

# **Программирование в Scilab. Основные операторы sci-языка**

Для создания программы (программу в Scilab иногда называют сценарием) необходимо:

1. Вызвать команду Editor из меню
2. В окне редактора Scipad набрать текст программы.
3. Сохранить текст программы с помощью команды File – Save в виде файла с расширением sce , например, file.sce .
4. После этого программу можно будет вызвать, набрав в командной строке `exes`, например, `exes("file.sce")`. Другие способы вызова — воспользоваться командой меню File – Exec. . . или, находясь в окне Scipad, выполнить команду Execute – Load into Scilab (Ctrl+L ).

```
a=3;
b=2;
c=5;
d=-4;
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;
```

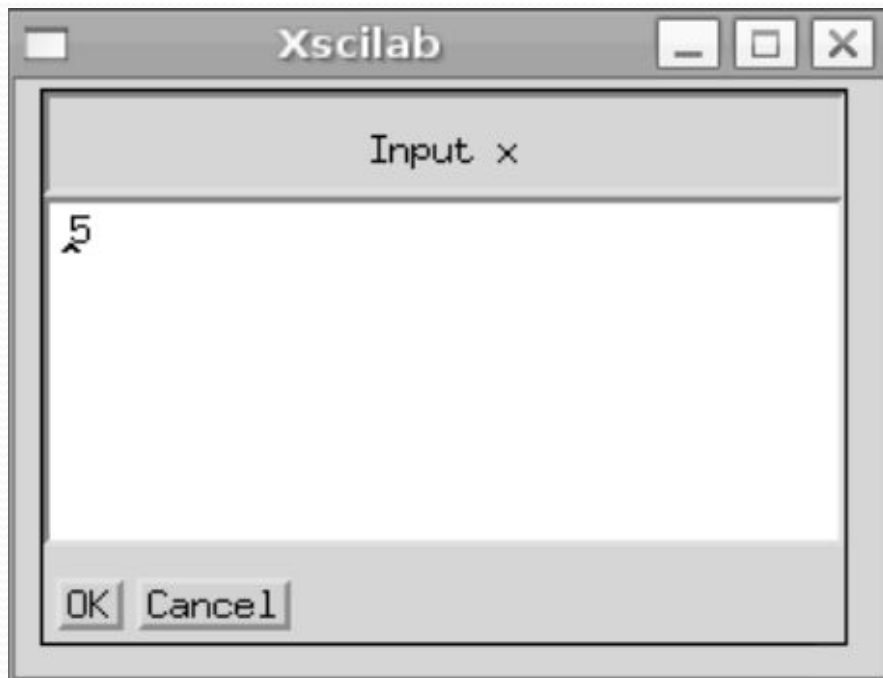
## Функции ввода-вывода в Scilab

Для организации простейшего ввода в Scilab можно воспользоваться функциями

```
x=input('title');
```

или

```
x=x_dialog('title', 'stroka')
```



```
x=x_dialog('Input X','5')
```

# Оператор присваивания

Оператор присваивания имеет следующую структуру

$$a=b$$

# Условный оператор

Существует обычная и расширенная формы оператора if в Scilab.

Обычный if имеет вид

```
if условие  
операторы1  
else  
операторы2  
end
```

## Расширенная форма оператора if

```
if условие1  
  операторы1  
else  
  if условие2  
    операторы2  
  else  
    if условие 3  
      операторы3  
    ...  
  Else  
    if условие n  
      операторы n  
    else  
      операторы  
    end  
end
```

## Задача 1.

В качестве примера программирования разветвляющегося процесса рассмотрим решение биквадратного уравнения  $ax^4 + bx^2 + c = 0$



Алгоритм состоит из следующих этапов:

1. Ввод коэффициентов уравнения  $a$ ,  $b$  и  $c$ ;
2. Вычисление дискриминанта уравнения  $d$ ;
3. Если  $d < 0$ , определяются  $y_1$  и  $y_2$ , в противном случае выводится сообщение «Корней нет».
4. Если  $y_1 < 0$  и  $y_2 < 0$ , то вывод сообщения «Корней нет».
5. Если  $y_1 > 0$  и  $y_2 > 0$ , то вычисляются четыре корня по формулам  $\pm\sqrt{y_1}$ ,  $\pm\sqrt{y_2}$  и выводятся значения корней.
6. Если условия 4) и 5) не выполняются, то необходимо проверить знак  $y_1$ .
7. Если  $y_1$  неотрицательно, то вычисляются два корня по формуле  $\pm\sqrt{y_1}$ , иначе оба корня вычисляются по формуле  $\pm\sqrt{y_2}$ .

