Основы создания программ-функций в среде MathCAD

Реализовать тот или иной алгоритм вычисления в среде MathCAD можно двумя способами:

1 способ:

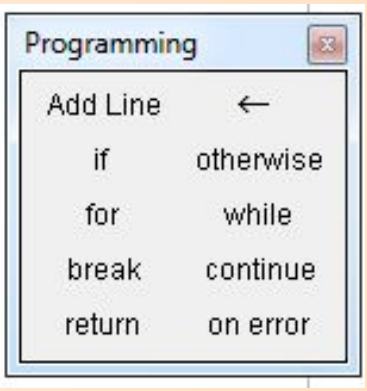
размещать обращение к соответствующим переменным, выражениям, операторам, функциям непосредственно в тексте документа MathCAD без какой-либо организационной структуры; данный способ называется *программированием в тексте документа*;

2 способ:

использовать организационные структуры, так называемые программы-функции, которые содержат конструкции, во многом подобные конструкциям таких языков программирования высокого уровня как FORTRAN, Pascal, C++, C# и др.

В данной лекции будем рассматривать второй способ реализации алгоритмов, который называется ***программированием в программе-функции***.

Для создания и использования программ-функций применяется панель инструментов Programming (Программирование).



Кнопка	Название	Назначение кнопки (команды)
Add Line	Add Program Line	позволяет *создать и/или удлинять вертикальную линию*, справа от которой формируется запись программного блока;
←	Local Definition	позволяет вставить *оператор внутреннего присваивания* (активен в теле модуля);
if	If Statement	позволяет вставить *условный оператор*;
otherwise	Otherwise Statement	позволяет вставить *оператор иного выбора*;
for	For Loop	позволяет вставить *оператор цикла с заданным числом повторений*;
while	While Loop	позволяет вставить *оператор цикла типа «пока»* (итерационный цикл);
break	Break Statement	позволяет вставить *оператор прерывания*;
continue	Continue Statement	позволяет вставить *оператор продолжения*;
return	Return Statement	позволяет вставить *оператор-функцию возврата*;
on error		

Примеры программ-функций:

Пример программы-функции, вычисляющей сумму ряда:

$$S = 1 + \frac{1}{2^2} - \frac{1}{3^2} + \frac{1}{4^2} - \ldots \pm \frac{1}{n^2} \ldots$$

$$\begin{array}{l} x \leftarrow 1 \\ \varepsilon \leftarrow 0.001 \\ \text{while } \dfrac{1}{x^2} > \varepsilon \\ \quad \left| \begin{array}{l} s \leftarrow s + \dfrac{1}{x^2} \\ x \leftarrow x + 1 \end{array} \right. \\ s \end{array} = 1.613$$

Пример программы-функции, реализующей итерационную процедуру приближенного вычисления корня квадратного из числа a;

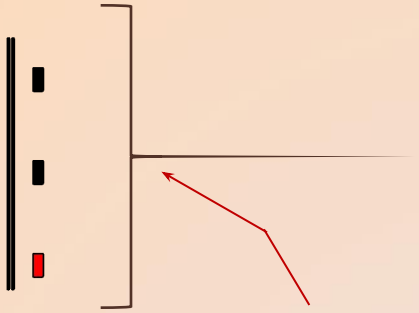
$$\text{sqroot}(a, eps) := \left| \begin{array}{l} xc \leftarrow 10000000000 \\ xn \leftarrow a \\ \text{while } |xn - xc| > eps \\ \quad \left| \begin{array}{l} xc \leftarrow xn \\ xn \leftarrow \dfrac{xc + \dfrac{a}{xc}}{2} \end{array} \right. \\ xn \end{array} \right.$$

Процесс программирования в программе-функции включает в себя два этапа:

1) описание программы-функции;
2) обращение к программе-функции (вызов).

