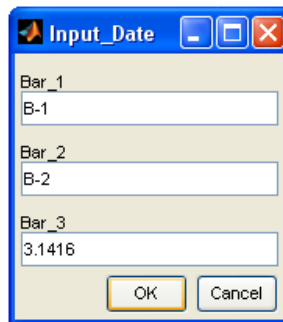


Рис. 4.6.

Пример: создание кнопки Диапазон и связывание с ее событием Callback функции btnData:

```
uicontrol('Style','pushbutton',...
    'String','Диапазон',...
    'Units','pixels',...
    'Position',[190 10 60 20],...
    'Callback',@btnData_Callback);

function btnData_Callback (hObject, eventdata)
% получение структуры указателей на объекты приложения
handles = guihandles(hObject);
% делаем оси текущими
axes(handles.axMain)
% очищаем оси
cla
% функция обработки события Callback кнопки Помощь
M = inputdlg({'Функция', 'Диапазон'});
[x,y]=fplot(M{1},str2num(M{2}));
```

Полоса прогресса. Полоса прогресса создается с помощью функции **waitbar**:

```
Name = waitbar(z, 'Text', Property_Name, Property_Value).
```

Функция формирует окно типа figure с полосой (рис. 4.7), над которой прописывается текст Text. Число z от нуля до единицы указывает, какая доля полосы должна быть закрашена. В аргументах Property_Name и Property_Value соответственно указываются свойства фигуры и их значения (табл. 2.7). Так, например, после выполнения команды

```
W = waitbar(20/100, 'Wait', 'Color', 'g', 'Name',...
    'Single_process', 'Position', [100,100,300,60])
```

в нижней левой части экрана появится окно, изображенное на рис. 4.7, размерами 300×60 пикселей. В верхней части окна будет отображаться текст, указанный в переменной `Single-process`, а над полосой – надпись `Wait`. 1/5 часть полосы будет закрашена в красный цвет, а область вокруг полосы – в зеленый. После выполнения команды

```
waitbar(70/100, W, 'Almost there')
```

полоса, указатель на которую `W`, будет закрашена на 7/10 своей длины, а над ней пропишется текст, указанный в переменной `Almost there`.



Рис. 4.7.

Диалоговое окно с запросом. Окно с тремя кнопками (по умолчанию Yes, No, Cancel) создается функцией `questdlg` (рис. 4.8):

```
button = questdlg('Question', 'Title', 'N1', 'N2', 'N3', 'default').
```

`button` принимает значения N1, N2 или N3 при нажатии одноименной кнопки и `button=[]`, если нажата кнопка закрытия окна.

Пример: использование «Окна с запросом»

```
% четвертый параметр – выбор по "Enter"
```

```
button=questdlg('Очистить оси', 'График', 'Да', 'Нет', 'Да')
```

Окно для выбора директорий. Для просмотра и выбора директорий предназначена функция `uigetdir`:

```
directory = uigetdir('path', 'Text'),
```

где `path` – директория по умолчанию (отмечена в окне выбора директории), `Text` – текст, выводимый в окно функции `uigetdir` (рис. 4.9).

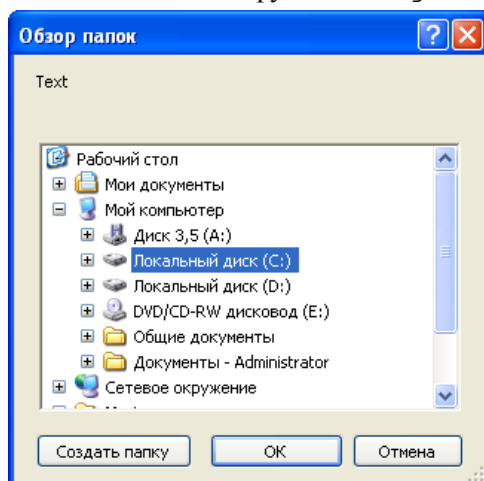


Рис. 4.9.

