

ЛАБОРАТОРНА РОБОТА №8. Основи програмування та анімація в MatLab.

Організація діалогу з користувачем

Мета: Вивчити основи програмування в середовищі MatLab.

1. Теоретичні відомості

Програмування. Кожного разу при створенні М-файлу ми пишемо комп'ютерну програму, використовуючи мову програмування MatLab. Як відомо (див. лабораторну роботу №2), всі команди MatLab, з яких складаються програми, виконуються *послідовно*. Однак, для розв'язання складніших задач потрібні програми, в яких дії повторюються, а також, залежно від певних умов, виконуються різні частини програми.

Виконання в MatLab схожих дій, що повторюються, здійснюється за допомогою *операторів циклу* for та while. Цикл for призначений для виконання заданої кількості дій, що повторюються, а цикл while – для дій, кількість яких наперед невідома, але відома умова продовження циклу. Записуються ці оператори таким чином:

<pre>for count = start : step : final команди MatLab end</pre>	<pre>while <умова циклу> команди MatLab end</pre>
--	---

Тут count – змінна циклу; start та final – її початкове й кінцеве значення; step – крок, на який збільшується count при кожному наступному вході до циклу for. Цикл for закінчується, якщо count > final; count може приймати цілі й дійсні значення будь-якого знаку, за замовчуванням count=1. Наприклад, цикл for у прикладі 1 виконується 5 разів. Допустимі вкладені цикли (приклад 2) та цикли з векторною змінною (приклад 3 – формування вектора, елементи якого – степені $2^{v(i)}$).

% Приклад 1

```
for i=2:6
    x(i)=2*x(i-1)
end
```

% Приклад 2

```
for i=1:m
    for j=1:n
        A(i,j)=1/(i+j-1);
    end
end
```

% Приклад 3

```
x=[];
for v=[0 2 3 1]
    x=[x 2^v]
end
```

% Приклад 4

```
n=1;
while prod(1:n)<1.e100
    n=n+1;
end
```

Цикл while працює доки виконується <умова циклу>. Для задання умови циклу використовують *операції відношення* (<, >, ==, <=, >=, ~=) та *логічні оператори* (and (&), or (|), not(~)). Для задання порядку виконання логічних операцій потрібно використовувати круглі скобки. Наприклад, знайти найменше число n, значення факторіалу якого записується числом, що містить 100 знаків (приклад 4).

Організація дій, що повторюються, у вигляді циклів робить програму простою та зрозумілою, але часто потрібно виконати певні дії залежно від деяких умов. В цих випадках використовуються *оператори розгалуження*.

Ці оператори дозволяють створити гнучкий алгоритм виконання команд, що розгалужується, в якому при виконанні певних умов працює відповідний блок операторів або команд MatLab.

Застосування умовного оператора виглядає таким чином:

if <умова> команди MatLab end	if <умова> команди MatLab else команди MatLab end	if <умова> команди MatLab elseif <умова> команди MatLab else команди MatLab end
-------------------------------------	---	---

В першому, найпростішому, випадку, якщо умова вірна, тоді виконуються команди MatLab, розміщені між if та end, інакше відбувається перехід до команд, розташованих після end. У двох інших випадках залежно від виконання тої чи іншої умови працює відповідна гілка програми. Якщо всі умови невірні, тоді виконуються команди, розташовані після else. При записі умови використовуються операції відношення та логічні оператори. В прикладі 5 розглянута М-функція fun1 обчислення виразу $\sqrt{x^2 - 1}$, що працює для будь-яких значень x, причому для $-1 < x < 1$ результат є комплексним числом.

% Приклад 5

```
function f=fun1(x)
% обчислює sqrt(x^2-1)
% перевірка аргументу
function f=fun1(x)
if abs(x)<1
    warning('результат комплексний');
end
% обчислення функції
f=sqrt(x^2-1);
```

% Приклад 6

```
switch num
case -1
    disp('мінус один')
case 0
    disp('нуль')
case 1
    disp('плюс один')
otherwise
    disp('інше значення')
end
```

% Приклад 7

```
switch var
case 1
    disp('1')
case {2,3,4}
    disp('2, або 3, або 4')
case 5
    disp('5')
otherwise
    disp('щось інше')
end
```

Оператор перемикавання (вибору) switch виконує розгалуження залежно від значень деякої змінної або виразу і має вигляд: switch <вираз>

```
case <значення_1>
    команди MatLab
...
case <значення_n>
    команди MatLab
otherwise
    команди MatLab
end
```

де <вираз> – це скаляр або рядок. Оператор switch працює, порівнюючи значення обчисленого виразу зі значенням груп case. Для числових виразів команди MatLab виконуються, якщо <значення> == <вираз>, для рядкових виразів – якщо <вираз> істинний. Якщо збігу немає, тоді виконується група otherwise, якщо вона існує. Наприклад, оператор switch у прикладі 6 перевіряє змінну num; якщо num дорівнює -1, 0 або 1, тоді на екран виводиться відповідне повідомлення, інакше виконання переходить до оператора otherwise. Оператор switch може використовувати множинну умову в єдиній групі case за допомогою включення виразу case, якщо вираз для цієї умови записаний у вигляді масиву елементів (приклад 7). Оператор switch зручно застосовувати у випадках, коли є відповідність між дискретними значеннями деякої змінної та наступними діями.