<u>Описание программы – функции</u> размещается в рабочем документе перед *<u>вызовом программы-функции</u>* и имеет следующую структуру:

$$\text{name_function}(\text{form_p1}, \text{form_p2}, \ldots, \text{form_pN}) :=$$



имя функции

имена формальных параметров

тело программы-функции (местозаполнители для ввода операторов)

Обращение к программе-функции (вызов) располагается правее или ниже описания программы-функции и может иметь следующую структуру:

с использованием фактических параметров

Формат 1.1:

name_function(<fact_p1,fact_p2,...,fact_pN>)=

Формат 1.2:

name_var := name_function(<fact_p1,fact_p2,…,fact_pN>)
name_var =

без использования фактических параметров

Формат 2.1:

name_function=

Формат 2.2:

name_var := name_function
name_var =

Между формальными и фактическими параметрами должно быть установлено *взаимно-однозначное соответствие по*:

- количеству;
- порядку следования;
- типу.

$$\text{name_function}(\text{form_p1}, \text{form_p2}, \ldots, \text{form_pN}) :=$$

name_function(<fact_p1,fact_p2,...,fact_pN>)=

Варианты обращений к программе-функции, реализующей итерационную процедуру приближенного вычисления корня квадратного из числа a:

1

$$\text{sqroot}(a, eps) := \left| \begin{array}{l} xc \leftarrow 10000000000 \\ xn \leftarrow a \\ \text{while } |xn - xc| > eps \\ \quad \left| \begin{array}{l} xc \leftarrow xn \\ xn \leftarrow \dfrac{xc + \dfrac{a}{xc}}{2} \end{array} \right. \\ xn \end{array} \right.$$

$\text{sqroot}(25, 0.001) = 5$

$\text{sqroot}(123, 0.00001) = 11.091$

$\text{sqroot}(9, 0.0001) = 3$

2

$$\text{sqroot}(a, eps) := \left| \begin{array}{l} xc \leftarrow 10000000000 \\ xn \leftarrow a \\ \text{while } |xn - xc| > eps \\ \quad \left| \begin{array}{l} xc \leftarrow xn \\ xn \leftarrow \dfrac{xc + \dfrac{a}{xc}}{2} \end{array} \right. \\ xn \end{array} \right.$$

$y := \text{sqroot}(25, 0.001) \qquad y = 5$

$y := \text{sqroot}(144, 0.01) \qquad y = 12$

3

$a := 25 \qquad eps := 0.01$

$$\text{sqroot} := \left| \begin{array}{l} xc \leftarrow 10000000000 \\ xn \leftarrow a \\ \text{while } |xn - xc| > eps \\ \quad \left| \begin{array}{l} xc \leftarrow xn \\ xn \leftarrow \dfrac{xc + \dfrac{a}{xc}}{2} \end{array} \right. \\ xn \end{array} \right.$$

$\text{sqroot} = 5$

Линейный алгоритм (следование) **- вычислительный процесс, в котором необходимые действия выполняются строго последовательно и каждое один раз.**

В реализации алгоритмов линейной структуры в программах-функциях используется ***оператор внутреннего присваивания*** (локальный оператор присваивания), с помощью которого осуществляется внутри программы-функции присваивание значений (числовых, строковых) переменным.

Формат оператора:

< имя_переменной > ← < выражение >

$$f(x) := \left| \begin{array}{l} x \leftarrow x + 2 \\ z \leftarrow x^{\frac{1}{3}} \\ z \end{array} \right.$$

выражение, определяющее возвращаемое через имя программы-функции значение

Замечание: использование "обычного" оператора присваивания (: =) в теле программы-функции приводит к синтаксической ошибке.

