

О. Г. Ревинская

# ОСНОВЫ ПРОГРАММИРОВАНИЯ В MatLab



- *MatLab как первый/базовый язык программирования*
- *Графические возможности MatLab*
- *Интерфейс современных версий MatLab*
- *Большое количество примеров*



УЧЕБНОЕ ПОСОБИЕ



ozon.ru

О. Г. Ревинская

# ОСНОВЫ ПРОГРАММИРОВАНИЯ В MatLab

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки: 03.03.02 — «Физика»

Санкт-Петербург

«БХВ-Петербург»

2016

УДК 004.43(075.8)  
ББК 32.973.26-018.1я73  
P32

**Ревинская О. Г.**

P32 Основы программирования в MatLab: учеб. пособие. —  
СПб.: БХВ-Петербург, 2016. — 208 с.: ил. —  
(Учебное пособие)

ISBN 978-5-9775-3564-9

Учебное пособие предназначено для изучения основ программирования в среде MatLab студентами младших курсов. Пособие содержит: описание основных элементов и структур языка, стандартные функции, базовые конструкции программирования. Рассмотрены графические возможности MatLab и работа с файлами. Теоретический материал сопровождается большим количеством примеров использования различных конструкций языка при обработке числовых данных.

*Для студентов и преподавателей  
классических и технических университетов*

УДК 004.43(075.8)  
ББК 32.973.26-018.1я73

**РЕЦЕНЗЕНТЫ:**

- С. Н. Колупаева*, д-р физ.-мат. наук, зав. кафедрой прикладной математики  
Томского государственного архитектурно-строительного университета;  
*А. Ф. Тузовский*, д-р техн. наук, проф. кафедры оптимизации систем управления  
Национального исследовательского Томского политехнического университета

Подписано в печать 31.03.16.  
Формат 60×90<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 13.  
Тираж 500 экз. Заказ №  
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.  
Первая Академическая типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3564-9

© Ревинская О. Г., 2016  
© Издательство "БХВ-Петербург", 2016

# ОГЛАВЛЕНИЕ

Предисловие .....	6
<b>Глава 1. Среда разработки программных решений MatLab .....</b>	<b>7</b>
1.1. Запуск среды MatLab.....	7
1.2. Структура основного окна среды MatLab .....	7
1.2.1. Панель инструментов среды MatLab.....	8
1.2.2. Командное окно среды MatLab.....	10
1.2.3. Информационные окна .....	11
1.3. Создание m-файла .....	11
1.4. Редактирование программы.....	12
1.5. Сохранение программы на диске .....	13
1.6. Выполнение программы.....	14
1.7. Просмотр результатов работы программы.....	15
1.8. Встроенная справочная система.....	15
<b>Глава 2. Основные элементы языка программирования</b>	
<b>MatLab.....</b>	<b>16</b>
2.1. Структура программы .....	16
2.2. Идентификаторы.....	17
2.3. Типы данных .....	17
2.4. Константы и переменные.....	18
2.5. Операции .....	19
2.6. Комментарии.....	21
<b>Глава 3. Элементы структурного программирования</b>	
<b>в MatLab.....</b>	<b>22</b>
3.1. Ввод информации с клавиатуры в диалоговом режиме .....	23
3.2. Вывод информации .....	24
3.3. Оператор присвоения .....	29
3.4. Оператор двоеточие (:)	32
3.5. Операторы условного перехода.....	32
3.5.1. Оператор условного перехода IF .....	32
3.5.2. Оператор выбора SWITCH.....	36
3.6. Операторы цикла .....	40
3.6.1. Оператор FOR-цикла .....	40
3.6.2. Оператор цикла WHILE.....	41
3.6.3. Особенности использования операторов циклов FOR, WHILE .....	42
3.6.4. Вычисление сумм и произведений .....	46

