МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ В MATHCAD

Сущность и преимущества модульного программирования

Модульная программа – программа, реализующая алгоритм вычислений в виде модулей и обращения к ним.

Сущность:

- в разбиении алгоритма решения задачи на слабо зависимые фрагменты вычислений (на подзадачи) и реализация каждого фрагмента в виде программных модулей;
- в вызове в нужных местах «основной» программы соответствующих модулей с передачей необходимых данных.

Преимущества:

- возможность распараллеливания разработки программы;
- простота отладки;
- простота сопровождения и модификации;
- возможность использования библиотек «готовых» модулей.

Описание подпрограммы-функции

ПФ предназначены для многократного выполнения фрагментов программы без повторения их записи.

- ***имя подпрограммы-функции***

Оригинальное имя, используемое для обращения к ПФ и «возврата» результата ее выполнения.

- ***список формальных параметров***

Передает данные, необходимые для выполнения вычислений внутри ПФ: переменные, массивы, функции.

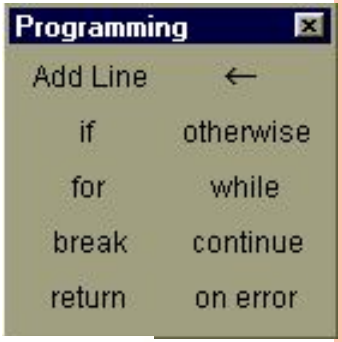
Формальные параметры заключаются в круглые скобки и отделяются запятой (список формальных параметров может отсутствовать).

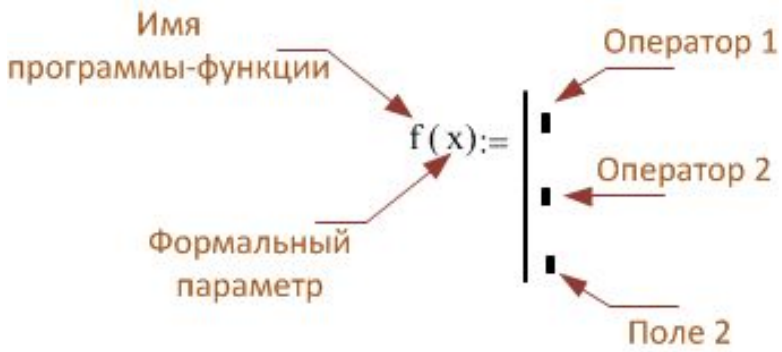
- ***тело подпрограммы-функции***

Включает любое число операторов: локальных операторов присваивания, условных операторов и операторов цикла, а также вызовов других ПФ и функций пользователя.

Порядок описания подпрограммы-функции

- Ввести имя ПФ и список формальных параметров, заключенный в круглые скобки.
- Ввести символ присваивания «:=».
- Открыть панель инструментов **Programming** (Программирование) и нажать кнопку **Add Line** (Добавить линию)





- В поле 1 вводится оператор тела ПФ.

Увеличение числа полей для ввода дополнительных операторов осуществляется с помощью кнопки **Add Line** или с клавиатуры «]»

Удаление оператора или поля ввода осуществляется нажатием клавиши **Delete**

- Заполнить поле 2 (нижнее поле), введя выражение, определяющее возвращаемое через имя ПФ значение

Локальный оператор присваивания

<Имя переменной> ← *<Выражение>*

Обращение к подпрограмме функции

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

Обращение к П-Ф должно находиться после ее описания, и ***к моменту обращения фактические параметры должны быть определены***.

$$x := 7$$

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

Варианты обращения к ПФ

$f(x) = 3$ $\quad x = 7$

$f(5) = 2.646$

$z := f(x + 7)$ $\quad z = 4$

Программирование линейных алгоритмов

Под линейным алгоритмом понимается вычислительный процесс, в котором необходимые операции выполняются строго последовательно.

$$\text{root_poly2}(a,b,c) := \left| \begin{array}{l} d \leftarrow b^2 - 4\cdot a\cdot c \\ x1 \leftarrow \dfrac{-b + \sqrt{d}}{2\cdot a} \\ x2 \leftarrow \dfrac{-b - \sqrt{d}}{2\cdot a} \\ \begin{pmatrix} x1 \\ x2 \end{pmatrix} \end{array} \right.$$

