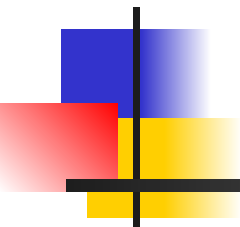


Тема 4

Основные алгоритмические структуры



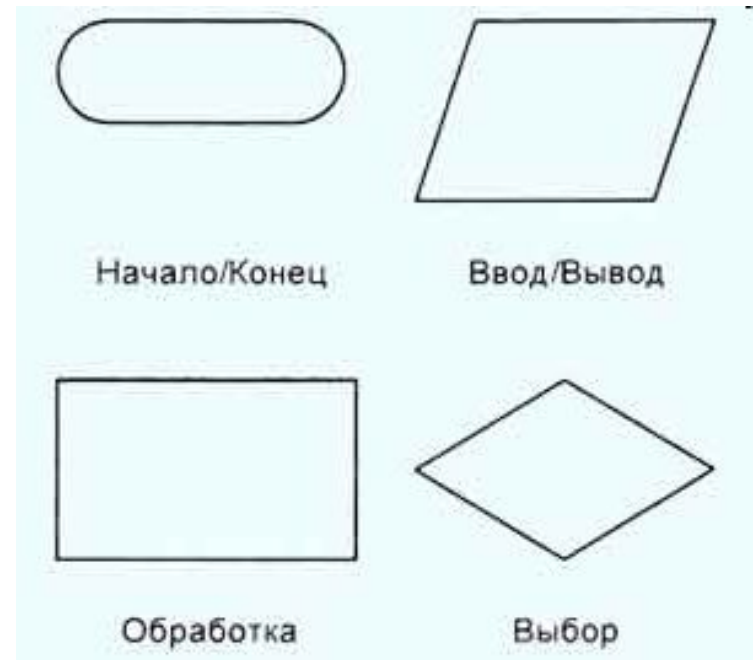
1. Понятие алгоритма
2. Структура ***Следование***
3. Структура ***Развилка***
4. Структура ***Цикл***

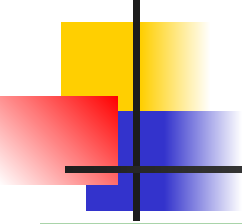
Технологическая цепочка решения задачи на ЭВМ

1. Постановка задачи.	Работа без применени я ЭВМ
2. Математическая формализация.	
3. Построение алгоритма.	
4. Составление программы на ЯП.	Работа на ЭВМ
5. Отладка и тестирование программы.	
6. Проведение расчетов и анализ результатов	

Алгоритм

Алгоритм - это точно определенная последовательность действий, которые необходимо выполнить над исходной информацией, чтобы получить решение задачи





Свойства алгоритма

Понятность - *указания, которые понятны исполнителю*

Дискретность - *возможность разбиения алгоритма на отдельные элементарные действия*

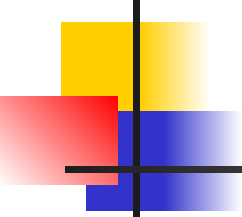
Массовость - *возможность применения алгоритмов для решения целого класса конкретных задач, отвечающих общей постановке задачи*

Конечность - *завершение работы алгоритма в целом за конечное число шагов*

Однозначность (определенность) - *единственность толкования правил выполнения действий и порядка их выполнения*

Результативность - *получение требуемого результата за конечное число шагов*

Правила построения алгоритма

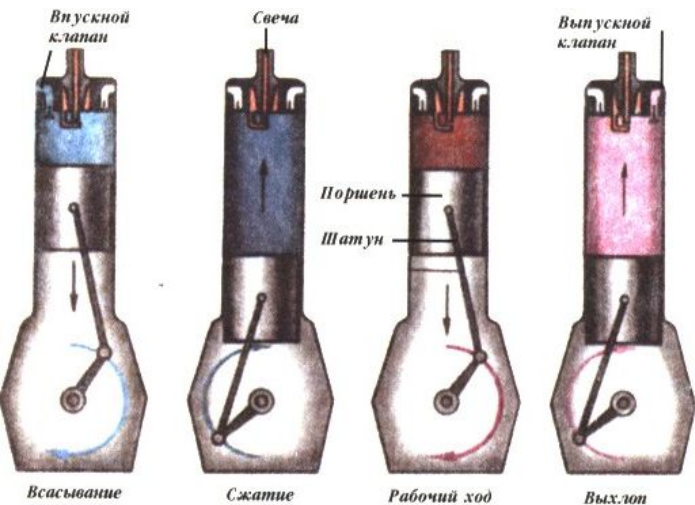
- 
1. При построении алгоритма прежде всего необходимо задать **множество объектов**, с которыми будет работать алгоритм. Алгоритм приступает к работе с некоторым набором данных, которые называются входными, и в результате своей работы выдает данные, которые называются выходными.
 2. Для работы алгоритма требуется **память**. В памяти размещаются входные данные, промежуточные данные и выходные данные. Поименованная ячейка памяти носит название переменной
 3. **Дискретность**. Алгоритм строится из отдельных шагов, множество которых конечно.
 4. **Детерменированность**. После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки.
 5. **Сходимость (результативность)**. Алгоритм должен завершать работу после конечного числа шагов и получить результат.

