



Scientific Computer Software

(*MatLab*)

Topic 11. Creating a **G**raphical **U**ser Interface

Курбатова Наталья Викторовна, к.ф.-м.н.,
доцент кафедры математического
моделирования, мехмат, ЮФУ



Contents:

- I. GUI Creation Strategies
- II. Parent objects of the controls.
- III. Features. Modification
- IV. Unified Control Designer
- V. The controls are static and dynamic. Changing properties, event handling
- VI. Axes. Features
- VII. Functions that provide access to the file system



Two approaches to GUI creation GUI

I – evolutionary, II – revolutionary; ideally - $I \cup II$

I. All elements and procedures are created by the user

Advantages:

- fully controlled process
- saves time at the stage of debugging and modification of the GUI
- in the process of GUI development, a systematic understanding of the functionality is achieved

Disadvantages:

- Significant time expenditures at the stage of building the GUI structure (appearance)

expenditure [ɪks'pendɪtʃə]расходы

II. GUI elements and procedures are created using the editor guide

Advantages:

- it is easy to create elements in the editor
- it is easy to edit elements, the structure of the entire window (ensemble: align elements, etc.)
- the system generates gui name.m and executable guiname.fig

Disadvantages:

- A higher level of understanding is required
- Difficulties arise at the stage of functional improvement GUI



figure – the parent object of all GUI elements, the field for the interface

uicontrol – a *single control elements constructor* (CEC), it creates the following

Control elements: *editableText*, *statictext*, *checkboxes*, *listboxes*, *popupmenus*, *sliders*, *pushbuttons*, *radiobuttons* (*togglebuttons*);

axes – the axis constructor

uigetdir, **uigetfile** – the functions provide an *interactive* selection of files on the media



Root и Figure

nvkurbatova@sfedu.ru

➤ parent object GUI – **figure!**

enter the descriptor for figure: **f=figure;**

important for multiwindows GUI

➤ root (descriptor - **0**) – parent object for figure;

It is important to identify the monitor size - design, default commands (format, hold, echo...)!

➤ size figure, position, *depends on screen size* :

`[x0,y0,hx,hy]=get(0, 'screensize')`

➤ the color of the figure (background) default *inherited by descendants, (check in your version!)*

`colorfigure=get(f, 'color ')`

descendants background - field BackgroundColor

➤ `set(gcf, 'MenuBar', 'none')` without title in figure

➤ `set(gcf, 'NumberTitle', 'off')` – without title number in figure

➤ `set(gcf, 'name', 'My interface')` – user's title

➤ `set(gcf, 'Resize', 'off')` prohibition on changing the size figure

prohibition[prəʊhɪ'biʃn] запрет



Style as type of control element. Constructor of CE

Field style	Value of field style	The purpose of the control element
style	edit	editable text, information input, dynamic element (de)
style	text	explanatory text (unchangeable), static element (se)
style	checkbox	(0 or 1) – dynamic element (branching the algorithm)
style	list	list (de)
style	sliders	scaling engine, slider (de)
style	pushbutton	the button to start the algorithm process (de)
style	radiobutton	the switch, a sign, takes the values 0 or 1 (de)
style	frame	a frame for visually highlighting a group of objects in the figure field (se)
style	popup	mini menu designer (de)



GUI Building Strategy (I-th approach)

- 1) we create a head file, for example, guiname.m – procedure without parameters
- 2) we describe the variables as global: **global** var1 var2; if they need to be used in external procedures; either we operate with the parameters of the subprocedures (**vari** – including CE descriptors)
- 3) we build a window for all GUI elements, using **default properties** f=figure, or change the properties according to need
- 4) change the default properties, for example, **units** of measurement: set(f, 'units', 'normalized') **The latest version is all friendly!**
- 5) creating all GUI elements (CE, axes, etc.)
- 6) we are asking! (at the debugging stage, it is advisable) or we do not programmatically set the "initial" values preceding the dialogue
- 7) we program that part of the code that is based on processing events that provide input and output of information, the choice of proposed transitions, the content of the problematic algorithm
- 8) we program procedures (in separate files, handle_function or under procedures) that ensure the launch of processes



Changing the properties of graphic objects

- **set** – setting the properties (fields of the structure) of a graphic object;
- **set(descriptor)** – getting possible properties of fields with default indication
- **get** – output (getting) the current properties of the fields of the graphic object structure;
- **reset(descriptor)** – restoring the default properties of a graphic object;
- **delete(descriptor)** – deleting a graphic object that is an argument of the delete function.



