

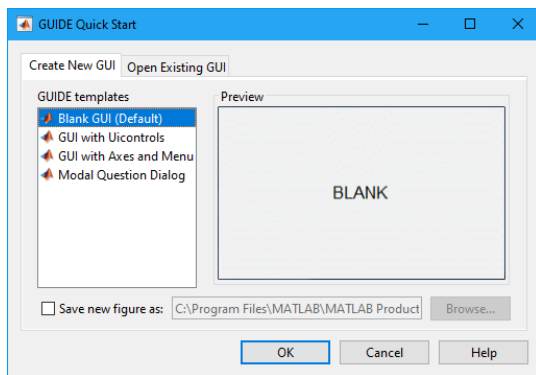
ИЗПОЛЗВАНЕ И СЪЗДАВАНЕ НА ГРАФИЧЕН ИНТЕРФЕЙС

Всяко по-значимо приложение в Matlab е желателно да бъде нагледно в резултатите и удобно при въвеждане на данни от страна на потребителя, да извежда собствени съобщения за грешки, да дава възможност за избор от менюта и да даде усещане на потребителя, че е в едно истинско windows приложение. Всичко това е предвидено в съвременните версии на Matlab.

Ние ще се запознаем с най-основните графични компоненти, функции за четене (get) и промяна (set) на техните свойства (Properties) и възможности за реагиране на възникнали събития (например избор на обекта, еднократното или двукратно щракване върху обекта с мишката и т.н.).

Считам, че е най-добре да го покажем чрез примерни кодове.

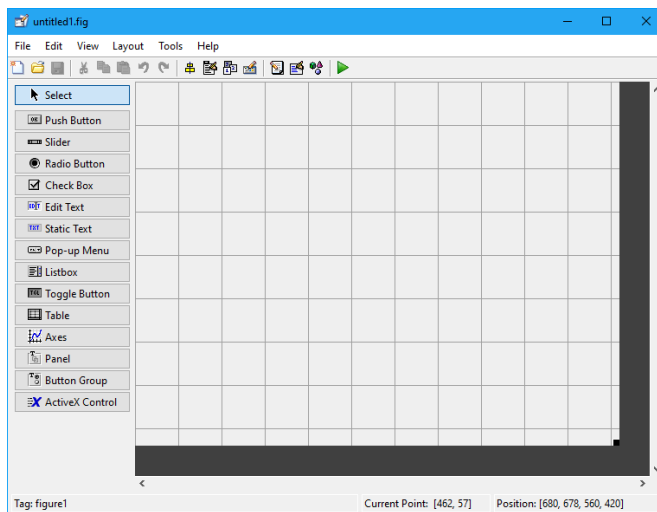
В Command Window стартираме приложението за създаване графичен потребителски интерфейс >>guide.



Избираме празен Blank GUI и натискаме ОК.

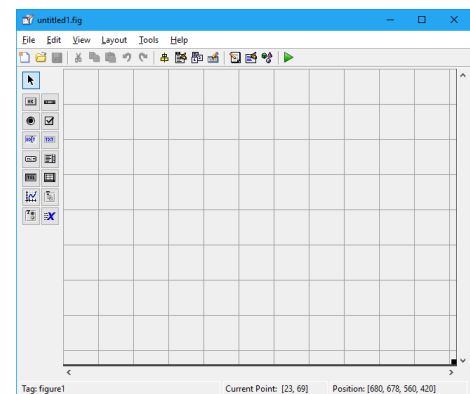
(Ако имаме вече създаден избираме Open Existing GUI).

Появява се следния прозорец, в който вече ще създаваме своя интерфейс.

