

ПРЕЗЕНТАЦИЯ ПО ИНФОРМАТИКЕ НА ТЕМУ: «АЛГОРИТМЫ»

Подготови
л:
Гараев К.Р.

Немного из истории

Слово «алгоритм»
происходит от латинского
написания имени
арабского математика
аль-Хорезми (Algorithmi),
который впервые описал
правила выполнения
четырёх арифметических
действий. 9 век н.э.



Понятие алгоритма

Алгоритм – это точное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Алгоритмизация – процесс создания алгоритмов.

Исполнитель алгоритма

Исполнитель – тот, кто исполняет

АЛГОРИТМ
Исполнитель

Формальный

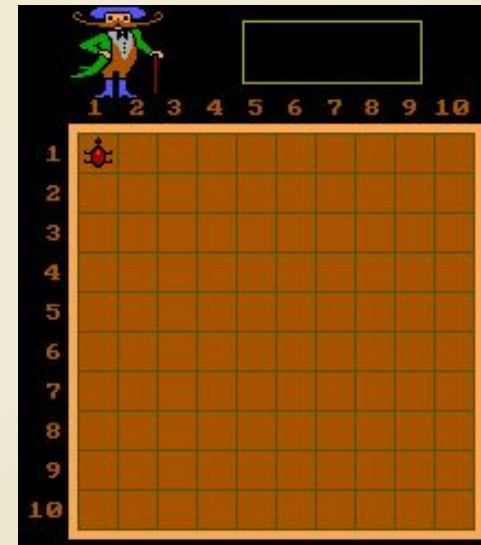
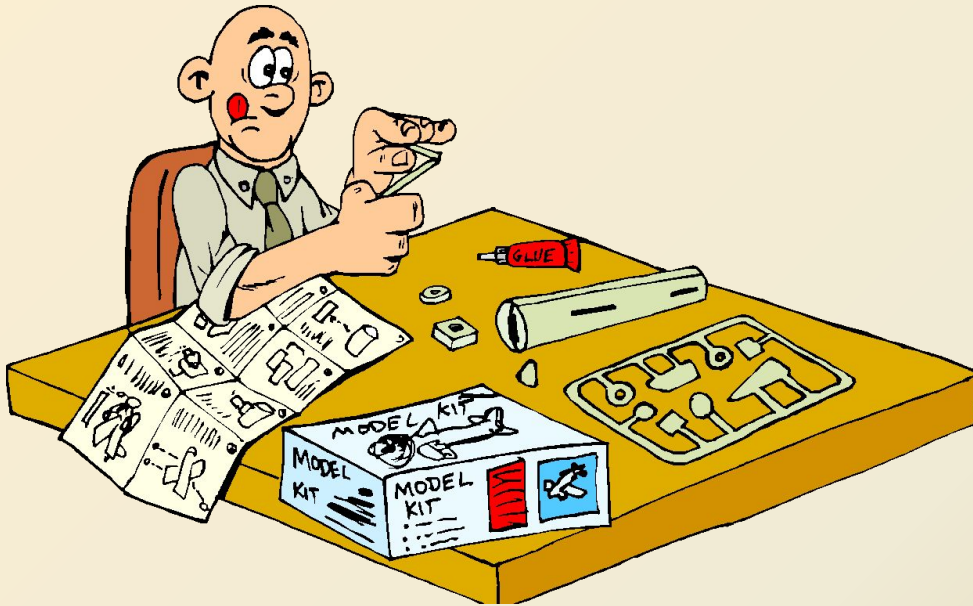
Неформальный



Формальный исполнитель следует указаниям в виде программы, не думая о результате.

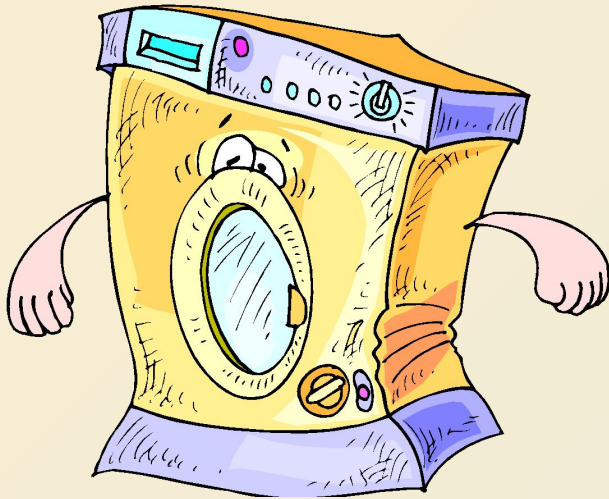
Исполнителя характеризуют

1. Среда , т. е. обстановка, в которой работает исполнитель.