Control Element - text

static, unchangeable

nvkurbatova@sfedu.ru

Required properties :

```
text1 = uicontrol(f, 'Style', 'text', 'String', ' надпись ', 'Position', [x0t y0t lxt lyt])
```

- descriptor f or gcf) optional if the element is being built in the active figure;
- **text1** – descriptor of CE 'text'
- the first letters in the field names can be uppercase and/or lowercase (field names are case-insensitive, but in notation **name.NameField** –?)
- the values of the other properties in this example are for **text1** selected by default (see set(CE_name))

Additional properties :

```
set(text1 , 'fontname', 'tex ', 'fontsize', 14, 'backgroundcolor', get(f, 'color'), ...,  
        'HorizontalAlignment', 'left', 'FontAngle', 'italic')
```

- properties are set at creation or as needed
- get(f, 'color') provides a choice of background color figure In case you need it :)

in new versions, it is not necessary



Example: text

nvkurbatova@sfedu.ru

- 1) function partGui
- 2) f=figure
- 3) tt=uicontrol('style','text','Position',[1, 1 100 50],...
'String','Ypa!','fontsize',14) % **Check for case-free (register)!**

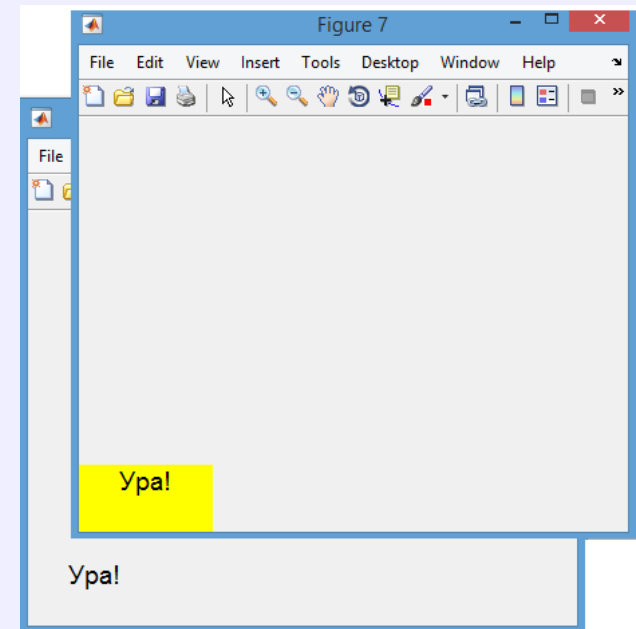
By default, the background color of the text matches the background figure!

- 4) set(tt) % this is how we find out the properties of fields, including by default
% HorizontalAlignment: {'left' 'center' 'right'}
- 5) tt.HorizontalAlignment % 'center'
- 6) tt.BackgroundColor % [1 1 0]
- 7) tt.Units % 'pixels'
- 8) tt.BackgroundColor='yellow'

Questions:

What are the coordinates of the lower-left corner of the figure?

Explain the parameters (elements) of the position vector! Check the 'VerticalAlignment' field in tt (removed?)





Control Element - **edit** (dynamic)

nvkurbatova@sfedu.ru

Required properties :

ed = uicontrol(f, '**Style**', 'edit', '**String**', ' x0 ', '**Position**', [x0e y0e lxe lye])

field *edit* it is intended for data entry, more often empty, at the debugging stage, it is advisable to set the value, f.e.

'String', ' x0 ' или 'string', num2str(double_value)

Additional properties (similar to text) change at the will of the programmer!

Additional properties :

this is getting the entered information and converting it to the required data type

Charvalue=get(**ed**, 'String') - information (Char) retrieved from the field *String*

DoubleValue=str2num(**Char**value) - converted to double



Control Element - **list** (dynamic)

Required properties :

lst = uicontrol(f, '*Style*', 'list', '*String*', Cell of String, [x0l y0l lx1 ly1])

- list - is a switch that allows you to implement several provided transition
- Cell of String - an array of cells from strings that describe a set of "transitions" and are displayed on this GUI element GUI
- a window is created, and scrolling is provided systematically if necessary
- the user selects a specific "transition"

Event processing (multi-functional transition):

Index= get(**lst**, 'value') % Index - the number of the selected item in the list
Cell of String, which defines the "transition"

The entire list of features is extracted from the *String* field:

ListNames=get(**lst**, '*String*') % Cell array

StepIN=ListNames{Index} % a custom transition is selected



Control Elements : **checkbox**, **radiobutton** (dynamic)

Required properties:

checkbox

chbox = uicontrol('Style', 'checkbox ', '*String*', 'explanation',...
'*Position*', [x0chb y0chb lxchb lychb]);

radiobutton

rb = uicontrol('Style', ' radiobutton ', '*String*', 'explanation',...
'*Position*', [x0rb y0rb lxrb lyrb]);

the purpose of both elements are switches, branching of the algorithm
(presence or absence of a specific option)

checkbox и **radiobutton** differ in the "topology" of the optional selection

Event processing:

Event: option choice – is generated 1, otherwise – 0

The system **passes** the option as a field property *value* of structures **chbox** or **rb**; are retrieved according to: **r**=get(**rb**, '*value*') or **r**=get(**chbox**, '*value*')

What is equal to r?

are retrieved[ri'tri:vd] извлекаются



Control Elements: **pushbutton** (dynamic)

Required properties:

pb = uicontrol('Style', 'pushbutton', 'String', 'explanation text',...
'Position', [x0pb y0pb lxp b lypb], 'Callback', **action**);

gcbo (GraphicCurrentButtonObject) – descriptor of **pushbutton**,
Callback of this CE currently in progress

The purpose of the element: is to launch an algorithm or process at the touch of a button

action can take the following values :

- a string that contains a sequence of commands, for example:
'cla, load XY, plot(X,Y) '
- the name of the subfunction, including handle_function
- the name of the external function (including for a complex multi-window GUI)

Event processing:

Performing the action provided for **action**



Control Elements: slider (dynamic)

Required properties:

```
sl = uicontrol('Style', 'slider', 'Min', vmin, 'Max', vmax, ...  
             'SliderStep', [a b], 'Value', position_slider, ...  
             'Position', [x0sl y0sl lxls lysl], 'Callback', action);
```

Appointment :

Scaling of objects, including graphical ones.
The zoom factor varies within [vmin,vmax]

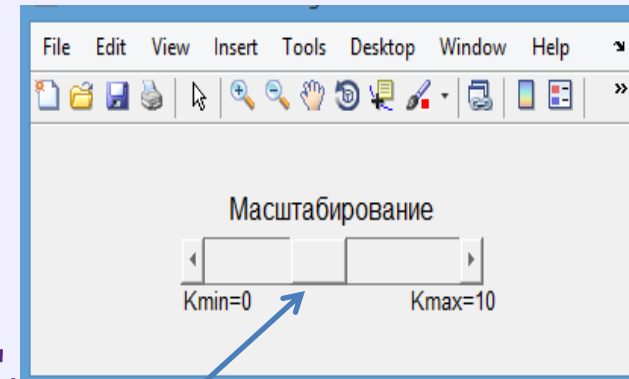
Event processing:

similarly **action** for **pushbutton**



Features of the engine parameters

```
function myslider
global sl scaling
sl = uicontrol('Style', 'slider', 'Min',0,'Max',10,...
    'SliderStep',[0.1 0.3], 'Value', 0.5, ...
    'Position', [100 100 200 25], 'Callback', 'CHf' );
tsl=uicontrol('Style', 'text','position',[100 125 200 25],...
    'string', ' Масштабирование ', 'fontsize',12)
tsl1=uicontrol('Style', 'text','position',[85 65 80 35], 'string',...
    'Kmin=0', 'fontsize',10, 'fontname', 'Latex')
tsl2=uicontrol('Style', 'text','position',[240 65 80 35],
    'string',...
    'Kmax=10', 'fontsize',10)
end
```



```
function val=CHf
global sl scaling
val=get(sl,'value')
scaling=get(sl,'Min')+(get(sl,'Max')-get(sl,'Min'))*val
end
```



Creating a pop up menu

nvkurbatova@sfedu.ru

Popup – a control with a list property and a button

Required properties:

```
pp = uicontrol('Style', 'popup',...  
               'String', {'array', ' Cell ', ' of Char'}, ...  
               'Position', [x0pp y0pp lcpp lypp],  
               'Callback', action);
```

Event processing:

accumulates the properties of the **list**, i.e. the required branch is selected, and the **buttons** - it starts *action*:

```
index=get(pp,'value')  
cellchar=get(pp,'string')  
newval=cellchar{index}
```

If the size of the selected area for the **popup** is not enough, then the control element is implemented as a pop-up menu!