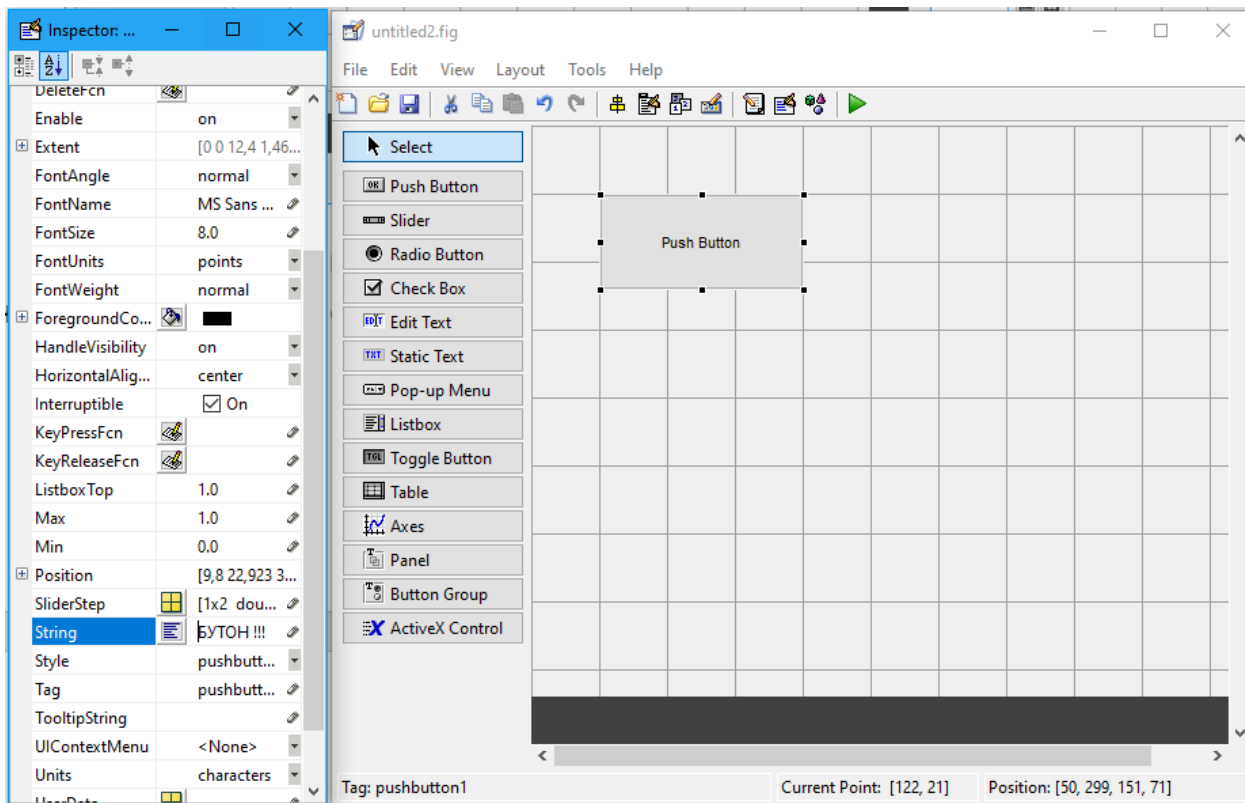


Изборът на обект и позиционирането му става чрез влачене с мишката или с двукратно щракване върху него.

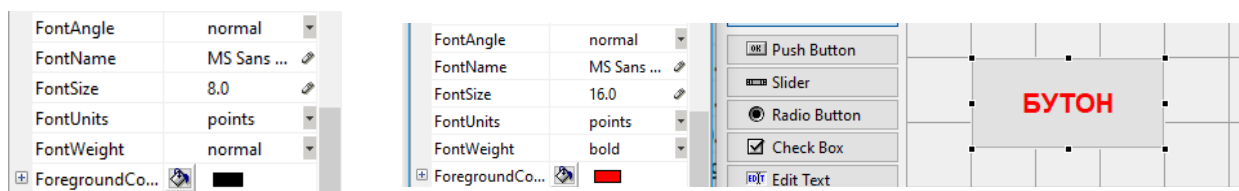
Ако обектите в ляво нямат надписи, можете да ги добавите от меню File->Preferences->Show names in Component Palette.



След позиционирането на графичния обект в работното поле е нужно да се променят някои негови свойства, като големина, надписи, цвят на шрифта и т.н. За целта се щрака два пъти върху него и се отваря прозореца Properties Inspector. Там се правят промените. В нашия пример сме поставили един бутон (Push Button) и сме сменили името му в String от PushButton1 в БУТОН.