Исполнителя характеризуют

2. Система команд исполнителя(СКИ), т. е. набор понятных исполнителю команд.



```
4ВВЕРХ 5ВЛЕВО 6ЭТО 7ПОВТОРИ 8ЕСЛИ 9ТО
ВВЕРХ
ВНИЗ
ВЛЕВО
ВПРАВО
рус  строк 5  строка 1  столбец 1
```

Исполнителя характеризуют

3. Система допустимых действий исполнителя, т. е. те действия, которые исполнитель может совершить. При использовании недопустимых действий возникают «сбои» в системе



Свойства алгоритмов

1. Дискретность – переход к следующему действию возможен только после выполнения предыдущего.
2. Понятность – точность и подробность в написании алгоритма.
3. Определенность – исполнитель должен знать, к какому пункту ему переходить после определенного действия.
4. Массовость – применение одного алгоритма к решению многих однотипных задач.
5. Результативность – направленность на получение конкретного результата.

Способы записи алгоритмов

1. Словесно-формульный (естественный язык) – предназначен для исполнения алгоритма человеком. Форма записи команд – произвольная.

Пример. Алгоритм приготовления чая для робота-повара:

- 1) Налить молоко
- 2) Налить заварку
- 3) Налить кипяток
- 4) Насыпать сахар
- 5) Помешать

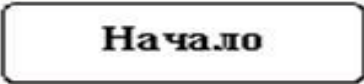
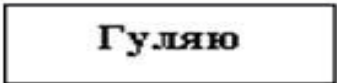

Словесный способ не имеет широкого распространения, так как такие описания:

- 1) Строго не формализуемы;
- 2) Страдают многословностью записей;
- 3) Допускают неоднозначность толкования отдельных предписаний.



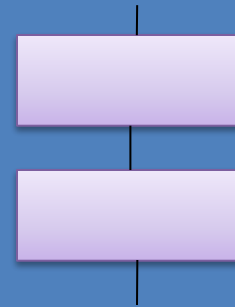
Способы записи алгоритмов

2. Графический – способ представления алгоритма с помощью блок-схем.

Вид стандартного графического объекта	Назначение
	Начало алгоритма
	Конец алгоритма
	Выполняемое действие записывается внутри прямоугольника
	Условие выполнения действий записывается внутри ромба
	Счетчик кол-во повторов
	Последовательность выполнения действий.

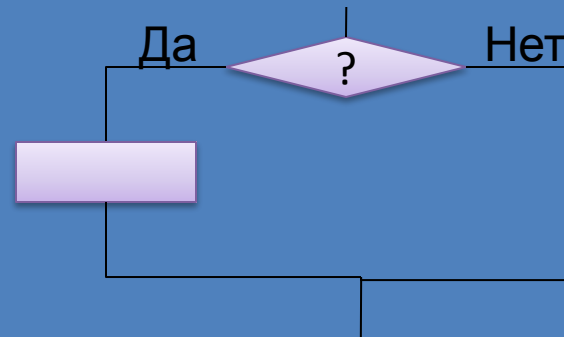
Структура СЛЕДОВАНИЕ

```
{оператор;  
оператор;  
...  
}
```



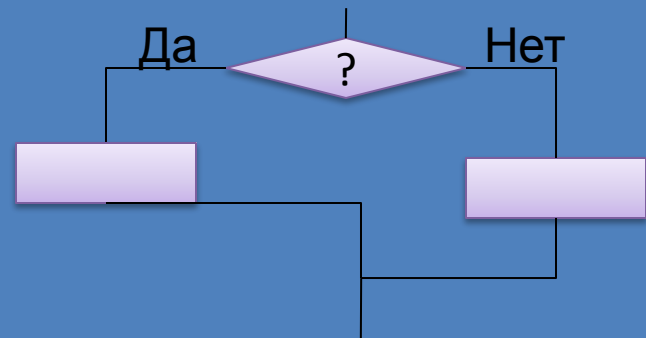
Структура ВЕТВЛЕНИЕ (в неполной форме)

```
Если (условие), то  
{оператор;  
оператор;  
...  
}
```



