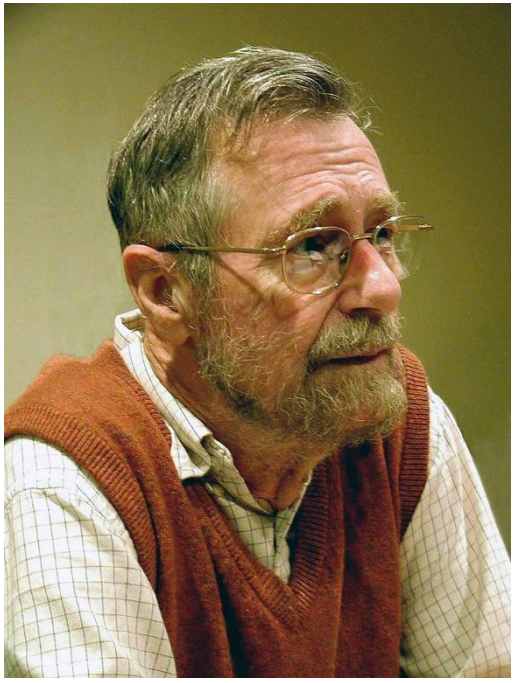
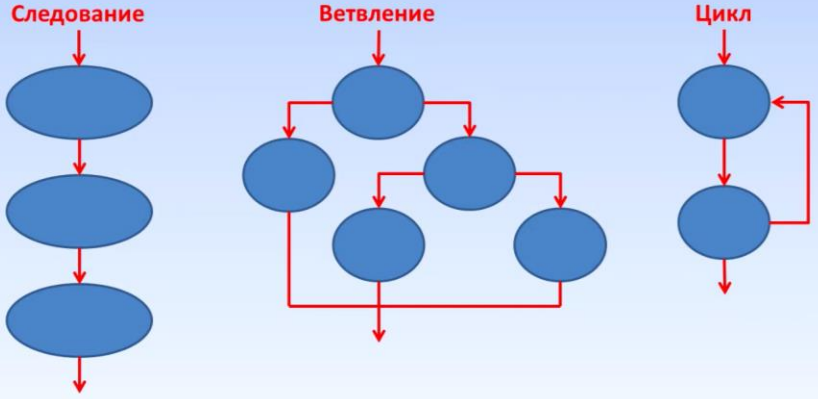


Э. Дейкстра
1930–2002 гг.

Базовые алгоритмические структуры

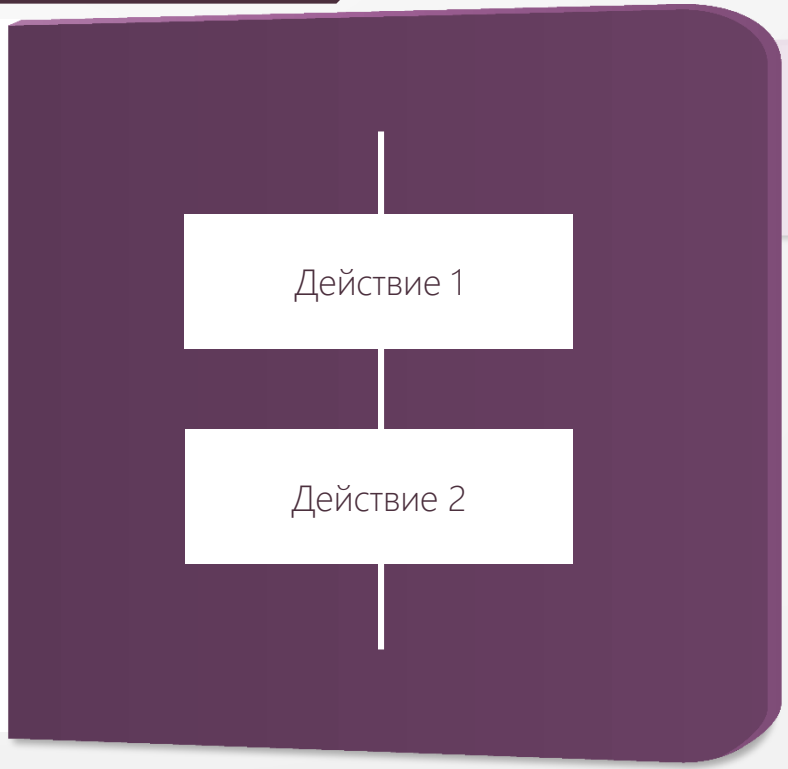


Задание
Составить конспект с блок-схемами для каждой конструкции

Для учащихся Уинской СОШ

ЛИНЕЙНЫЙ АЛГОРИТМ

Следование — алгоритмическая конструкция, отображающая естественный, последовательный порядок действий. Алгоритмы, в которых используется только структура «следование», называются линейными алгоритмами.



