



Этапы решения задачи на ЭВМ

Этапы решения задачи на ЭВМ

1. Постановка задачи:

- сбор информации о задаче;
- формулировка условия задачи;
- определение конечных целей решения задачи;
- определение формы выдачи результатов;
- описание данных (их типов, диапазонов величин, структуры и т. п.).

2. Анализ и исследование задачи, модели:

- анализ существующих аналогов;
- анализ технических и программных средств;
- разработка математической модели;
- разработка структур данных.

3. Разработка алгоритма:

- выбор метода проектирования алгоритма;
- выбор формы записи алгоритма (блок-схемы, псевдокод и др.);
- выбор тестов и метода тестирования;
- проектирование алгоритма.

4. Программирование:


- выбор языка программирования;
- уточнение способов организации данных;
- запись алгоритма на выбранном языке программирования.

5. Тестирование и отладка:

- синтаксическая отладка;
- отладка семантики и логической структуры;
- тестовые расчеты и анализ результатов тестирования;
- совершенствование программы.

6. Анализ результатов решения задачи и уточнение в случае необходимости математической модели с повторным выполнением этапов 2-5.





Основы программирования. Алгоритм. Способы записи алгоритма. Виды алгоритмов.

Основные определения

Математическая модель — это упрощенное описание реальности с помощью математических понятий.

Программирование — процесс и искусство создания компьютерных программ и/или программного обеспечения с помощью языков программирования.

Алгоритм — это точный набор инструкций, описывающих последовательность действий исполнителя для достижения результата решения задачи за конечное время.

Способы записи алгоритмов

1. Словесный

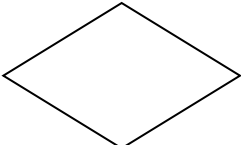
Например, приготовление любого блюда в кулинарной книге записано именно этим способом.


2. Блок-схема

Это способ записи алгоритмов при помощи специальных символов.

Обозначения блок-схемы

 – действия (команды);

 – условие;

 – начало/конец;

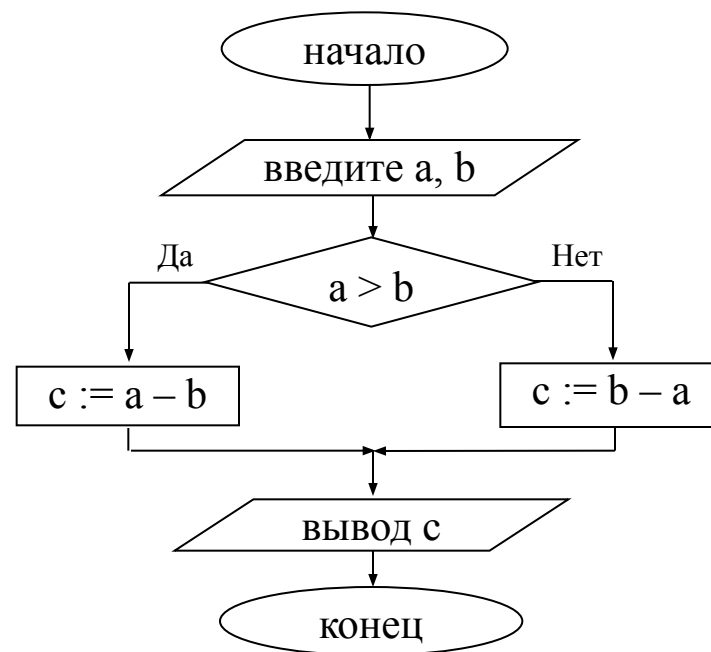
 – ввод/вывод данных.

На самом деле обозначений в блок-схемах гораздо больше. Но для наших целей достаточно этих четырёх.

3. Программа, написанная на языке программирования

Языков программирования существует очень много. Однако все их объединяют общие принципы. Отличаются они только синтаксисом, то есть названием тех или иных команд и способом их записи.

