

MATLAB.
СОЗДАНИЕ
ГРАФИЧЕСКИХ
ПРИЛОЖЕНИЙ.

**Кафедра Информатики и
Компьютерного Проектирования
РХТУ им. Д.И.Менделеева
Москва, 2012**

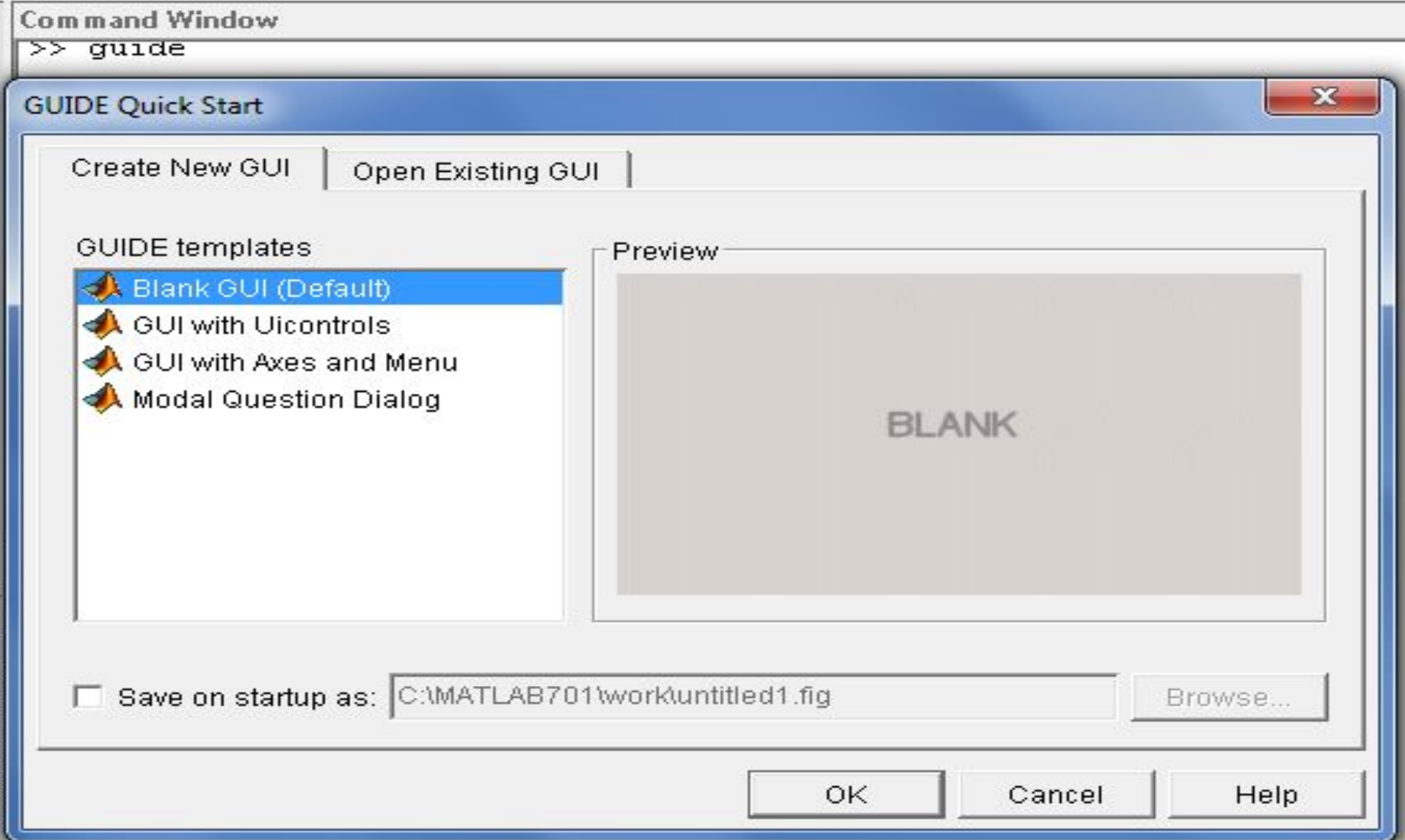
Графические объекты. 1

- Графические приложения содержат меню, кнопки, области ввода текста, переключатели, графики.
- Элементы, из которых состоят графические приложения, являются объектами. Указатель это переменная, в которой хранится адрес (в памяти компьютера) объекта. Пользователь пользуется указателями, как переменными. То есть для совершения действия над элементом графики, пользователь в соответствующей команде указывает имя переменной, означающей данный элемент графики (указатель на данный графический элемент)

Графические объекты. 2

- Для установки свойств объектов есть функция `set(указатель_на_объект, 'изменяемое_свойство', 'его_новое_значение', 'изменяемое_свойство_2', 'его_новое_значение_2')`.
- Для получения свойств объекта есть функция `get(указатель_на_объект, 'свойство')`.
- В окне Command Window следует набрать `guide` и тогда откроется окно для создания визуального интерфейса. В нем можно выбрать как уже существующие (Open Existing GUI) так и создание нового интерфейса.

