Модульное программирование

В пакете MathCad программный модуль реализуется с помощью подпрограммы-функции (П-Ф).

Для использования П-Ф составляется ее ***описание***.

Чтобы выполнить П-Ф, *ниже описания* организуется ***обращение*** (вызов) к П-Ф.

Описание подпрограммы-функции

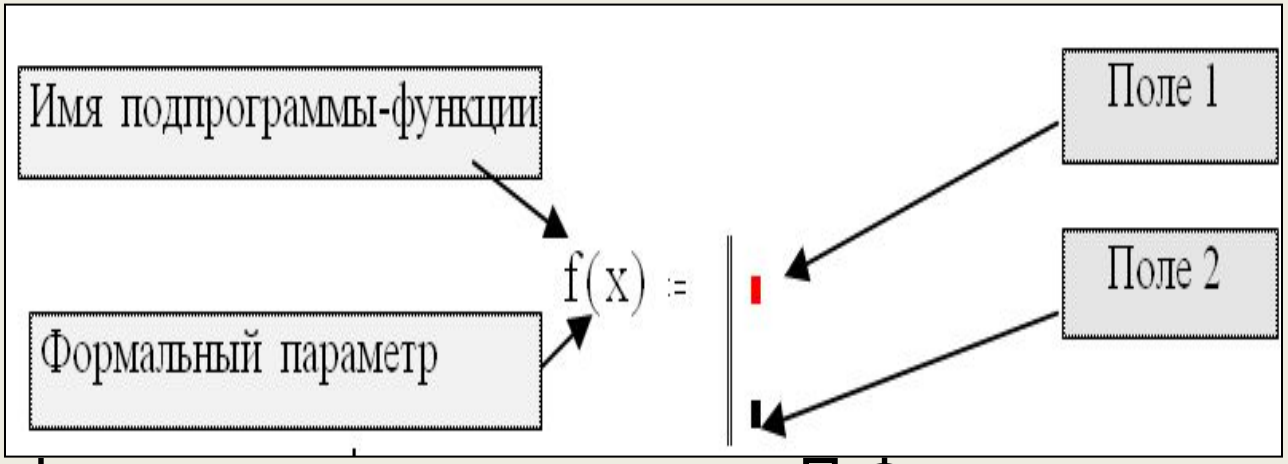
Описание П-Ф включает в себя:

- имя подпрограммы-функции;
- список формальных параметров;
- тело подпрограммы-функции.

Для ввода конструкций в тело П-Ф используется

Панель программирования

Структура П-Ф



Через формальные параметры в П-Ф передаются данные, необходимые для выполнения вычислений, т.е. все ***формальные параметры являются входными***. В качестве формальных параметров могут использоваться имена простых переменных, массивов и функций.

Вертикальная черта и *вертикальный столбец с полями* для ввода операторов, образующих тело П-Ф, появляются при щелчке на кнопке **Add Line,** расположенной на ***Панели программирования***.

Тело П-Ф включает в себя любое число операторов (локальных операторов присваивания, условных операторов и операторов цикла), а также вызов других П-Ф и функций

Замечание 1. П-Ф может не иметь формальных параметров, и тогда данные передаются через имена переменных, заданных выше описания П-Ф.

Замечание 2. Самое нижнее поле ввода в структуре

П-Ф служит для записи переменной или выражения, определяющих возвращаемое через имя П-Ф результат.

Замечание 3. Если результатом работы П-Ф являются несколько величин, то из них в теле П-Ф необходимо сформировать массив и его имя поместить в нижнее поле П-Ф.

Локальный оператор присваивания

Для задания внутри П-Ф значения или выражения

< имя переменной > □ < выражение >

оператор присваивания, имеющий вид:

Обращение к подпрограмме-функции

Для выполнения П-Ф необходимо обратиться к ней с указанием ***имени*** и ***списка фактических параметров*** (если в описании П-Ф присутствует список формальных параметров). Вызов П-Ф имеет

< имя П-Ф > (< список фактических параметров >)

Фактические параметры определяют конкретные значения, при которых выполняются вычисления в П-Ф.