Варианты обращений к программе-функции, реализующей вычисление значения переменной z:

$$f(x) := \left| \begin{array}{l} x \leftarrow x + 2 \\ z \leftarrow x^{\frac{1}{3}} \\ z \end{array} \right.$$

$$z := f(2)$$

$$z = 1.587$$

различные переменные, не связанные между собой

$$x := 2$$

$$f := \left| \begin{array}{l} x \leftarrow x + 2 \\ z \leftarrow x^{\frac{1}{3}} \\ z \end{array} \right.$$

используется значение x=2

значение переменной x изменяется

$$f = 1.587$$

вызов программы-функции

$$x = 2$$

вне программы-функции переменная сохраняет свое значение

Пример 1. Написать программу-функцию, вычисляющую значения корней квадратного уравнения $ax^2 + bx + c = 0$.

$$qq1(a,b,c,sig1) := \left| \begin{array}{l} d \leftarrow b^2 - 4\cdot a\cdot c \\ x \leftarrow \dfrac{-b + sig1\cdot\sqrt{d}}{2\cdot a} \end{array} \right.$$

$$qq1(9,-42,35,1) = 3.581$$

$$qq1(9,-42,35,-1) = 1.086$$

$$qq1(9,-4,5,-1) = 0.222 - 0.711i$$

Пример 2. Написать программу-функцию, вычисляющую произведение цифр четырехзначного положительного целого числа N.

$$g(n) := \left| \begin{array}{l} n1 \leftarrow mod(n,10) \\ n2 \leftarrow mod\left(trunc\left(\dfrac{n}{10}\right),10\right) \\ n3 \leftarrow mod\left(trunc\left(\dfrac{n}{100}\right),10\right) \\ n4 \leftarrow trunc\left(\dfrac{n}{1000}\right) \\ p \leftarrow n1\cdot n2\cdot n3\cdot n4 \end{array} \right.$$

Примечание к примеру 2: для решения задачи следует использовать функции:

mod(x,y) – возвращает остаток при делении числа x на y (x и y – действительные числа, результат наследует знак числа x) ;

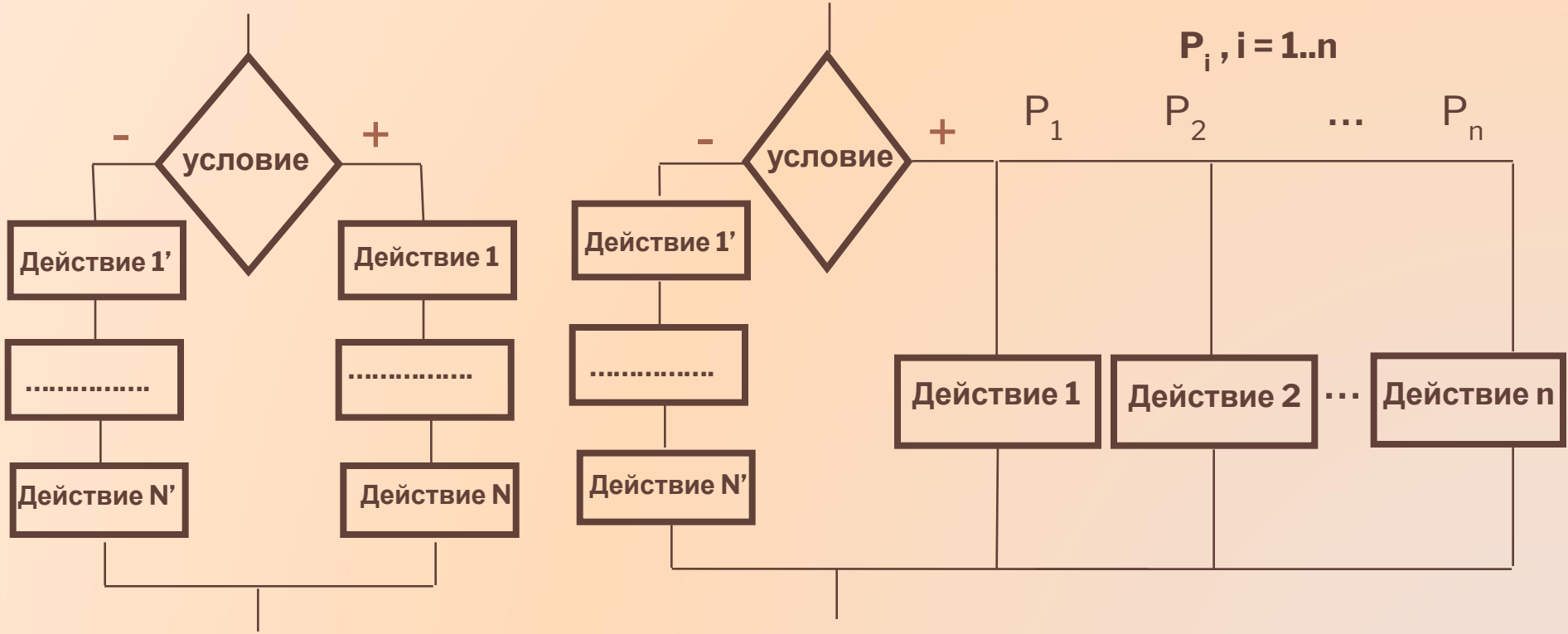
trunc(x) – возвращает целую часть действительного числа x.