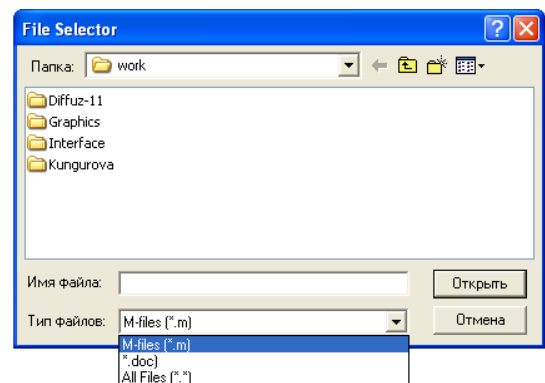


Рис. 4.10.

Переменная `directory` принимает значение выбранной директории, если нажата кнопка `OK`, и 0, если нажата кнопка `Close` или кнопка закрытия окна. Рис. 4.9 соответствует команда

```
directory = uigetdir('C:\', 'Text').
```

Окно для чтения имени файла вызывается функцией **uigetfile** (рис. 4.10):

```
[FileID, PathID, FilterIndex] = uigetfile('Filter', ...  
'Title', 'Name_Path'),
```

где Title – пропись в верхней части окна, Filter – список расширений файлов.

Например,

```
Filter = {'*.doc', '*.m', '.*', ...} (рис. 4.10).
```

Name_Path – название директории, в которой ищется файл (текущая директория по умолчанию). После выбора файла и нажатия кнопки *Open* в переменной FileID будет содержаться имя выбранного файла, в PathID – путь, в FilterIndex – порядковый номер расширения файла в списке расширений.

Рис. 4.10 соответствуют команды

```
Filter = {'*.doc', '*.m', '.*'};  
[FileID, PathID] = uigetfile(Filter, 'FileSelector')
```

Окно для записи файлов. Окно для записи файлов вызывается функцией **uiputfile**. Обращение к ней не отличается от обращения к функции **uigetfile**. В отличие от функции **uigetfile** при выборе имени файла можно указать несуществующее имя.

Окно для выбора цвета. Выбор цвета осуществляется функцией **uicontrol**:

```
Color = uicontrol(Name, 'Palitre'),
```

где Name – триплет по шкале RGB или указатель на него, а Palitre – текст, который выводится в окно палитры цветов (рис. 4.11), появляющееся при обращении к функции **uicontrol**. Окно с цветом, соответствующим триплету Name, "высвечивается" на палитре цветов. После выбора цвета и нажатия Ok вектору Color будут возвращены три числа, соответствующие выбранному цвету по шкале RGB.

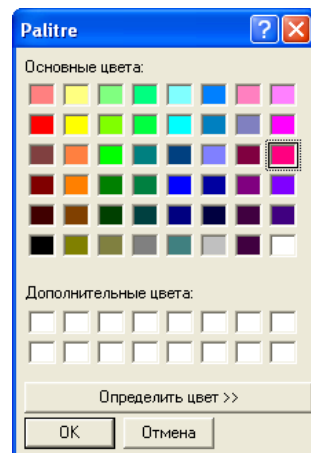


Рис. 4.11.

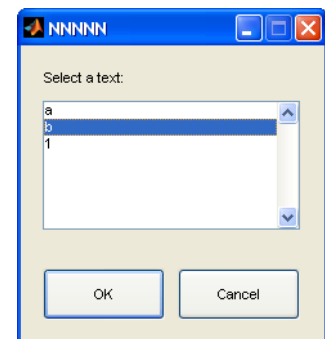


Рис. 4.12.

Диалоговое окно со списком для выбора. Диалоговое окно со списком, из которого можно выбрать какой-нибудь элемент, создается функцией **listdlg** (рис. 4.12):

```
[Selection, Res] = listdlg('ListString', S, 'Parametr', Value,  
...),
```

где S – список, отображаемый в окне функции **listdlg**. Параметры Parametr и их значения приведены в табл. 4.2.

Если нажата кнопка Cancel или кнопка закрытия окна, Res принимает значение 0, а Selection = []. После выбора элемента списка и нажатия кнопки Ok, Res принимает значение 1, а Selection – номер выбранной строки списка (для случая

SelectionMode = single) или номера выбранных строк списка (для случая SelectionMode = multiple).