Между фактическими и формальными параметрами должно быть соответствие по ***количеству, порядку следования и типу***.

Обращение задается ниже описания, и к моменту вызова фактические параметры *должны*

Программирование линейных алгоритмов

Пример. Составить П-Ф для вычисления значения функции $z(x)$, определяемую выражением:

$$z(x) = ay^{5+x} + b\cos|y|, \text{ где } y = b\sin^2 x + b^2\sin^4 x^2$$

Вычислить значения функции при:

а) **a = 1.2;** **b = 3;** x **= 0.45;**

б) **a = 1.2;** **b = 3;** x **= –8.34**

Решение

$a := 1.2 \qquad b := 3$

$$z(x) := \left\| \begin{array}{l} y \leftarrow b\cdot\sin(x)^2 + b^2\cdot\sin\left(x^2\right)^4 \\ a\cdot y^{5+x} + b\cdot\cos(|y|) \end{array} \right.$$

$z(0.45) = 2.569 \qquad z(-8.34) = -2.588$

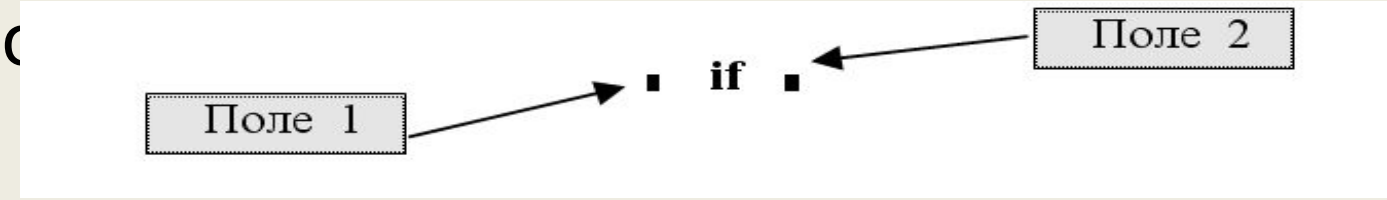
Программирование разветвляющихся алгоритмов

Для программирования разветвляющихся алгоритмов в П-Ф можно использовать:

- *условную функцию if;*
- *условный оператор if* (только в П-Ф).

Реализация структуры ЕСЛИ-ТО

Испо



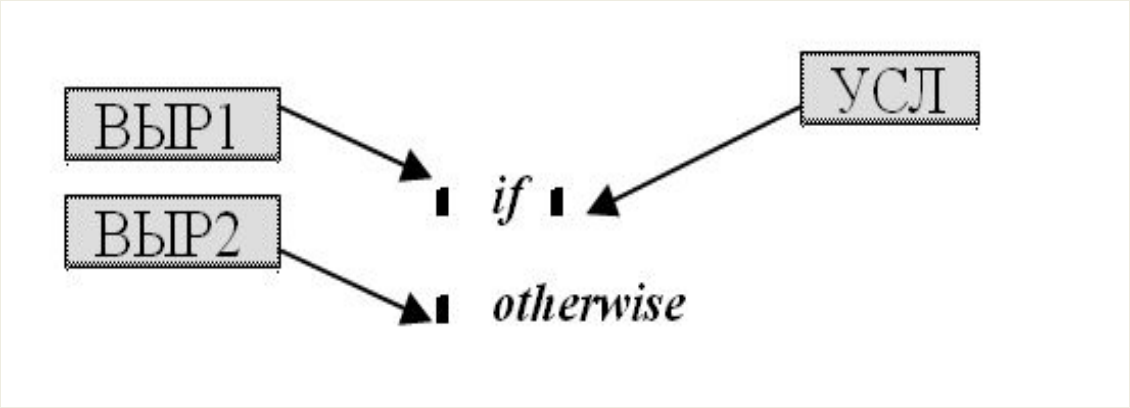
Для ввода условного оператора надо щелкнуть на кнопке **if** на **Панели программирования**.