Виды алгоритмов как логико-математических средств

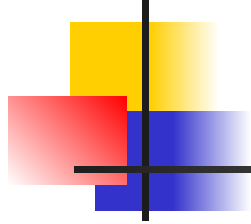
□ *Механические*

□ *Гибкие*

- *Вероятностный (стохастический)*
- *Эвристический*



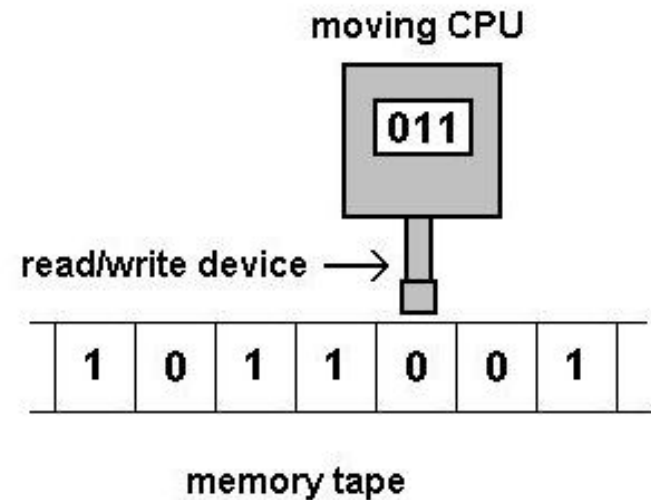
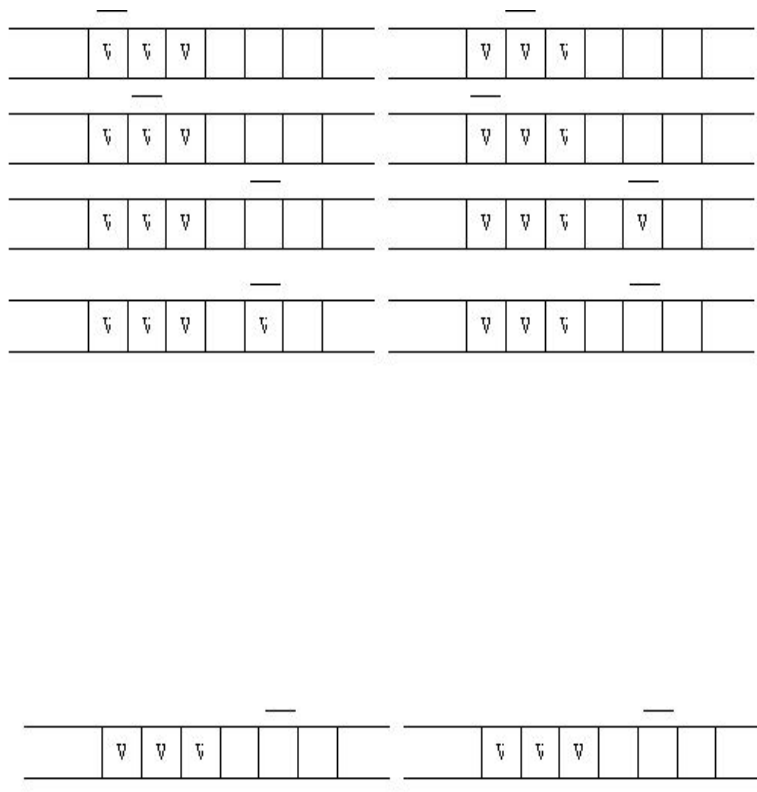
Способы записи алгоритмов



- *Словесный*
- *Формульный*
- *Табличный*
- *Графический* (с
помощью блок-схем)

Конечный автомат

Движение головки на одну клетку вправо $n \rightarrow m$.
Движение головки на одну клетку влево $n \leftarrow m$.
Нанесение метки на клетку, над которой находится головка $n M m$.
Стирание метки из клетки, над которой находится головка $n C m$.
Проверка наличия метки в клетке, над которой находится головка. Если метка отсутствует, то управление передается команде $m1$, а иначе $m2$. <pre> graph LR n --> m1 n --> m2 </pre>
Остановка машины n Стоп m





Виды вычислительных процессов

Последовательный

команды выполняются
последовательно одна за другой

Разветвляющийся

та или иная серия команд выполняется в зависимости от истинности **условия**.

