

Севастопольский государственный университет
Кафедра информационных систем

Курс лекций по дисциплине
«Алгоритмизация и программирование»

Лектор: Бондарев Владимир Николаевич

Лекция 4

Разветвляющиеся структуры алгоритмов. Циклические структуры алгоритмов.

Разветвляющиеся структуры алгоритмов

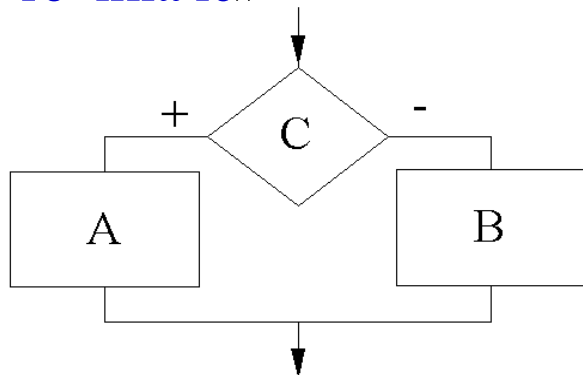
Алгоритмическая структура, которая позволяет исполнителю алгоритма, выбрать сценарий последующих действий в зависимости от истинности определенного утверждения, называется **ветвлением**.

Существуют две основные разновидности структур ветвления:

альтернативное ветвление — выбор одной из 2-х ветвей алгоритма;

множественное ветвление — выбор одной из N ветвей алгоритма.

Альтернативное ветвление определяется в виде управляющей конструкции «**если–то–иначе**»



если условие C **то**

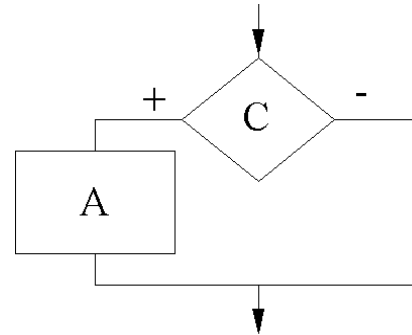
действие A

иначе действие B

Разветвляющиеся структуры алгоритмов

Альтернатива может быть не полной :

если условие C **то**
действие A



Множественное ветвление определяется в виде управляющей конструкции
выбор :

выбор

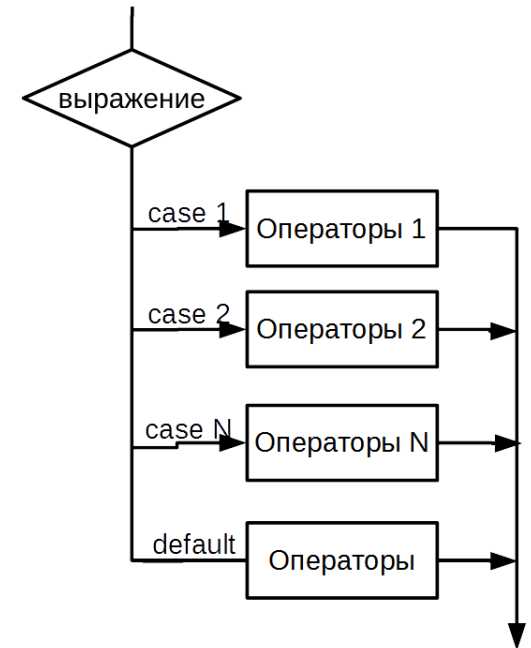
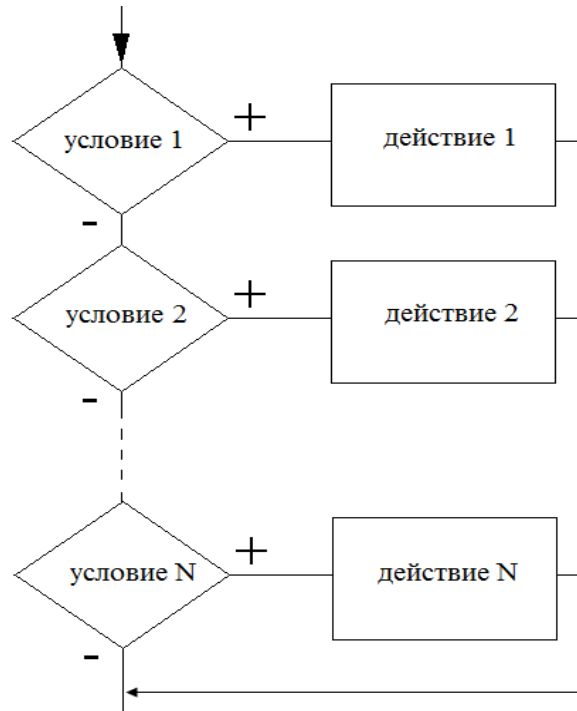
условие 1: действие 1;