---

---

3.7. Массивы.....	49
3.7.1. Ввод массива с клавиатуры.....	51
3.7.2. Вывод значений, хранящихся в массиве.....	53
3.7.3. Обсчет данных, расположенных в массивах.....	58
3.7.4. Простейшие задачи поиска элементов в массиве.....	62
3.7.5. Простейшие задачи по изменению расположения данных в массиве.....	64
3.8. Нестандартные функции.....	66
3.8.1. m-функции.....	66
3.8.2. Подфункции.....	71
3.8.3. InLine-функции.....	72
<b>Глава 4. Стандартные функции MatLab.....</b>	<b>73</b>
4.1. Генераторы случайных чисел.....	73
4.2. Операции с одномерными массивами.....	77
4.3. Операции с матрицами.....	78
<b>Глава 5. Базовые конструкции программирования,     реализованные на языке MatLab.....</b>	<b>81</b>
5.1. Присвоение значений.....	81
5.2. Ввод данных.....	85
5.3. Вывод данных.....	91
5.4. Вычисления.....	103
5.5. Заполнение массива.....	108
5.6. Вычисления с массивами.....	122
5.7. Поиск минимального и максимального элементов массива.....	134
5.8. Поиск информации в массиве.....	139
5.9. Изменение взаимного расположения элементов в массиве.....	145
5.10. Изменение значений элементов в массивах.....	150
<b>Глава 6. Графические возможности MatLab.....</b>	<b>155</b>
6.1. График функции одной переменной.....	155
6.2. График нескольких функций одной переменной.....	161
6.3. График функции двух переменных.....	163
6.4. Анимация.....	166
6.5. Настройка параметров графического окна и области отображения графика.....	169
6.5.1. Стандартные инструменты графического окна.....	169
6.5.2. Дескрипторная графика.....	173
6.6. Визуально-ориентированное проектирование Windows-интерфейса.....	179
6.6.1. Конструктор графического интерфейса.....	179

---

---

6.6.2. Добавление объектов .....	183
6.6.3. Редактирование свойств объектов .....	184
6.6.4. Программная обработка событий с объектами .....	185
6.6.5. Диалоговые окна .....	195
<b>Глава 7. Чтение данных из файла. Запись данных в файл.....</b>	<b>198</b>
7.1. Обращение к файлу .....	198
7.2. Чтение данных из файла .....	200
7.3. Запись данных в файл.....	203
Литература .....	207

# ПРЕДИСЛОВИЕ

MatLab (сокращение от англ. Matrix Laboratory) – пакет прикладных программ для решения задач технических вычислений и одноимённый язык программирования, используемый в этом пакете. Он работает в большинстве современных операционных систем, включая Linux, Mac OS и Microsoft Windows.

MatLab как язык программирования был разработан Кливом Моулером в конце 1970-х годов. Цель разработки: дать студентам возможность использования программных библиотек Linpack и EISPACK без необходимости изучения Фортрана. Новый язык быстро распространился среди университетов и был с большим интересом встречен учёными, работающими в области прикладной математики. С 1984 г. компанией The MathWorks под руководством Джона Литтла и Клива Моулера развивается коммерческая версия MatLab. Первоначально MatLab предназначался для проектирования систем управления, но быстро завоевал популярность во многих других научных и инженерных областях. Он также широко используется и в образовании, в частности, для преподавания линейной алгебры и численных методов.

Язык MatLab является высокоуровневым интерпретируемым языком программирования, включающим основанные на матрицах (двумерных массивах) структуры данных, широкий спектр функций, интегрированную среду разработки, объектно-ориентированные возможности и взаимодействие с программами, написанными на других языках программирования.

# Глава 1. СРЕДА РАЗРАБОТКИ ПРОГРАММНЫХ РЕШЕНИЙ MATLAB

Перед тем как начать изучение языка программирования MatLab необходимо овладеть элементарными навыками работы в интегрированной среде разработки программных решений и использования элементов пакета MatLab. Она объединяет в себе редактор, интерпретатор и отладчик, что позволяет писать, редактировать, выполнять программу и просматривать результаты ее работы, не покидая среды MatLab.