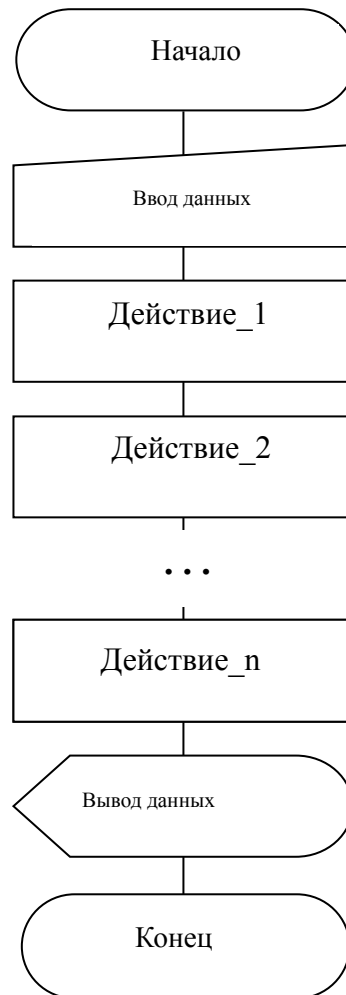
Пример:

ЕСЛИ хочешь быть здоров, ТО
закаляйся, ИНАЧЕ валяйся весь день на
диване.

Циклический

одна и та же последовательность действий
повторяется несколько раз до тех пор, пока
выполняется некоторое условие

Структура Следование



Структура Следование (пример)

Формализация задачи

Разработать информационную технологию, позволяющую вычислить значение площади круга и длины окружности при заданном радиусе.

Входные данные:

радиус круга – **R**, вещественная переменная.

Выходные данные:

длина окружности – **L**, вещественная переменная

площадь круга – **S**, вещественная переменная

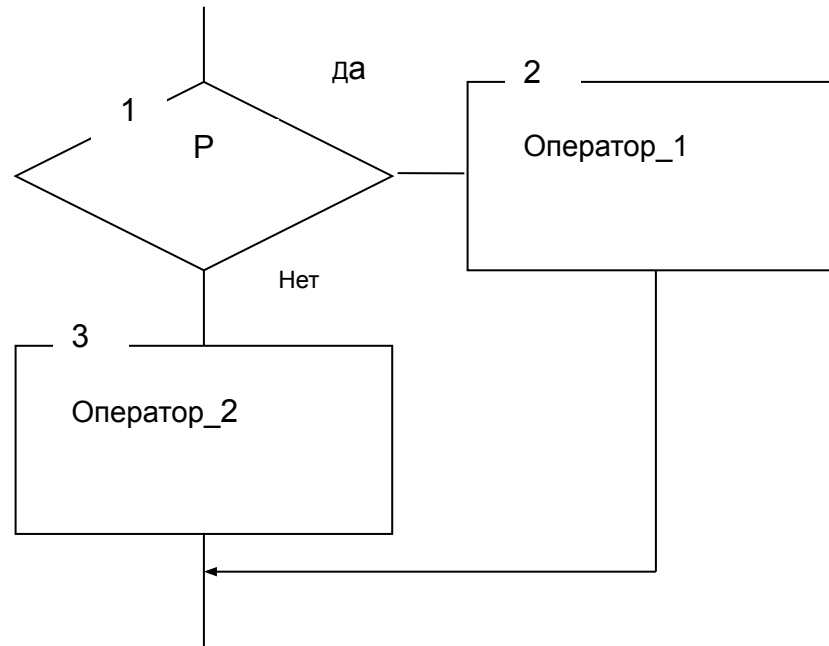
Математическое описание алгоритма

- Ввести в память компьютерной системы значение радиуса **R**,
- Вычислить значение длины окружности по формуле **$L = 2 * \pi * R$**
- Вычислить значение площади круга по формуле **$S = \pi * R^2$**
- Вывести на экран монитора значения **R, L и S**

Разработка схемы алгоритма



Классическая структура Развилка



Классическая структура Развилка (пример)

Функция, выполняемая информационной технологией, описывается следующими математическими и логическими зависимостями:

$$y = \begin{cases} \sin(x), & \text{если } x > 0 \\ \cos(x), & \text{если } x \leq 0 \end{cases}$$

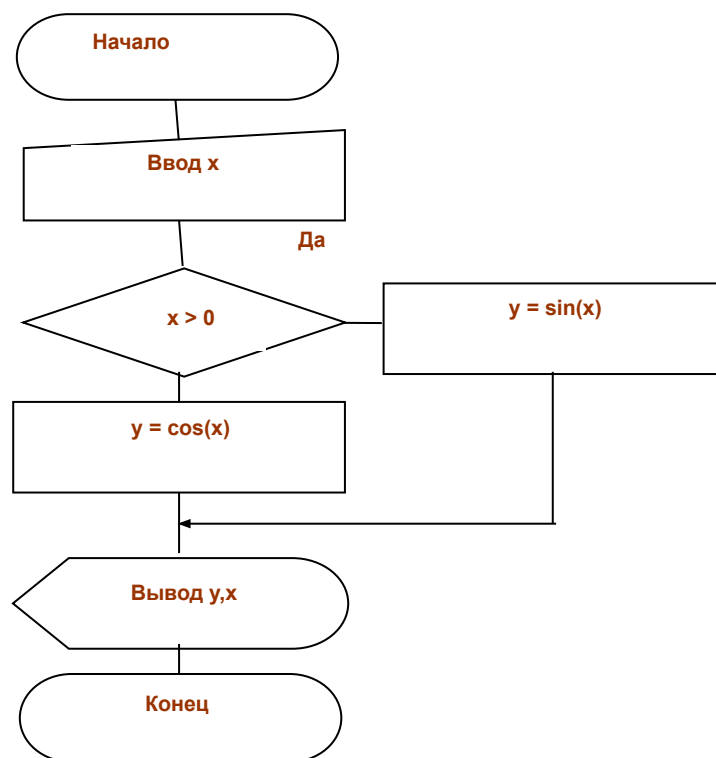
Входные данные: **x** - переменная вещественного типа.