$$g(4562) = 240$$

Разветвляющийся алгоритм (ветвление) – алгоритмическая конструкция, в которой в зависимости от значения логического выражения (условия) происходит выбор той или иной последовательности действий.

Ветвления

простые **сложные**



Для реализации ветвлений в программах-функциях используется:

- условная функция **if** (встроенная функция системы MathCAD);
- условный оператор **If Statement** и оператор иного выбора **Otherwise Statement** (с панели инструментов Программирование) .

Условие (логическое выражение)

простое

формат:

< выр. A > < опер. сравнения > < выр. B>

$d = 0 \quad x \geq 4 \quad \sin(x) \neq 0$

$x := 6$

$x \leq 2 = 0$

$x := 6 \qquad 2 \leq x \leq 8 = 1$

результаты вычисления логического выражения

сложное

формируется *из простых* с помощью логических операторов not, and, or, xor

$x := 2 \qquad y := 3$

$(x > 1) + (y < 1)\cdot\left(x^2 + y^2 \leq 4\right) = 1$

$[(x > 1) \vee (y < 1)] \wedge \left(x^2 + y^2 \leq 4\right) = 0$

Следует отметить, что при формировании *сложных условий* вместо операторов And, Or могут быть также использованы арифметические операторы сложения и умножения:

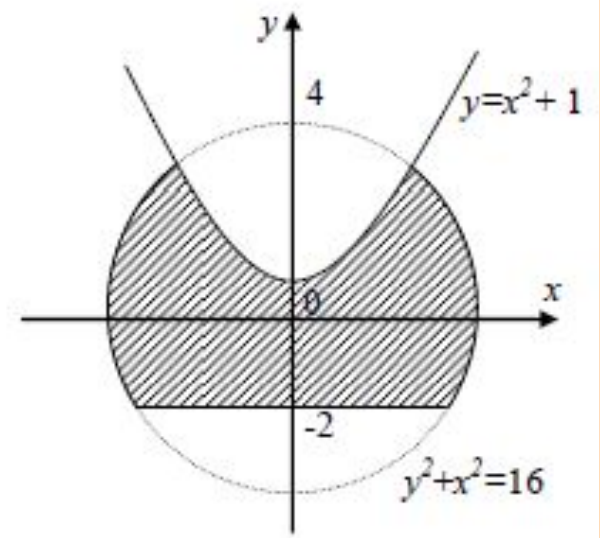
$$(x > 1) + (y < 1)\cdot\left(x^2 + y^2 \leq 4\right) = 1$$

Формат условной функции if:

if (< логич. выраж. > , < ариф.выраж.1> , < ариф.выраж.2 >)

Внутри программы-функции *условная функция* используется в арифметических выражениях, стоящих в правой части локального оператора присваивания (оператора внутреннего присваивания).

Пример 3. Написать программу-функцию, определяющую принадлежность точки с координатами x и y заштрихованной области на рисунке.