В **Поле 2** вводится *логическое выражение* (условие).

В **Поле 1** вводится *выражение* или *локальный*

Реализация структуры ЕСЛИ-ТО-ИНАЧЕ

Используются условный оператор **if** и оператор **otherwise**.



Для ввода операторов надо щелкнуть на кнопках **if** и **otherwise** на **Панели программирования**.

Конструкция **ВЫР1**, стоящая перед оператором **if**, выполняется, если логическое выражение (условие) равно **1** (**ИСТИНА**).

Конструкция **ВЫР2**, стоящая перед оператором **otherwise**, выполняется, если логическое выражение (условие) равно **0** (**ЛОЖЬ**).

Примеры программирования разветвляющихся алгоритмов

Пример 1. Составить описание П-Ф для вычисления функции $\mu(x, \varepsilon)$ по формуле:

$$\mu(x,\varepsilon) = \begin{cases} \dfrac{1}{2\sqrt{x+1}}, \text{если } |x-y| < \varepsilon; \\ \dfrac{1}{3}\sqrt[3]{x+y}, \text{если } |x-y| \geq \varepsilon, \end{cases}$$

где $y = |x|$

Решение

Вариант 1
(использование структуры ЕСЛИ-ТО)

$$\mu(x,\varepsilon) := \left| \begin{array}{l} y \leftarrow |x| \\ \dfrac{1}{2\cdot\sqrt{x+1}} \quad \text{if} \quad |x-y| < \varepsilon \\ \dfrac{\sqrt[3]{x+y}}{3} \quad \text{if} \quad |x-y| \geq \varepsilon \end{array} \right.$$

$$\mu(12.8,4) = 0.135 \qquad \mu(-12.8,4) = 0$$

Примеры программирования разветвляющихся алгоритмов

Вариант 2
(использование структуры ЕСЛИ-ТО-ИНАЧЕ)

$$\mu(x,\varepsilon) := \left| \begin{array}{ll} y \leftarrow |x| & \\ \dfrac{1}{2\cdot\sqrt{x+1}} & \text{if } |x-y| < \varepsilon \\ \dfrac{\sqrt[3]{x+y}}{3} & \text{otherwise} \end{array} \right.$$

$$\mu(12.8,4) = 0.135 \qquad \mu(-12.8,4) = 0$$

Примеры программирования разветвляющихся алгоритмов

Пример 2. Составить описание П-Ф для вычисления значения ***z*** по одной из трех ветвей:

$$z = \begin{cases} 30, & \text{если } x \le -1; \\ |x|, & \text{если } -1 < x \le 1; \\ x^2 - 30, & \text{если } x > 1 \end{cases}$$

Решение

$$z(x) := \begin{cases} 30 & \text{if } x \le -1 \\ |x| & \text{if } -1 < x \le 1 \\ x^2 - 30 & \text{otherwise} \end{cases}$$

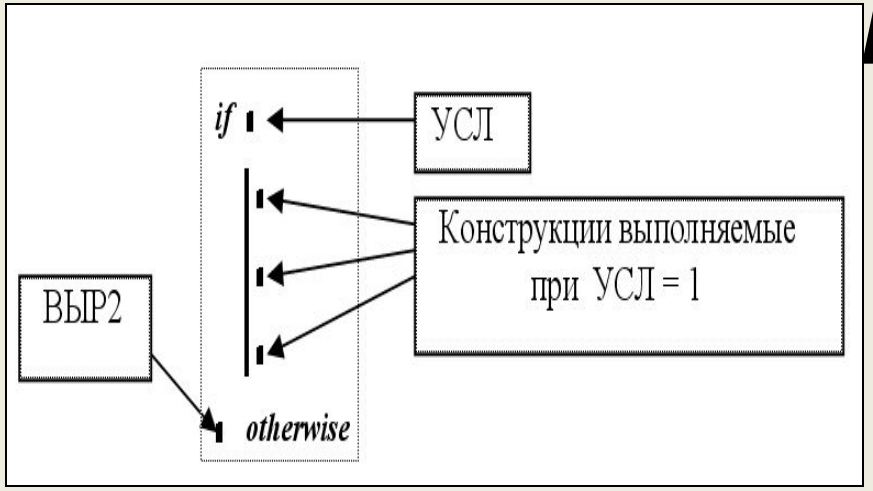
$z(-3) = 30 \qquad z(-0.5) = 0.5 \qquad z(3) = -21$