Выходные данные: **y** – переменная вещественного типа.

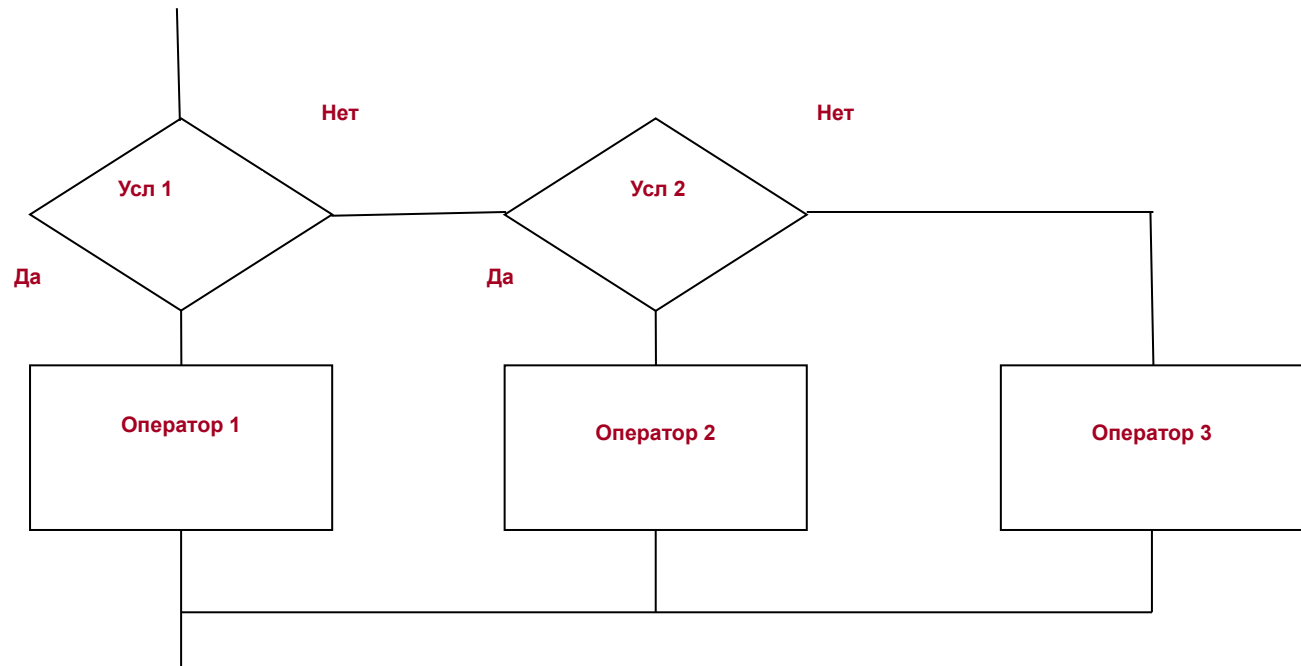
Математическое описание алгоритма

1. Ввести в память компьютера значение переменной **x**.
2. Если условие **x > 0** истинно, то вычислить значение **y = sin(x)**, если условие ложно, то вычислить значение **y = cos(x)**.
3. Вывести на экран монитора значения **x** и **y**.

Схема алгоритма:



Вложенная структура Развилка



Вложенная структура Развилка (пример)

Функция, выполняемая ИТ, описывается следующими математическими и логическими зависимостями:

$$y = \begin{cases} \text{ctg } x, & \text{если } x > 10 \\ 10, & \text{если } x = 10 \\ q * \cos x^3 & \text{если } x < 10 \end{cases}$$