условие 2: действие 2;

...

условие N: действие N;

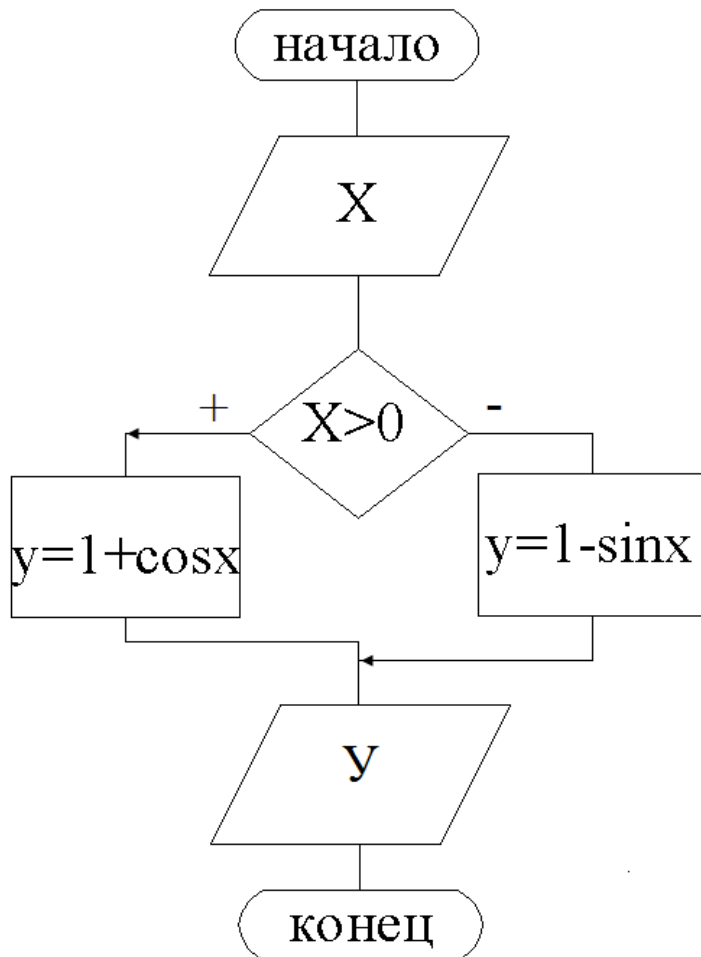
конец



Разветвляющиеся структуры алгоритмов

Пример 1. Пусть требуется вычислить значение выражения:

$$y = \begin{cases} 1 + \cos x, & x > 0 \\ 1 - \sin x, & x \leq 0 \end{cases}$$



ВВОД (x);

если $x > 0$ **то**
 $y := 1 + \cos(x)$

иначе

$y := 1 - \sin(x);$

ВЫВОД(y);

Разветвляющиеся структуры алгоритмов

Пример 2. Составим алгоритм решения квадратного уравнения

$$ax^2 + bx + c = 0$$

Решением в общем случае будут два корня x_1 и x_2 , которые вычисляются по формуле:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Алгоритм 1. Решение квадратного уравнения

ВВОД(a, b, c);

d := $b^2 - 4ac$;

x1 := $(-b + \sqrt{d}) / (2a)$;

x2 := $(-b - \sqrt{d}) / (2a)$;

ВЫВОД(x1, x2);

Слабость такого алгоритма видна невооруженным глазом. Он не обладает важнейшим свойством, предъявляемым к качественным алгоритмам, — **результативностью**.