Создание программ с визуальным интерфейсом. 1



Создание программ с визуальным интерфейсом. 2

- Для примера создадим вариант интерфейса. Он предназначен для ввода нескольких исходных величин (аргументов) и вычисления нескольких интересующих пользователя ответов (результатов или, иначе говоря, функций от введенных аргументов). Также предусмотрено построение графика какой-либо функции от какой-либо переменной по усмотрению пользователя, указывающего также минимальное и максимальное значения аргумента и шаг его изменения.

Создание программ с визуальным интерфейсом

exGuide_1.fig

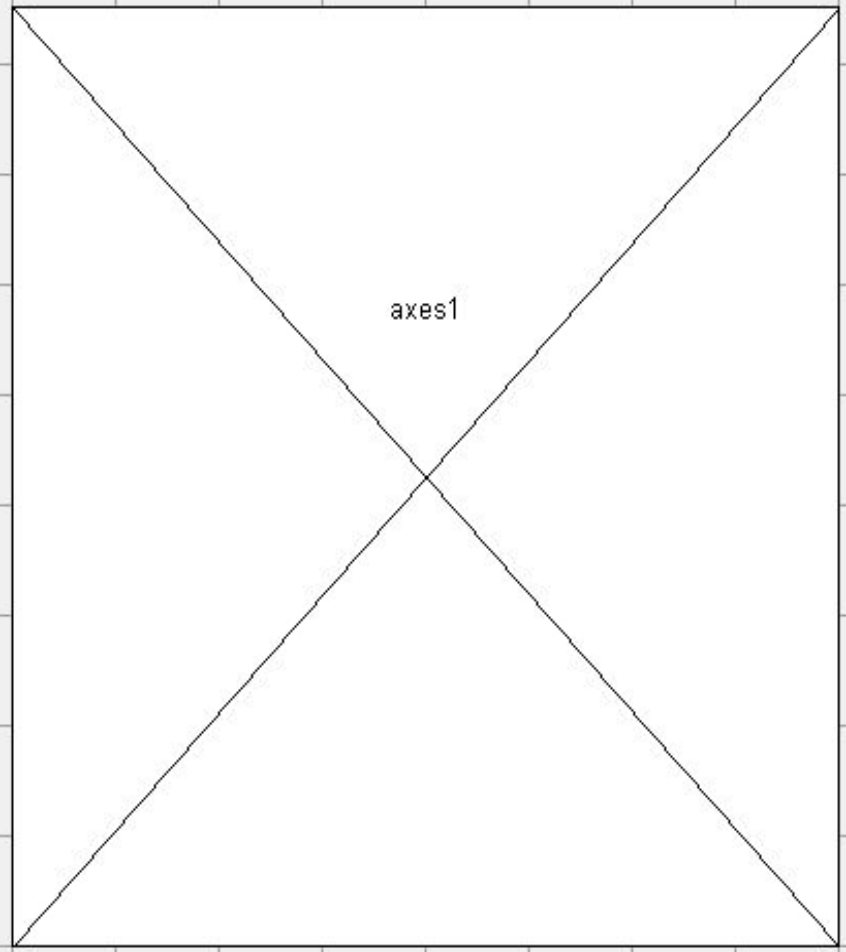
File Edit View Layout Tools Help



Arguments		Results	
<input type="text"/>	argument x	<input type="text"/>	result func1
<input type="text"/>	argument y	<input type="text"/>	result func2
<input type="text"/>	argument z	<input type="text"/>	result func3

Calculate/Plot	Calculate/Plot & Save	Open from file
<input type="button" value="Calculate"/>	<input type="button" value="Calculate & Save"/>	<input type="button" value="Open data"/>
<input type="button" value="Plot"/>	<input type="button" value="Plot & Save"/>	<input type="button" value="Open data & plot"/>

Plot parameters				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
NumFun	NumArg	MinArg	StepArg	MaxArg

