

Тема 4.6

Программирование на языке MATLAB

Вопросы для изучения

- 4.21 Сложные типы данных (структуры). Создание структур.
Доступ к полям структуры
- 4.22 Массивы структур, использование массивов в полях
структуры

4.21 Сложные типы данных (структуры). Создание структур. Доступ к полям структуры

Во многих случаях недостаточно объявить простую переменную или массив, а нужна более гибкая форма представления данных.

Таким элементом может быть структура, которая позволяет включать в себя разные типы данных и даже другие структуры.

Структуры задаются следующим образом:

- с использованием операторов присваивания;
- с использованием функции `struct`.

Применение оператора присваивания.

Для того чтобы сформировать простейшую структуру, необходимо присвоить данные соответствующим полям.

Система MATLAB автоматически формирует структуру по мере ее заполнения.

`имя структуры.поле структуры = выражение;`

Пример: сформируем структуру `student` размера `1x1`, состоящую из следующих полей:

```
student.name = 'Казаков';  
student.group = 171;  
student.ball = [8 7 9];
```

Таким образом, `student` - это пока массив из одной записи с тремя полями.

Для того чтобы расширить его, достаточно добавить индекс в имени структуры.

Создадим вторую запись в структуре student:

```
student(2).name = 'Коршунов';  
student(2).group = 171;  
student(2).ball = [6 6 7];
```

Теперь структура student имеет размер 1x2.

Если структура содержит более одной записи, при ее запросе, содержимое полей не выводится, а выводится только обобщенная информация.

Для получения информации можно использовать функцию **fieldnames**, которая возвращает массив ячеек, содержащий строки с именами полей.

По отношению к структуре выполняются следующие условия:

- все записи структуры имеют одинаковое количество полей;
- все имена полей одинаковы.

Доступ к полю структуры при записи или чтении организован с помощью оператора присваивания при этом идентификатор поля структуры формируется по следующему правилу: на первом месте указывается имя структуры, а после него через разделитель «.» указывается название изменяемого поля структуры

`student.name = 'Казаков';` - запись

`FIO= student.name` - чтение

Применение функции `struct`.

Если работа ведётся в интерактивном режиме, когда все данные вводятся с клавиатуры, рассмотренный процесс последовательного задания полей и элементов массива структур вполне оправдан. Однако в программном режиме он не достаточно хорош с точки зрения производительности. Вместо него лучше использовать функцию с именем `struct`.

Функция `struct` имеет следующий синтаксис:

```
Имя структуры = struct('<имя_поля1>', '<значения1>', '<имя_поля2>', '<значения2>',  
... '<имя_поля N>', '<значенияN>')
```

где `имя_поля 1...N` – название полей структуры;
`значения 1...N` – содержание полей.

Функция `struct` позволяет сформировать структуру с указанными значениями полей. В дальнейшем можно изменить значения полей, используя операторы присваивания.

Пример:

```
student=struct('name' , 'Казаков', 'group' , 171, 'ball' , [8 7 9])  
student(2)=struct('name' , 'Коршунов', 'group' , 171, 'ball' , [8 8 9])  
student(3)=struct('name' , 'Засинец', 'group' , 171, 'ball' , [9 9 9])
```

4.22 Массивы структур, использование массивов в полях структуры

Приведем пример, в котором использование структуры позволяет эффективно представить данные. Таким примером будет список студентов, в котором для каждого студента необходимо указать фамилию, номер группы и баллы.

Количество студентов может не более 100.

Для хранения информации об одной книге будем использовать структуру, которая задается следующим образом:

```
student=struct('name' , ' ', 'group' , 0, 'ball' , [0 0 0])
```

В итоге задается структура с тремя полями: name, group и ball. Каждое поле имеет свой тип данных и значение.

Однако по условиям задачи необходимо осуществлять запись не по одной, а по 100 книгам. В этом случае целесообразно использовать вектор (массив) структур `stud`, который можно задать следующим образом:

```
stud(100,1) = struct('name' , ' ', 'group' , 0, 'ball' , [0 0 0]);
```

и записывать информацию о студентах как было рассмотрено выше

```
stud(1).name = 'Казаков';  
stud(1).group = 171;  
stud(1).ball = [8 7 9];
```

	имя	группа	балл
stud (1)	stud (1).name	stud (1).group	stud (1).ball
stud (2)	stud (2).name	stud (2).group	stud (2).ball
stud (3)	stud (3).name	stud (3).group	stud (3).ball
...
stud (100)	stud (100).name	stud (100).group	stud (100).ball

Графическое представление массива структур по 100 студентам

При работе со структурами полезными являются следующие функции:

- `isstruct(stud)` – возвращает истину, если аргумент структура
- `isfield(stud , 'name')` – возвращает истину, если имеется такое поле
- `fieldnames(stud)` – возвращает массив строк с именами всех полей
- `rmfield(stud , 'name')` — возвращает структуру `stud` с удаленным полем `S. 'name'`;
- `rmfield(S,FIELDS)` — возвращает структуру `S` с несколькими удаленными полями.

Список удаляемых полей `FIELDS` задается в виде массива символов (строки) или строкового массива.

Рассмотрим как обратиться к полю `ball` первого элемента массива структур `stud`. Для этого надо применить две операции индексации и одну операцию доступа к полю структуры:

```
stud(1).ball( 1 , 2 )
```

имеет значение `7`, так как в поле `ball` расположен вектор `1x3`.