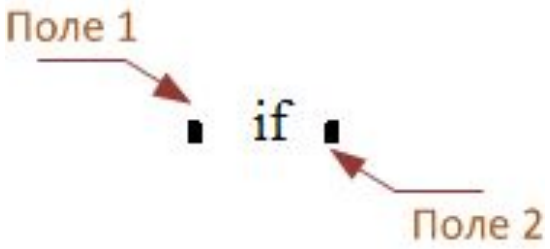
$$z := \text{root_poly2}(2,5,8) = \begin{pmatrix} -1.25 + 1.561i \\ -1.25 - 1.561i \end{pmatrix}$$

Программирование разветвляющихся алгоритмов

В разветвляющихся алгоритмах присутствует несколько ветвей вычислительного процесса. Выбор конкретной ветви зависит от выполнения (или невыполнения) заданных условий.

Условный оператор *if*

Условная структура **ЕСЛИ- ТО**

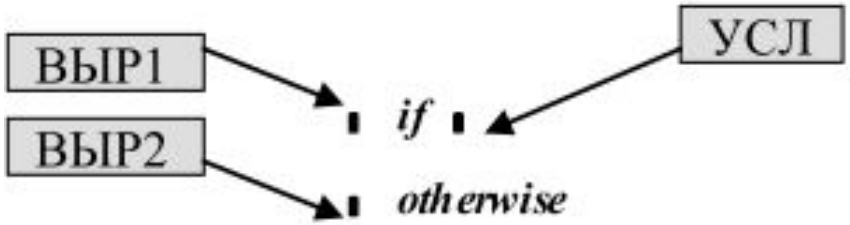


В поле 2 вводится логическое выражение (выражение отношений).

В поле 1 вводится выражение (как правило, арифметическое), значение которого используется, если проверяемое логическое выражение принимает значение 1 (истина)

Программирование разветвляющихся алгоритмов

Условная структура **ЕСЛИ- ТО - ИНАЧЕ**



ВЫР2 выполняется, если проверяемое логическое выражение (**УСЛ**) = **0**.

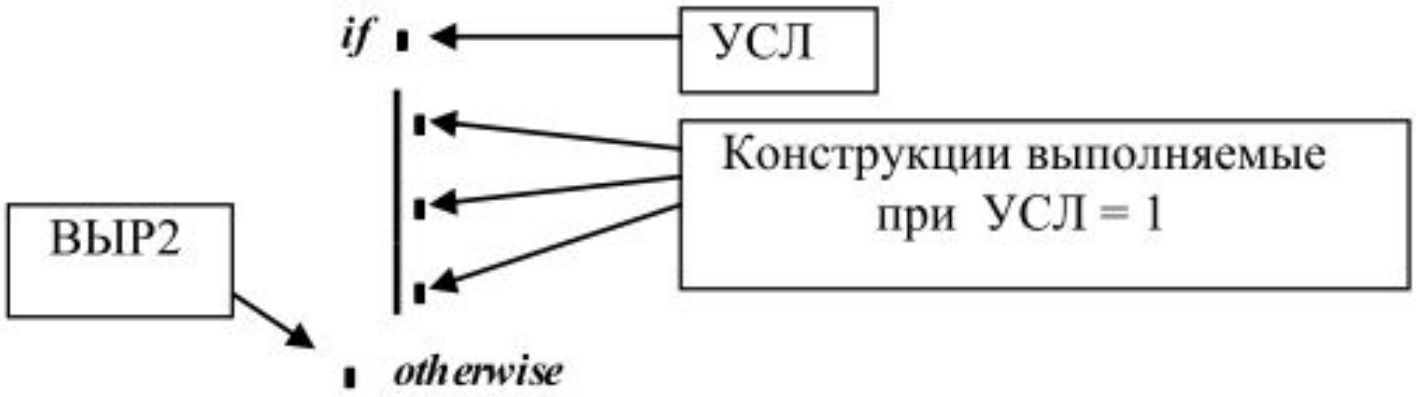
Оператор ***oterwise*** следует непосредственно после условного оператора ***if***

$$y(x) := \begin{cases} \ln(x) & \text{if } x > 0 \\ e^{x} & \text{otherwise} \end{cases}$$