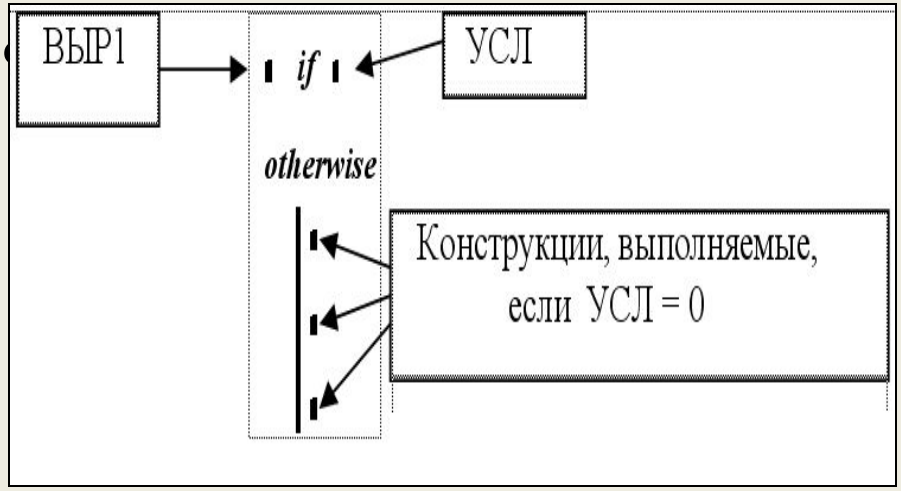
Выражение, в операторе **otherwise** будет вычисляться только в том случае, когда не выполнятся условия **в двух вышестоящих операторах if**.

Реализация сложной структуры разветвляющихся алгоритмов

В сложных алгоритмах в операторах ***if*** и ***otherwise*** требуется выполнять не одну, а *несколько конструкций*.

Для реализации такой структуры необходимо в поле перед оператором ***if*** или ***otherwise*** щелкнуть нужное число раз на кнопке **Add Line** на **Панели программирования**.





Примеры программирования разветвляющихся алгоритмов

Пример 3. Даны два числа ***x***, ***y***. Составить описание П-Ф, которая переменной ***x*** присваивает максимальное значение из этих двух чисел, а ***y*** – минимальное.

Решение

$$
\begin{array}{l}
P(x,y) := \left| \begin{array}{l}
\text{if } x > y \\
\quad \left| \begin{array}{l} v_0 \leftarrow x \\ v_1 \leftarrow y \end{array} \right. \\
\text{otherwise} \\
\quad \left| \begin{array}{l} v_0 \leftarrow y \\ v_1 \leftarrow x \end{array} \right. \\
v
\end{array} \right. \\
\\
\begin{pmatrix} x \\ y \end{pmatrix} := P(4,9) = \begin{pmatrix} 9 \\ 4 \end{pmatrix} \\
\\
x = 9 \qquad y = 4
\end{array}
$$

Результат в П-Ф оформлен в виде массива **v = (v_0, v_1)**, так как по правилам описания имени П-Ф может быть присвоено значение ***только одной переменной***

(в данном случае **v** – это имя одной *переменной-массива*).

