

Основные возможности Scilab, используемые при работе с пользователь- скими функциями

Функция, как правило, предназначена для неоднократного использования, она имеет входные параметры и не выполняется без их предварительного задания. Рассмотрим несколько способов создания функций в Scilab. Первый способ — это применение оператора deff, который в общем виде можно записать так:

```
deff('[имя1,...,имяN] =  
имя_функции(переменная_1,...,переменная_M)', 'имя1=выражение1;...;  
имяN=выражениеN')
```

где имя1,...,имяN — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений, имя_функции — имя с которым эта функция будет вызываться, переменная_1,...,переменная_M — входные параметры.

Далее приведен самый простой способ применения оператора deff. Здесь показано, как создать и применить функцию для вычисления выражения

$$z = \sqrt{\left| \sin \left(\frac{x}{y} \right) \right|} \cdot e^{xy}$$

```
-->deff('z=fun1(x,y)', 'z=sqrt(abs(sin(x/y)))*exp(x^y)');  
-->x=1.2;y=0.3;z=fun1(x,y)  
z = 2.5015073
```

Второй способ создания функции — это применение конструкции вида:

```
function[имя1,...,имяN]=имя_функции(переменная_1,...,переменная_M)
    тело функции
endfunction
```

где имя1,...,имяN — список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений; имя_функции — имя с которым эта функция будет вызываться, переменная_1, ..., переменная_M — входные параметры.

Все имена переменных внутри функции, а также имена из списка входных и выходных параметров воспринимаются системой как локальные, т. е. считаются определенными только внутри функции. Вообще говоря, функции в Scilab играют роль подпрограмм. Поэтому целесообразно набирать их тексты в редакторе и сохранять в виде отдельных файлов.

Причем имя файла должно обязательно совпадать с именем функции. Расширение файлам-функциям обычно присваивают sci или sce.

Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, т.е. из командной строки. Однако функции, хранящиеся

в отдельных файлах, должны быть предварительно загружены в систему, например, при помощи оператора exес(имя_файла) или командой главного меню

File – Exес. . . , что, в общем, одно и то же.

```
В качестве примера рассмотрим
следующую задачу — решить кубическое
уравнение:  $a \cdot x^3 + b \cdot x^2 + c \cdot x + d = 0$ 
//файл cub.sce
function [x1,x2,x3]=cub(a,b,c,d)
    r=b/a;
    s=c/a;
    t=d/a;
    p=(3*s-r^2)/3;
    q=2*r^3/27-r*s/3+t;
    D=(p/3)^3+(q/2)^2;
    u=(-q/2+sqrt(D))^(1/3);
    v=(-q/2-sqrt(D))^(1/3);
    y1=u+v;
    y2=-(u+v)/2+(u-v)/2*i*sqrt(3);
    y3=-(u+v)/2-(u-v)/2*i*sqrt(3);
    x1=y1-r/3;
    x2=y2-r/3;
    x3=y3-r/3;
endfunction
//Вызов функции и вывод результатов ее
работы:
-->exес('./cub.sce')
// Пример вызова функции из файла для ОС
Windows:
//-->exес('C:\Scilab\scilab-4.1.1\cub.sce');
-->disp('exес done');
Warning :redefining function: cub
exес done
-->[x1,x2,x3]=cub(3,-20,-3,4)
x3 =
0.3880206
x2 =
- 0.5064407
x1 =
6.7850868
```

Об авторе

Моисеенко Павел,
студент 2 курса группы
ИВТ, кафедры ИИТиТО