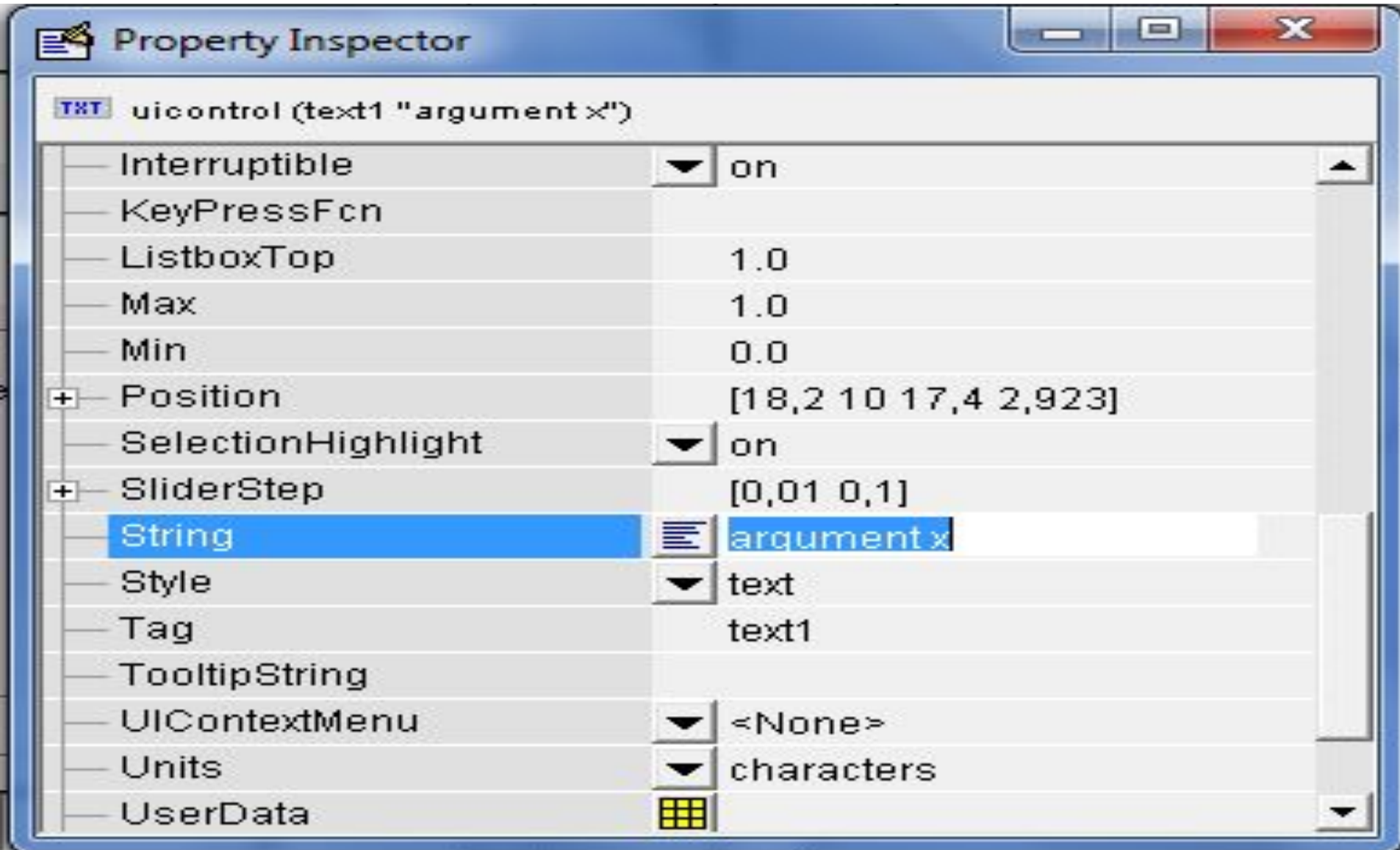


Создание программ с визуальным интерфейсом. 4

- Для редактирования свойств элементов интерфейса, определенный элемент дважды щелкнуть мышкой (левой кнопкой). Откроется редактор свойств (Property Inspector). В нем например имя элемента (под которым он фигурирует в компьютере) называется Tag. Найдя слово Tag в левой колонке, в правой увидим само имя (например text1). Надпись на элементе, которая видна в окне интерфейса, называется String. Найдя в левой колонке слово String, в правой ему соответствует сама надпись (например argument x).

Файл интерфейса имеет расширение fi

Создание программ с визуальным интерфейсом. 5



Подготовка M-файла для

интерфейса. 1

- Для того, чтобы интерфейс был связан с выполнением требуемых действий (например, при нажатии на кнопку Calculate происходило бы вычисление функций и вывод их значений в соответствующих окошках) необходимо предварительно описать в M-файле все, что требуется сделать. M-файл появится на экране после нажатия на соответствующую (четвертую справа) кнопку в ряду кнопок вверху экрана редактора интерфейса.

Подготовка M-файла для интерфейса. 2

- В M-файле, имеющем то же самое название, как и файл интерфейса, уже автоматически сгенерирован текст в соответствии с выбранными пользователем элементами интерфейса. Каждому элементу интерфейса соответствует абзац текста, начинающийся с упоминания названия (Tag) элемента интерфейса.

Подготовка M-файла для

интерфейса. 3

- Поскольку целесообразно начать с описания действий, выполняемых после нажатия на кнопку Calculate, то рассмотрим абзац текста в M-файле, описывающий их. Для этого в файле интерфейса щелкнем кнопку Calculate, открыв Property Inspector, и найдем Tag этой кнопки. Пусть, например, он оказался pushbutton1. Тогда в M-файле найдем абзац, озаглавленный `function pushbutton1_Callback(hObject, eventdata, handles)`
- В этом абзаце (то есть ниже строки его заголовка) укажем что должно произойти после ее нажатия

Подготовка M-файла для

интерфейса, 4

- Должна быть вызвана функция ввода аргументов, считывающая из соответствующих текстовых окошек исходные аргументы и возвращающая вектор (массив) аргументов. Затем должна быть вызвана функция, решающая задачу вычисления значений-результатов. Она получает вектор значений аргументов и возвращает вектор значений результатов. После этого должна быть вызвана функция для вывода в соответствующие окошки интерфейса значений из вектора результатов.