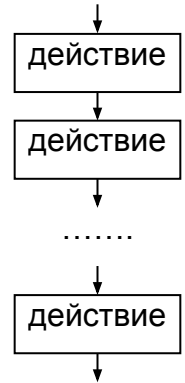
Пример



Виды алгоритмов:

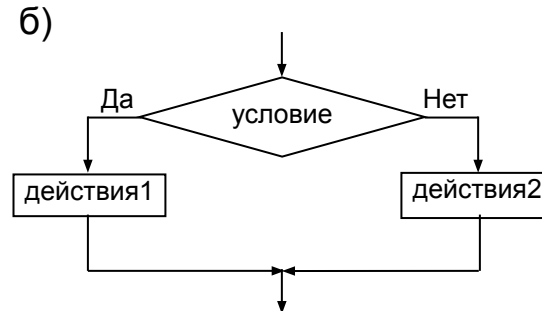
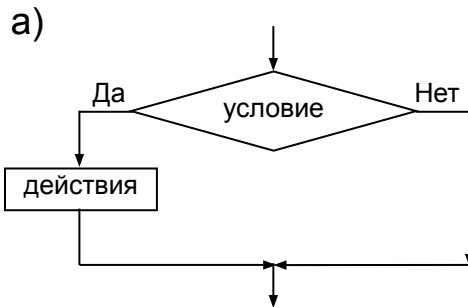
1. Линейный

В линейном алгоритме действия (команды) следуют подряд друг за другом.



2. Разветвляющийся

Это алгоритм, в котором есть условие. Выделяют два типа разветвляющихся алгоритмов.



В качестве примера можно привести следующий.

Я лежу на диване. За окном идёт дождь.

а) Если дождь прекратится, то я пойду гулять.

Здесь никаких действий в случае невыполнения условия не происходит!

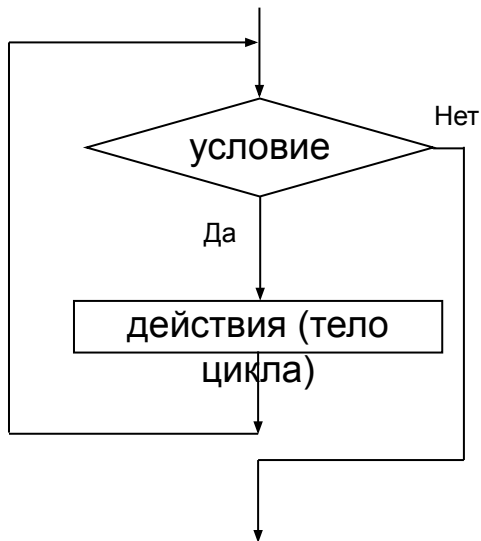
б) Если дождь прекратится, то я пойду гулять, иначе – буду смотреть телевизор.

3. Циклический

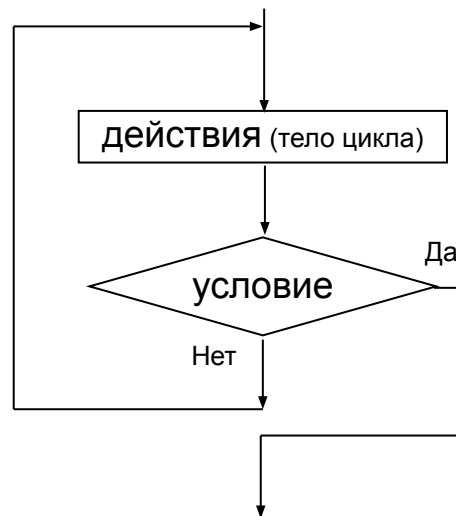
Это алгоритм, в котором есть повторяющиеся действия.

Во многих языках программирования есть три циклические конструкции (оператора):

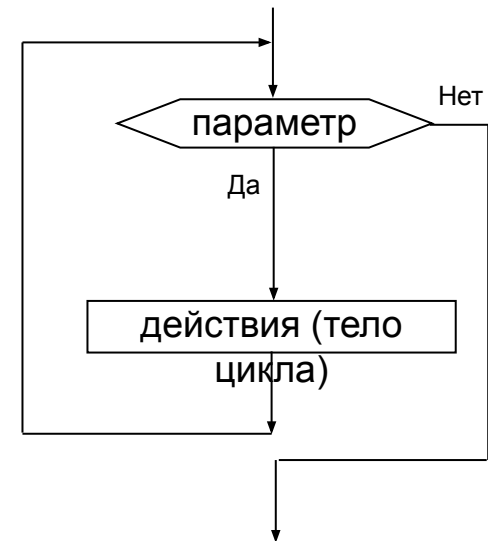
а) цикл с предусловием



б) цикл с постусловием



в) цикл с параметром



Отметим различия между первыми двумя типами циклических алгоритмов:

- 1) У цикла а) условие находится в начале, у б) – в конце;
- 2) У цикла а) условие продолжения выполнения цикла должно быть истинно, у цикла б) – ложно;
- 3) Тело цикла а) может ни разу не выполниться (если условие сразу окажется ложным), тело цикла б) выполнится обязательно по крайней мере один раз;