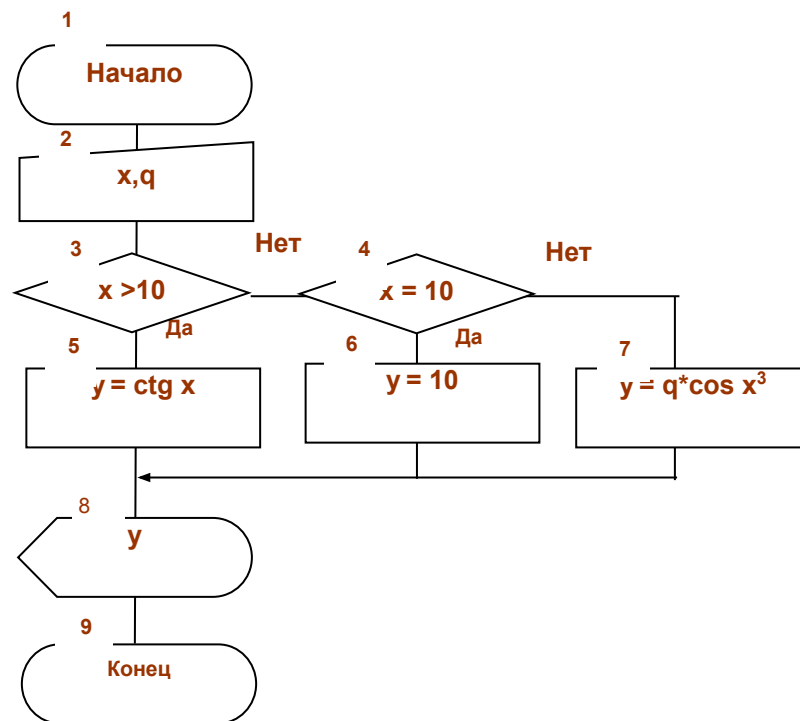
Входные данные: **x, q** – переменные вещественного типа.

Выходные данные: **y** – переменная вещественного типа.

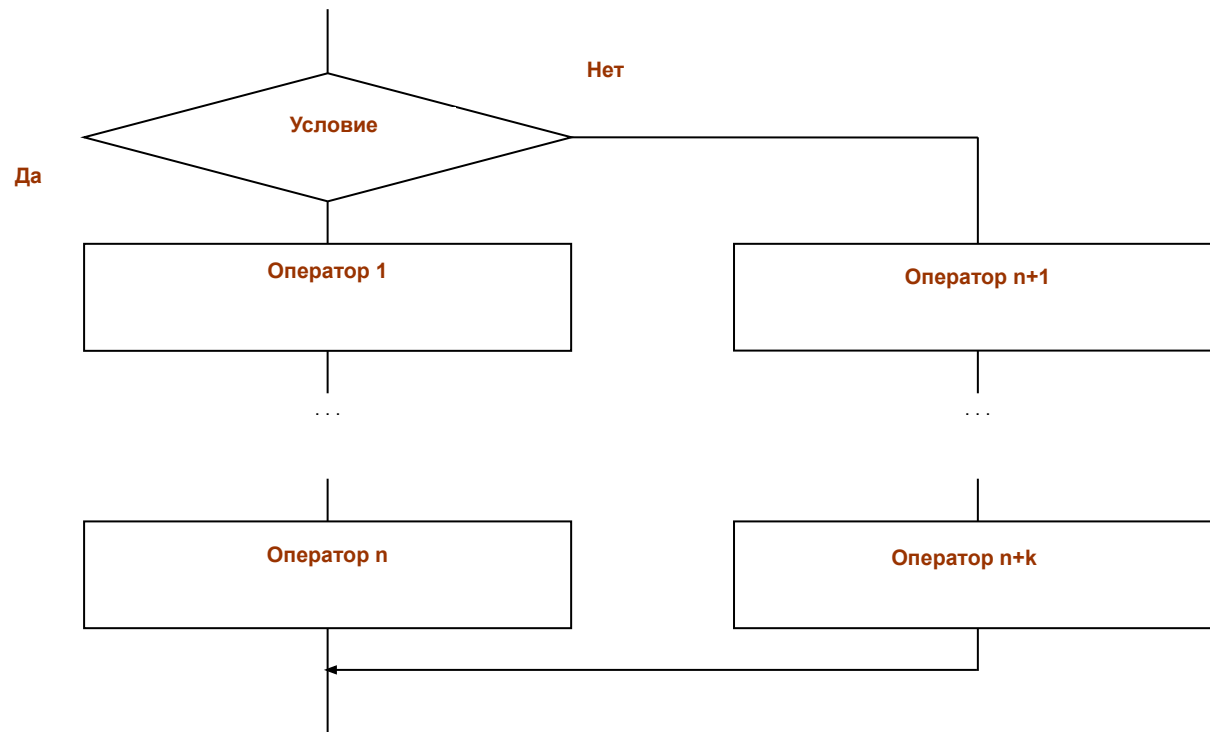
Математическое описание алгоритма

1. Ввести значения переменных **x** и **q**.
2. Если условие **x > 0** истинно, то вычислить значение **y = ctg(x)**,
3. если условие ложно, то проверить условие **x = 0**, если условие истинно, то присвоить **y** значение, равное **10**, если условие ложно, то вычислить значение **y = q*cos(x³)**.
4. Вывести на экран монитора значение **y**.