## 1.1. Запуск среды MatLab

Для запуска интегрированной среды MatLab необходимо активизировать исполняемый файл MatLab.EXE, входящий в комплект поставки. Способов его активизации может быть несколько в зависимости от настройки компьютера.

Если на компьютере установлена *операционная система Windows*, то для запуска среды MatLab можно:

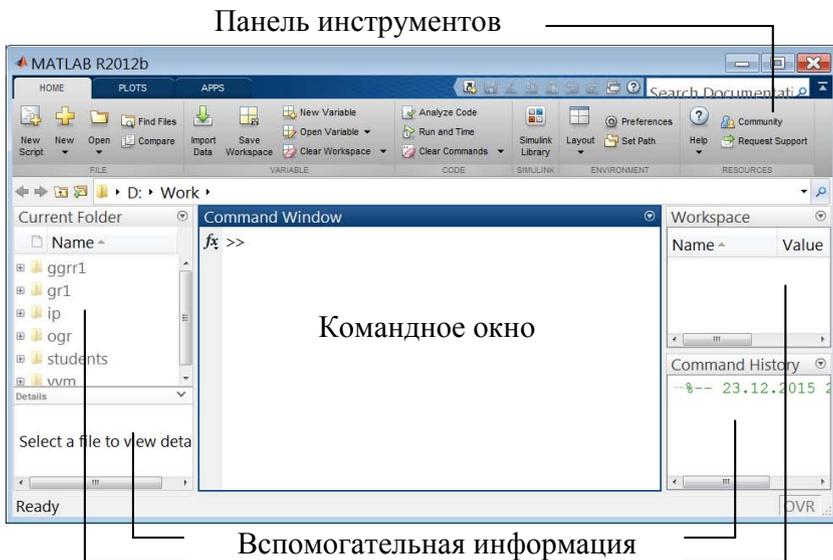
1. Сделать двойной щелчок левой кнопкой мыши по ярлыку MatLab на рабочем столе Windows.
2. Нажать кнопку Пуск на панели задач и выбрать последовательно среди пунктов всплывающего меню Все программы → MatLab → → MatLab.
3. Сделать двойной щелчок левой кнопкой мыши на иконке уже существующего файла с программой, написанной на языке MatLab (на иконке файла с расширением `.m`, например, `example1.m`).

В результате любого из этих действий после служебных сообщений появится основное окно среды MatLab.

## 1.2. Структура основного окна среды MatLab

Интерфейс интегрированной среды MatLab, начиная с версии 2012, существенно отличается от интерфейса предыдущих версий, которые по структуре и виду интерфейсных элементов были практически одинаково-

выми. Интерфейс версии MatLab 2012 наследуется всеми последующими версиями (2013, 2014, 2015 и т. д.). В пособии для иллюстрации материала использованы элементы интерфейса MatLab 2012, актуально и для последующих версий.



**Рис. 1.1.** Основное окно среды MatLab версии 2012 и выше

Основное окно среды MatLab состоит из трех различных по функциональному назначению частей (рис. 1.1): панель инструментов (верхняя часть), командное окно (центральная часть) и окна со вспомогательной информацией (справа и слева от командного окна).

### 1.2.1. Панель инструментов среды MatLab

Панель инструментов среды MatLab версии 2012 и выше содержит кнопки для управления средой, тематически сгруппированные на трех вкладках: Home, Plots, Apps (см. рис. 1.1). Кнопки на каждой из вкладок расположены на нескольких панелях. На панелях вкладки Home расположены следующие кнопки.

---

---

Панель **File** (содержит основные операции по управлению файлами):

- New Script**: позволяет создать новый m-файл с управляющей программой (script-файл, сценарий);
- New**: позволяет создать новый m-файл (как сценарий, так и функцию);
- Open**: позволяет открыть сохраненный ранее m-файл;
- Find Files**: позволяет искать сохраненные ранее файлы;
- Compare**: позволяет сравнить содержимое двух файлов.