Подготовка M-файла для интерфейса 4

```
% --- Executes on button press in pushbutton1. CALCULATE  
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)  
[VectorArg]=VvodArg(hObject, eventdata, handles);  
[VectorRes]=ReshitZadachu(VectorArg);  
VivodRes(VectorRes,hObject, eventdata, handles);
```

Подготовка M-файла для интерфейса. 5

- Упомянутые функции должны быть описаны. Рекомендуется описывать функции выше того места программы, где они вызываются.

```
function[VectorArg]=VvodArg(hObject, eventdata, handles)
    x=str2double(get(handles.edit1,'String'));
    y=str2double(get(handles.edit2,'String'));
    z=str2double(get(handles.edit3,'String'));
    VectorArg=[x, y, z];
```

Функции визуального интерфейса 1

• Команда

```
x=str2double(get(handles.edit1,'String'));
```

означает что в переменную x будет записано число, возвращаемое функцией str2double которая преобразует строку цифр в число.

Аргумент этой функции это возвращаемое значение функции get(handles.edit1,'String'); которая обращается к элементу интерфейса handles.edit1 где handles означает обращение к визуальному интерфейсу, edit1 это обозначение конкретного элемента интерфейса.

Функции визуального интерфейса 2

- Команда `S=sprintf('%g',f1);` означает что в переменную `S` будет записана строка символов оператором `sprintf('%g',f1);` который обращается к аргументу `f1` и `'%g'` указывает что аргумент является вещественным числом.
- Команда `set(handles.edit4,'String',S);` означает что в элемент интерфейса `edit4` (окошко вывода текста) будет выведена строка символов содержащаяся в переменной `S`.

Функции визуального интерфейса 3

```
function [VectorRes] = ReshitZadachu (VectorArg) |
    x=VectorArg(1);
    y=VectorArg(2);
    z=VectorArg(3);
    f1=x^2+y+z;
    f2=x+y^2+z;
    f3=x+y+z^2;
    VectorRes=[f1, f2, f3];
```

```
function [] = VivodRes (VectorRes, hObject, eventdata, handles)
    f1=VectorRes(1);
    f2=VectorRes(2);
    f3=VectorRes(3);
    S=sprintf('%g', f1);
    set(handles.edit4, 'String', S);
    S=sprintf('%g', f2);
    set(handles.edit5, 'String', S);
    S=sprintf('%g', f3);
    set(handles.edit6, 'String', S);
```

Построение графика. 1

- Для построения графика предназначена кнопка интерфейса Plot. Логика построения графика такова. Пользователь вводит в окошки ввода значения переменных NumFun (номер функции 1, 2 или 3), NumArg (номер аргумента), MinArg (минимальное значение аргумента, с него начинаются подписи к горизонтальной оси), MaxArg (максимальное значение аргумента, до него идут подписи к горизонтальной оси), StepArg (шаг изменения аргумента). Затем пользователь нажимает кнопку Plot и происходит построение графика в соответствии с введенными данными.

Построение графика. 2

- В М-файле опишем соответствующий этой кнопке абзац текста. После нажатия на кнопку вызывается функция VvodArg, считывающая исходные аргументы и возвращающая вектор их значений.
- Потом вызывается функция PostrGraf. Ей передается вектор аргументов. Она строит график. Для построения графика функция PostrGraf считывает из окошек текстового ввода интерфейса соответствующие переменные.

Построение графика. 3

• Затем подсчитывает число шагов, вычитая минимальное значение из максимального и деля на величину шага. При этом команда `round` округляет до целого значения результат деления.

`NumberSteps=round((MaxArg-MinArg)/StepArg);`
после этого для каждого элемента массива горизонтальных координат `horis` точек графика и для каждого элемента массива вертикальных координат `vert` точек графика определяются значения. Когда все готово, то команда `plot` строит график и затем на график накладывается координатная сетка

Построение графика. 4

```
% --- Executes on button press in pushbutton2. PLOT
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[VectorArg]=VvodArg(hObject, eventdata, handles);
PostrGraf(VectorArg,hObject, eventdata, handles);
```

Построение графика. 5

```
function [] = PostrGraf (VectorArg, hObject, eventdata, handles)
    NumFun = str2double (get (handles.edit7, 'String'));
    NumArg = str2double (get (handles.edit11, 'String'));
    MinArg = str2double (get (handles.edit12, 'String'));
    StepArg = str2double (get (handles.edit13, 'String'));
    MaxArg = str2double (get (handles.edit14, 'String'));
    NumberSteps = round ((MaxArg - MinArg) / StepArg);
    horis (1) = MinArg;
    VectorArg (NumArg) = horis (1);
    [VectorRes] = ReshitZadachu (VectorArg);
    vert (1) = VectorRes (NumFun);
```