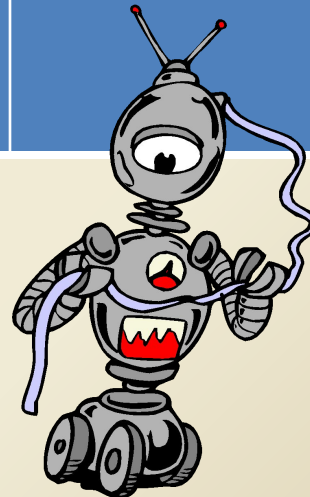
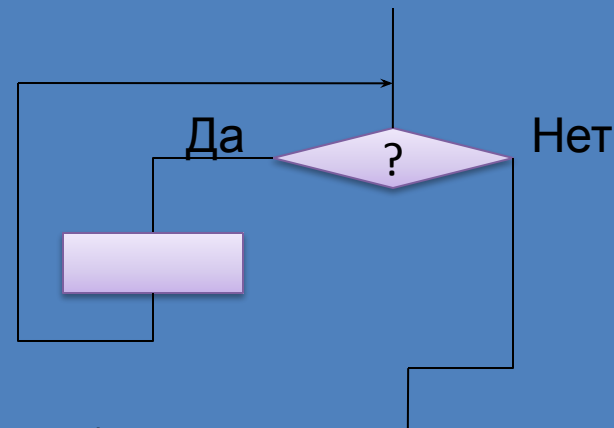
Структура ВЕТВЛЕНИЕ (в полной форме)

```
Если (условие), то  
{оператор;  
оператор;  
...  
}  
иначе  
{оператор;  
оператор;  
...  
}
```



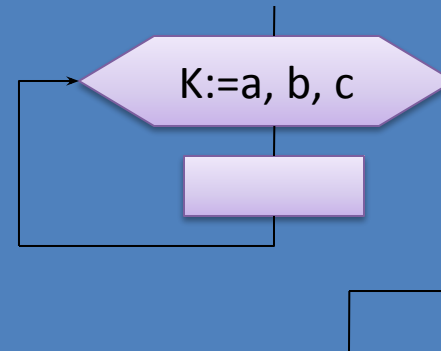
Структура ЦИКЛ В ФОРМЕ «ПОКА»

```
Делать пока (условие)  
{оператор;  
 оператор;  
 ...  
}  
(*конец цикла*)
```



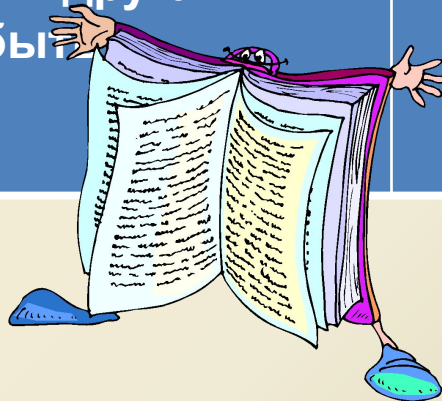
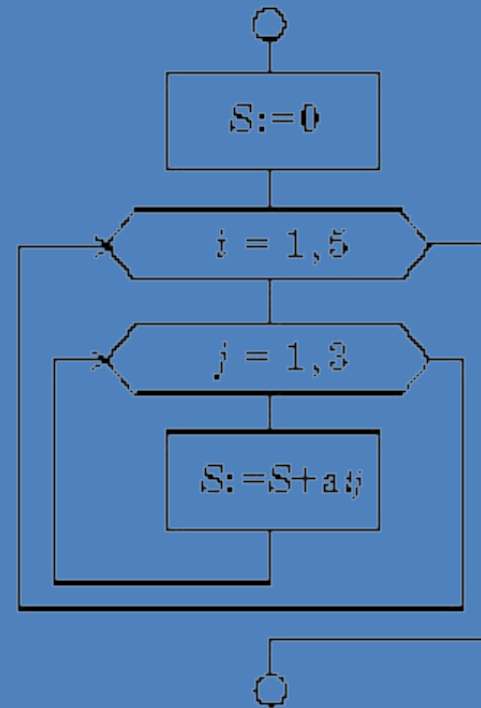
Структура ЦИКЛ СО СЧЕТЧИКОМ

```
Делать от k:=a до b  
с шагом с  
{оператор;  
оператор;  
...  
}  
(*конец цикла*)
```