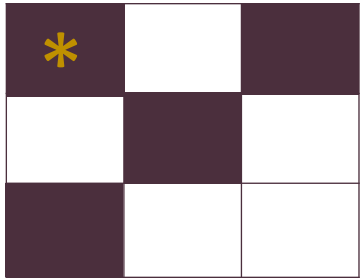
На языке программирования Pascal

```
begin
 {операторы}
end.
```

Следование

Исполнитель: Робот
Команды: вверх, вниз, влево, вправо, закрасить.
Задача: Составить линейный алгоритм действий Робота, нарисовавшего узор и вернувшегося в исходное положение.

```
алг узор
нач
   закрасить
   вниз
   вниз
   закрасить
   вправо
   вверх
   закрасить
   вправо
   вверх
   закрасить
   влево
   влево
кон
```

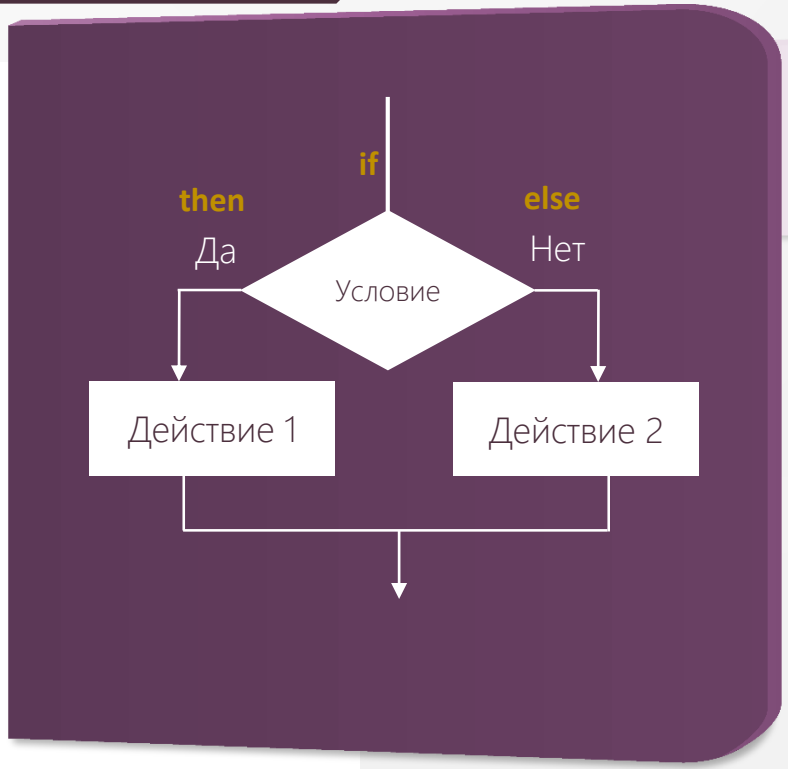


* <-- Робот

ВЕТВЛЕНИЕ (ВЫБОР)

Логическое выражение (условие) может принимать только два значения:
- true (истина, да, 1)
- false (ложь, нет, 0)

Ветвление — алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной из двух последовательностей действий (ветвей). Алгоритмы, в основе которых лежит структура «ветвление», называют разветвляющимися.