$y(-3) = 0.05$ $\qquad$ $y(2.71) = 0.997$

Программирование сложных разветвляющихся алгоритмов

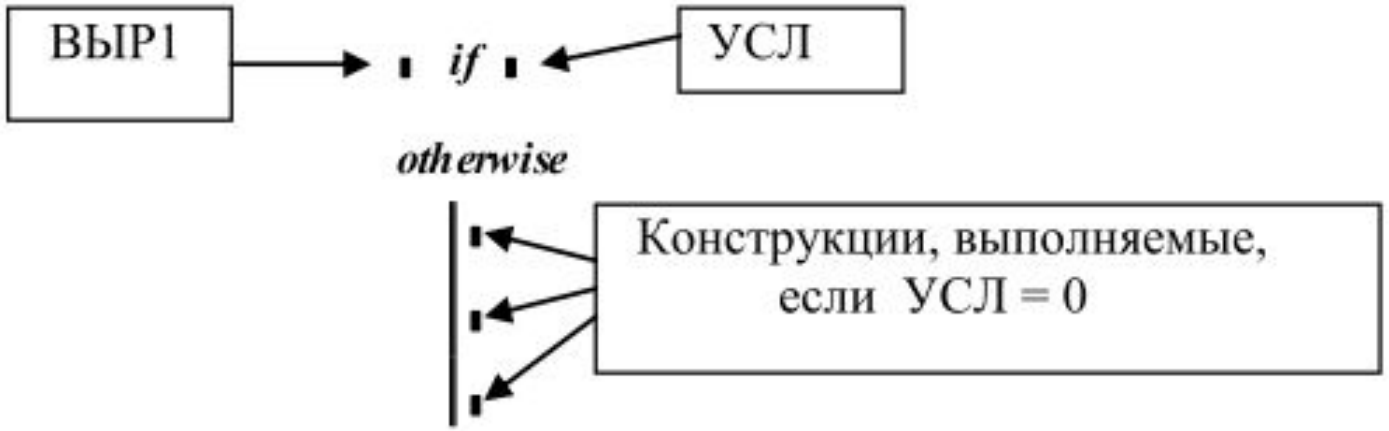
***Вариант 1*: При выполнении заданного условия УСЛ необходимо выполнить несколько конструкций**



- выделить поле 1 условного оператора ***if***
- нажать кнопку ***Add line*** нужное число раз
- заполнить появившиеся поля

Программирование сложных разветвляющихся алгоритмов

Вариант 2: При невыполнении заданного условия УСЛ необходимо выполнить несколько конструкций



- выделить поле оператора ***otherwise***
- нажать кнопку ***Add line*** нужное число раз
- заполнить появившиеся поля

Программирование сложных разветвляющихся алгоритмов

Составить ПФ, возвращающую значения двух переменных a и b, возведенных во вторую или третью степень. Степень задается переменной n. Если $n < 2$ или $n > 3$, то значения переменных принять равными 0.

$$\text{poly}(n,a,b) := \left|\begin{array}{l} \text{if } n = 2 \\ \quad \left|\begin{array}{l} x \leftarrow a^2 \\ y \leftarrow b^2 \end{array}\right. \\ \text{if } n = 3 \\ \quad \left|\begin{array}{l} x \leftarrow a^3 \\ y \leftarrow b^3 \end{array}\right. \\ \text{otherwise} \\ \quad \left|\begin{array}{l} x \leftarrow 0 \\ y \leftarrow 0 \end{array}\right. \\ \begin{pmatrix} x \\ y \end{pmatrix} \end{array}\right.$$

$$\text{poly}(-1,2,-3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\text{poly}(0,2,-3) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\text{poly}(2,2,-3) = \begin{pmatrix} 4 \\ 9 \end{pmatrix}$$