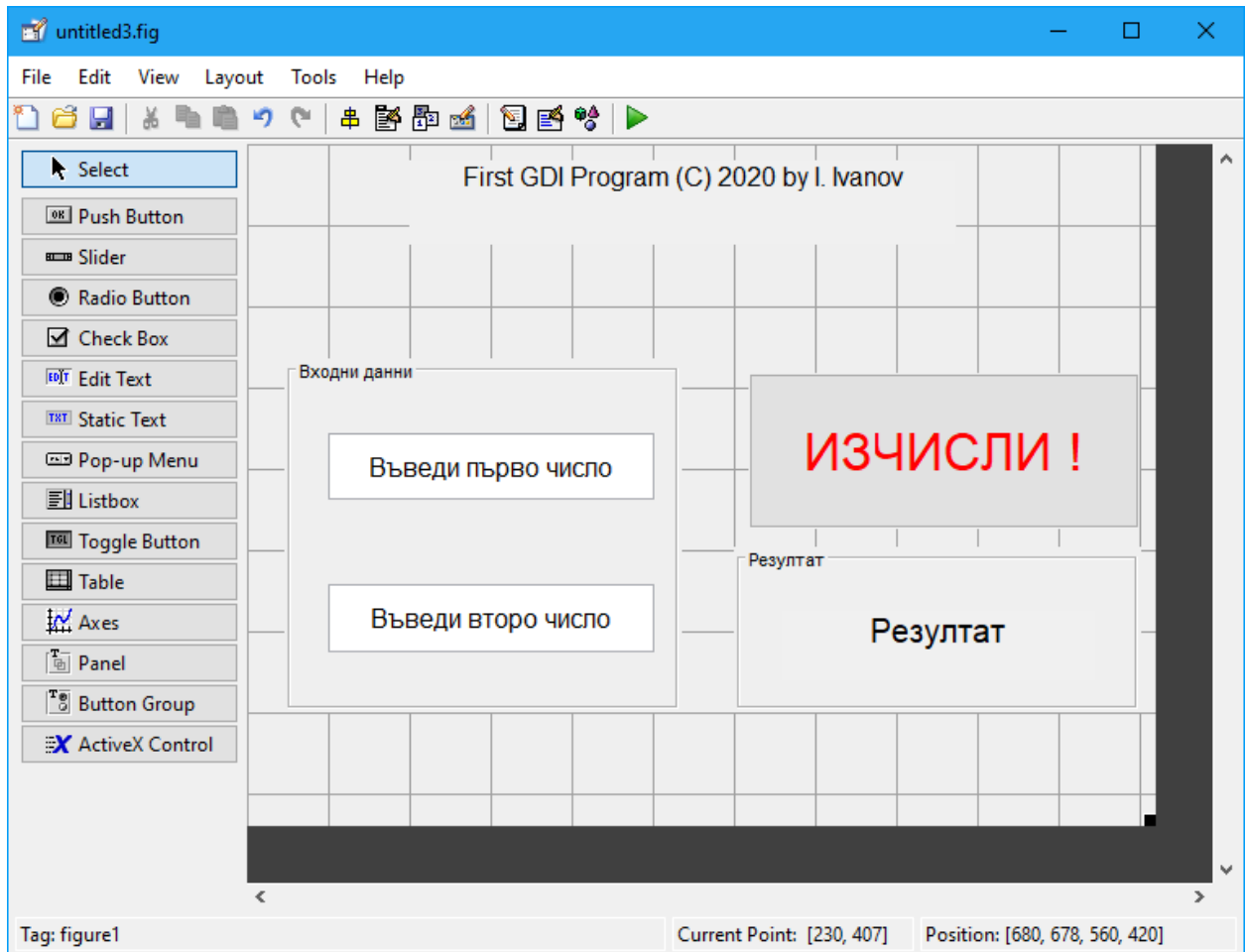
Смяната на шрифта, размера му, цвета му и изписването – подчертан, наклонен и т.н. става чрез нужната промяна на следните свойства.