Building an axis object by the constructor **axes**

axes (name), name = gca (*descriptor of current axes*)

Creating axes:

```
ha=axes( 'parent', f, 'Units', 'points', 'Position', [ x0ha y0ha lxha lyha ], ...  
'Color', [ 1 1 1 ], 'FontSize', 14 );
```

little tricks:

- axes(ha) ~ ha – an active axis
- cla – clear the active axis
- set(ha, 'Xtick', [], 'Ytick', []) – refusal to apply markings on the axes
(when is it important to give up?)

**axis([Xmin Xmax Ymin Ymax]) –
figure it out on your own!**

Search for controls with specified properties :

Obj= findobj(f, 'style', 'list'), **f.e.**

Obj= findobj(f, 'tag', 'specific'),

'tag' - a label is assigned by the user to any controls, including heterogeneous ones, connected by a certain logic – for example, to make a group of CE invisible or apply some operation to a group of CE



Functions providing access to the file system

selecting the working-current directory:

folder_name = uigetdir

Navigating the file system and
selecting a folder in the dialog box:

folder_name = uigetdir(start_path, dialog_title)

selecting a file in the current folder:

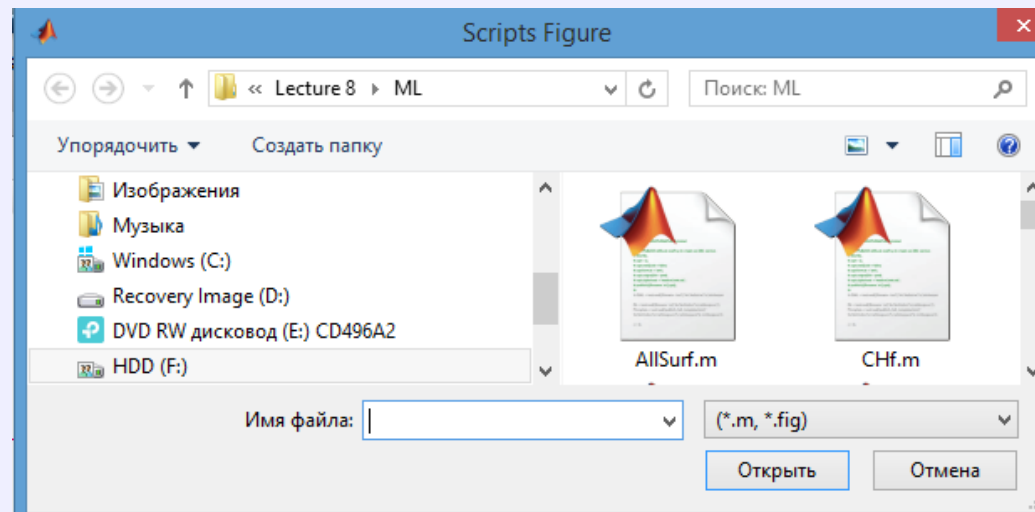
filename = uigetfile

Search in the dialog box:

**[filename, pathname] =
uigetfile(FilterSpec, DialogTitle, Name)**

The specified files are displayed in a dialog box, for example :

[FileName, PathName] = uigetfile({'*.m', '*.fig'}, 'Scripts Figure');



Sample code (written in the netgui.m file)

```
function netgui
```

```
f = figure('visible','off')
set(f,'resize','off') % fix size figure
```

```
ax = axes('Units','pixels');
surf(peaks)
```

```
% Create pop up menu
```

```
popup = uicontrol('style','popup',...
    'String',{'parula','jet','hsv','hot','cool','gray'},...
    'Position',[20 340 100 50],...
    'Callback',@setmap);
```

```
% Create push button for clear axes
```

```
btn = uicontrol('style','pushbutton','String','Clear',...
    'Position',[20 20 50 20],'Callback','cla');
```

```
% Create slider
```

```
sld = uicontrol('Style','slider',...
    'Min',1,'Max',45,'Value',41,...
    'Position',[400 20 120 20],...
    'Callback',@surfzlim);
```

```
% Add a text uicontrol to label the slider
```

```
txt = uicontrol('Style','text',...
    'Position',[400 45 120 20],...
    'String','Vertical Exaggeration');
```

```
% Make figure visible after adding all components
```

```
set(f,'Visible','on');
```

```
function setmap(source,event)
```

```
    val = source.Value;
    maps = source.String;
    newmap = maps{val};
    colormap(newmap);
```

```
end
```

```
function surfzlim(source, event)
```

```
    val = 51 - source.Value;
    zlim(ax,[-val val]);
```

```
end
```