Алгоритм 2. Решение квадратного уравнения

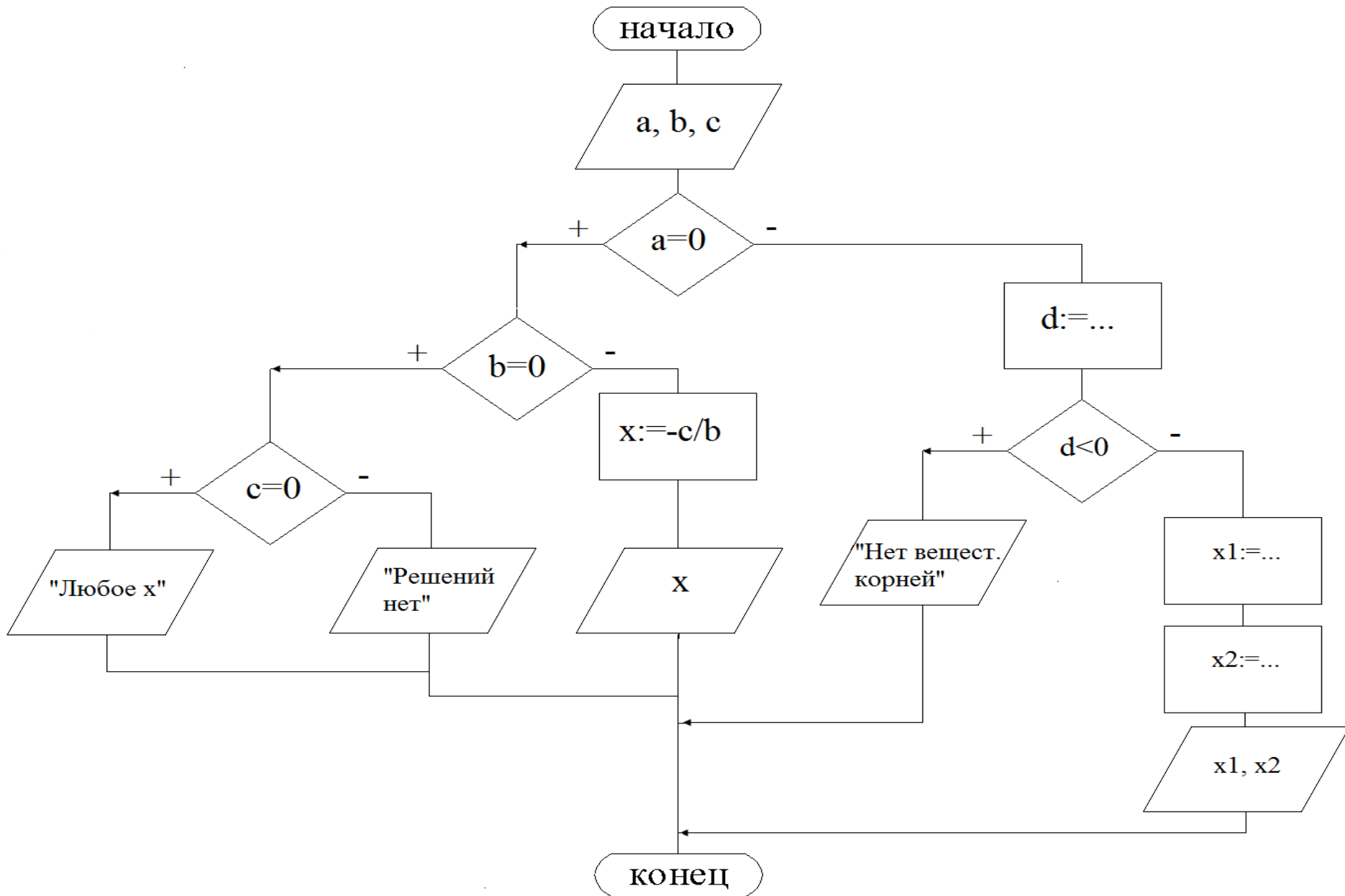
Чтобы построить универсальный алгоритм, сначала требуется тщательно проанализировать математическое содержание задачи.