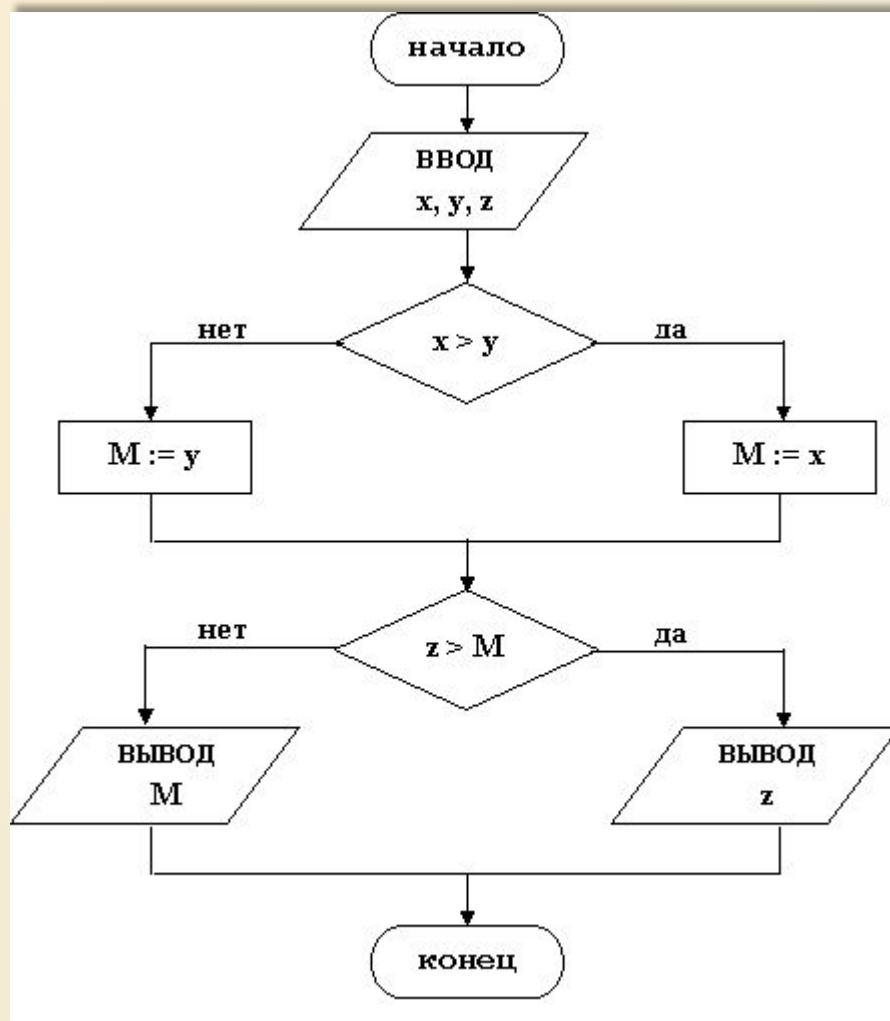


Вложенный цикл

Возможны случаи, когда внутри тела цикла необходимо повторять некоторую последовательность операторов, т. е. организовать внутренний цикл. Такая структура получила название цикла в цикле или вложенных циклов. Глубина вложения циклов (то есть количество вложенных друг в друга циклов) может быть различной.



Пример графической записи алгоритма



Способы записи алгоритмов

3. На алгоритмическом (формальном) языке

Общий вид алгоритма:

алг название алгоритма

дано описание исходных переменных

нач описание промежуточных величин

| последовательность команд (тело алгоритма)

кон конец алгоритма



Пример записи алгоритма на формальном языке

```
File Edit Search Run Compile Debug Tools Options Window Help
NONAME00.PAS
Error 10: Unexpected end of file.
begin
writeln(<Привет еще раз!>);
end
4:1
Output
Привет!
Привет!
Привет еще раз!
Привет еще раз!
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Ma
```

```
Program Dve polovinky v obratnom napravleniy;
const N=8;
Var a :array [1..N]of integer;
i,k:integer;
Begin
For i:=1 to N do
readln(a[i]);
For i:=1 to (N div 2) div 2 do Begin
k:=a[i];
a[i]:=a[(N div 2)-i+1];
a[(N div 2)-i+1]:=k;end;
For i:=(N div 2)+1 to ((N div 2)+((N div 2) div 2)) do Begin
k:=a[i];
a[i]:=a[N-i+(N div 2)+1];
a[N-i+(N div 2)+1]:=k;end;
For i:=1 to N do
Write( a[i], ' ' );
end.
```

Последовательность написания алгоритмов

1. Поставить задачу
2. Придумать список команд для исполнителя(СКИ)
3. Составить алгоритм из СКИ

Критерии качества алгоритма

1. Связанность – определяется количеством промежуточных результатов, подлежащих запоминанию.
2. Объем алгоритма – количество операций (шагов), которые необходимо выполнить для достижения конечного результата.
3. Длительность решения – определяется как количеством, так и сложностью шагов.
4. Разветвленность алгоритма – характеризует логическую сложность и определяется количеством путей, по которым может реализовываться алгоритм.
5. Цикличность алгоритма – заключается в том, что фактическое количество операций, которые должны быть выполнены, превышает количество операций, содержащихся в записи алгоритма.