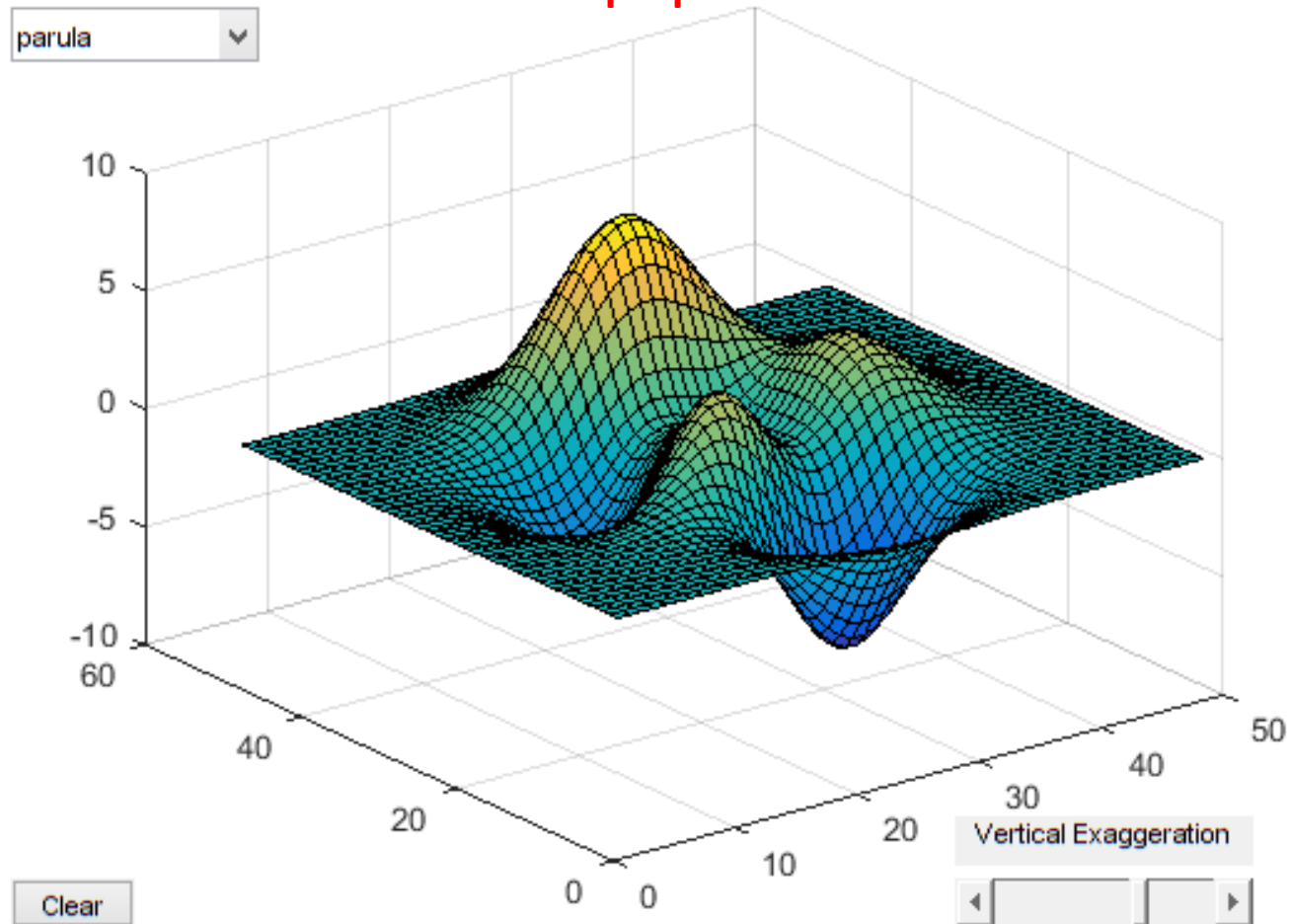
```
end
```



GUI - netgui.m

nvkurbatova@sfedu.ru

About the properties of fields in the source!