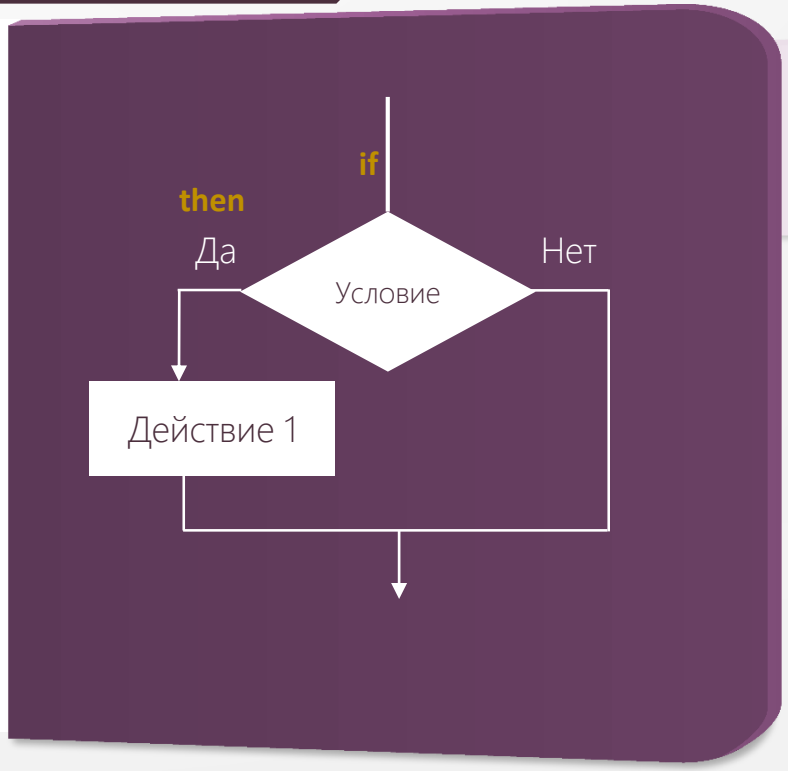


На языке программирования Pascal

```
begin
 if {логическое выражение} then
  {действие1}
 else
  {действие2};
end.
```

Ветвление

Ветвление неполное — алгоритмическая конструкция, в которой в зависимости от результата проверки условия («да» или «нет») предусмотрен выбор одной последовательности действий (ветви).



На языке программирования Pascal

```
begin
 if {логическое выражение} then
  {действие1};
end.
```

Алгоритмическая форма записи ветвления

Полная форма ветвления:

если <условие>
то <действия 1>
иначе <действия 2>
всё

На языке программирования Pascal

```
begin
 if {логическое выражение} then
  {оператор1}
 else
  {оператор2};
end.
```

Пример **полной** формы ветвления:

алг правописание приставок НЕ, НИ
нач
  если приставка под ударением
    то писать НЕ
    иначе НИ
  всё
кон

Алгоритмическая форма записи ветвления

Неполная форма ветвления:

если <условие>
то <действия 1>
всё

На языке программирования Pascal

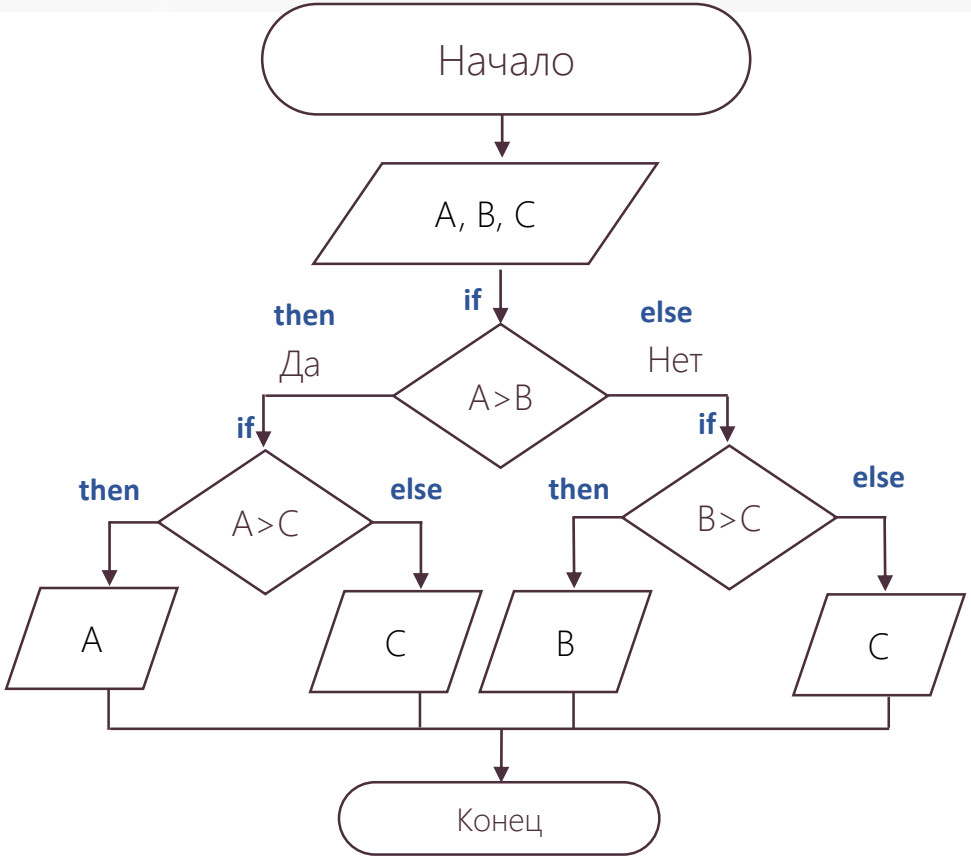
```
begin
  if {логическое выражение}
   then {оператор1};
end.
```