$$\text{poly}(3,2,-3) = \begin{pmatrix} 8 \\ -27 \end{pmatrix}$$

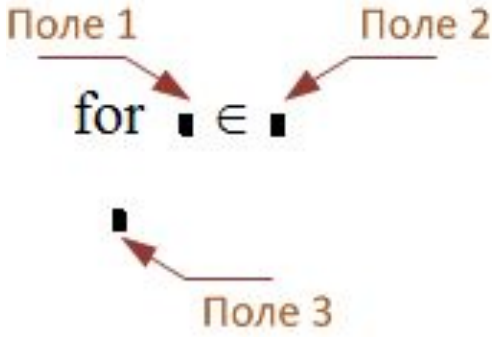
Программирование циклических алгоритмов

Циклические алгоритмы (циклы) содержат повторяющиеся вычисления, зависящие от некоторой переменной.

Такая переменная называется *параметром цикла*, а сами повторяющиеся вычисления составляют *тело цикла*.

Программирование цикла типа арифметической прогрессии

Оператор цикла *for*



- В поле 1 вводится имя переменной, являющейся параметром цикла;
- В поле 2 вводится закон изменения параметра цикла, используя для этого описание дискретной переменной или описание массива;
- В поле 3 вводятся операторы, составляющие тело цикла.

Вычислить значения функции

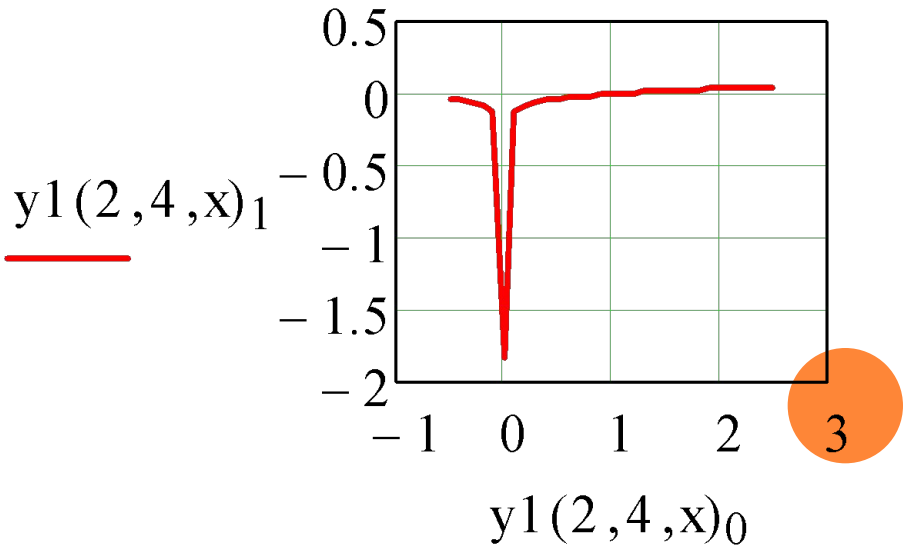
$$y(x)=\frac{\ln|x|}{a^2+b^2}$$

для всех $\boldsymbol{x}$, изменяющихся в интервале [–0.5, 2.5] с шагом $\Delta\boldsymbol{x}$ = 0.1;
$\boldsymbol{a}$, $\boldsymbol{b}$ – заданные вещественные числа.
Построить график этой функции

$x0 := -0.5 \qquad xk := 2.5 \qquad d := 0.1$

$$y1(a,b,x) := \left|\begin{array}{l} i \leftarrow 0 \\ \text{for } x \in x0, x0+d .. xk \\ \quad \left|\begin{array}{l} z_i \leftarrow \dfrac{\ln(|x|)}{a^2+b^2} \\ X_i \leftarrow x \\ i \leftarrow i+1 \end{array}\right. \\ \begin{pmatrix} X \\ z \end{pmatrix} \end{array}\right.$$