Пример

1. Постановка задачи

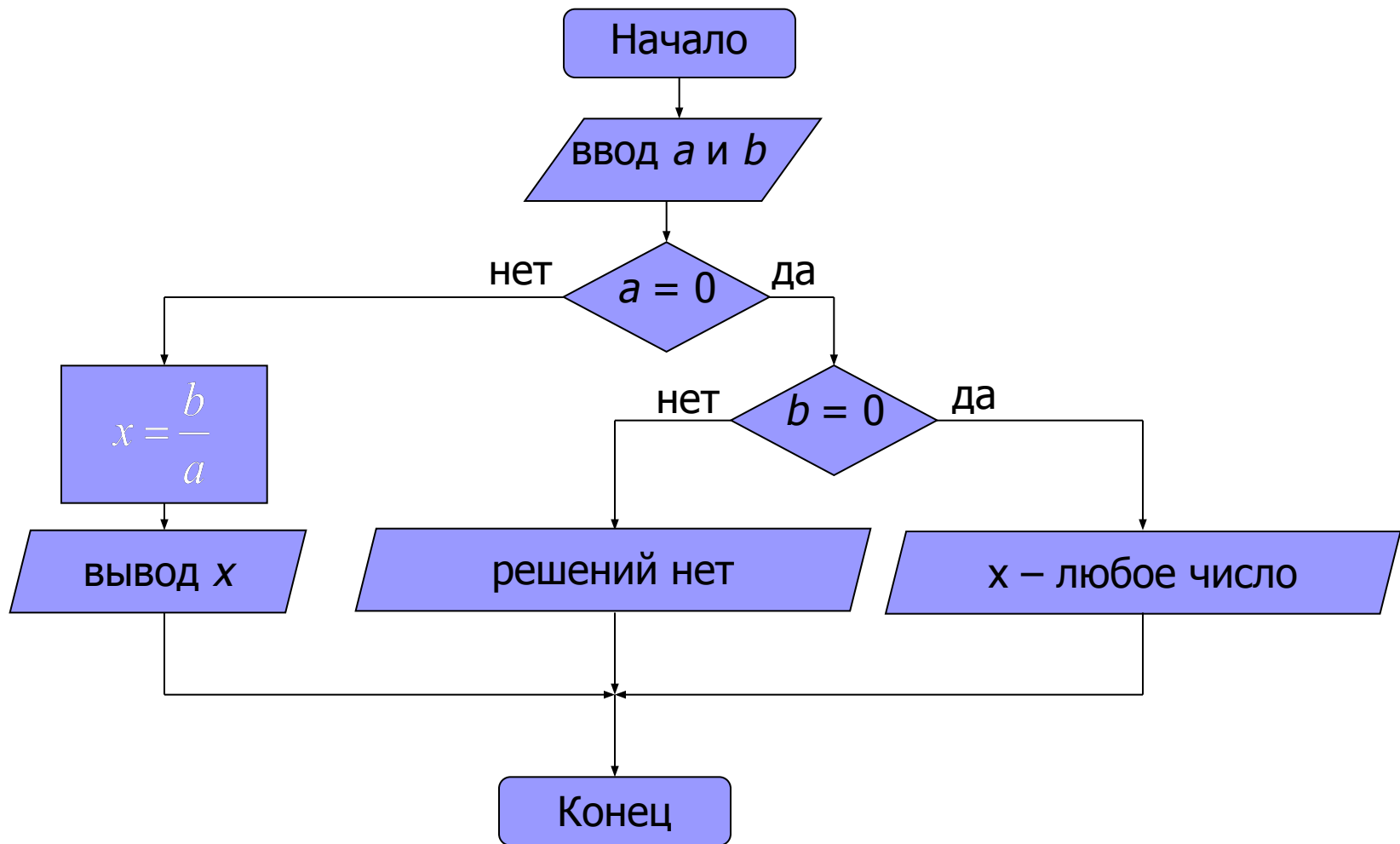
Решите уравнение: $ax = b$, где a и b – любые числа.

2. Анализ и исследование задачи

- 1) Если $a = 0$, $b \neq 0$, то решений нет.
- 2) Если $a = 0$, $b = 0$, то x – любое число.
- 3) Если $a \neq 0$, $b \neq 0$, то $x = \frac{b}{a}$.

3. Разработка алгоритма.

Составим алгоритм в виде блок-схемы.