Программирование циклических алгоритмов

По способам организации циклов в П-Ф также выделяются две группы:

а) *циклы типа арифметической прогрессии*;

б) *итерационные циклы*.

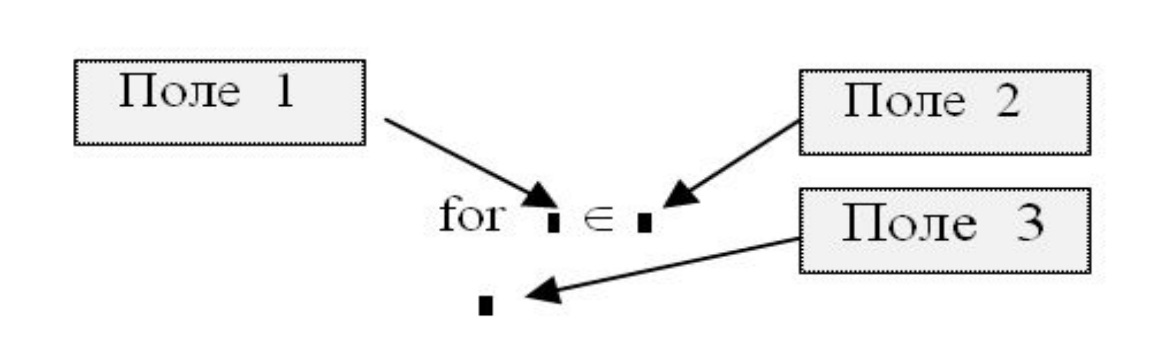
Программирование циклов типа арифметической прогрессии

Для программирования таких циклов используется *оператор цикла с параметром* ***for***.

Особенности оператора цикла for

1. Параметр цикла может принимать значения различных типов: численные (целые или вещественные), текстовые и др.
2. Значения параметра цикла могут задаваться *дискретной переменной, последовательностью чисел, массивом*

Структура оператора цикла for



Для ввода оператора цикла с параметром надо щелкнуть на кнопке ***for*** на **Панели программирования**.

В **Поле 1** вводится *имя переменной*, являющейся параметром цикла.

В **Поле 2** задается *закон изменения параметра* цикла.

В **Поле 3** вводятся операторы, составляющие тело цикла. Если одного поля недостаточно, то дополнительные поля для ввода операторов

Примеры программирования циклических алгоритмов с параметром цикла

Пример 1. Составить описание П-Ф, реализующей формирование вектора ***z*** из ***n*** (*n*=5) элементов, определяемых по правилу:

$$z_i = \frac{1}{i+4}$$

Решение

ORIGIN := 1

vect(n) := | for $i \in 1..n$

$z_i \leftarrow \frac{1}{i+4}$

z

$z := vect(5)$ $\quad z^T = (0.2 \quad 0.167 \quad 0.143 \quad 0.125 \quad 0.111)$

Примеры программирования циклических алгоритмов с параметром цикла

Пример 2. Для переменной ***x***, изменяющейся от **0.5** до **1.5** с шагом **0.2** сформировать вектор ***q***, состоящ[illegible] значений функции:

$$y(x) = \frac{\ln|x|}{a^2 + b^2}$$

где a и b – заданные вещественные числа.

Примеры программирования циклических алгоритмов с параметром цикла

Пример 2 (решение)

Вариант 1. Параметр цикла – переменная ***x***.

$$\mathrm{ORIGIN} := 1$$

$$f1(a,b) := \left| \begin{array}{l} i \leftarrow 1 \\ \text{for } x \in 0.5, 0.7 .. 1.5 \\ \quad \left| \begin{array}{l} y_i \leftarrow \dfrac{\ln(|x|)}{a^2 + b^2} \\ i \leftarrow i + 1 \end{array} \right. \\ y \end{array} \right.$$

$$q := f1(2.1, 4.56)$$

$$q^T = \begin{pmatrix} -0.028 & -0.014 & -4.18 \times 10^{-3} & 3.782 \times 10^{-3} & 0.01 & 0.016 \end{pmatrix}$$