Решение уравнения зависит от значений коэффициентов a , b , c .

Ограничимся только поиском вещественных корней:

- 1) если $a = 0$, $b = 0$, $c = 0$, то любое x — решение уравнения;
- 2) если $a = 0$, $b = 0$, $c \neq 0$, то уравнение решений не имеет;
- 3) если $a = 0$, $b \neq 0$, то уравнение имеет одно решение $x = -c/b$;
- 4) если $a \neq 0$ и $d \geq 0$, то уравнение имеет два вещественных корня (формулы приведены выше);
- 5) если $a \neq 0$ и $d < 0$, то уравнение не имеет вещественных корней.

Алгоритм решения квадратного уравнения



Алгоритм решения квадратного уравнения

ВВОД(a,b,c);

если $a=0$ **то**

если $b=0$ **то**

если $c=0$ **то**

ВЫВОД(«Любое x – решение»)

иначе

ВЫВОД(«Решений нет»)

иначе **начало**

$x := -c/b$;

ВЫВОД(x)

конец

иначе **начало**

$d := \dots$;

если $d < 0$ **то**

ВЫВОД(«Действительный корней нет»)

иначе **начало**

$x_1 := \dots$;

$x_2 := \dots$;

ВЫВОД(x_1, x_2);

конец

конец

Множественное ветвление (выбор)

Пример 3. Вывести названия дней недели в зависимости от значения переменной x . Если $x=1$, то вывести «Понедельник», если $x=2$ – вторник и т.д.

выбор

$x=1$: **вывод**(«Понедельник»);

$x=2$: **вывод**(«Вторник»);

...

$x=7$: **вывод**(«Воскресенье»);

конец

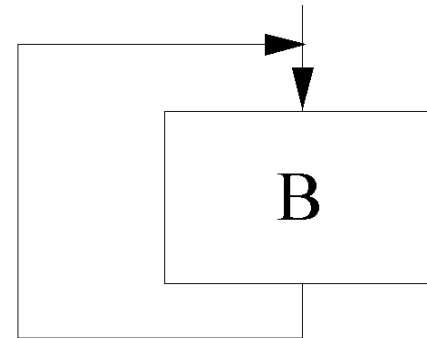
Циклические структуры алгоритмов

Алгоритмическая структура, которая обеспечивает многократное выполнение некоторой совокупности действий, называется **циклом**. Многократно повторяемые действия образуют **тело цикла**.

Существует следующие разновидности циклов:

- 1) бесконечный цикл;
- 2) цикл, управляемый условием «пока» (цикл с предусловием);
- 3) цикл, управляемый условием «до» (цикл с постусловием);
- 4) цикл с параметром

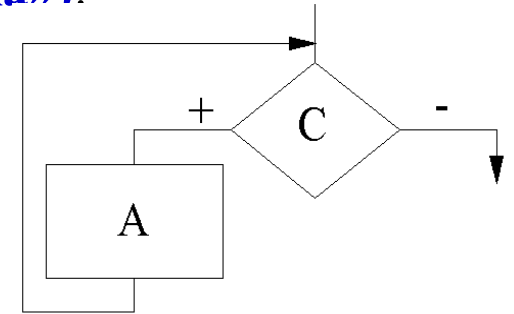
1. Бесконечный цикл
повторять бесконечно
действие В;



Циклические структуры алгоритмов

2. Цикл с предусловием (управляемый условием «пока»).