Схема алгоритма



Модифицированная структура Развилка



Модифицированная структура Развилка (пример)

Функция, выполняемая информационной технологией, описывается следующими математическими и логическими зависимостями:

$y1 = \operatorname{tg} x2$; ввести значение $y2$, если $x \leq 0$

$y1 = \cos x + \sin^2 x$; $y2 = 0$, если $x > 0$

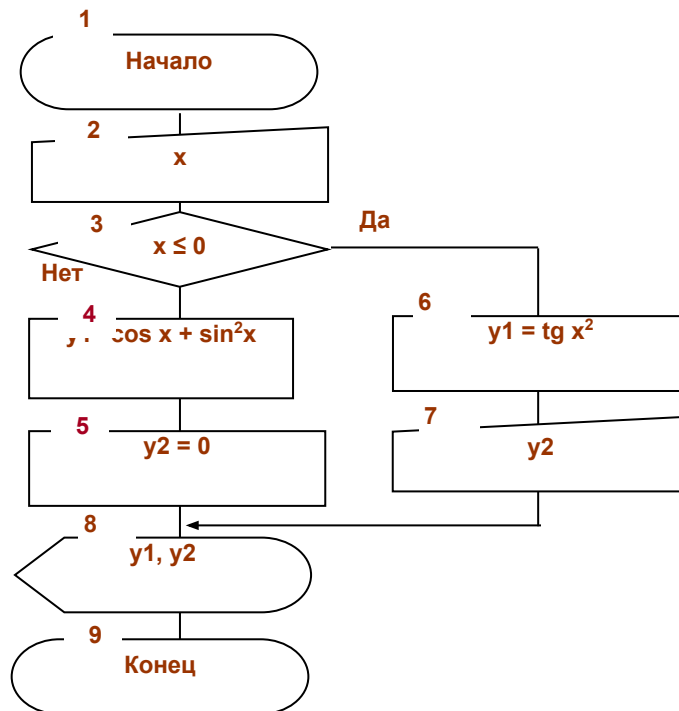
Входные данные: x – переменная вещественного типа.

Выходные данные: $y1$, $y2$ – переменные вещественного типа.

Математическое описание алгоритма

1. Ввести значение переменной x .
 2. Если условие $x \leq 0$ истинно, то вычислить значение $y1 = \operatorname{Tg}(x2)$ и ввести значение $y2$, если условие ложно, то вычислить значение $y1 = \cos(x)+\sin(x)2$ и $y = 0$.
- Вывести на экран монитора значения x , $y1$, $y2$.

Схема алгоритма

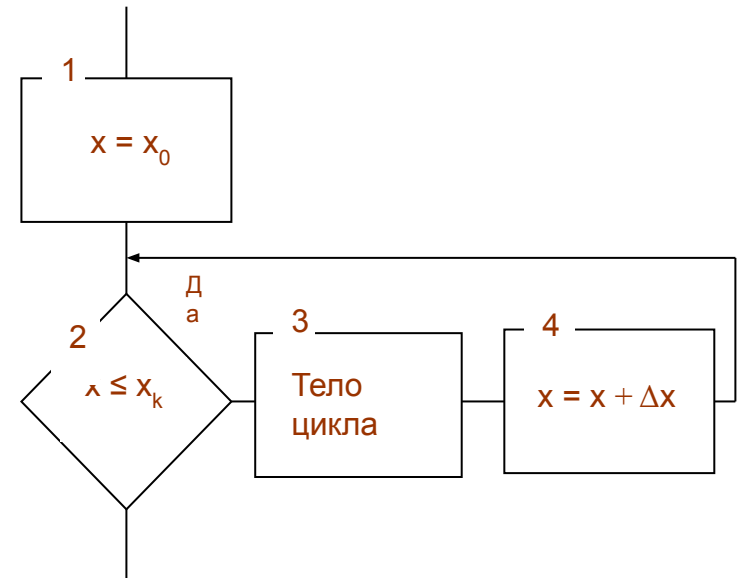


Структура Цикл (с параметром)

В циклических процессах с параметром выполнение циклического процесса заканчивается, когда условие его окончания становится истинным.

При описании циклического процесса с параметром используются следующие понятия:

- параметр цикла (обозначим его x);
- начальное значение параметра цикла (обозначим его x_0);
- конечное значение параметра цикла (обозначим его x_k);
- шаг изменения параметра цикла (обозначим его Δx).
- условие окончания цикла,
- тело цикла.



Структура Цикл с параметром (пример)

Разработать информационную технологию, позволяющую вычислить значение функции $y = \sin(x)$, при изменении значения параметра цикла x от начального значения a до конечного значения b с шагом $= \pi/6$.