Примеры программирования циклических алгоритмов с параметром цикла

Пример 2 (решение)

Вариант 2. Параметр цикла – переменная ***i***.

$$n := \mathrm{trunc}\left(\frac{1.5 - 0.5}{0.2}\right) + 1 = 6$$

n - число элементов массива **q**

$$f2(a,b) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \left| \begin{array}{l} x_i \leftarrow 0.5 + (i-1)\cdot 0.2 \\ y_i \leftarrow \dfrac{\ln(|x_i|)}{a^2 + b^2} \end{array} \right. \\ y \end{array} \right.$$

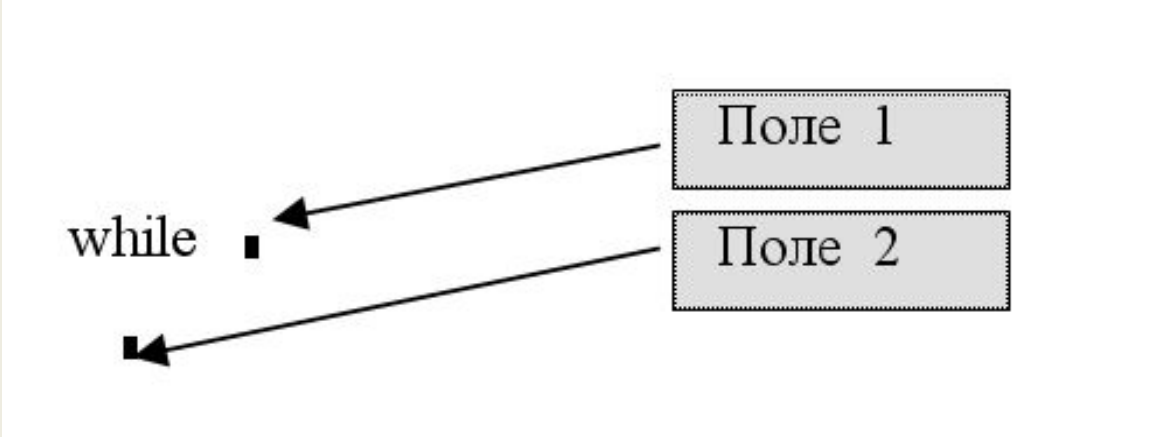
$$q := f2(2.1, 4.56)$$

$$q^T = \begin{pmatrix} -0.028 & -0.014 & -4.18 \times 10^{-3} & 3.782 \times 10^{-3} & 0.01 & 0.016 \end{pmatrix}$$

Программирование итерационных циклов

Для программирования итерационных циклов используется *оператор цикла* ***while***.

Структура оператора цикла while



Для ввода оператора надо щелкнуть на кнопке ***while*** на **Панели программирования**.

В **Поле 1** вводится условие выполнения цикла.

В **Поле 2** вводятся операторы тела цикла. В теле цикла должны присутствовать операторы, которые *изменяют значение переменной, управляющей циклом*

Программирование итерационных циклов

Пример

Составить П-Ф, реализующую приближенное вычисление корня квадратного $\sqrt{a}$ по итерационной формуле:

$$x_{n+1} = 0.5 \cdot \left(x_n + \frac{a}{x_n}\right), \; n = 0, 1, \ldots; \; x_0 = a.$$

В качестве приближенного значения x_{n+1} принимается , удовлетворяющее условию:

$$\left| x_{n+1}^2 - a \right| \le \varepsilon$$

где ε – заданная точность вычисления корня квадратного.

Программирование итерационных циклов

Пример (*решение*)

При составлении П-Ф нет необходимости запоминать все приближенные решения x_0, x_1, x_2,…на каждой итерации. Достаточно хранить два последних значения. Обозначим:

x – решение, полученное на текущей итерации;

x1 – решение, полученное на предыдущей итерации.