Панель **Variable** (содержит основные операции по управлению переменными в командном окне):

- Import Data**: позволяет загрузить данные из файла;
- Save Workspace**: позволяет сохранить значения используемых переменных в файле;
- New Variable**: позволяет создать новую переменную;
- Open Variable**: позволяет редактировать значения используемых переменных;
- Clear Workspace**: позволяет удалить используемые переменные.

Панель **Code** (содержит основные операции по управлению программным кодом):

- Analyze Code**: позволяет проанализировать код файлов в текущей папке;
- Run and Time**: позволяет выполнить и измерить время выполнения программного кода;
- Clear Commands**: позволяет очистить командное окно.

Панель **Simulink** (позволяет запускать библиотеку Simulink).

Панель **Environment** (содержит основные операции по управлению средой MatLab):

- Layout**: позволяет настроить расположение и отображение рабочих окон среды;
- Preferences**: позволяет выполнять специальные настройки среды;
- Set Path**: позволяет изменить папку, которую среда использует для поиска файлов;
- Parallel**: позволяет выполнять настройки среды при параллельных вычислениях.

Панель **Resorurces** (содержит основные операции для получения справочной информации):

- Help**;
- Community**;
- Request Support**.

На панелях вкладки `Plots` расположены кнопки для построения различных графиков и диаграмм. На панелях вкладки `Apps` расположены кнопки для использования специальных пакетов и библиотек.

На отдельную панель (см. рис. 1.1) в виде пиктограмм вынесены инструменты, связанные с редактированием программного кода:

-  создать новую команду – `Create New`;
-  сохранить программный код в файл – `Save (Ctrl-S)`;
-  вырезать – `Cut (Ctrl-X)`;
-  скопировать – `Copy (Ctrl-C)`;
-  вставить – `Paste (Ctrl-V)`;
-  отменить команду – `Undo (Ctrl-Z)`;
-  отменить отмену команды – `Restore (Ctrl-Y)`;
-  показать все открытые окна.

При выборе некоторых кнопок на панели инструментов среды `MatLab` раскрывается дополнительный набор опций, выбор которых может осуществляться либо из списка, либо в режиме диалогового окна.

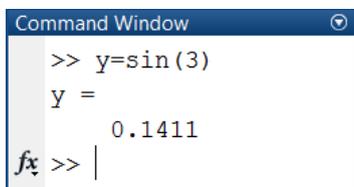
## 1.2.2. Командное окно среды `MatLab`

Командное окно (`Command Window`) среды `MatLab` версии 2012 и выше занимает центральную часть экрана (см. рис. 1.1) и предназначено для ввода программного кода и вывода результатов.

Программный код вводится в активной строке, отмеченной символом `>>`. Результаты выводятся в пустой строке после каждого фрагмента кода, не запрещающего отображение результата (рис. 1.2).

Командное окно среды также используется для отображения результатов выполнения программ, записанных в `m`-файлах (сценариях, функциях). В этом случае вывод результатов выполнения записанной в `m`-файле программы может начинаться с отображения названия файла (рис. 1.3).

Инициировать выполнение программы можно, написав в командном окне название `m`-файла, в котором она хранится.

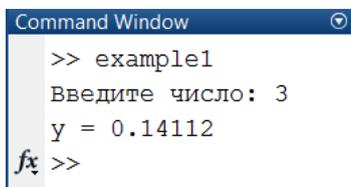


```

Command Window
>> y=sin(3)
y =
    0.1411
fx >> |

```

**Рис. 1.2.** Результаты вычислений в командном окне MatLab



```

Command Window
>> example1
Введите число: 3
y = 0.14112
fx >>

```

**Рис. 1.3.** Результаты выполнения в командном окне MatLab записанной в m-файле программы

### 1.2.3. Информационные окна

Вспомогательные окна `Current Folder`, `Details`, `Workspace`, `Command History`, расположенные справа и слева от командного окна (см. рис. 1.1), содержат следующую информацию:

- окно `Current Folder` отображает содержание текущей папки;
- окно `Details` – дополнительную информацию;
- окно `Workspace` – список и текущее значение используемых переменных;
- окно `Command History` – историю выполнения программного кода, сгруппированную по датам.