Решение: заштрихованная область – это точки, удовлетворяющие системе неравенств:

$$\begin{cases} y \ge -2 \\ y \le x^2 + 1 \\ x^2 + y^2 \le 16 \end{cases}$$

$$w(x,y) := \left| \begin{array}{l} z \leftarrow \text{if}\left[(y \ge -2)\cdot\left(y \le x^2+1\right)\cdot\left(y^2 \le 16 - x^2\right), \text{"точка принадлежит области"}, \text{"точка не принадлежит области"}\right] \\ z \end{array} \right.$$

$w(-2,-3.5) =$ "точка не принадлежит области"

$w(0,0) =$ "точка принадлежит области"

Формат использования

условного оператора **if** и оператора иного выбора **Otherwise**

$$f(x) := \begin{cases} \blacksquare \quad \text{if} \ \blacksquare \\ \blacksquare \end{cases}$$

поле для ввода логического выражения

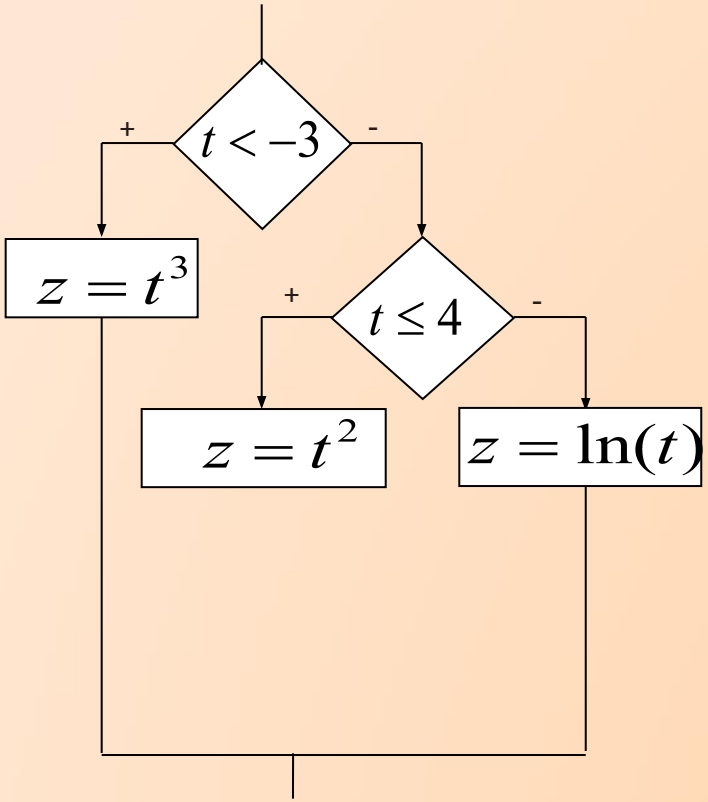
поле для ввода выражения, значение которого используется, если значение логического выражения true

$$f(x) := \begin{cases} \blacksquare \quad \text{if} \ \blacksquare \\ \blacksquare \quad \text{otherwise} \end{cases}$$

поле для ввода выражения, значение которого используется, если значение логического выражения false

Данные операторы могут находиться <u>только внутри тела программы-функции!</u>

Пример 4. Написать программу-функцию, вычисляющую значение кусочно-заданной функции *z*:



$$z(t) = \begin{cases} t^3, t < -3; \\ t^2, -3 \le t \le 4; \\ \ln(t), t > 4. \end{cases}$$

Предлагаются следующие варианты описания программы-функции:

1

$$f(t) := \left| \begin{array}{l} z \leftarrow t^3 \ \text{ if } \ t < -3 \\ z \leftarrow t^2 \ \text{ if } \ t \le 4 \\ z \leftarrow \ln(t) \ \text{ otherwise} \\ z \end{array} \right.$$

$$f(-4) = 16$$

2

$$f(t) := \left| \begin{array}{l} z \leftarrow t^3 \ \text{ if } \ t < -3 \\ \text{otherwise} \\ \quad \left| \begin{array}{l} z \leftarrow t^2 \ \text{ if } \ t \le 4 \\ z \leftarrow \ln(t) \ \text{ otherwise} \end{array} \right. \\ z \end{array} \right.$$

$$f(-4) = -64$$

Необходимо выбрать верный вариант ☝

Циклический алгоритм (цикл) - **алгоритмическая конструкция, которая содержит упорядоченную совокупность действий, повторяющихся многократно.**