Добре написана програма запобігає помилковим діям, які призводять до її дострокового припинення. Перевірка входних і вихідних аргументів М-функції є одним із способів контролю. MatLab надає ще два способи: переривання циклу й обробку виключних ситуацій. Функція break *перериває виконання циклів* for та while. Всередині будь-якого з цих циклів можна використовувати break для того, щоб змусити програму MatLab зупинити виконання циклу та перейти до наступного після кінця циклу рядка. У випадку вкладених циклів переривання можливе лише з «найглибшого» внутрішнього циклу. Проте команда break найчастіше використовується у складі виразу if.

Конструкція try...catch дозволяє обійти *виключні ситуації*, що призводять до помилки, та провести певні дії у випадку їх виникнення. Схема використання try...catch така:

```

    % оператори, виконання яких може призвести до помилки
catch
    % оператори, які потрібно виконати при виникненні помилки в блоці між try та catch
end

```

Команда return виконує *повернення* або до викликаючої функції, або до режиму роботи з клавіатурою, а також дозволяє завершити режим роботи з клавіатурою.

Практично в усіх мовах програмування є оператори, аналогічні вищенаведеним операторам циклу та їх переривання, розгалуження й обробки виключних ситуацій та повернення.

У MatLab є можливість сформувати необхідну команду у вигляді рядка символів, а потім виконати її, як звичайну команду, набрану в командному вікні. Для цього призначена вбудована функція eval(<текст>). Метод задавання команд у вигляді текстових рядків та їх виконання за допомогою цієї функції, використовується при написанні m-файлів, оскільки такий метод дозволяє реалізовувати ефективніші алгоритми розв'язання задач.

Наприклад, `eval('2*2+3')` виводить на екран 7. Аргументом функції може бути також рядкова змінна.

Команда `eval('<ім'я скрипт-файлу>')` виведе на екран результат виконання операторів, розміщених у файлі з відповідним ім'ям.

Функція `feval` застосовується для обчислення зовнішніх функцій, значення яких потрібно переобчислювати на кожному кроці у тілі іншої процедури (функції). Так, функція `feval('<ім'я функції>', x1,...,xn)` передає аргументи x_1, \dots, x_n функції, яка викликається.

Анімація. Як відомо з лабораторної роботи №2, для візуалізації руху точки по траєкторії використовується команда `comet`. При цьому рухома точка нагадує ядро комети з хвостом. Є ще одна команда, яка дозволяє спостерігати рух точки, але вже у тривимірному просторі. Це команда `comet3`.

В системі MatLab немає процедур, що дозволяють безпосередньо створювати анімовані зображення, проте є засоби, що дозволяють запускати мультфільми, кадрами яких є заздалегідь побудовані графіки. Тому для більш складних випадків анімації можливе застосування техніки мультиплікації. Вона зводиться до побудови ряду кадрів зображення, причому кожен кадр з'являється на деякий час, потім стирається і замінюється на новий кадр, який дещо відрізняється від попереднього. Якщо ця відмінність є незначною, то створюється ілюзія плавного переміщення об'єкта.

Основні команди, що реалізують анімацію в MatLab, такі: `capture` – захоплення графічної фігури; `getframe` – створення фрейму (кадру) для анімації; `movie` – виконання анімації; `rotate` – обертання фігури; `frame2im` – перетворення фрейма (кадра) на графічний образ; `im2frame` – перетворення графічного образу на фрейм (кадр).

Побудова анімації в MatLab здійснюється у в два етапи. Спочатку організується цикл вигляду: `for k=1:n`

```

        Довільний блок команд,
        що забезпечують побудову
        потрібного графіка;
        M(k) = getframe;
    end

```

де n – число кадрів майбутнього мультфільму, M – масив, в який функцією `getframe` послідовно складаються кадри.

Потім зібраний мультфільм програється функцією `movie`, при виклику якої першим аргументом вказується масив кадрів, другим та третім може вказуватись, відповідно, кількість повторних циклів анімації та швидкість відтворення (число кадрів за секунду).

Наведені нижче фрагменти програмного коду демонструють побудову двовимірної (приклад 8) та тривимірної (приклад 9) анімації.

% Приклад 8

```
% Анімація функції на площині
% формування масиву рівновіддалених вузлів
t=linspace(0,pi,300);
x=0:0.01:1;
for k=1:300
    y=sin(2*t(k)).*sin(3*pi*x);
    plot(x,y);
    axis([0 1 -1 1]);
    grid on;
    M(k)=getframe;
end
% Програвання побудованої анімації
movie(M);
```

% Приклад 9

```
% Анімація функції у просторі
[x,y]=meshgrid([0:0.01:1]);
t=linspace(0,pi,100);
for k=1:100
    z=sin(2*pi*x).*sin(3*pi*y).*sin(2*t(k));
    surf(x,y,z);
    shading flat;
    colormap(jet);
    axis([0 1 0 1 -1 1]);
    M(k)=getframe;
end
% Програвання анімації
movie(M)
```

Організація діалогу з користувачем. Для забезпечення зручної взаємодії з користувачем в процесі виконання М-файлів в системі MatLab застосовують такі функції, як: `disp` – виведення значень вказаної змінної або вказаного тексту у командне вікно, `input` – введення інформації користувачем з клавіатури у діалоговому режимі, `pause` – призупинення виконання М-файлу, `menu` – створення меню користувача тощо.