Опитайте сега да позиционирате в работното поле различни графични обекти, променяйте техните свойства чрез Properties Inspector, наблюдавайте какво се променя, и после ги изтрийте чрез избор с мишката и бутона от клавиатурата Delete или чрез десен бутон и избор от контекстното меню на командата Clear.

Нека сега направим най-проста програма, която задава две дробни числа в две текстови полета тип Edit, (текстът в които може да се променя) и отпечатва името на автора и резултата от произведението на числата в други две текстови полета (тип Static text), в които текста не може да се променя. Това се случва след като се натисне бутон с име Изчисли.

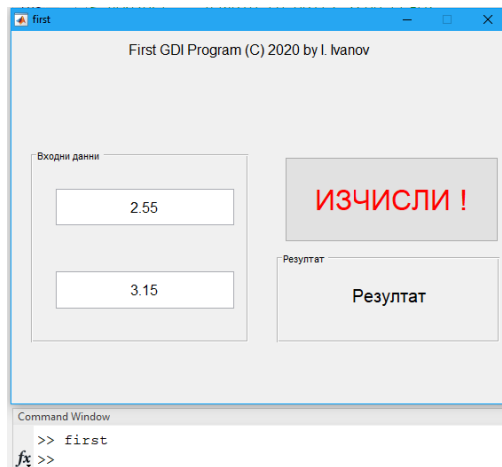
Графично прозорецът може да изглежда по следния начин:



Разбира се, големината на символите, разположението на обектите може да бъде друго!

Сега е необходимо да запишем постигнатото. Това става чрез меню File->Save или Save as... Избираме First. Създават се два файла: first.fig (графична информация) и first.m(код).

Сега може да стартирате от Command window вашето приложение с неговото име First и да изпробвате как се държи графичната част. Засега резултати от сметките не може да има.



Във функцията First.m е генерирания код от GUIDE на Matlab. Сега ще направим в него корекции, така че да се изпълни предвидения от нас алгоритъм.

Действието се реализира след натискане на бутона и следователно търсим функция свързана с него. Намираме `function pushbutton1_Callback(...)`.

След края ѝ, но в нейното тяло трябва да добавим нашия код.

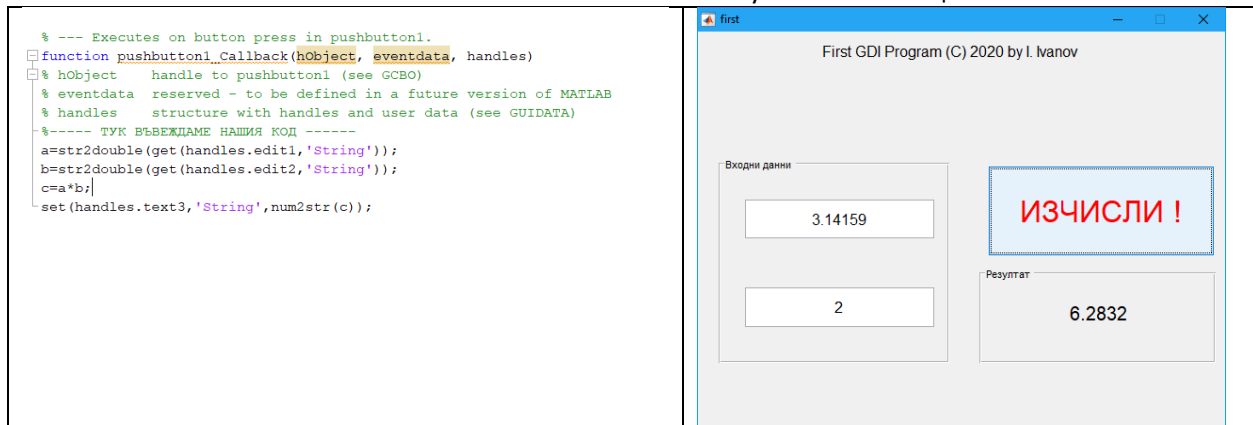
Чрез функцията `get(...)` четем съдържание. В случая чрез `get(handles.edit1, 'String')` четем съдържанието на текстовото поле edit1 във вид на символи. Трябва да го преобразуваме в число за да събираме, умножаваме и т.н. Това става чрез функцията `str2num()`. Същото правим и от текстовото поле edit2.