$$y1(2,4,x) = \begin{pmatrix} \{31,1\} \\ \{31,1\} \end{pmatrix}$$



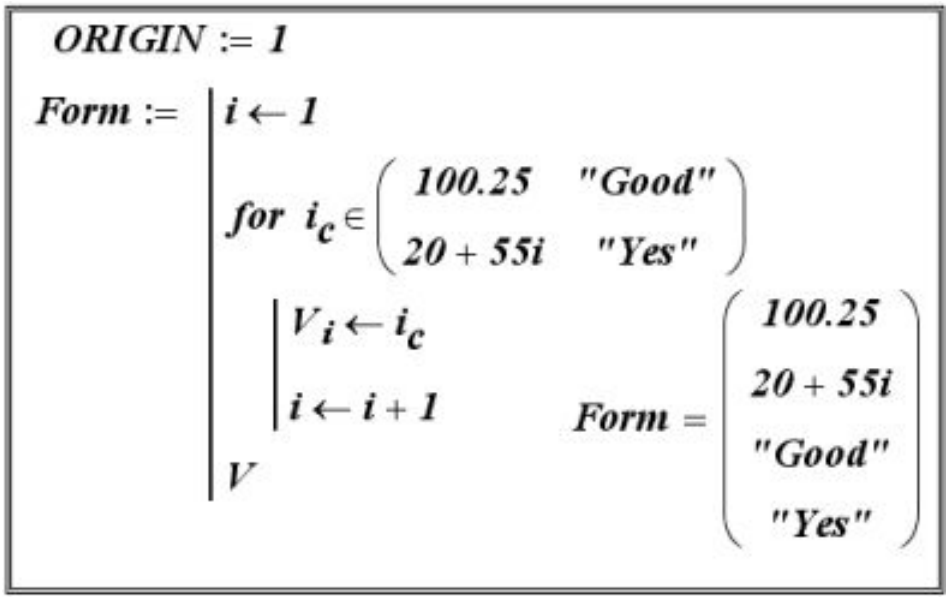
Составить описание П-Ф, вычисляющей сумму вектора с проекциями |2,5,7|

$$\begin{array}{l} sum_vec_3 := \left| \begin{array}{l} S \leftarrow 0 \\ \text{for } i \in \begin{pmatrix} 2 \\ 5 \\ 7 \end{pmatrix} \\ \quad S \leftarrow S + i \\ S \end{array} \right. \\ sum_vec_3 = 14 \end{array}$$

Составить описание П-Ф, вычисляющей сумму следующих слагаемых 2, 3, 5, 9, 11, 17, 21.

$$\begin{array}{l} sum_1 := \left| \begin{array}{l} S \leftarrow 0 \\ \text{for } i \in 2,3,5,9,11,17,21 \\ \quad S \leftarrow S + i \\ S \end{array} \right. \\ sum_1 = 68 \end{array}$$

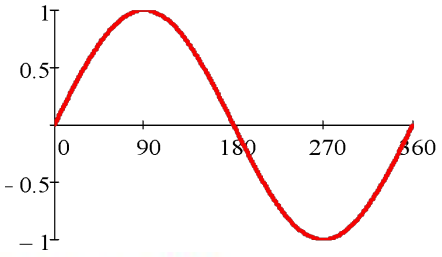
Составить П-Ф, которая формирует двумерный массив из объектов разных типов.



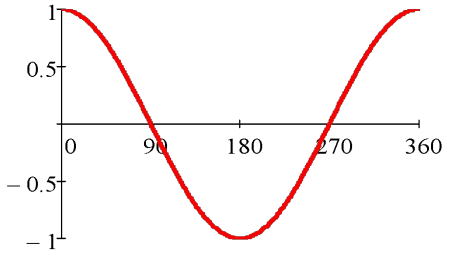
Составить подпрограмму-функцию, осуществляющую построение графика в соответствии с рисунком (выполнить модульное программирование в MathCAD).

Примеры графиков функций $f(t) = Sin(t)$ (рисунок а) и $f(t) = Cos(t)$ (рисунок б).