Построение графика. 6

```
for i=2:NumberSteps+1
    horis(i)=horis(i-1)+StepArg;
    VectorArg(NumArg)=horis(i);
    [VectorRes]=ReshitZadachu(VectorArg);
    vert(i)=VectorRes(NumFun);
end

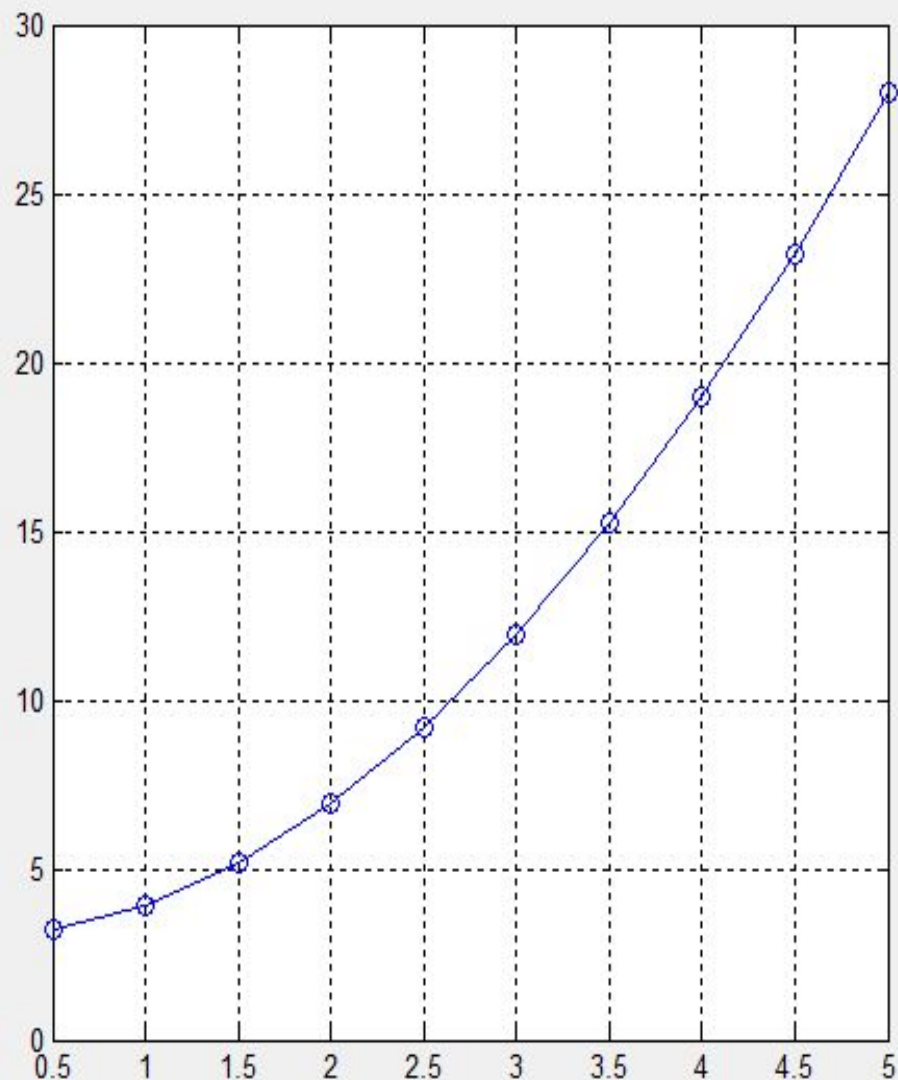
plot(horis,vert,'o-');
set(gca,'XGrid','on');
set(gca,'Ygrid','on');
```

Построение графика. 7

Arguments		Results	
<input type="text" value="1"/>	argument x	<input type="text" value="6"/>	result func1
<input type="text" value="2"/>	argument y	<input type="text" value="8"/>	result func2
<input type="text" value="3"/>	argument z	<input type="text" value="12"/>	result func3

Calculate/Plot	Calculate/Plot & Save	Open from file
<input type="button" value="Calculate"/>	<input type="button" value="Calculate & Save"/>	<input type="button" value="Open data"/>
<input type="button" value="Plot"/>	<input type="button" value="Plot & Save"/>	<input type="button" value="Open data & plot"/>

Plot parameters				
<input type="text" value="3"/>	<input type="text" value="3"/>	<input type="text" value="0.5"/>	<input type="text" value="0.5"/>	<input type="text" value="5"/>
NumFun	NumArg	MinArg	StepArg	MaxArg



Вычислить и сохранить

результаты. 1

• Для того, чтобы вычислить результаты и сразу же сохранить их в файл, предназначена кнопка Calculate & Save. В соответствующем этой кнопке тексте программы, происходит открытие файла

```
[f,p] = uiputfile ( 'Путь к файлу \ Имя файла.  
расширение', 'Окно выбора');
```

где среди возвращаемых значений p путь, f имя файла. После получения этих возвращаемых значений функции uiputfile, следует объединить эти данные с помощью команды