Conversion to dimensionless controls

- Controls are created by default in **pixels**;
- Adding a field and a property when creating all elements (even if position is already set in pixels): **'units','normalized'** or redefining **name.units='normalized'**, we provide the ability to change the size of the figure, i.e. redefine : **set(f,'resize','on')**; even if they were "locked up" before that `resize → off`

Creating a control in default units, changing the units of measurement, for example:

% Create slider

```
sld = uicontrol('Style','slider',...  
'Min',1,'Max',45,'Value',41,...  
'Position',[400 20 120 20],...  
'Callback', @surfzlim, 'units','normalized' );
```

It seems to be a contradiction?! Yes, but the goal is - do not injure the user!!

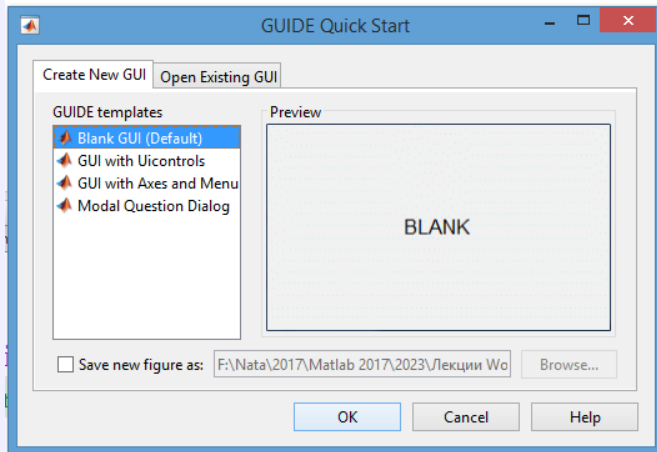
injure['ɪndʒə] - ранить



GUI Builder

nvkurbatova@sfedu.ru

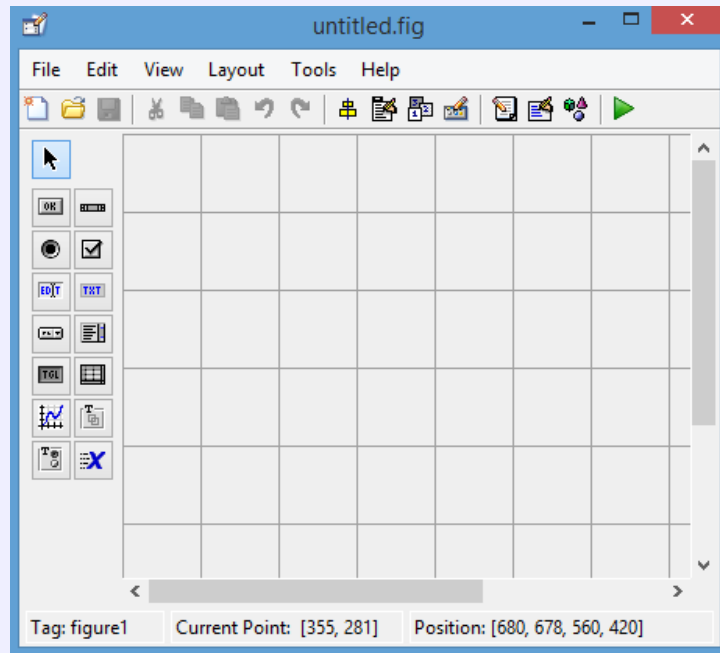
K>> guide



Creating an CE in the Blank field → OK

Advantages: easy editing, interactive setting of properties. A script with the selected name is automatically created, for example, nameGUI.m

Disadvantages: the required depth of comprehension of the process; the complexity of modifying links and adding new elements.



The simplicity of the GUI constructor is compensated by the complexity of the implementation, especially for multi-window algorithms, if need GUI restructurization!



Graphic object management functions

nvkurbatova@sfedu.ru

gco – returns the descriptor ("pointer") of the current graphic object;

gcbo – returns the descriptor of the object (pushbutton) whose function is currently running;

gcbf – returns the descriptor of the figure window containing the object whose function is currently running – important for a multi-window interface;

drawnow – executing a queue of delayed graphic commands is important when programming animations;

findobj – find objects with the specified properties (the field is a property of the structure);

copyobj – copy the object and the objects generated by it.

**Help on creating a GUI in the classic way (I-th approach)
get into Moodle or Teams!**



Спасибо за внимание!

«Dreams, when they get old, certainly become nightmares» - Romain Gary ~ Eugene Azhar

«Don't delay your dreams!» NB