Тема 2. «Математические объекты и их представления»

Вариативная самостоятельная работа

Задание 2.2 (2)

**Переменная**

Переменная в Scilab — это именованный массив всего с одним полем, которое хранит данные некоторого типа.

Среди типов данных можно выделить:

* Числа
  + Целые числа
  + Вещественные числа
  + Комплексные числа
* Строки
* Логические переменные

Создать переменную в среде не составляет труда. Для этого достаточно ввести ее имя и присвоить ей какое-либо начальное значение. Для переменной будет автоматически выделено место в памяти, а ее область видимости по умолчанию становится локальной.

Для того, чтобы посмотреть текущее значение переменной, достаточно просто обратиться к ней по имени, либо воспользоваться редактором переменных. В дальнейшем вы убедитесь, что редактор крайне удобен для матриц.

Существуют следующие правила, которым должны удовлетворять имена переменных и вообще любых объектов среды:

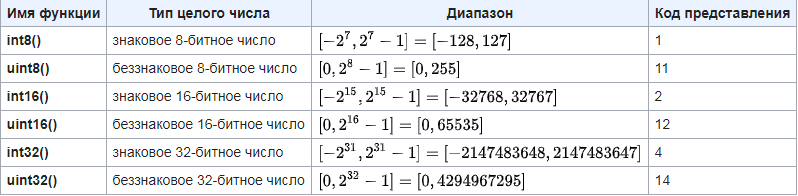
* Имя переменной может состоять из букв латинского алфавита (верхнего и нижнего регистра) и цифр;
* Имя переменной не может начинаться с цифры, но может начинаться с символов '%', '\_', '#', '!', '$', '?';
* Регистр в имени играет роль, то есть переменные с именами var,VAR ,Var и т. п. разные;
* Запрещено совпадение имени переменной с зарезервированными словами, такими как имена объявленных функций, констант и др.;

**Целые числа**

В Scilab целые числа возможно создавать только через специальные функции. Во всех остальных случаях числовому значению всегда будет присваиваться вещественный тип данных.

Для хранения целого числа в памяти может быть использовано разное число битов, а именно 8, 16 и 32. От количества используемых битов зависит диапазон целых чисел. Кроме того, имеется возможность включения и отключения знакового бита, что бывает полезно, когда отрицательные целые числа не требуются.

В таблице ниже приведены функции для создания целых чисел и диапазоны возможных значений. Во всех случаях функция возвращает целое число, указанное в качестве аргумента, с определенным способом его хранения в памяти.



Какой тип целочисленной переменной следует применить, пользователь решает сам. Целые числа могут быть использованы как в чисто математических задачах, так и программировании, в качестве счетчиков.

Для работы с целочисленными переменными существует две функции:

* iconvert(X, itype) — меняет представление в памяти в общем случае матрицы из вещественных или целых чисел. Функции при этом следует передавать имя этой матрицы X и код внутреннего представления целого числа itype из последней колонки таблицы выше;
* inttype(X) — возвращает код внутреннего представления в общем случае матрицы целочисленных данных.

**Функции в Scilab**

Все функции, используемые в Scilab, можно разделить на два класса:

* встроенные;
* определенные пользователем.

Вобщем виде обращение к функции вScilab имеет вид:

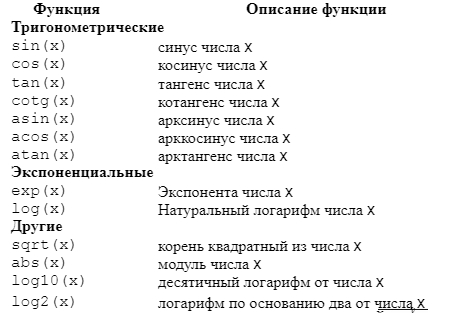
имя\_переменной = имя\_функции(переменная1 [, переменная2, …])

где

* имя\_переменной – переменная, в которую будут записаны результаты работы функции; этот параметр может отсутствовать, тогда значение, вычисленное функцией будет присвоено системной переменнойans ;
* имя\_функции – имя встроенной функции или ранее созданной пользователем;
* переменная1,переменная2, … – список аргументов функции.