4. Программирование

Программа на языке Pascal

```
program lin_ur;
uses crt;
var
a,b,x:Real;
BEGIN
ClrScr;
Writeln;
Writeln;
Writeln;
Writeln;
Writeln('Vvedite a');
Readln(a);
Writeln('Vvedite b');
Readln(b);
      If a=0 then
      Begin
        If b=0 then Writeln('x - luboe chislo')
          else Writeln('Uravnenie
resheniy ne imeet');
      End
      else
      Begin
        x:=b/a;
        Writeln('Koren uravneniya: ',x:8:3);
      End;
Readln;
END.
```

5. Тестирование и отладка программы

6. Анализ результатов

Задачи для самостоятельного решения.

Задача 1

Даны два числа a и b , которые являются сторонами прямоугольника. Напишите алгоритм (в виде блок-схемы), с помощью которого находятся площадь и периметр такого прямоугольника.

Задача 2

Даны три числа a , b и c , которые являются сторонами треугольника. Напишите алгоритм (в виде блок-схемы), с помощью которого находится периметр такого треугольника (площадь).

Основы программирования на языке Pascal

Разделы программы:

1. Заголовок программы.

Начинается со служебного слова

`Program <имя программы>;`

Имя программы не должно содержать спец. символов и символа «пробел».

Пример: `Program treugolnik;`

2. Раздел описания модулей.

Начинается со служебного слова `Uses <Имя модуля>`

Пример: `Uses CRT;`

3. Раздел описания переменных.

Переменные описываются после служебного слова `"Var"`.

`<переменная>:<тип>.`

Пример:

Var

a,b:Integer; (целые числа)

c:Real; (действительные числа)

4. Текст программы.

BEGIN

Программа

END.

Текст программы начинается со служебного слова `"BEGIN"` и заканчивается служебным словом `"END"`, после которого ставится точка.

В конце каждой строки ставится `;"`. После служебных слов `;"` не ставится.

Основные операторы языка Pascal

1. Операторы ввода/вывода

Write(In) - оператор вывода;

Read(In) - оператор ввода;

Если писать в конце этих операторов "In", то курсор будет переходить на следующую строку.

Пример:

```
Write('Vvedite znachenie a: ');
```

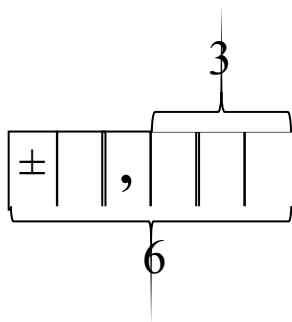
```
Readln(a);
```

Текст в операторе вывода заключается в апострофы.

Для того чтобы вывести на экран значение переменной надо ввести:

```
Write('Znachenie a: ', a:6:2);
```

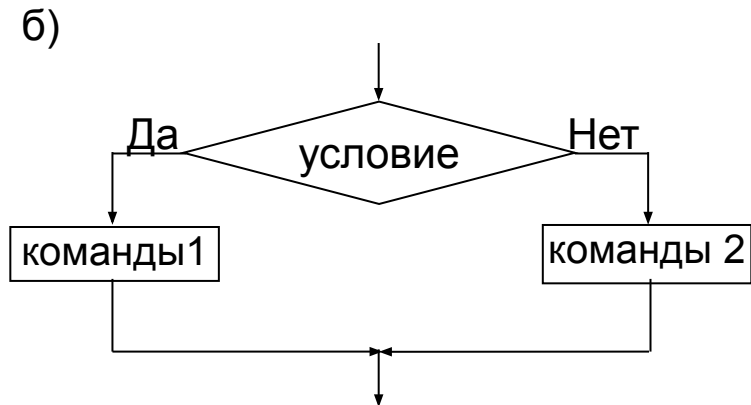
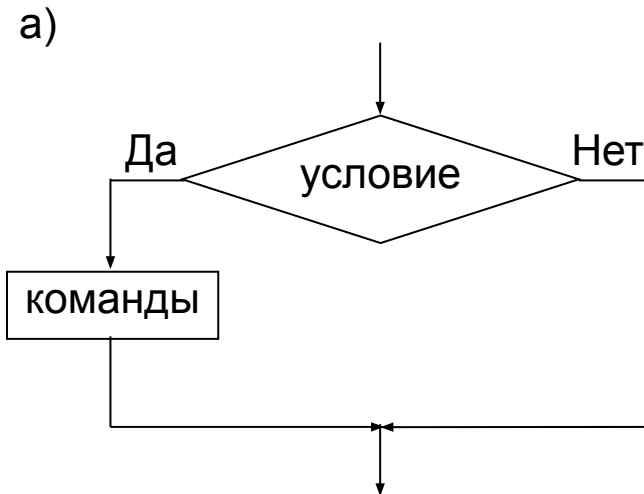
В данном примере сначала на экране появится строка Znachenie a:, а затем появится само значение числа a. 6 означает, что под вывод числа a выделяется 6 позиций, включающих в себя позицию под знак числа, позицию под запятую и количество знаков после запятой, 3 - число позиций после запятой.