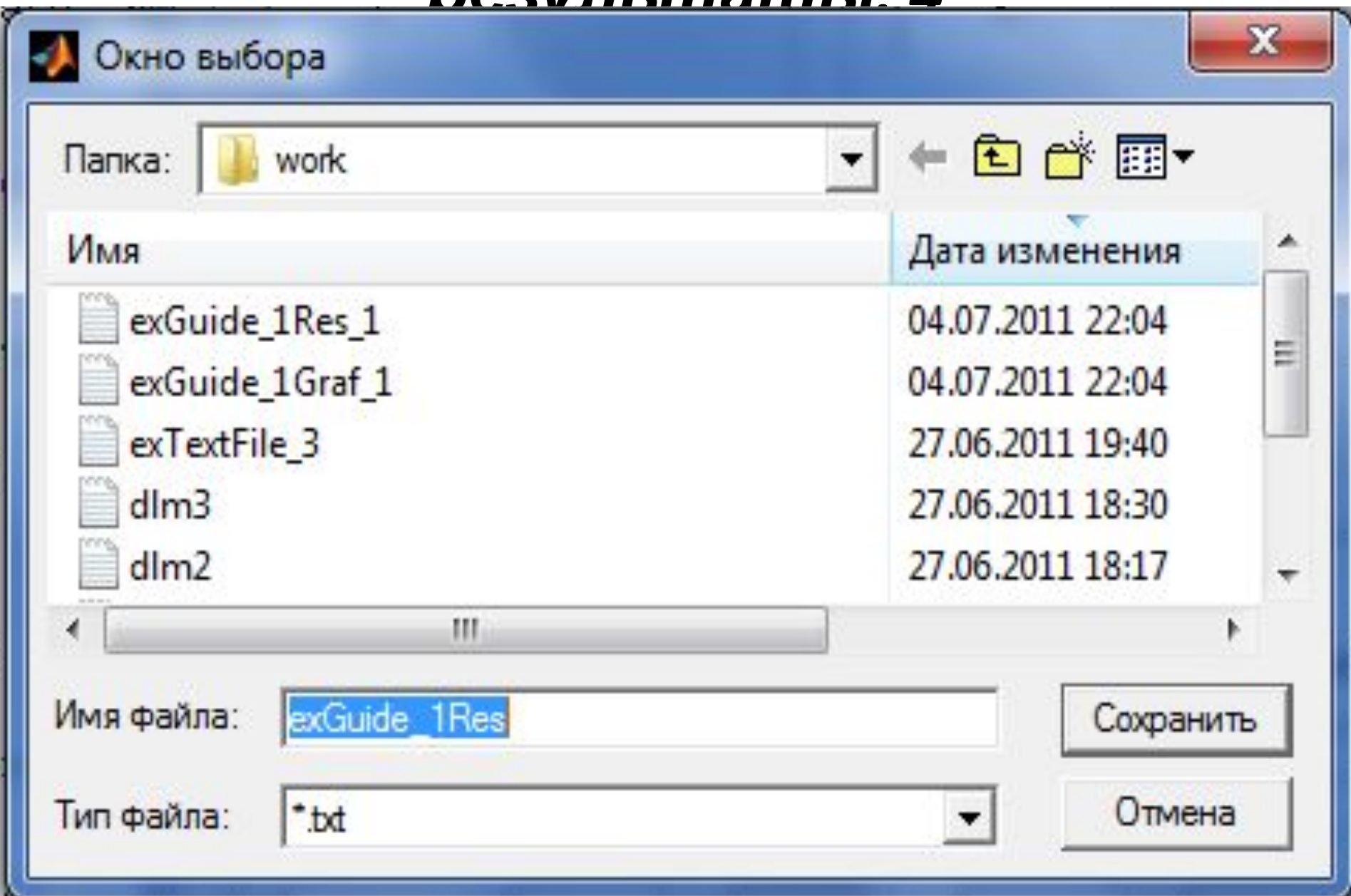
```
KudaZapisat=strcat(p,f);
```

дальнейшие действия так же, как в ранее рассмотренных упражнениях о работе с файлами

```
% --- Executes on button press in pushbutton3. CALCULATE & SAVE
function pushbutton3_Callback(hObject, eventdata, handles)
    %CALCULATE
    [VectorArg]=VvodArg(hObject, eventdata, handles);
    [VectorRes]=ReshitZadachu(VectorArg);
    %OPEN FILE WT
    [f,p]=uiputfile('C:\MATLAB701\work\exGuide_1Res.txt','Окно выбора');
    KudaZapisat=strcat(p,f);
    f=fopen(KudaZapisat,'wt');
    %SAVE VALUES OF ARGUMENTS AND RESULTS
    for i=1:3
        fprintf(f,'%g\t',VectorRes(i));
    end
    for i=1:3
        fprintf(f,'%g\t',VectorArg(i));
    end
end
```

```
%sohranit v fail rezultati vichislenij
fprintf(f, '\nexGuide_1Res\n');
fprintf(f, 'RESULTATI VICHISLENIY\n');
TextString=strcat('Funct_#1\t', 'Funct_#2\t', 'Funct_#3\n');
fprintf(f, TextString);
fprintf(f, '%g\t\t\t', VectorRes(1));
fprintf(f, '%g\t\t\t', VectorRes(2));
fprintf(f, '%g\n', VectorRes(3));
%sohranit v fail ishodnije argumenti
fprintf(f, 'ARGUMENTI VICHISLENIY\n');
TextString=strcat('argument#1_x\t', 'argument#2_y\t', 'argument#3_z\n');
fprintf(f, TextString);
for i=1:3
    fprintf(f, '%g\t\t\t', VectorArg(i));
end
fclose(f);
```

Вычислить и сохранить результаты. 4



Построить и сохранить график. 1

•Для того, чтобы построить график и сразу же сохранить его, предназначена кнопка Plot & Save. В соответствующем этой кнопке тексте программы сначала происходит построение графика. Можно было бы просто поставить вызов функции, уже описанной выше, которая строит график. Но все же более понятно для изучения, если все переменные вычисляются здесь же. Для контроля строится и сам график.