Створення меню користувача. Найпростішим способом виведення на екран вікна з випадаючим меню, що має заголовок і кнопки з назвами пунктів меню, є функція `menu`. Кількість кнопок, яке дорівнює кількості пунктів меню, визначається кількістю параметрів функції `menu`: `k=menu('<заголовок меню>','<назв. пункту 1>',...,'<назв. пункту n>')` Номер обраної кнопки є результатом функції `menu`.

Нехай, наприклад, необхідно побудувати графік функції $x = \sin t$ на заданому інтервалі $[0, 2\pi]$ кольором, вибраним з меню. Script-файл розв'язання даної задачі наведений у прикладі 10. Функція `menu` виводить на екран вікно вигляду (рис.8.1):

% Приклад 10

```
% Побудова графіку функції кольором,
% вибраним в меню
color=['r' 'g' 'b'];
n=1;
while n<4
    n=menu('меню','червоний','зелений','синій','вихід');
    if(n==4) break
    else feval('fun',color(n));
end
end
```

function fun(c)

```
% Побудова графіка вибраним кольором c
t=0:0.1:2*pi;
s=sin(t);
plot(t,s,c)
end
```

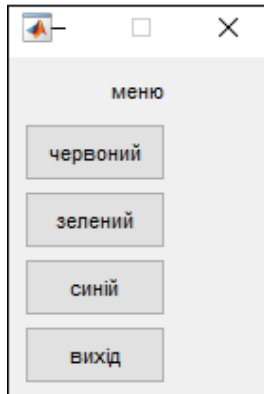


Рис.8.1 – Меню користувача

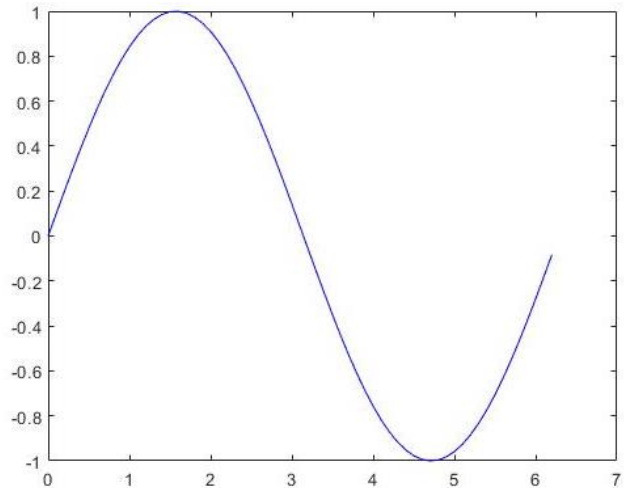


Рис.8.2 – Результат вибору синього кольору

При натисканні на одну з кнопок з назвою кольору (наприклад, «синій»), номер цієї кнопки ($n=3$) стає результатом функції `menu` і потрапляє у змінну n . Вектор `color` містить атрибути кольору для побудови графіка. За номером n необхідний атрибут вибирається з масиву `color` і вказується в команді `plot` функції побудови графіка синуса `fun`. В результаті, вибравши потрібний пункт меню, отримаємо графік функції, побудований вибраним кольором (рис.8.2).

Функція `feval` передає відповідний вибраний в меню аргумент кольору `color(n)` функції побудови графіка `fun`.

Іншим способом створення графічного інтерфейсу користувача є робота в середовищі `GUIDE`, що входить до складу `MatLab`. Робота в цьому середовищі полягає в тому, що елементи управління (кнопки, списки, що розкриваються, тощо) розміщуються за допомогою миші, а потім програмуються події, які виникають при зверненні користувача до даних елементів управління. Створюваний об'єкт може складатися, як з одного основного вікна, так і декількох вікон та здійснювати виведення графічної і текстової інформації в основне вікно та в окремі вікна. Ряд функцій `MatLab` призначений для створення стандартних діалогових вікон відкриття і збереження файлу, друку, вибору шрифту, вікна для введення даних та ін., якими можна користуватися у власних програмах.

Для створення програми з користувацьким меню застосовують два методи:

- команда `guide` у вікні *Command Window*;
- вибір в меню *HomeFileNewGraphical User Interface (GUI)*.

В обох випадках генерується діалогове вікно створення GUI, яке містить дві вкладки: *Creating New GUI* – створення нового GUI та *Open Existing GUI* – відкриття існуючого GUI.

На вкладці створення нового GUI є дві панелі: ліворуч список шаблонів для вибору, праворуч – попередній перегляд. Можна вибрати один із запропонованих шаблонів (рис.8.3):

- *Blank GUI (Default)* – порожній GUI (за замовчуванням);
- *GUI with Uicontrols* – GUI з елементами управління інтерфейсом користувача;
- *GUI with Axes and Menu* – GUI з об'єктом Axes (осі) і меню;
- *Modal Question Dialog* – модальне вікно діалогу з обов'язковою відповіддю на питання.