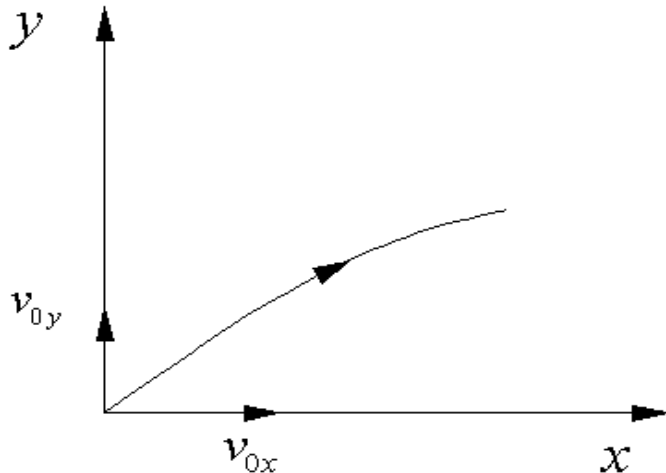
пока условие С повторять
действие А;



Особенности цикла :

- 1) когда условие С становится ложным происходит выход из цикла;
- 2) если условие ложно изначально, то действие А не выполняется ни разу;
- 3) действие А должно изменять условие С, иначе цикл будет бесконечным.

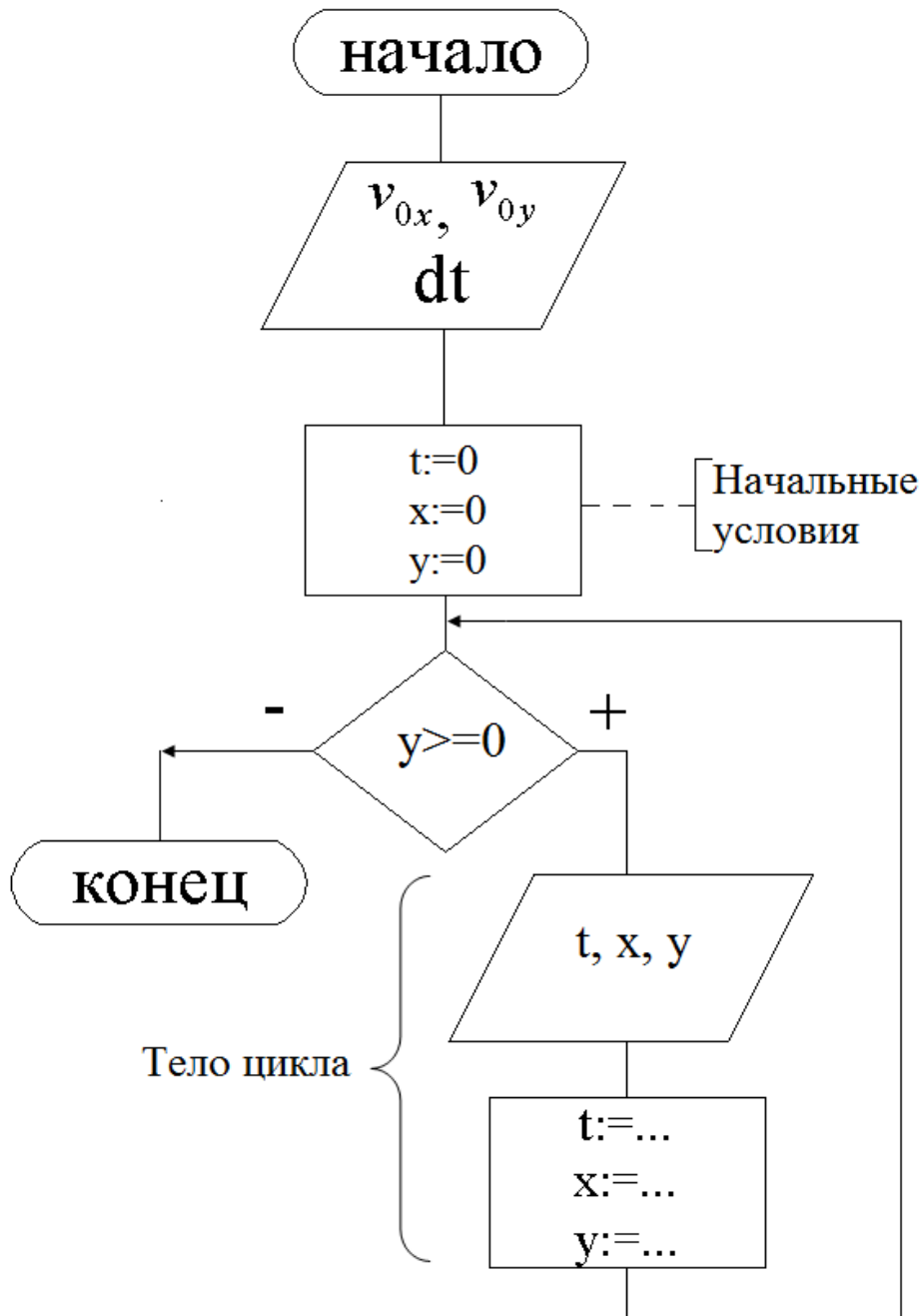
Пример 1. Пусть требуется определить координаты снаряда, выпущенного под углом к горизонту.