а)

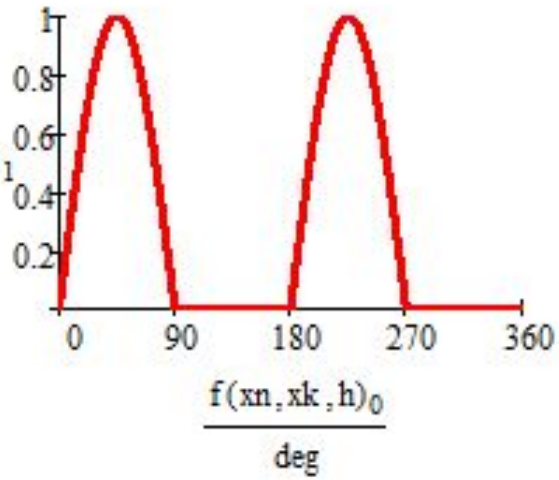


б)



$xn := 0 \quad xk := 2 \cdot \pi \quad h := 0.001$

$$f(xn, xk, h) := \left| \begin{array}{l} k \leftarrow 0 \\ \text{for } t \in xn, xn + h .. xk \\ \quad \left| \begin{array}{l} y_k \leftarrow \left| \begin{array}{ll} \sin(2t) & \text{if } \left(0 \le t < \frac{\pi}{2}\right) \vee \left(\pi \le t < \frac{3 \cdot \pi}{2}\right) \\ 0 & \text{otherwise} \end{array} \right. \\ x_k \leftarrow t \\ k \leftarrow k + 1 \end{array} \right. \\ \begin{pmatrix} x \\ y \end{pmatrix} \end{array} \right.$$



Программирование итерационных циклов

Оператор цикла *while*

– внутренние операторы цикла будут исполняться до тех пор, пока будет истинным условие, следующее за ключевым словом ***while***.

- В поле 1 вводится условие выполнение цикла;
- В поле 2 вводятся операторы тело цикла.

В теле цикла должны присутствовать операторы, которые могут изменить значение условия цикла, иначе цикл будет продолжаться бесконечно.

Решить методом Ньютона

$$f(x) := 5\cdot\sin(x) - 4\cdot x^2 + 5.8 \qquad \varepsilon := 10^{-4} \qquad x_0 := 3$$

$$\text{root_Nyuton} := \left|\begin{array}{l} i \leftarrow 1 \\ df(x) \leftarrow \dfrac{d}{dx} f(x) \\ x_0 \leftarrow 3 \,, x_1 \leftarrow x_0 - \dfrac{f(x_0)}{df(x_0)} \\ \text{while } \left|x_i - x_{i-1}\right| > \varepsilon \\ \quad \left|\begin{array}{l} x_{i+1} \leftarrow x_i - \dfrac{f(x_i)}{df(x_i)} \\ i \leftarrow i + 1 \end{array}\right. \\ x \end{array}\right.$$

$$\text{root_Nyuton}^T = (3 \quad 1.9811938 \quad 1.6833021 \quad 1.6429796 \quad 1.6421985 \quad 1.6421982)$$

Программирование итерационных циклов

Оператор *break*

– служит для преждевременного завершения цикла, чтобы, например, избежать зацикливания или слишком продолжительных вычислений.

Оператор ***break*** используется в левом поле ввода условного оператор ***if***, а в правом размещается условие, при выполнении которого происходит прекращение работы цикла или программы.

$$\text{root_Nyut}(x0, \varepsilon) :=$$

ierr ← 1 , x_0 ← x0

$$df(x) \leftarrow \frac{d}{dx} f(x)$$

$$x_1 \leftarrow x_0 - \frac{f(x_0)}{df(x_0)}$$

for $i \in 1 .. 5$

  if $|x_i - x_{i-1}| < \varepsilon$

    ierr ← 0

    break

  $$x_{i+1} \leftarrow x_i - \frac{f(x_i)}{df(x_i)}$$

x ← "not solve" if ierr = 1

$$\begin{pmatrix} x \\ ierr \end{pmatrix}$$

$$\text{root_Nyut}(3, 10^{-4})^T = \begin{bmatrix} \begin{pmatrix} 3 \\ 1.9811938 \\ 1.6833021 \\ 1.6429796 \\ 1.6421985 \\ 1.6421982 \end{pmatrix} & 0 \end{bmatrix}$$

$$\text{root_Nyut}(3, 10^{-10})^T = (\text{"not solve"} \quad 1)$$

Программирование двойных циклов

for ∎ ∈ ∎	for ∎ ∈ ∎	while ∎
for ∎ ∈ ∎	while ∎	for ∎ ∈ ∎
∎	∎	∎

Сформировать матрицу (двумерный массив) ***B*** по следующему правилу:

$$B_{i,j} = \frac{1}{i + j + 1};$$

ORIGIN := 1

$$\text{form_M}(n, m) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad \text{for } j \in 1..m \\ \quad\quad B_{i,j} \leftarrow \dfrac{1}{i + j + 1} \\ B \end{array} \right.$$

$$\text{form_M}(4, 2) = \begin{pmatrix} 0.333 & 0.25 \\ 0.25 & 0.2 \\ 0.2 & 0.167 \\ 0.167 & 0.143 \end{pmatrix}$$