Циклы можно условно разделить на две группы:

- **циклы типа арифметической прогрессии (с заданным числом повторений), реализуются** *с помощью оператора for*;
- **итерационные циклы, реализуются** *с помощью оператора while.*

Формат использования оператора for

$$f(x) := \left| \begin{array}{l} \text{for } \blacksquare \in \blacksquare \\ \quad \blacksquare \\ \blacksquare \end{array} \right.$$

поле для диапазона изменений параметра

поле для ввода параметра цикла

поле для ввода тела цикла

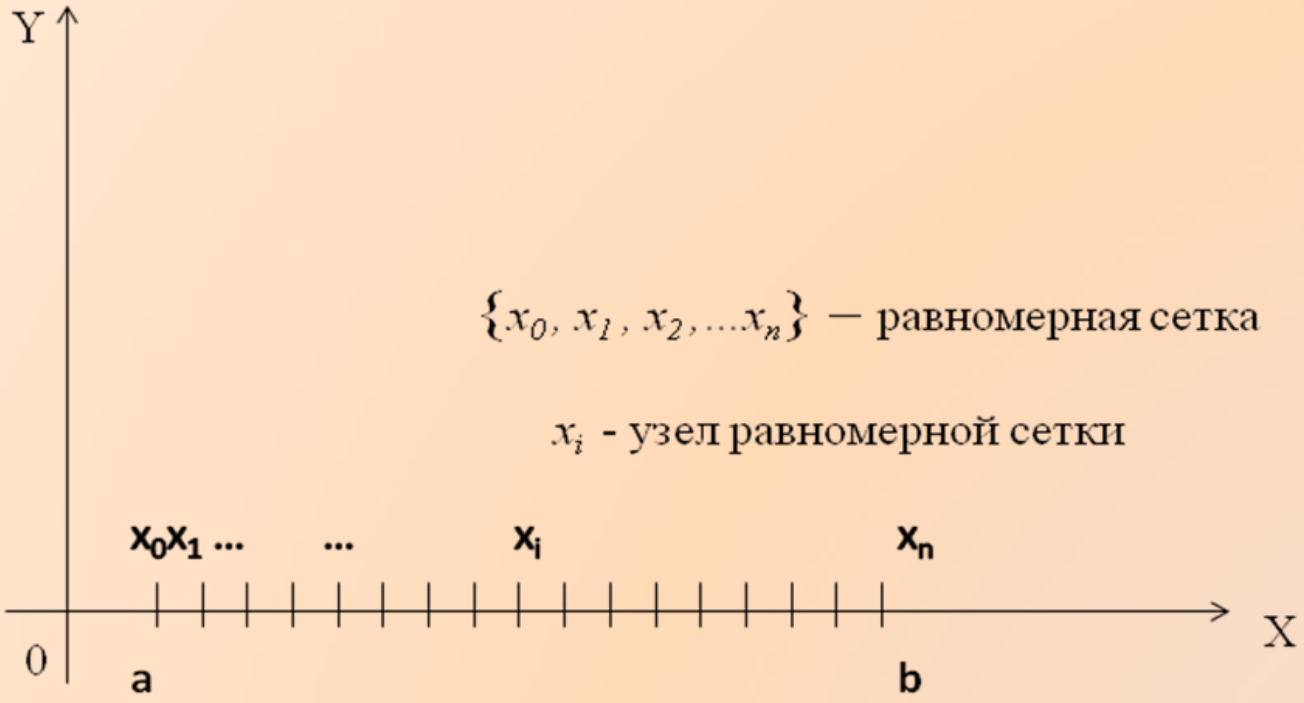
$$f(x) := \left| \begin{array}{l} \text{for } \blacksquare \in x0, x0 + h .. xk \\ \quad \blacksquare \\ \blacksquare \end{array} \right.$$