$$x = v_{0x} \cdot t$$

$$y = v_{0y} \cdot t - \frac{gt^2}{2}$$

Циклические структуры алгоритмов



ВВОД (V_{0x}, V_{0y}, dt);

t:= 0;

x:=0;

y:=0;

пока y >= 0 **повторять**
начало

ВЫВОД(t, x, y);

t := t + dt;

$x := v_{0x} \cdot t;$

$y := v_{0y} \cdot t - \frac{gt \cdot t}{2};$

конец;

Циклы, для которых число повторений тела цикла заранее неизвестно, называют **итерационными**.

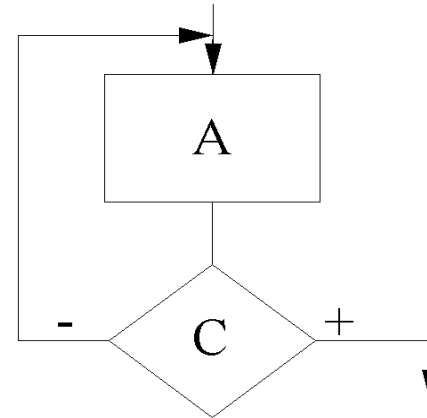
Циклические структуры алгоритмов

3. Цикл с постусловием (управляемый условием «до»).

повторять

действие А

до условие С;



Особенности цикла :

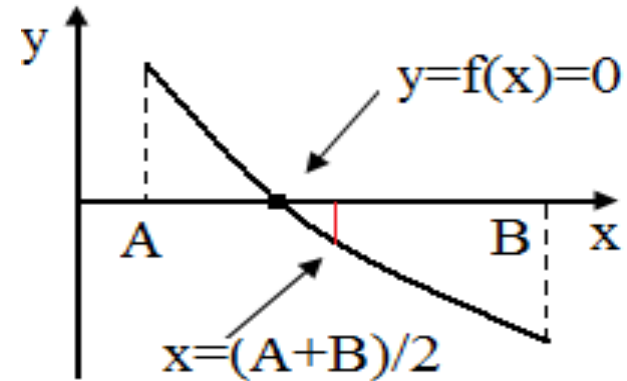
- 1) когда условие С становится истинным происходит выход из цикла;
- 2) действие А выполняется хотя бы один раз не зависимо от условия;
- 3) действие А должно изменять условие С, иначе цикл будет бесконечным.

Циклические структуры алгоритмов

Пример. Найти корень уравнения $f(x)=0$,
где $f(x)$ – монотонно убывающая функция на отрезке $[A,B]$.

Решим задачу **методом деления отрезков пополам**.