## 1.3. Создание m-файла

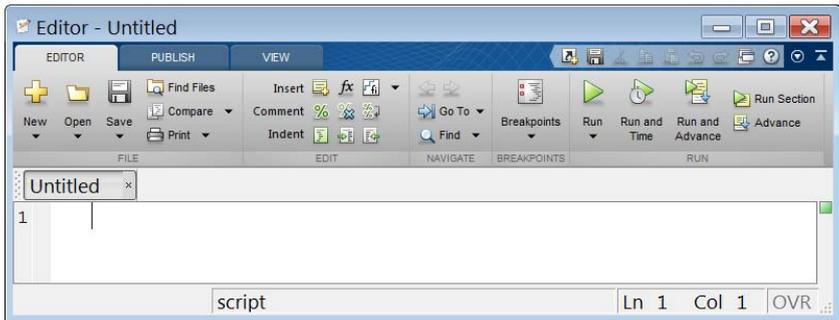
MatLab позволяет создавать файлы с программным кодом двух типов: m-функции и m-сценарии (управляющие программы).

Программный код функции имеет фиксированную структуру и хранится в виде самостоятельного m-файла. Это позволяет использовать m-функции в других программах, написанных на языке MatLab, без дополнительного подключения. Методы написания и использования m-функций описаны в разделе 3.8.

Программный код, не являющийся описанием функции, сохраненный в самостоятельном m-файле, называется сценарием (script) или управляющей программой. Структура script-файла в MatLab не регламентируется.

Кнопка `New Script` на панели инструментов `File` (см. рис. 1.1) позволяет создать новый файл сценария (управляющей программы).

В результате нажатия на кнопку `New Script` в отдельном окне запускается редактор `m`-файлов (рис. 1.4).



**Рис. 1.4.** Окно редактора `m`-файлов (версии MatLab 2012 и выше)

Окно редактора содержит панель инструментов, а также область ввода и редактирования текста программного кода. Между панелью инструментов и текстом программного кода в виде вкладок расположен список открытых в данный момент файлов, который позволяет переключаться между файлами.

## 1.4. Редактирование программы

Если окно редактора `m`-файлов пустое, в нем можно набирать новый программный код. Если окно редактора содержит текст программы, его можно редактировать.

Мигающий курсор указывает то место на экране, в котором будет появляться текст.

Движение по тексту программы можно осуществлять с помощью манипулятора мышь, а также клавиш клавиатуры, например, `←`, `→`, `↑`, `↓` (на одну позицию влево, вправо, на одну строку вверх, вниз), `PageUp`, `PageDn` (на один экран вверх, вниз).

Нажатие клавиши `Enter` на клавиатуре приводит к вставке дополнительной строки в позиции курсора. Строки автоматически нумеруются.

Удалить неправильно набранный символ можно с помощью клавиш на клавиатуре: `Del` (удаляет символ в позиции курсора) и `BackSpace` (удаляет символ, расположенный перед курсором).

---

Эти и другие кнопки позволяют облегчить процесс набора и редактирования программы.

Редактор m-файлов позволяет копировать, переносить и стирать большие фрагменты программы. Выделить необходимый фрагмент можно, либо, перемещая мышь, удерживая нажатой левую клавишу, либо с помощью комбинации клавиш клавиатуры Shift – ↓ (→, ↑, ←). Далее можно воспользоваться пиктограммами редактирования (Cut, Copy, Paste) на дополнительной панели инструментов, соответствующими комбинациями клавиш клавиатуры (Ctrl-X, Ctrl-C, Ctrl-V) или пунктами контекстного меню.