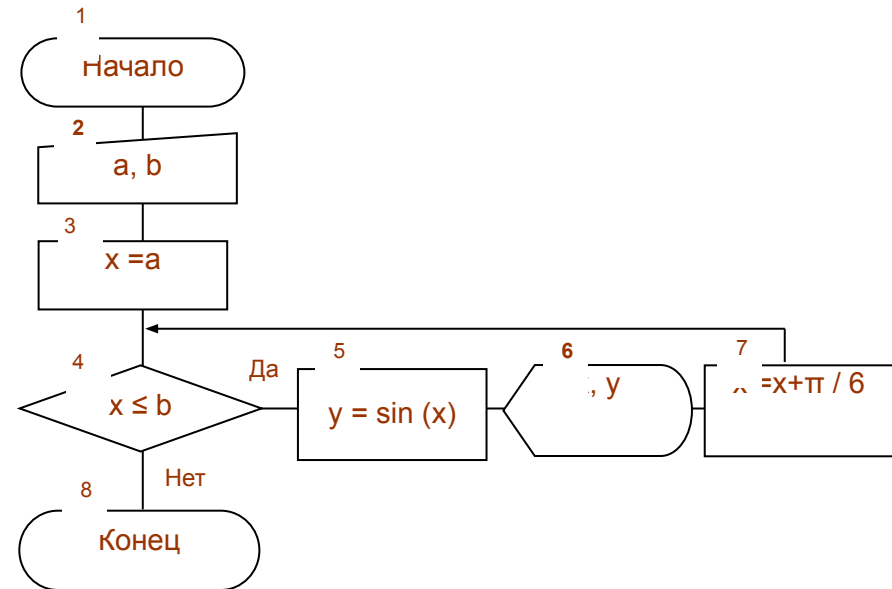
Входные данные: a, b – переменные вещественного типа.

Выходные данные: y – переменная вещественного типа.

Математическое описание алгоритма

1. Ввести начальное и конечное значения параметра цикла (a, b)
2. Присвоить параметру цикла начальное значение ($x = a$).
3. Проверить условие окончания цикла $x \leq b$. Если условие окончания цикла истинно, то выполнить тело цикла:
 - 1) **вычислить** значение функции $y = \sin(x)$
 - 2) **вывести** на экран монитора полученное значение функции y и значение параметра цикла x .
4. Увеличить значение параметра цикла на величину шага $\Delta x = \pi/6$ и перейти к проверке условия окончания цикла. Если условие окончания цикла **истинно**, то вновь выполнить тело цикла, если условие окончания цикла **ложно**, то закончить вычислительный процесс.

Схема алгоритма

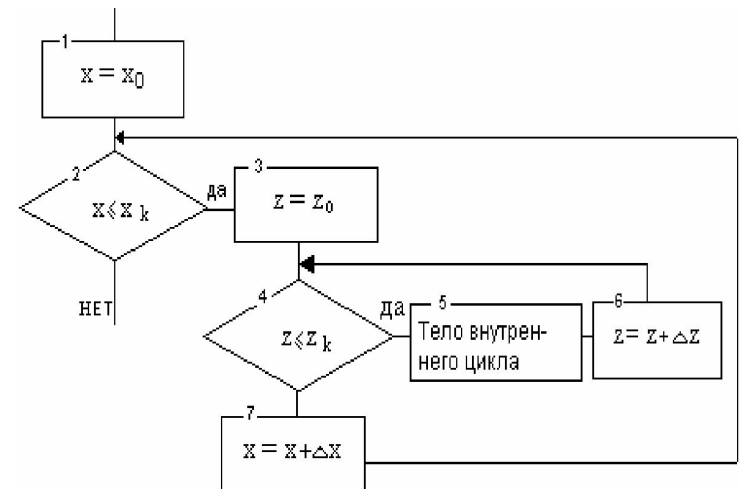


Структура Цикл в цикле

Циклический процесс называется **цикл в цикле**, если использовано два параметра цикла

Используются следующие понятия:

- параметр внешнего цикла (x) и параметр внутреннего цикла (z),
- начальное значение параметра внешнего цикла (x_0)
- начальное значение параметра внутреннего цикла (z_0)
- конечное значение параметра внешнего цикла (x_k)
- конечное значение параметра внутреннего цикла (z_k)
- шаг изменения параметра внешнего цикла (Δx)
- шаг изменения параметра внутреннего цикла (Δz),
- условие окончания внешнего цикла,
- условие окончания внутреннего цикла,
- тело внешнего цикла,
- тело внутреннего цикла.



Структура Цикл в цикле

Пусть требуется вычислить значение функции

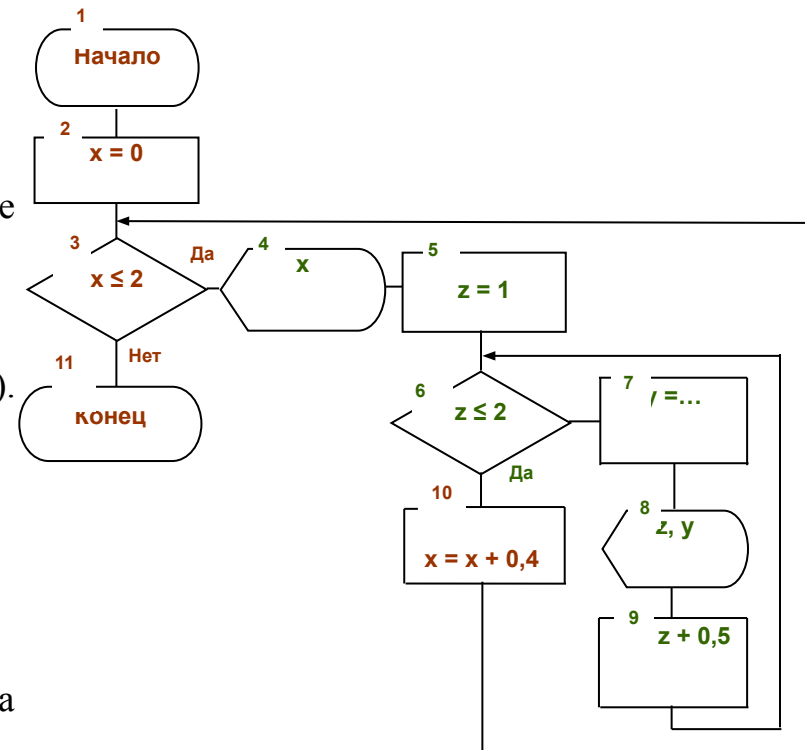
$y = \sin(x) + \cos(z)$ при изменении $0 \leq x \leq 2$, с шагом $\Delta x = 0,4$, $1 \leq z \leq 2$ с шагом $\Delta z = 0,5$.