диапазон изменений параметра формируется с помощью ранжированной переменной

$$f(x) := \left| \begin{array}{l} \text{for } \blacksquare \in \blacksquare \\ \quad \left| \begin{array}{l} \blacksquare \\ \blacksquare \\ \blacksquare \end{array} \right. \\ \blacksquare \end{array} \right.$$

добавление полей с помощью Add Line

Пример 5. Табулирование функции одной переменной *y=f(x)* на отрезке [a;b] с шагом h.



$X_0 = a$
$X_1 = X_0 + h$
$X_2 = X_1 + h$
.....
$X_i = X_{i-1} + h$

Для решения задачи необходимо найти значения заданной функции в каждом узле получившейся совокупности точек и представить их в виде таблицы.

Конкретизируем постановку задачи: на отрезке от -2 до 2 с шагом 0,5 протабулировать функцию $f(x) = e^{-x} \cdot \mathrm{Cos}(2x)$.

$$\mathrm{form_tab}(x0, xk, d) := \left| \begin{array}{l} i \leftarrow 1 \\ \text{for } x \in x0, x0 + d .. xk \\ \quad \left| \begin{array}{l} z \leftarrow e^{-x} \cdot \cos(2 \cdot x) \\ y_i \leftarrow z \\ i \leftarrow i + 1 \end{array} \right. \\ y \end{array} \right.$$

$\mathrm{ORIGIN} := 1$

<описание программы-функции>

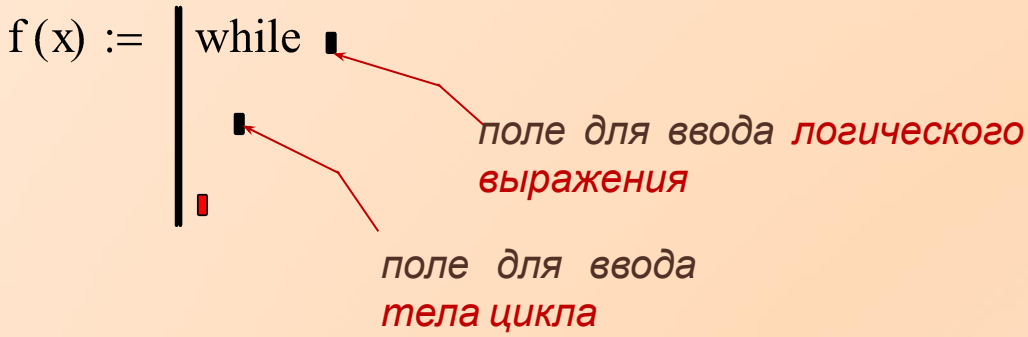
$$\mathrm{form_tab}(-2, 2, 0.5) = \begin{pmatrix} -4.83 \\ -4.437 \\ -1.131 \\ 0.891 \\ 1 \\ 0.328 \\ -0.153 \\ -0.221 \\ -0.088 \end{pmatrix}$$