Листинг 9.1. Программа решения биквадратного уравнения

```
//Ввод значений коэффициентов биквадратного уравнения.
a=input('a=');
b=input('b=');
c=input('c=');
//Вычисляем дискриминант.
d=b*b-4*a*c;
//Если дискриминант отрицателен,
if d<0
//то вывод сообщения,
disp('Real roots are not present');
else
//иначе вычисление корней соответствующего
//квадратного уравнения.
x1=(-b+sqrt(d))/2/a;
x2=(-b-sqrt(d))/2/a;
//Если оба корня отрицательны,
if (x1<0)&(x2<0)
//вывод сообщения об отсутствии действительных корней
disp('Real roots are not present');
//иначе, если оба корня положительны,
elseif (x1>=0)&(x2>=0)
```

```
//вычисление четырех корней.
disp('Four real roots');
y1=sqrt(x1);
y2=-y1;
y3=sqrt(x2);
y4=-y2;
disp(y1,y2,y3,y4);
//Иначе, если оба условия (x1<0)&(x2<0) и (x1>=0)&(x2>=0)
//не выполняются,
else
//то вывод сообщения
disp('Two real roots');
//Проверка знака x1.
if x1>=0
//Если x1 положителен, то вычисление двух корней биквадратного
//уравнения извлечением корня из x1,
y1=sqrt(x1);
y2=-y1;
disp(y1);
disp(y2);
//иначе (остался один вариант - x2 положителен),
//вычисление двух
//корней биквадратного уравнения извлечением корня из x2.
else
y1=sqrt(x2); y2=-y1;
disp(y1); disp(y2);
end
end end
```

### Листинг 9.3. Решение биквадратного уравнения

```
a=input('a=');  
b=input('b=');  
c=input('c=');  
d=b*b-4*a*c;  
x1=(-b+sqrt(d))/2/a;  
x2=(-b-sqrt(d))/2/a;  
y1=sqrt(x1);  
y2=-y1;  
y3=sqrt(x2);  
y4=-y3;  
disp(y1,y2,y3,y4);
```

Результат работы программы, представленной на листинге 9.3 представлены ниже (см. листинг 9.4).

### Листинг 9.4. Комплексные корни биквадратного уравнения

```
-->a=3  
-->b=8  
-->c=-1  
-1.6692213i  
1.6692213i  
-0.3458800  
0.3458800
```

# Оператор альтернативного выбора

Оператор альтернативного выбора select имеет следующую структуру:

```
select параметр  
case значение1 then операторы1  
case значение2 then операторы2  
...  
else операторы  
end
```

## Задача 2.

Вывести на печать название дня недели, соответствующее заданному числу  $D$ , при условии, что в месяце 31 день и 1-е число — понедельник.

Листинг 9.5. Решение задачи 9.2

```
D=input('Enter a number from 1 to 31');  
//Вычисление остатка от деления D на 7, сравнение его с числами  
//от 0 до 6.  
select D-int(D/7)*7  
case 1 then disp('Monday');  
case 2 then disp('Tuesday');  
case 3 then disp('Wednesday');  
case 4 then disp('Thursday');  
case 5 then disp('Friday');  
case 6 then disp('Saturday');  
else  
disp('Sunday');  
end
```

Листинг 9.6. Вызов функции для решения задачи 9.2

```
-->exec('G:\Lecture Scilab EG\2\l2.sci');disp('exec done');  
Enter a number from 1 to 31-->19  
Friday
```

# Оператор while

Оператор цикла while имеет вид:

```
while условие  
операторы  
end
```

# Оператор for

Оператор цикла for имеет вид:

```
for x=xn:hx:xk  
операторы  
end
```