## 1.5. Сохранение программы на диске

Панель инструментов File редактора m-файлов содержит кнопку Save (см. рис. 1.4), позволяющую сохранить файл на рабочем диске компьютера. Раскрывающийся список, расположенный ниже кнопки Save, позволяет также воспользоваться для сохранения файла кнопками Save as..., Save All и т. д.

Операции Save соответствует комбинация клавиш клавиатуры Ctrl-S. Операции Save as... и Save All можно вызвать только с помощью раскрывающегося списка. Операции Save и Save as... применяются только к активному файлу, а операция Save All – ко всем открытым файлам сразу.

*При первоначальном сохранении программы* результаты применения кнопок Save и Save as... не отличаются. В обоих случаях появится диалоговое окно, содержащее поле для ввода имени файла, а также информацию об уже существующих файлах и их расположении.

*При повторном сохранении программы* результаты применения кнопок Save и Save as... существенно отличаются. Повторное использование кнопки Save для сохранения программы с заданным именем файла приводит к сохранению новой версии программы со старым именем файла. Использование кнопки Save as... для сохранения программы с уже заданным именем файла приведет к появлению еще одного файла, название которого следует ввести в появившемся диалоговом окне.

Кнопка Save All используется при работе с несколькими файлами, если есть уверенность, что все открытые файлы необходимо сохранить

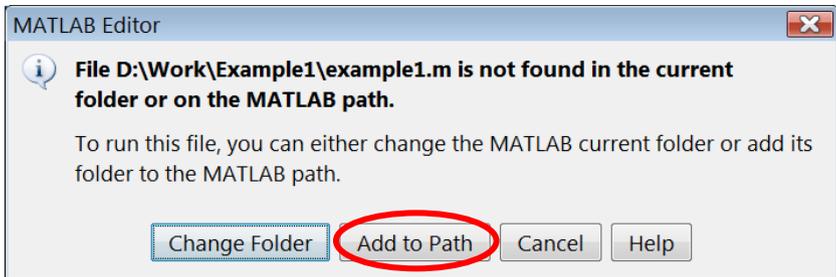
в том виде, в каком они существуют в настоящий момент, и с прежними названиями файлов.

## 1.6. Выполнение программы

Чтобы обеспечить сохранность текста набранной или измененной программы, перед выполнением ее необходимо сохранить на диске компьютера!

Запуск программы на выполнение из редактора m-файлов осуществляется клавишей F5 на клавиатуре или с помощью кнопки Run на панели инструментов Run (см. рис. 1.4).

Если программа сохранена в папке, отличающейся от папки Current Folder, используемой по умолчанию, при первом выполнении программы MatLab предложит (рис. 1.5) или изменить папку, используемую по умолчанию (Change Folder), или добавить папку, в которой сохранен файл, к списку папок, используемых средой MatLab (Add to Path). Если в одной из папок, используемых средой MatLab, существует файл с таким же именем, то предложение Add to Path в соответствующем диалоговом окне будет отсутствовать.



**Рис. 1.5.** Добавление папки, в которой сохранен файл, к списку папок, используемых средой MatLab по умолчанию

Перед выполнением текст программы автоматически переводится в машинные коды построчно. Этот процесс называется интерпретацией. Во время интерпретации происходит проверка синтаксиса программы. Если интерпретатор обнаружит в программе синтаксическую ошибку, он выдает соответствующее сообщение (Error...) в командном окне, указав номер строки, в которой произошла ошибка. После исправления ошибки необходимо вновь запустить программу на выполнение. Если

---

синтаксических ошибок не обнаружено, строка программы выполняется, а интерпретатор переходит к анализу следующей строки.

Выполнение программы происходит в командном окне среды MatLab.

Если в процессе выполнения программы возникает необходимость прервать ее работу, можно воспользоваться комбинацией клавиш клавиатуры `Ctrl-Break`.