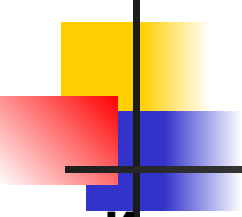
Входные данные: данных, подлежащих вводу, нет

Выходные данные: x, z, y – переменные вещественного типа

Математическое описание алгоритма

- Присвоить параметру **внешнего** цикла начальное значение ($x = 0$).
- Проверить условие окончания **внешнего** цикла ($x \leq 2$). Если условие окончания внешнего цикла **истинно**, то выполнить **тело внешнего** цикла:
 - вывести на экран монитора значение x ,
 - присвоить параметру **внутреннего** цикла начальное значение ($z = 1$).
 - проверить условие окончания **внутреннего** цикла ($z \leq 2$). Если условие окончания **внутреннего** цикла **истинно**, то выполнить **тело внутреннего** цикла:
 - вычислить значение функции $y = \sin(x) + \cos(z)$
 - вывести на экран монитора полученное значение функции y и значение параметра **внутреннего** цикла z .
 - увеличить значение параметра **внутреннего** цикла на величину шага $\Delta z = 0,5$ и проверить условие окончания **внутреннего** цикла.
- Если условие окончания **внутреннего** цикла **ложно**, то увеличить значение параметр **внешнего** цикла на величину шага $\Delta x = 0,4$.
- Проверить **условие окончания внешнего** цикла. Если условие окончания **внешнего** цикла **ложно**, то закончить вычислительный процесс.





Структура Итерационный Цикл

Итерация – результат повторного применения какой-либо математической операции.

Так, если $y = f(x) \equiv f_1(x)$ есть некоторая функция от x , то функции

$$f_2(x) = f[f_1(x)], f_3(x) = f[f_2(x)], f_4(x) = f[f_3(x)], \dots, f_n(x) = f[f_{n-1}(x)]$$

называются, соответственно, второй, третьей, ..., n -й итерациями функции $f(x)$.

Использование понятия итерации позволяет вычислять значения функций с заданной точностью.

Для этого вычисляются абсолютные значения разности между двумя итерациями, которые сравниваются с заданной точностью ξ :

$$|f_1(x) - f_2(x)| \geq \xi$$

$$|f_2(x) - f_3(x)| \geq \xi$$

$$|f_3(x) - f_4(x)| \geq \xi$$

и так далее, пока абсолютное значение разности не будет превосходить значение заданной точности вычисления ξ .

Следовательно, **особенность итерационных процессов заключается в том, что они являются циклическими и вычисления заканчиваются при достижении заданной точности.**

Структура Итерационный Цикл

Разработать информационную технологию, позволяющую вычислять корень уравнения $x - \cos(x) = 0$ с заданной точностью, и при начальном приближении к корню $= 0,6$.

Входные данные:

начальное приближение к корню x_0 ,
переменная вещественного типа,
заданная точность вычисления ξ , переменная вещественного типа.

Выходные данные: приближённое значение корня уравнения x .

Математическое описание алгоритма:

Преобразуем исходное уравнение к виду: $x = \cos(x_0)$.

Вычислим **первое приближение** к корню: $x = \cos(0,6) = 0,825336$ ($0,82$).

Вычислим **разность** $|x_0 - x|$ и сравним полученную разность с заданной точностью ξ , $|0,6 - 0,825336| = 0,225336 \leq 0,01$.

Присвоим x значение, равное $0,82$, и вычислим **второе приближение** к корню $x = \cos(0,82) = 0,682221$ ($0,68$). Вычислим разность $|x_0 - x|$ и сравним полученную разность с заданной точностью ξ , $|0,82 - 0,68| = 0,137779 \leq 0,01$. Процесс повторяется, пока не будет достигнута заданная точность.

	A	B	C
1	x	Cos(x)	Точность
2	0,6	0,825336	0,225336
3	0,82	0,682221	0,137779
4	0,68	0,777573	0,097573
5	0,77	0,717911	0,052089
6	0,71	0,758362	0,048362
7	0,75	0,731689	0,018311
8	0,73	0,745174	0,015174
9	0,74	0,738469	0,001531

Схема алгоритма