2. Оператор условия (или условный оператор)

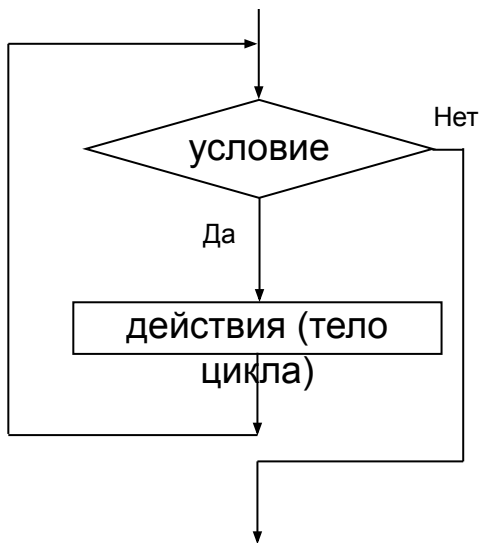
a)
If <условие>
Then
 Begin
 Команды
 End;

б)
If <условие>
Then
 Begin
 Команды
 End
Else
 Begin
 Команды
 End;

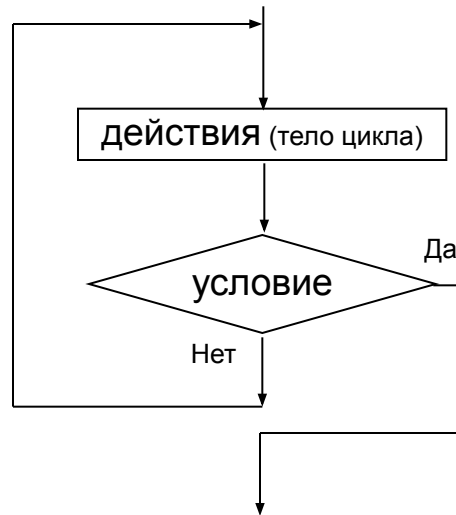


Операторы цикла

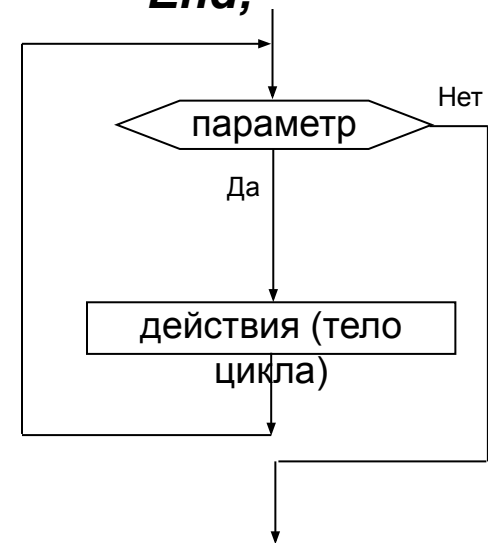
1.
While <условие>
Begin
 Команды
End;



2.
Repeat
 Команды
Until <условие>;



3.
For $i:=1$ to n **do**
 Begin
 Команды
 End;



Отметим различия между первыми двумя типами циклических алгоритмов:

- 1) У цикла а) условие находится в начале, у б) – в конце;
- 2) У цикла а) условие продолжения выполнения цикла должно быть истинно, у цикла б) – ложно;
- 3) Тело цикла а) может ни разу не выполниться (если условие сразу окажется ложным), тело цикла б) выполнится обязательно по крайней мере один раз;

Тестирование и отладка программы

При нажатии на кнопку F9 запустится компилятор. Если компилятор обнаружит в тексте программы синтаксические ошибки (чаще всего они связаны с пропуском ; или с несовпадением типов данных), то он выдаст сообщение с кодом ошибки. После того как все ошибки будут устранены, следует нажать CTRL+F9. Тогда программа начнёт работать.