Изчисляваме $c=a*b$; съгласно нашия алгоритъм.

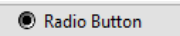
Отпечатването в статичното текстово поле става чрез функцията `set(...)`.

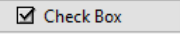
```
set(handles.text3, 'String', num2str(c));
```

`text3` е името на полето, в което отпечатваме резултата. При вас може да е с друго име, ако сте въвели повече или по-малко статични текстови полета. Резултатът е налице.

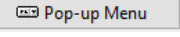


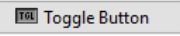
Извикването и на друга функция или файл-функция създадена от нас става по аналогичен начин, на мястото където ни е необходимо, най-често след натискане на даден бутон.

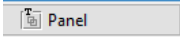
Обекта  обикновено се използва когато потребителят трябва да избира между несъвместими (алтернативни) неща. (пол мъж/жена, гражданство българско/чужденец и т.н.).

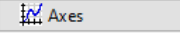
Обекта  прави възможен избор на няколко неща заедно, те не са взаимноизключващи се.

Обекта  прави възможен избор от списък подреден по някакъв признак.


Обекта  дава възможност на потребителя да създава менюта и разклонения при избора.

Обекта  е подобен на радио бутона, но визуално остава включен(натиснат) или изключен.

Обекта  е удобен за подреждане на визуална информация, например панел за избор от опции, състоящ се от няколко check box. Вече го използвахме в горната програма за по-прегледно разположение на входни данни и резултати.

Обекта  позиционира прозорец, в който ще се строи 2D или 3D графична зависимост.

Обекта  позволява представянето на подредени данни в табличен вид.

Обекта  предизвиква изпълнение на друг код (графичен и не само), написан не на езика на Matlab, който се обработва/използва от вашето графично приложение.

Често използвани диалогови кутии

1. Съобщение за грешки

```
errorDlg('Съобщение, което да се появява','Error');
```

Например, когато се проверява дали потребителят е въвел число в текстовото поле от тип edit, ако не е число да го нулира.

```
var = str2double(get(hObject, 'String'));  
if isnan(var)  
    set(hObject, 'String', 0);  
    errorDlg('Input must be a number','Error');  
end
```

2. Диалогова кутия за въвеждане на данни

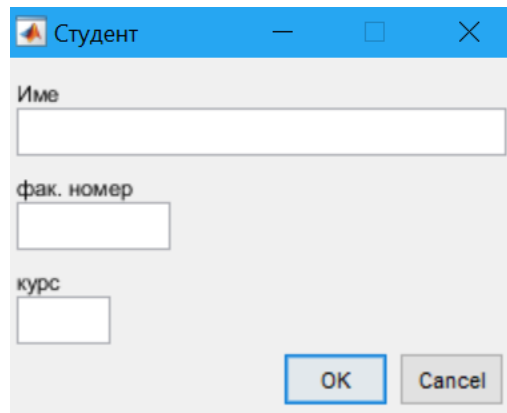
Друг удобен графичен обект е inputdlg

```
answer = inputdlg(prompt)  
answer = inputdlg(prompt,dlg_title)  
answer = inputdlg(prompt,dlg_title,num_lines)
```

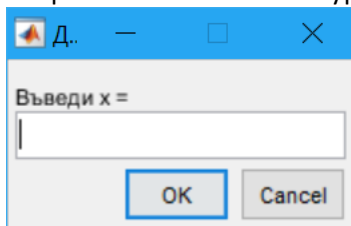
Например:

```
>> x = inputdlg({'Име','фак. номер','курс'},...  
'Студент', [1 40; 1 12; 1 7])
```

Ето как изглежда при извикване:



Или удобен вариант за въвеждане на променливи от клавиатурата:

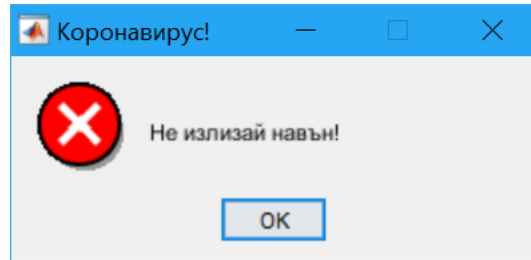


3. Диалогова кутия MessageBox за съобщения към потребителя!

```
h = msgbox(Message,Title,Icon)
```

Например:

```
h = msgbox('Не излизай навън!', 'Коронавирус!', 'Error');
```



4. Диалогово меню. Използва се за избор от няколко възможности и връща номера на избраната.