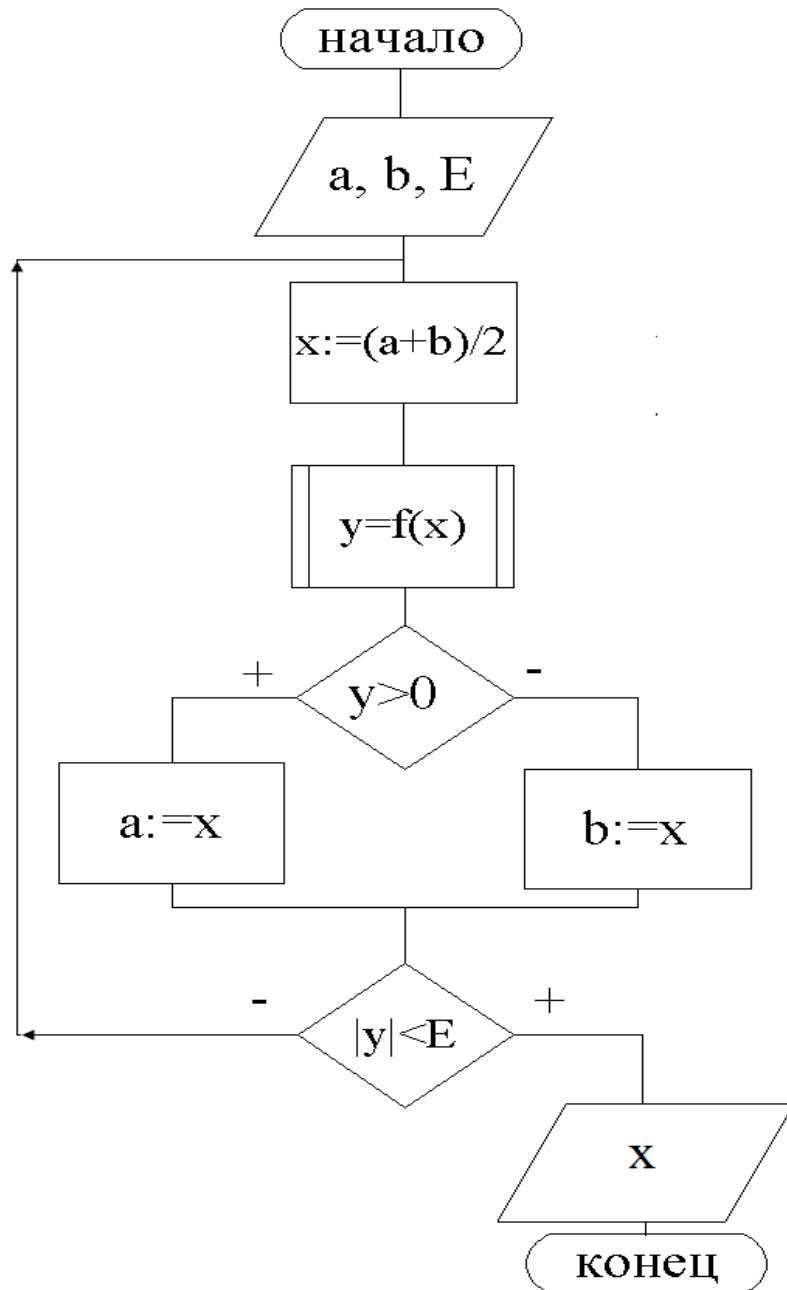
1. $x:=(A+B)/2$;
2. $y:=f(x)$;
3. Если $y>0$ то $A:=x$;
4. Если $y<0$ то $B:=x$.



Действия выполняются многократно, пока не будет достигнута заданная точность решения E , т.е.

$$y \pm E \approx 0$$

Алгоритм поиска корня уравнения



ВВОД (A, B, E);

ПОВТОРЯТЬ

$x := (A + B)/2;$

$y := f(x);$

ЕСЛИ $y > 0$ **ТО**

$A := x$

ИНАЧЕ

$B := x;$

ДО $|y| < E$

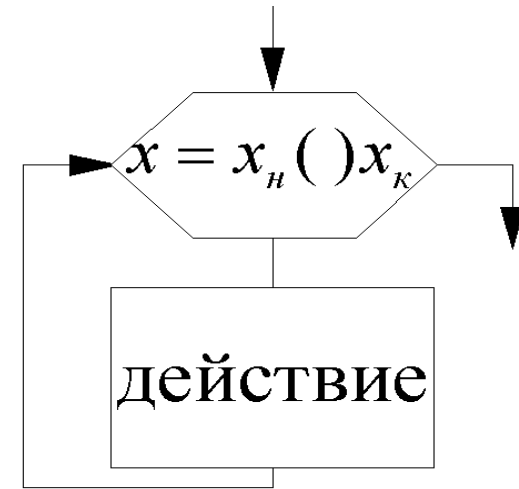
ВЫВОД (x);

Циклические структуры алгоритмов

4. Цикл с параметром.