Функция **listdlg** допускает обращение к ней в общей форме:

```
[Selection, ok] = listdlg('ListString', массив ячеек строк,  
    парам1, знач1, парам2, знач2, ...)
```

Рис. 4.12 соответствуют следующие команды:

```
str = {'a', 'b', '1'};  
[s,v] = listdlg('PromptString', 'Select Text: ', 'Name',  
'NNNNN', ...  
'SelectionMode', 'single', 'InitialValue', 2, ...  
'ListSize', [200 100], 'uh', 40, 'fus', 10, ...  
'CancelString', 'Cancel', 'OKString', 'OK', ...  
'ListString', str).
```

После выполнения этих команд будет выведено окно размерами 200×100 пикселей, с именем NNNNN и подписью 'Select Text: ' над полем для выбора строк. Выбор предусмотрен из трех строк с текстами a, b или 1 (listdlg(...'ListString', str)). В списке строк "по умолчанию" отмечена строка b (listdlg(...'InitialValue', 2, ...)). Предусмотрен выбор только одной строки (listdlg(...'SelectionMode', 'single')). Под полем со списком располагаются две кнопки шириной 40 и высотой 10 пикселей с подписями Cancel на левой кнопке и OK на правой.

Табл. 4.2.

Параметр	Назначение	Переменная
PromptString	Сообщение, появляющееся над списком. Значения: строка или массив строк / ячеек из строк (для многострочного текста). По умолчанию пустой массив ячеек.	<i>string</i>
Name	Заголовок окна (по умолчанию пустая строка).	<i>string</i>
SelectionMode	Число выбираемых строк.	<i>single</i> или <i>multiple</i>
InitialValue	Номер выделенной строки списка при его открытии (по умолчанию 1).	<i>numerical</i>
ListSize	Размер области окна (пикселей), значение задается вектором [ширина высота] (по умолчанию [160 300]).	два числа ([a b])
uh	Высота кнопок, значение задается в пикселях (18 по умолчанию).	<i>numerical</i>
ffs	Расстояние между областью списка и границей окна, значение задается в пикселях (8 по умолчанию).	<i>numerical</i>
fus	Расстояние между окном списка и кнопками <i>Cancel</i> и <i>OK</i> , значение задается в пикселях (18 по умолчанию).	<i>numerical</i>
OKString	Текст на кнопке <i>OK</i>	<i>string</i>
CancelString	Текст на кнопке <i>Cancel</i>	<i>string</i>
ListString	???	<i>cell array</i>

Пример:

```
[Selection,ok] = listdlg('ListString', {'Linear'; 'Quadratic';  
    'Cubic'})
```

– создает диалоговое окно выбора со списком из трех строк: Linear, Quadratic, Cubic. Если пользователь выбрал, к примеру Quadratic и нажал кнопку *OK*, то Selection = 2 и ok = 1. Если пользователь выбрал Linear и Cubic и нажал кнопку *OK*, то Selection = [1 3] и ok = 1 и т. д.

Меню с кнопками для выбора. Функция **menu** создает диалоговое окно (рис. 4.13) с *n* кнопками

```
Num = menu('Title', 'Text_1', 'Text_2', ..., 'Text_n').
```

В окне будет подпись Title и *n* кнопок с надписями Text_1, Text_2, ... Text_n. Графическое окно закрывается после нажатия любой из этих кнопок или кнопки закрытия окна. Переменная Num принимает целочисленное значение от 1 до *n*, соответствующее номеру нажатой кнопки. Кнопке закрытия окна соответствует 0.

Рис. 4.13 соответствует команде

```
Num = menu('Title', 'B1', 'B2', 'B3', 'B4', 'B5').
```

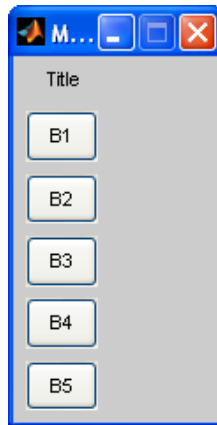


Рис. 4.13.