## **1.7. Просмотр результатов работы программы**

Если командное окно среды MatLab не очищалось, результаты работы программы (т. е. выводимую ею информацию, а не сам текст программы) можно посмотреть не только в процессе, но и после выполнения программы в командном окне. Из основного окна среды MatLab в редактор `m`-файлов и обратно можно вернуться, стандартным образом переключаясь между окнами в операционной системе Windows.

## **1.8. Встроенная справочная система**

В разрешении различных затруднений, возникающих при разработке программ, может помочь встроенная справочная система среды MatLab (`Help`). Справочная система среды MatLab содержит информацию о зарезервированных словах и операторах языка MatLab, директивах интерпретатора, стандартных функциях, ошибках и о самой среде MatLab. Вызов различных разделов справочной системы осуществляется с помощью кнопки `Help` на панели инструментов `Resources` (см. рис. 1.1) либо с помощью клавиши `F1` на клавиатуре.

# Глава 2. ОСНОВНЫЕ ЭЛЕМЕНТЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ MATLAB

## 2.1. Структура программы

Структура программы, написанной на языке MatLab, определяется логической последовательностью операций, для выполнения которых пишется программа. Программа выполняется последовательно сверху вниз.

В языке MatLab, как в любом языке программирования, использование любого объекта (переменной, вектора, матрицы и т. д.) в качестве исходных данных требует, чтобы его значение было предварительно задано каким-либо способом.

В процессе выполнения любая программа, как правило, должна осуществлять обработку информации. Всякая обработка информации предполагает три этапа:

1. Ввод данных (информации).
2. Обработка данных (информации).
3. Вывод результатов.

Изменение последовательности этапов или исключение хотя бы одного из них, как правило, делает бессмысленными оставшиеся действия.

Поэтому даже для самых простых задач содержание программного кода (программы) тоже должно состоять из трех частей, реализующих каждый из этих этапов. При решении более сложных задач перечисленные этапы обработки информации могут выполняться несколько раз, но всегда в указанной последовательности – как логически завершенные блоки. Тогда результаты одного блока обработки информации могут выступать в качестве исходных данных для следующего блока.

Этап ввода данных реализуется, в основном, с помощью оператора `input`. Для вывода результатов используют оператор `disp`. Выбор операторов для реализации этапа обработки данных зависит от предполагаемого способа обработки информации. Это могут быть *операторы*

*присвоения, условных переходов, циклы* и их комбинации. Отдельные фрагменты программы могут быть оформлены как *функции*.

## 2.2. Идентификаторы

**Идентификатор** – уникальный набор символов, предназначенный для обозначения переменной, константы, команды или функции.

В качестве символов идентификатора можно использовать все буквы латинского алфавита (как строчные, так и прописные), цифры и символ `_`. Идентификаторы НЕ МОГУТ содержать пробел, точку, запятую, точку с запятой, начинаться с цифры.

Идентификаторы подразделяются на стандартные и нестандартные (идентификаторы, выбираемые программистом). Часть идентификаторов зарезервирована в MatLab для обозначения стандартных команд, операторов и функций. Такие идентификаторы называют стандартными идентификаторами или зарезервированными словами. Использование стандартных идентификаторов не по назначению может вызвать ошибку при интерпретации программного кода.

Нестандартные идентификаторы не связаны с каким-либо фиксированным значением или объектом среды MatLab, их предназначение и методы использования определяются программистом по своему усмотрению.

Примеры нестандартных идентификаторов:

X	Tau	NAME2
x	tau	name2
X_min	Tau_Q	Name_Student

Идентификаторы, написанные строчными и прописными буквами, считаются разными.

## 2.3. Типы данных

Данные, используемые в программе, могут относиться к различным типам – числовые, символьные и т. д. в зависимости от содержания информации, хранящейся в них. В среде MatLab определено 6 базовых (стандартных) типов данных: `double`, `char`, `sparse`, `uint8`, `cell`, `struct`. Эти типы предназначены для хранения следующей информации:

□ **Double** – числовые данные (как вещественные, так и целочисленные);