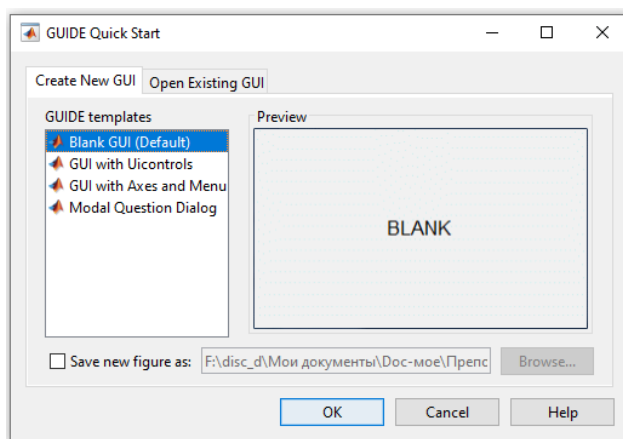


Рис.8.3 – Вікно створення GUIDE

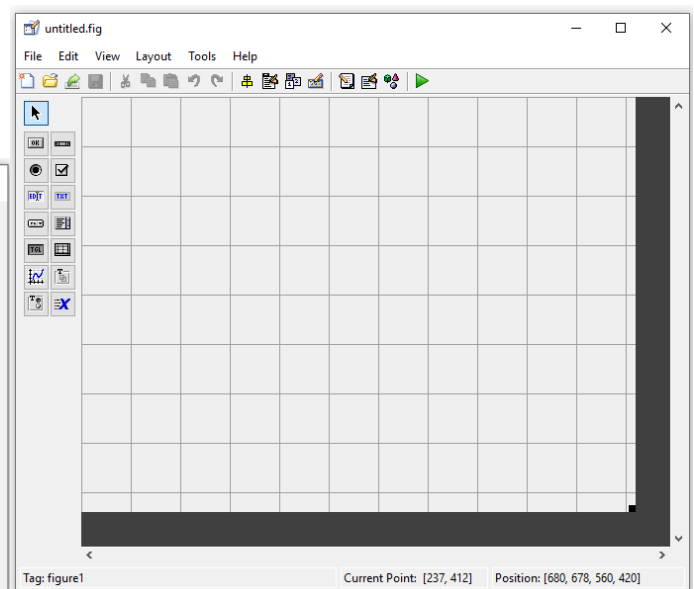


Рис.8.4 – Вікно редактору GUI

При роботі з GUIDE:

- компоненти GUI обираються з палітри компонентів інструмента GUIDE мишкою і переносяться до робочої області GUI;
- GUIDE зберігає GUI у двох файлах: *.fig (рисунок) та *.m (М-файл).
- М-файл, що керує роботою GUI, генерується автоматично при збереженні проекту.

Вікно редактора GUI (рис.8.4) містить: меню; панель інструментів; набір доступних компонентів ліворуч; робоче поле (*Layout Area*) з координатної сіткою.

Команда *File→Preferences* викликає вікно вподобань MatLab, в якому будуть відображені параметри GUI за замовчуванням. У ньому потрібно встановити позначку відображення імен компонентів в палітрі (*Show names in component palette*) (рис.8.5).

Після цього вікно редактора набуде іншого вигляду (рис.8.6). При цьому, звичайно, розміри робочого поля зменшуються.

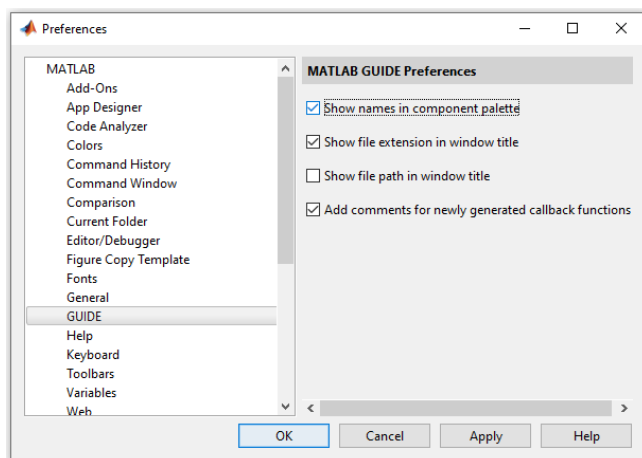


Рис.8.5 – Вікно інструментів

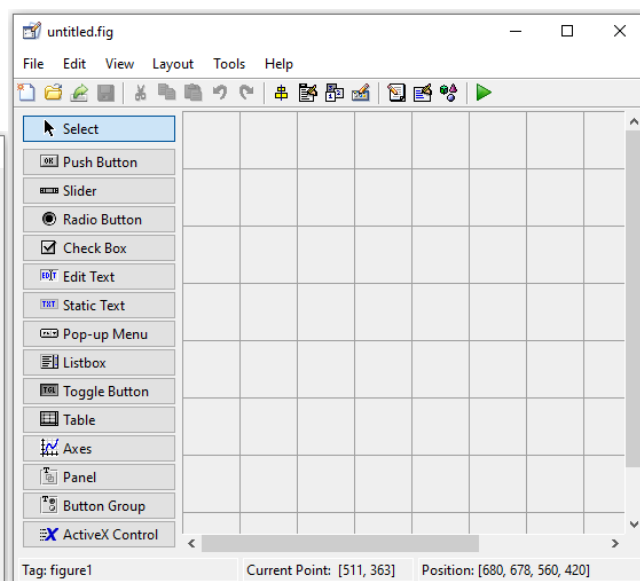


Рис.8.6 – Новий вигляд вікна редактору GUI

Для зміни властивостей компонентів можна використовувати різні способи:

- прямо у вікні GUI, переміщуючи компонент або змінюючи його розміри мишкою, можна змінити розміри та позицію;
- використовувати браузер об'єктів *Object Browser*, в якому обирається потрібний об'єкт. При цьому викликається інспектор об'єктів;
- *Property Inspector* – інспектор об'єктів (відповідна піктограма розміщена на панелі інструментів вікна редактору GUI), в якому відображаються властивості компонентів. Якщо обирається конкретний компонент, то в інспекторі він виділяється кольором;
- пряме редагування коду в файлі GUI.

Інспектор об'єктів (рис.8.7) призначений для встановлення доступних на етапі розробки властивостей, які визначають поведінку компонента під час виконання програми. Вікно інспектора об'єктів складається з двох основних частин: в лівій частині розміщені найменування властивостей, закріплені за виділеним об'єктом, у правій частині – поля значень, доступні для редагування – вибору або введення значення властивості.

Розглянемо приклад створення простого об'єкту за допомогою GUIDE – графічного інтерфейсу first, запуск якого з командного рядка призводив би до появи програмного вікна з кнопкою ОК. Після натискання на кнопку в вікно повинен виводитися текст «Hi, student!».

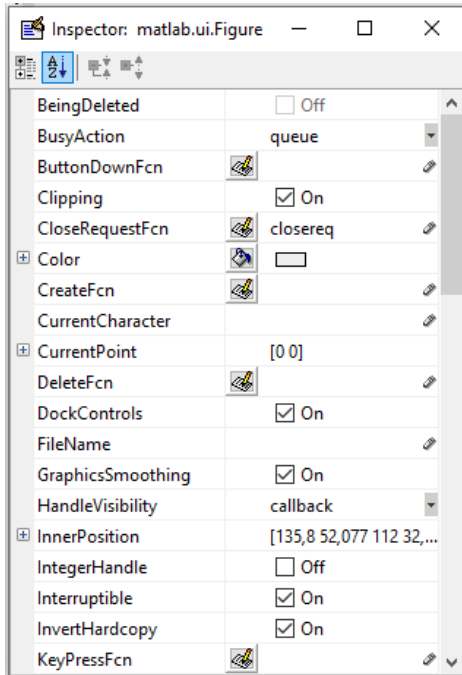


Рис.8.7 – Інспектор об'єктів

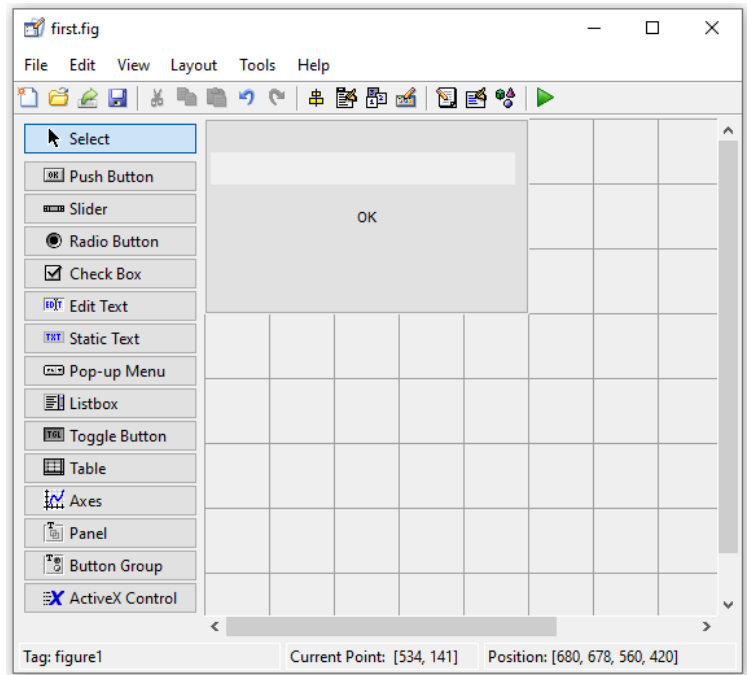


Рис.8.8 – Вікно редактору GUI із заготовкою first

Для розв'язання задачі виконаємо такий алгоритм:

1. Перейти до середовища GUIDE, виконавши команду: `guide`
2. Вибрати на вкладці *Create New GUI* рядок *Blank GUI* і натиснути *OK*.
3. Додати на заготовку вікна програми кнопку за допомогою інструменту *Push Button*.
4. Задати тег (ім'я) об'єкту, який ідентифікуватиме даний елемент серед інших об'єктів. Для задання імені необхідно перейти до інспектору об'єктів (клацнути лівою клавішею миші по відповідній піктограмі на панелі інструментів вікна редактору GUI). Далі знайти в лівому стовпчику таблиці властивість *Tag* і в області вводу праворуч від нього змінити поточне значення *pushbutton1* на *Button_ok* та натиснути *Enter*.
5. Змінити надпис на кнопці: обрати властивість *String* у вікні інспектора об'єктів і замість *Push Button* ввести *OK*.
6. Задати також значення інших властивостей кнопки (колір фону, колір та розмір надпису тощо) у вікні інспектора об'єктів – *BackgroundColor*, *ForegroundColor*, *FontSize* та ін., відповідно.
7. Для розміщення тексту розташувати у створеному вікні програми елемент *Static Text*. Задати цьому елементу тег *Text_ok* області виводу тексту, а у властивість *String* встановити порожній рядок, просто видаливши в інспекторі об'єктів те значення, яке було за замовчуванням.