**Элементарные математические функции**

Пакет Scilab снабжен достаточным количеством всевозможных встроенных функций, знакомство с которыми будет происходить в следующих разделах. Здесь приведем, только элементарные математические функции, используемые чаще всего.



**Функции, определенные пользователем**

В первой главе мы уже упоминали о файлах-сценариях и даже создавали небольшую программу, которая решала конкретное квадратное уравнение. Но в эту программу невозможно было передать входные параметры, то есть это был обычный список команд, воспринимаемый системой как единый оператор.

Функция, как правило, предназначена для неоднократного использования, она имеет входные параметры и не выполняется без их предварительного задания. Рассмотрим несколько способов создания функций вScilab.

Первый способ это применение оператора

*deff('[имя1,...,имяN]=имя\_функции(переменная\_1,...,переменная\_M)', 'имя=выражение;...;имя1=выражение1;...;имяN=выражениеN')*

где имя1,...,имяN – список выходных параметров (от 1 доN), то есть переменных, которым будет присвоен конечный результат вычислений,имя\_функции – имя с которым эта функция будет вызываться,переменная\_1,...,переменная\_M – входные параметры (от 1 доM).

Второй способ создания функции это примение конструкции вида:

*function[имя1,...,имяN]=имя\_функции(переменная\_1,...,переменная\_M)*

*тело функции*

*endfunction*

где имя1,...,имяN – список выходных параметров (от 1 доN), то есть переменных, которым будет присвоен конечный результат вычислений,имя\_функции – имя с которым эта функция будет вызываться,переменная\_1,...,переменная\_M – входные параметры (от 1 доM).

Все имена переменных внутри функции, а так же имена из списка входных и выходных параметров воспринимаются системой как локальные, то есть эти переменные считаются определенными только внутри функции.

Вообще говоря, функции в Scilsb играют рольподпрограмм. Поэтому целесообразно набирать их тексты в редакторе и сохранять в виде отдельных файлов. Причем имя файла должно обязательно совпадать с именем функции. Расширениефайлам-функциямобычно присваивают\*.sci или\*.sce.

Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, то есть из командной строки. Однако функции, хранящиеся в отдельных файлах должны быть предварительно загружены в систему, напрмер при помощи оператораexec(иям\_файла) или командой главного менюFile\Exec..., что в общем, одно и то же.

**Матрицы**

Матрица в Scilab — это двухмерный массив однотипных элементов. Можно понимать матрицу как несколько векторов-строк, записанных столбцом.

Создать матрицу в Scilab можно одним из нескольких способов:

* Матрицу можно создать из составляющих ее элементов;
* Из имеющихся векторов, упорядочив их строками или столбцами;
* Одной из специальных функций.

В общем случае синтаксическая конструкция имеет вид

[x11, x12, …, x1n; x21, x22, …, x2n; …; xm1, xm2, …, xmn]

Таким образом, вы создаете векторы-строки, которые отделяете точкой с запятой. Запятая в этом случае, как и с вектором, не обязательна.

Матрицу можно собрать из векторов. Вот два поучительных примера

--> [1:5; 5:-1:1]  
ans =  
 1. 2. 3. 4. 5.  
 5. 4. 3. 2. 1.  
--> [(1:5)' (5:-1:1)']  
ans =  
 1. 5.  
 2. 4.  
 3. 3.  
 4. 2.  
 5. 1.

Обратите внимание на второй случай, как сначала мы транспонируем векторы, создаваемые особой конструкцией, а затем присоединяем их столбцами друг к другу.

В любом случае, вы всегда должны помнить, что нельзя допускать конфликты с размерами векторов. Например, если вы образуете матрицу из векторов-строк, то число элементов в них должно быть одинаковым, иначе произойдет ошибка.