Пример **неполной** формы ветвления:

алг сборы на прогулку
нач
если на улице дождь
то взять зонтик
всё
кон

Ветвление

Нахождение наибольшего числа из трёх: A, B, C.
Дано: A, B, C.



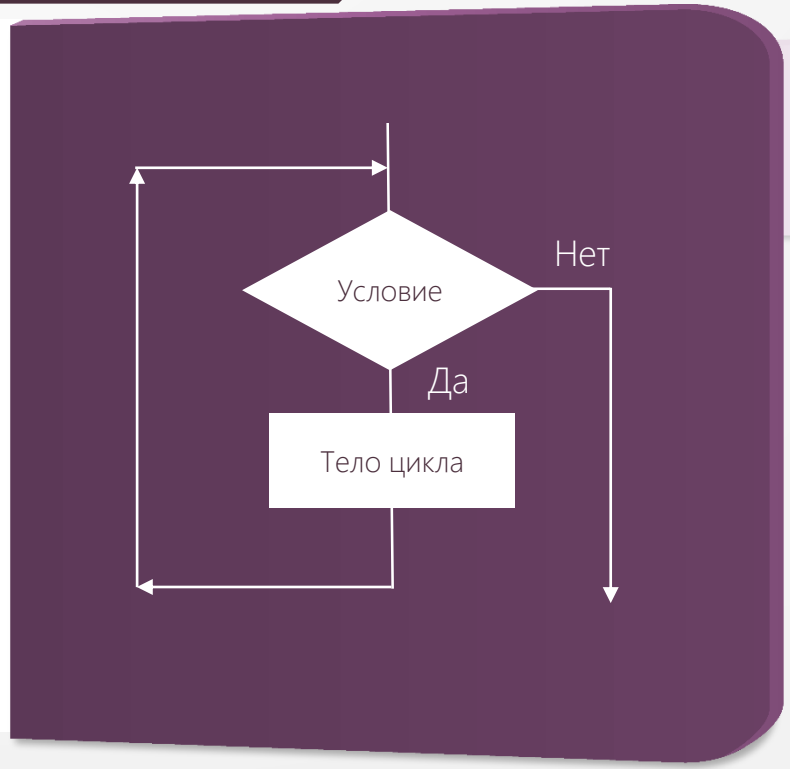
На Pascal

```
var  a, b, c: real;
begin
  readln(a, b, c);
  if a > b then
    if a > c  then
      writeln(a)
    else
      writeln(c)
  else
  if b > c then
    writeln(b)
  else
    writeln(c);
end.
```

ЦИКЛ

Повторение — алгоритмическая конструкция, представляющая собой последовательность действий, выполняемых многократно.

Алгоритмы, содержащие конструкцию повторения, называют циклическими или циклами. Последовательность действий, многократно повторяющаяся в процессе выполнения цикла, называется **телом цикла**.