8. Далі в меню *File* середовища GUIDE обрати пункт *Save as* і у з'явившомуся діалоговому вікні збереження файлу вибрати папку або створити нову та задати ім'я файлу *first*, після чого автоматично додасться розширення *fig* (рис.8.8). За замовчуванням, створений об'єкт складається з двох файлів: з розширенням *fig* (графічного вікна з розміщеними на ньому елементами управління) та з розширенням *m* (файлу з функціями, які обробляють різні події, що виникають в ході взаємодії програми з користувачем).

9. При натисканні користувачем на кнопку ОК повинна виникати подія *Callback*, яка виводитиме текст у вікно, для чого необхідно встановити властивість *String* області виводу тексту у значення «*Hi, student!*» та задати колір і розмір шрифту. Для цього слід перейти до заготовки вікна програми і в контекстному меню кнопки обрати в пункті *View Callbacks* підпункт *Callback*. Відбудеться перехід у редакторі М-файлів до підфункції обробки події *Button_ok_Callback*, заголовок якої та коментарі генерується автоматично:

```
% --- Executes on button press in Button_ok.
function Button_ok_Callback(hObject, eventdata, handles)
% hObject    handle to Button_ok (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

10. Видалити коментарі та після заголовку підфункції *Button_ok_Callback* розмістити оператори, які змінюють значення властивості *String* області виводу тексту (об'єкту *Uicontrol*), встановлюючи його у «*Hi, student!*», а також значення кольору та розміру шрифту (рис.8.9):

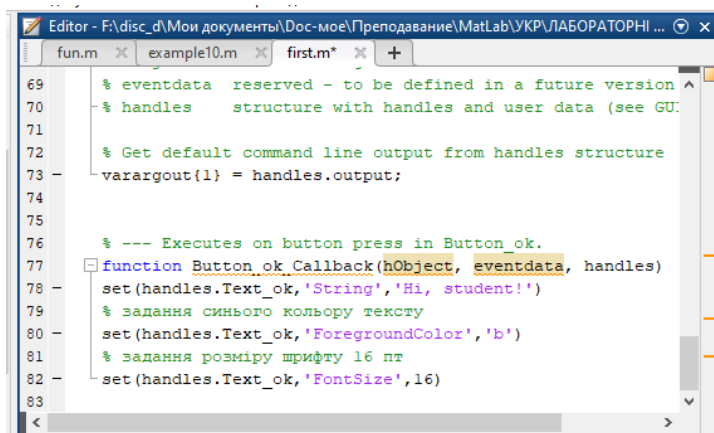


Рис.8.9 – Вікно редактору М-файлів
з кодом програми *first*

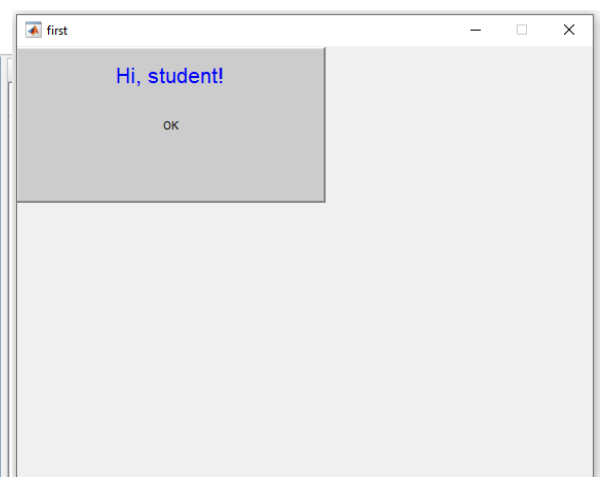


Рис.8.10 – Вікно з результатом
натискання на кнопку ОК

11. Запустити програму, скориставшись кнопкою *Run* на панелі управління середовища GUIDE (рис.8.10).

2. Порядок виконання роботи

Виконати наступне завдання згідно з номером свого варіанту.

1. а) Вивести множину кривих для $x \in [0, 2\pi]$, заданих функцією, що залежить від параметра $y(x, a) = e^{-ax} \sin x$, для значень параметра від -0.1 до 0.1 .

б) Анімувати функцію $y = e^{-x} \sin(20x)$, $x \in [0, 1]$, на площині.

2. а) Замінити в рядку одні символи на інші. Початковий рядок, символи, що замінюються, та зразки заміни вводити з екрану після відповідних запитів). Перевірити правильність роботи програми на прикладі.

б) Анімувати поверхню з лаб.2,б. Оберіть спосіб візуалізації поверхні зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

3. а) Вивести таблицю значень синуса при 21 значенні аргументу від 0.2 до 4 з кроком 0.2 різними операторами циклу. Вивести також позначення стовпців.

б) Анімувати функцію $y = (1 - x) \cos(20x)$, $x \in [0, 2]$, на площині.

4. а) Заповнити в діалоговому режимі й вивести на екран масив структур з інформацією про студентів вашої групи в такому вигляді: номер, прізвище, ім'я, оцінки по n предметам (взяти $n=3$). Знайти середній бал успішності в групі.

б) Анімувати поверхню $x = (x^2 + y^2) \cos(x^2 + y^2)$ для $x, y \in [-1; 1]$.

5. а) Обчислити суму ряду $S(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$ для $x = \pi/2$. Накопичувати суму, доти, поки доданки більші по модулю, ніж 10^{-10} . Для перевірки побудувати графіки функцій $S(x)$ та $\sin(x)$ на відрізку $[-\pi, \pi]$ в одному вікні.

б) Візуалізувати процес обертання сфери з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

6. а) Написати М-функцію, яка по коефіцієнтах квадратного рівняння знаходить лише його дійсні корні, а для комплексних видає повідомлення про помилку. Перевірити роботу функції для рівнянь $x^2 + x - 6 = 0$ та $x^2 + 1 = 0$.

б) Візуалізувати процес обертання циліндра з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

7. а) Нехай в деякому файлі зберігаються такі дані: рік навчання, кількість предметів у поточному семестрі, кількість екзаменів, кількість заліків. Написати фрагмент програми, яка: зчитує у деяку змінну дані з файлу, будує в різних вікнах дво- та тривимірну кругові діаграми в палітрі pink та відокремлює від них сектор 1, а потім проводить