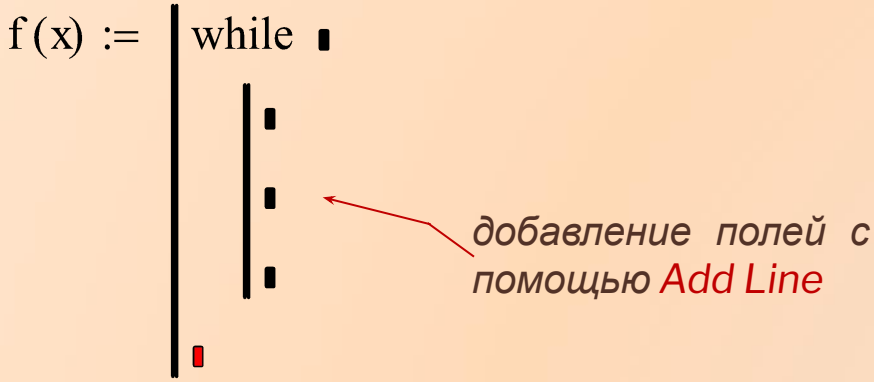
form_tab (−2, 2, 0.5) =

	0
0	0
1	-4.83
2	-4.437
3	-1.131
4	0.891
5	1
6	0.328
7	-0.153
8	-0.221
9	-0.088

отображается значение y_0=0, т.к. значение системной переменной ORIGIN по умолчанию равно 0

Формат использования оператора while:





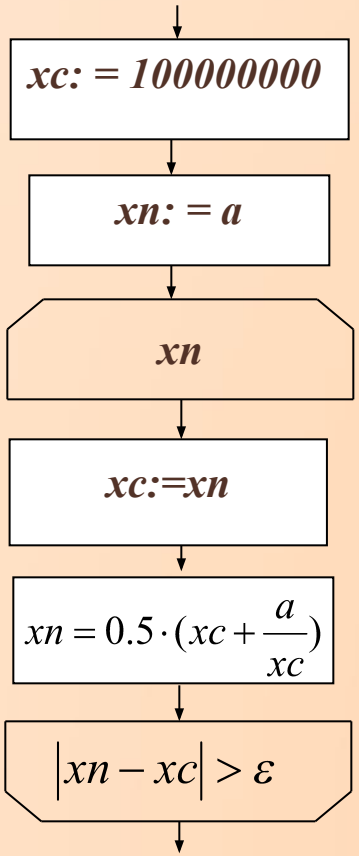
Для итерационных циклов нельзя априори определить количество повторений тела цикла. Это обусловлено тем, что окончание таких циклов определяется не выходом параметра цикла за конечное значение, а более сложными условиями.

Пример 6. Вычислить приближенное значение корня квадратного $x = \sqrt{a}$, используя итерационную процедуру:

$$x_n = 0.5 \cdot (x_{n-1} + \frac{a}{x_{n-1}}), \text{ где } n = 1,2,3..., x_0 = a$$

Известно, что в качестве приближенного значения корня квадратного берется такое значение x_n, которое удовлетворяет условию $|x_n - x_{n-1}| \le \varepsilon$

ε – *заданная точность.*




```
sqroot(a, eps) := | xc ← 10000000000
                  | xn ← a
                  | while |xn − xc| > eps
                  |   | xc ← xn
                  |   | xn ← (xc + a/xc)/2
                  | xn
```

$\text{sqroot}(9, 0.001) = 3$ $\qquad$ $\text{sqroot}(49, 0.01) = 7$

$\text{sqroot}(123, 0.0001) = 11.091$

$\text{sqroot}(-9, 0.001) =$

Формат использования оператора прерывания break:

$$f(x) := \left| \begin{array}{l} \text{break if } \blacksquare \\ \blacksquare \end{array} \right.$$

поле для ввода логического выражения

прерывание цикла, если значение логического выражения true

поле для ввода оператора, выполняемого если значение лог. выражения false

$$\text{sqroot}(a, eps) := \left| \begin{array}{l} xc \leftarrow a \\ \text{for } i \in 1..1000 \\ \quad \left| \begin{array}{l} xn \leftarrow \dfrac{xc + \dfrac{a}{xc}}{2} \\ \text{break if } |xn - xc| \le eps \\ xc \leftarrow xn \\ ierr \leftarrow \left| \begin{array}{ll} 1 & \text{if } i = 1000 \\ 0 & \text{otherwise} \end{array} \right. \\ x_0 \leftarrow xn \\ x_1 \leftarrow ierr \end{array} \right. \\ x \end{array} \right.$$

$$\text{sqroot}(-9, 0.001) = \begin{pmatrix} -0.909 \\ 1 \end{pmatrix}$$

$$\text{sqroot}(25, 0.001) = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

значение ierr

Спасибо за внимание