Построить и сохранить график. 2

- Затем, когда все переменные, описывающие график, определены, то открываем файл. Функция `uiputfile` возвращает путь к файлу и имя файла. Затем их объединяем в одну переменную. Записываем в файл количество элементов массивов горизонтальных и вертикальных координат (очевидно они одинакового размера). Затем записываем сами массивы координат точек графика. Потом записываем аргументы, а также номер функции и номер аргумента, для которых построен график. Затем файл закрываем.

```

% --- Executes on button press in pushbutton4. PLOT & SAVE
function pushbutton4_Callback(hObject, eventdata, handles)
    %PLOT
    [VectorArg]=VvodArg(hObject, eventdata, handles);
    NumFun=str2double(get(handles.edit7,'String'));
    NumArg=str2double(get(handles.edit11,'String'));
    MinArg=str2double(get(handles.edit12,'String'));
    StepArg=str2double(get(handles.edit13,'String'));
    MaxArg=str2double(get(handles.edit14,'String'));
    NumberSteps=round((MaxArg-MinArg)/StepArg);
    horis(1)=MinArg;
    VectorArg(NumArg)=horis(1);
    [VectorRes]=ReshitZadachu(VectorArg);
    vert(1)=VectorRes(NumFun);

```

```
for i=2:NumberSteps+1
    horis(i)=horis(i-1)+StepArg;
    VectorArg(NumArg)=horis(i);
    [VectorRes]=ReshitZadachu(VectorArg);
    vert(i)=VectorRes(NumFun);
end
plot(horis,vert,'o-');
set(gca,'XGrid','on');
set(gca,'Ygrid','on');
%OPEN FILE WT
[f,p]=uiputfile('C:\MATLAB701\work\exGuide_1Graf.txt','Окно выбора');
KudaZapisat=strcat(p,f);
f=fopen(KudaZapisat,'wt');
%gotovim dannije dlja avtomaticheskogo chitivanija
fprintf(f,'%d\t',NumberSteps);
for i=1:(NumberSteps+1)
    fprintf(f,'%g\t',horis(i));
end
fprintf(f,'\n');
```


Открыть результаты вычислений из файла. 1

•Для открытия файла с результатами вычислений предназначена кнопка Open Data. В соответствующем ей месте программы опишем необходимые действия. Функция `uigetfile` подготавливает данные об открываемом файле для чтения. При этом открывается окно выбора, где определяется имя файла и путь к нему. Эти данные являются возвращаемыми значениями для функции `uigetfile`. Получив их, объединяем их в одну переменную `OtkudaChitat`

Открыть результаты вычислений из файла. 2

- Затем открываем файл. После открытия файла, считываем интересующие нас данные из файла в переменные с соответствующими названиями. Необходимо теперь вывести значения этих переменных в соответствующие окошки текстового вывода в интерфейсе. Для этого используем функцию `VivodRes` и ранее не использовавшуюся функцию `VivodArgumentovNaEkran`, которую необходимо описать выше описания кнопки.