для $x \in X$ повторять
действие A;

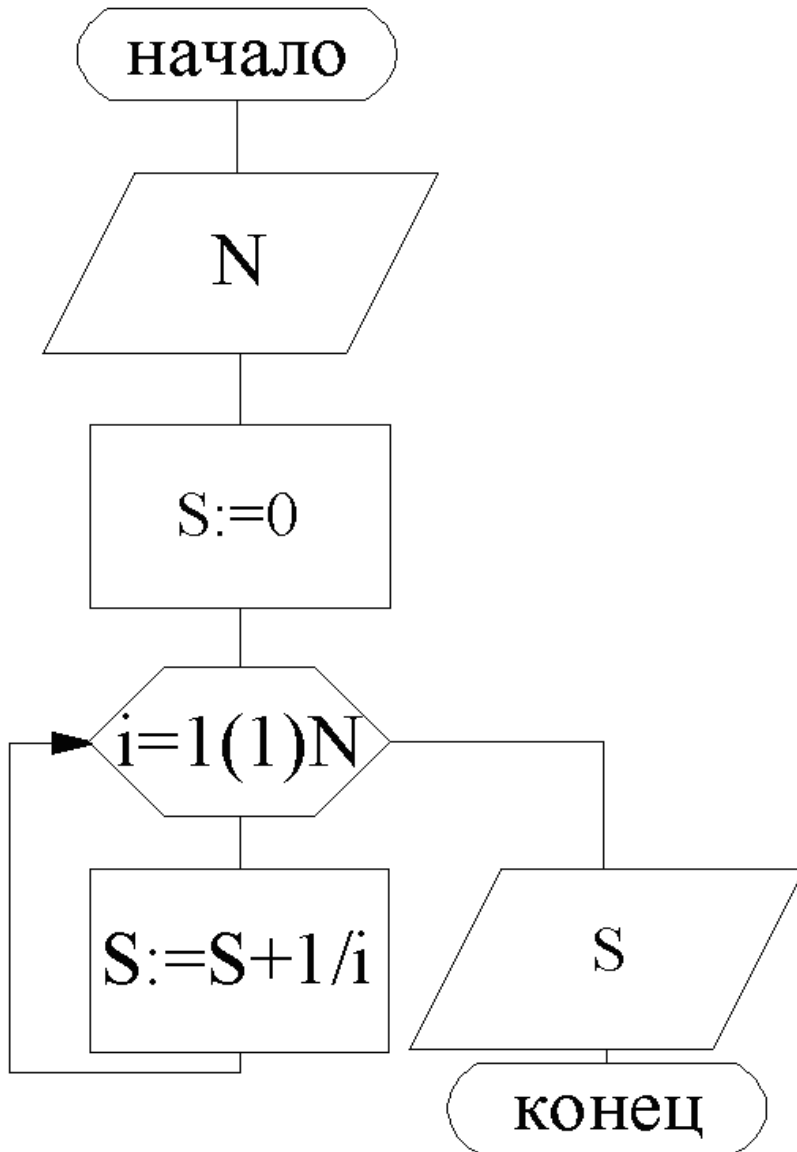
для $x := x_{нач}$ до $x_{кон}$ повторять
действие A;



Пример. Вычислим сумму элементов гармонического ряда:

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{N}$$

Алгоритм вычисления суммы ряда



ВВОД (N);

S := 0;

для i := 1 **до** N **повторять**
S := S + 1/i;

ВЫВОД(S);

Пошаговое выполнение

S:=0;

1. i:=1;

S:=S+1/i=0+1=1.

2. i:=2;

S:=S+1/i=1+1/2;

3. i:=3;

S:=S+1/i=1+1/2+1/3;

4. ...