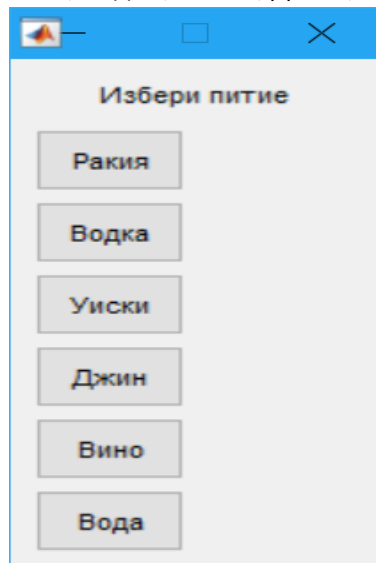
Обикновено се обработва с оператор `switch` и `case`

```
choice=menu('mtitle','opt1','opt2',...,'optn')
```

```
choice = menu('mtitle',options)
```

Пример:

```
choice=menu('Избери питие','Ракия','Водка','Уиски','Джин','Вино','Вода')
```



5. Диалог за въпроси `questdlg`

```
button = questdlg('qstring','title',default)
```

Например:

```
options.Interpreter = 'tex';
```

```
% Включва отговор по подразбиране Default answer
```

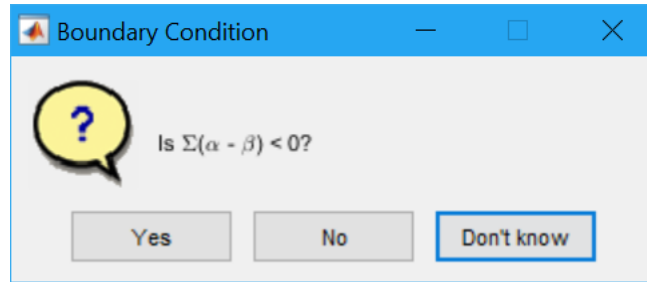
```
options.Default = 'Don't know';
```

```
% Създава (Ла) TeX string за въпроса когато съдържа гръцки и специални символи
```

```
qstring = 'Is  $\Sigma(\alpha - \beta) < 0?$ ';
```

```
choice = questdlg(qstring, 'Boundary Condition',...
```

```
    'Yes', 'No', 'Don't know', options)
```

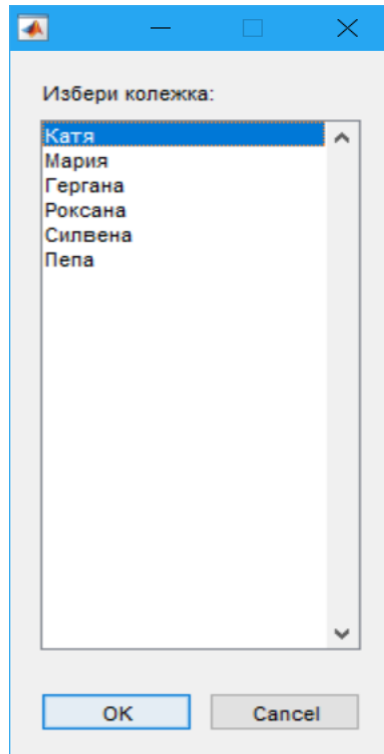


6. Избор от списък listdlg

[Selection,ok] = listdlg(Name,Value,...)

Пример:

```
>> str={'Катя' 'Мария' 'Гергана' 'Роксана' 'Силвена' 'Пепа'};  
>> [s,v] = listdlg('PromptString','Избери колежка:', 'SelectionMode','single',...  
    'ListString',str)
```



Съществуват и други, за които можете да прочетете в ръководството за Matlab.