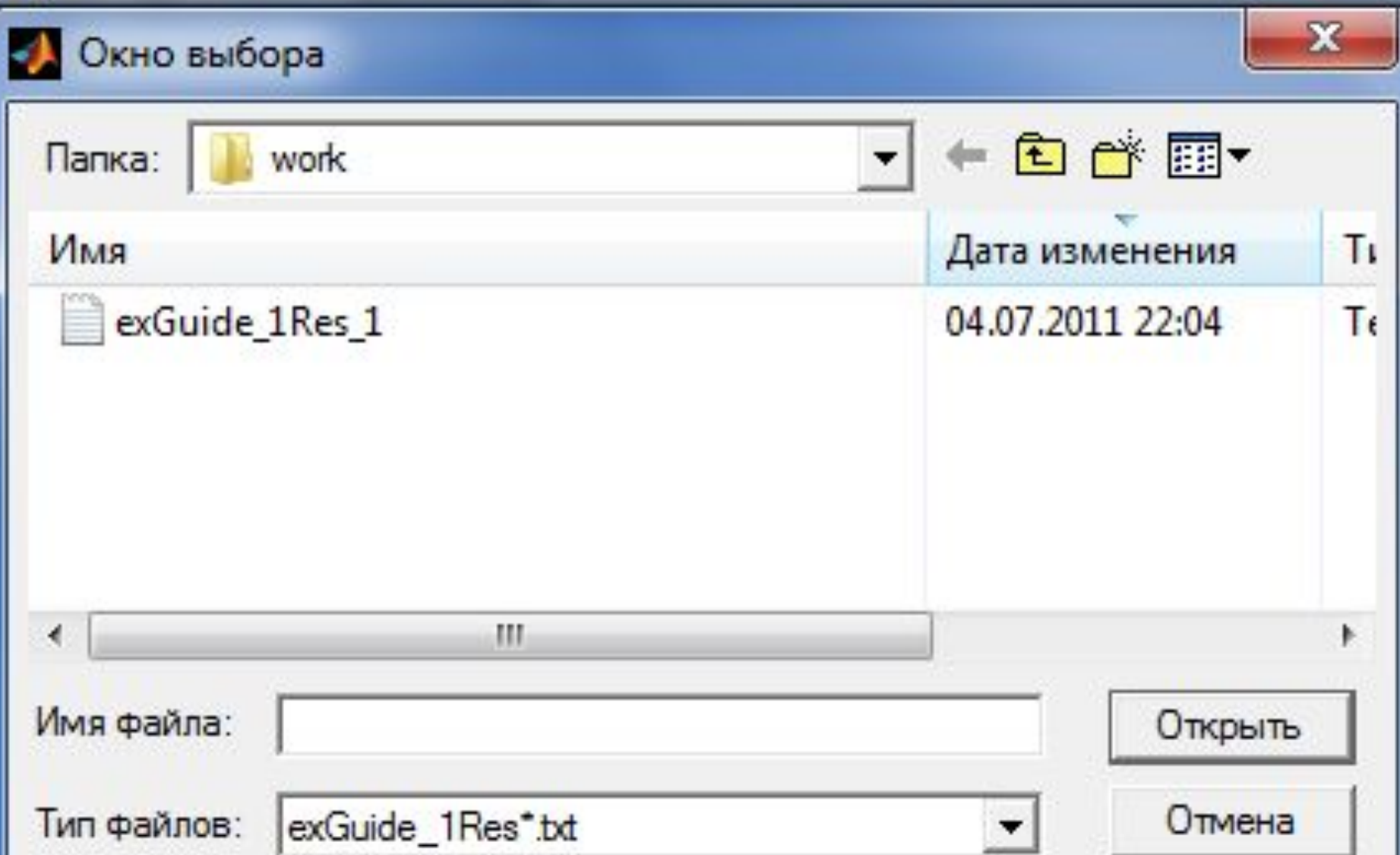
Открыть результаты вычислений из файла. 3

```
function []=VivodArgumentovNaEkran(VectorArg,hObject, eventdata, handles);  
    x=VectorArg(1);  
    y=VectorArg(2);  
    z=VectorArg(3);  
    S=sprintf('%g',x);  
    set(handles.edit1,'String',S);  
    S=sprintf('%g',y);  
    set(handles.edit2,'String',S);  
    S=sprintf('%g',z);  
    set(handles.edit3,'String',S);
```

Открыть результаты вычислений из файла. 4

```
% --- Executes on button press in pushbutton5. OPEN DATA
function pushbutton5_Callback(hObject, eventdata, handles)
    %OPEN FILE RT
    [f,p]=uigetfile('C:\MATLAB701\work\exGuide_1Res*.txt','Окно выбора');
    OtkudaChitat=strcat(p,f);
    f=fopen(OtkudaChitat,'rt');
    %chitat is faila
    for i=1:3
        VectorRes(i)=fscanf(f,'%g',1);
    end
    for i=1:3
        VectorArg(i)=fscanf(f,'%g',1);
    end
    fclose(f);
    %vivod na ekran
    VivodRes(VectorRes,hObject, eventdata, handles);
    VivodArgumentovNaEkran(VectorArg,hObject, eventdata, handles);
```

Открыть результаты вычислений из файла. 5



Открыть график и данные из файла.

1

- Открытие данных, необходимых для построения графика, из файла осуществляется кнопкой Open Data & Plot. В соответствующем ей тексте программы логика действий примерно такая же как и при открытии результатов вычислений из файла. После получения всех необходимых данных из файла, осуществляется построение графика.

Открыть график и данные из файла.

2

- Кроме того, выводятся данные об исходных аргументах и, кроме того, о номере функции и номере аргумента, для которых построен график. При желании можно также доработать это упражнение и выполнить подписи к осям графика в соответствии с названиями аргументов и функций.

Открыть график и данные из

файла ?

```
% --- Executes on button press in pushbutton6. OPEN DATA & PLOT
function pushbutton6_Callback(hObject, eventdata, handles)
    %otkrit grafic is faila
    [f,p]=uigetfile('C:\MATLAB701\work\exGuide_1Graf*.txt','Окно выбора');
    OtkudaChitat=strcat(p,f);
    f=fopen(OtkudaChitat,'rt');
    %chitat is faila
    NumberSteps=fscanf(f,'%d',1);
    for i=1:NumberSteps+1
        horis(i)=fscanf(f,'%g',1);
    end
    for i=1:NumberSteps+1
        vert(i)=fscanf(f,'%g',1);
    end
    for i=1:3
        VectorArg(i)=fscanf(f,'%g',1);
    end
end
```


Открыть график и данные из файла 4

```
NumFun=fscanf(f,'%g',1);
NumArg=fscanf(f,'%g',1);
fclose(f);
%stroim grafic
plot(horis,vert,'o-');
set(gca,'XGrid','on');
set(gca,'YGrid','on');
%vivod na ekran
VivodArgumentovNaEkran(VectorArg,hObject, eventdata, handles);
S=sprintf('%g',NumFun);
set(handles.edit7,'String',S);
S=sprintf('%g',NumArg);
set(handles.edit11,'String',S);
```

Конец презентации