обчислення, не пов'язані зі зчитаними даними, наприклад, обчислює вираз

$$t = e^{-2.5} (\ln 11.3)^{0.3} - \sqrt{\frac{\sin 2.45\pi + \cos 3.78\pi}{\operatorname{tg} 3.3}}. \quad \text{Передбачити ситуацію відсутності}$$

потрібного файлу, але продовження виконання програми.

б) Візуалізувати процес обертання конусу з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

8. а) Заповнити в діалоговому режимі масив структур з інформацією про студентів вашої групи в такому вигляді: номер, прізвище, ім'я, оцінки по 4 предметах. Знайти середній бал успішності для кожного студента й додати до початкового масиву структур. Вивести отриманий масив структур.

б) Візуалізувати процес обертання гіперболоїду з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

9. а) Створити вікно меню користувача, що дозволяє, залежно від вибору, будувати графік функції $z = xe^{(-x^2-y^2)}$ у вигляді: сітчастої (каркасної), суцільної й затіненої поверхонь, а також двовимірних ліній рівня. Передбачити можливість повторного вибору.

б) Візуалізувати процес обертання об'ємної фігури з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

10. а) Дослідити границі спектру матриці Гільберта для різних порядків матриці (від 1 до n) та вивести результат для n=12 у вигляді графіків значень максимального й мінімального власних чисел залежно від розміру матриці.

б) Візуалізувати процес обертання об'ємної фігури з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

11. а) Створити функцію, яка залежно від значення аргументу візуалізує функцію $e^{-|xy|} \sin \pi x \cdot \cos \pi x^2$ різними способами: каркасною, суцільною поверхнею, лініями рівня, освітленою поверхнею, або в різних вікнах відразу всіма наведеними способами, $x, y \in [-2, 2]$. Передбачити можливість вводу кольорової палітри.

б) Візуалізувати процес обертання поверхні з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

12. а) Знайти кількість і позиції всіх цифр, що входять у введеній з екрана рядок. Перевірити правильність роботи програми на прикладі.

б) Візуалізувати процес обертання фігури
$$\begin{cases} x = \frac{u}{2} - \frac{u^3}{6} + \frac{uv^2}{2} \\ y = \frac{-v}{2} + \frac{v^3}{6} - \frac{vu^2}{2} \\ z = \frac{u^2}{2} - \frac{v^2}{2} \end{cases}, \text{ для } u, v \in [-2, 2]$$

навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні.

13. а) Створити функцію для обчислення таких сум: $S_1 = \sum_{k=1}^{10} \frac{1}{k!}$; $S_2(x) = \sum_{k=1}^{10} \frac{x^k}{k!}$ для

$x=1.5$.

б) Візуалізувати процес обертання тора з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

14. а) Обчислити значення функції $y(x) = \begin{cases} \sin x - 1, & x < -\pi \\ x/\pi, & -\pi \leq x < \pi \\ -\cos x, & x \geq \pi \end{cases}$ та побудувати її графік

на відрізку $[-3\pi, 3\pi]$. Проаналізувати результати використання функцій fplot та plot для побудови графіка функції $y(x)$.

б) Функція двох змінних $f_t(x, y) = \sin t \frac{\sin(x^2 + y^2)}{x^2 + y^2}$, $t, x, y \in [-\pi, \pi]$, має параметр t .

Побудувати просторову анімацію (рух мембрани) за умови, що параметр t змінюється в інтервалі $[0, 2\pi]$. Оберіть спосіб візуалізації поверхні зі згладжуванням кольорів. Підсвічення поверхні здійснити за допомогою нескінченно віддаленого джерела білого світла. Використати блискучий матеріал поверхні.

15. а) Побудувати фігури Лісажу для різних відношень ω_1/ω_2 і для різних значень амплітуд a_1 та a_2 . В заголовки відповідних графіків виводити відношення ω_1/ω_2 . Для визначеності взяти $a_1=1.2$, $a_2=1.0$, $\omega_1/\omega_2 = 1.25, 1.5, 1.75, 2.0$, $t \in [0, 15]$.

б) Візуалізувати процес обертання фігури
$$\begin{cases} x = \cos u + v \cos\left(\frac{u}{2}\right) \cos u \\ y = \sin u + v \cos\left(\frac{u}{2}\right) \sin u \\ z = v \sin\left(\frac{u}{2}\right) \end{cases}, \text{ для}$$