Дополнительные операторы, используемые при программировании циклов

Оператор *continue*

Обычно используется для продолжения выполнения цикла путем возврата в начало тела цикла

Составить описание П-Ф, формирующую новый вектор из положительных элементов исходного вектора.

$$\text{form_V}(v) := \left| \begin{array}{l} i \leftarrow 0 \,, j \leftarrow 0 \\ \text{while } i < \text{last}(v) \\ \quad \left| \begin{array}{l} i \leftarrow i + 1 \\ \text{continue} \quad \text{if } v_i \leq 0 \\ j \leftarrow j + 1 \\ w_j \leftarrow v_i \end{array} \right. \\ w \end{array} \right.$$

$$v := \begin{pmatrix} 2 \\ -6 \\ -1 \\ 12 \\ 5 \end{pmatrix}$$

$$\text{form_V}(v) = \begin{pmatrix} 2 \\ 12 \\ 5 \end{pmatrix}$$

Оператор *return*

– преждевременное завершение ПФ и возвращение операнда, стоящего в поле 1

$$f(x) := \left| \begin{array}{l} y \leftarrow x^2 \\ z \leftarrow y + 1 \\ \text{return } z \end{array} \right. \qquad f(2) = 5$$

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

Составить описание П-Ф, находящую первую положительную проекцию исходного вектора.

$$pol_1(v) := \left| \begin{array}{l} i \leftarrow 1 \\ \text{while } v_i \le 0 \\ \quad i \leftarrow i + 1 \\ v_i \end{array} \right.$$

$$pol_2(v) := \text{for } i \in 1 .. last(v) \quad \text{return } v_i \text{ if } v_i > 0$$

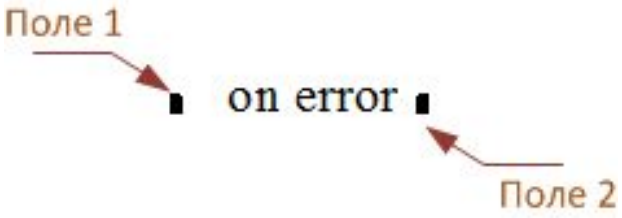
$$pol_1(v) = 2$$

$$pol_2(v) = 2$$

Оператор *on error*

– перехват ошибок.

< конструкция 1 > on error < конструкция 2 >



В поле 2 вносится выражение, ошибка в вычислении которого должна быть зарегистрирована.

В поле 1 вносится условие, которое необходимо выполнить при возникновении ошибки.

$$f(x) := \left| \begin{array}{l} z \leftarrow x \\ \text{"деление на ноль"} \quad \text{on error} \quad \frac{1}{z} \end{array} \right.$$

$$f(-2) = -0.5$$

$$f(0) = \text{"деление на ноль"}$$

Функция *error*

Используется для вывода диагностических сообщений при возникновении в вычислениях ошибки и записывается в виде

error ("< диагностическое сообщение пользователя >")

Сформировать вывод диагностического сообщения при попытке спроецировать вектор v на нулевой вектор w

$$\text{proj}(v, w) := \begin{cases} \text{error}(\text{"Проекция на нулевой вектор"}) & \text{if } |w| < 10^{-10} \\ \frac{w}{|w|} \cdot (v \cdot w) & \text{otherwise} \end{cases}$$

$$x := \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \qquad w := \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix} \qquad \text{proj}(x, w) = \begin{pmatrix} 16.036 \\ 10.69 \\ 5.345 \end{pmatrix}$$

$$x := \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} \qquad w := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$\text{proj}(x, w) =$ ▪▪

Проекция на нулевой вектор