$u \in [0, 2\pi]$, $v \in [-0.1, 0.1]$ навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні.

16. а) Написати функцію, що обчислює \min , \max матриці та суму її елементів, незалежно від розмірності двома способами (за допомогою функцій створення матриць спеціального вигляду, а також користуючись операторами циклу й розгалуження).

б) Візуалізувати рух точки по траєкторії $x(t) = t^2 \cos(\cos(5t))$, $y(t) = t^2 \sin(\sin(5t))$, $t \in [0, 3\pi]$, $\Delta t = 0.01$.

17. а) Створити М-функцію, яка залежно від значення відповідного аргументу візуалізує функцію $z(x, y) = 4 \sin 2\pi x \cdot \cos 1.5\pi y \cdot (1-x)^2 \cdot y \cdot (1-y)$ різними способами: каркасною, суцільною поверхнями, двовимірними лініями рівня, освітленою поверхнею, або в різних вікнах відразу всіма наведеними способами, $x \in [-1, 1]$, $y \in [0, 1]$. Передбачити можливість вводу кольорової палітри.

б) Анімувати функцію $y = \sqrt{x} \sin^2(10x)$, $x \in [0, 4]$, на площині.

18. а) Різними способами вивести в таблицю значення функції

$$y(x) = \frac{\sin^2(x)}{1 + \cos x} + e^{-x} \ln x \text{ при 21 значенні аргументу на відрізку } [2, 3] \text{ з кроком } 0.05.$$

Вивести позначення стовпців.

б) Візуалізувати рух точки по траєкторії $x(t) = \cos^2(t) \cos(\cos(10t))$, $y(t) = \sin^2(t) \sin(\sin(10t))$, $t \in [0, 3\pi]$, $\Delta t = 0.01$.

19. а) Створити функцію, що заповнює незалежно від розмірності квадратну матрицю

$$\text{вигляду } A = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix} \text{ двома способами (за допомогою функцій створення матриць}$$

спеціального вигляду, а також користуючись операторами циклу й розгалуження). Результат вивести для матриці 7×7 .

б) Візуалізувати процес обертання фігури з лаб.4,б навколо координатних осей зі згладжуванням кольорів та підсвіченням поверхні за допомогою джерела світла.

20. а) Створити функцію, що заповнює незалежно від розмірності квадратну матрицю

$$\text{вигляду } M = \begin{pmatrix} 1 & 0 & 0 & 2 & 2 & 2 \\ 0 & 1 & 0 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 & 2 & 2 \\ 3 & 3 & 3 & 4 & 0 & 0 \\ 3 & 3 & 3 & 0 & 4 & 0 \\ 3 & 3 & 3 & 0 & 0 & 4 \end{pmatrix} \text{ двома способами (за допомогою функцій створення}$$

матриць спеціального вигляду, а також користуючись операторами циклу й розгалуження), та розраховує суму елементів матриці. Результат вивести для матриці 8×8 .

б) Анімувати функцію $y = \sin^2 x \sin(10x)$, $x \in [0, \pi]$, на площині.

21. а) Заповнити в діалоговому режимі масив структур з інформацією про студентів вашої групи в такому вигляді: номер, прізвище, ім'я, оцінки по 5 предметах. Сформувати новий масив структур з інформацією про студентів, середній бал успішності яких більше 4, в який додати середній бал успішності. Вивести отриманий масив структур.

б) Візуалізувати процес обертання сітчастої поверхні
$$\begin{cases} x = u \\ y = v, \text{ для } u, v \in [-40, 40], \\ z = c \end{cases}$$

$c=10$, навколо координатних осей.

22. а) Порівняти різні способи множення двох випадкових матриць $n \times n$ в MatLab.

б) Візуалізувати процес обертання каркасної поверхні із заливкою
$$\begin{cases} x = \text{airy}(u)\cos(v) \\ y = \cos(u)\sin(v) \\ z = \sin(u)\text{ones}(\text{size}(v)) \end{cases} \quad (u, v \in [0, 2\pi], \text{ airy} - \text{функція Єйрі})$$
 навколо координатних осей.

3. Контрольні питання

1. Які оператори циклу, що використовуються в MatLab, Ви знаєте? В яких випадках вони застосовуються?

2. Назвіть оператори розгалуження, що використовуються в MatLab. В яких випадках вони застосовуються?

3. Які існують способи попередження помилкових дій, що призводять до дострокового припинення програм в MatLab? За допомогою яких операторів вони реалізуються?

4. Як створити анімацію в MatLab?

5. Введення та виведення інформації у діалоговому режимі

6. Створення меню користувача в MatLab.