В зависимости от способа организации повторений различают 4 типа циклов:

Цикл с заданным условием продолжения работы (цикл **ПОКА**)

Цикл с заданным условием окончания работы (цикл **ДО**)

Цикл с заданным числом повторений (цикл **ДЛЯ** или цикл **ПОВТОРИ**)

Цикл с заданным условием продолжения работы

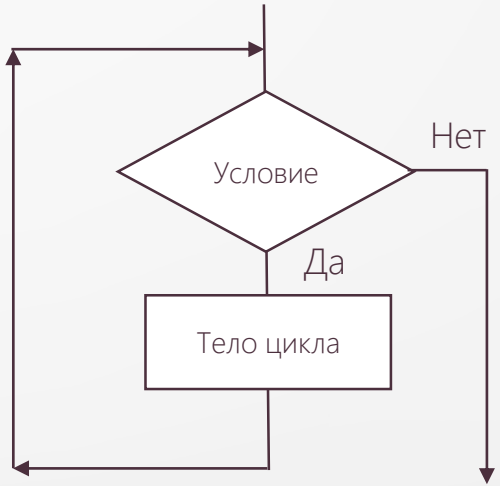
Алгоритмическая форма записи:

нц пока <условие>
<тело_цикла (последовательность действий)>
кц

На Pascal

```
begin
  while {логическое выражение} do
    {операторы};
end.
```

Цикл «ПОКА»



Алгоритм выполнения цикла «ПОКА»

1. Проверяется условие (вычисляется значение логического выражения).

2. Если условие удовлетворяется, то выполняется тело цикла и снова осуществляется переход к проверке условия; если же условие не удовлетворяется, то выполнение цикла заканчивается.

Пример цикла «ПОКА»

Алгоритм, по которому из всех имеющихся кубиков отбираются только красные и складываются в корзину.

```
алг отбор
нач
    нц пока есть кубики
    взять один кубик
    если кубик красный
       то положить его в корзину
    иначе отложить кубик в сторону
    все
  кц
кон
```

Цикл с заданным условием окончания работы

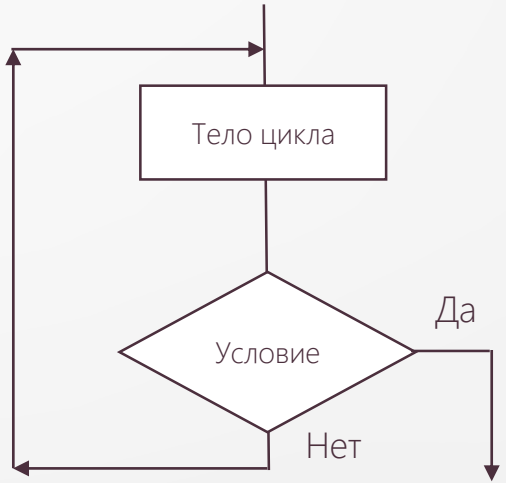
Алгоритмическая форма записи:

нц
<тело_цикла (последовательность действий)>
кц при <условие>

На Pascal

```
begin
  repeat
    {операторы};
  until {логическое выражение};
end.
```

Цикл «ДО»



Алгоритм выполнения цикла «ДО»

1. Выполняется тело цикла.

2. Проверяется условие (вычисляется значение логического выражения); если условие не удовлетворяется, то снова выполняется тело цикла и осуществляется переход к проверке условия; если же условие удовлетворяется, то выполнение цикла заканчивается.

Пример цикла «ДО»

Алгоритм по заучиванию таблицы умножения

алг таблица умножения
нач
нц
прочитать таблицу умножения по учебнику 1 раз
повторить таблицу умножения с закрытым учебником
кц при не сделал ошибку
кон

Цикл с заданным числом повторений

Алгоритмическая форма записи:

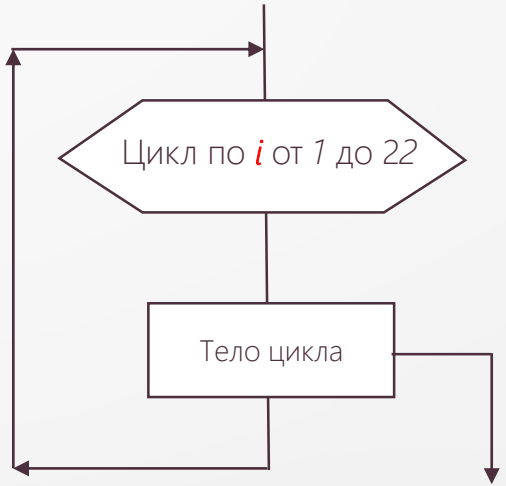
нц для i **от** 1 **до** 22
<тело цикла (последовательность действий)>
кц

На Pascal

```
begin
  for var i:=1 to 22 do
      {операторы};
end.
```

```
begin
  loop 22 do
      {операторы};
end.
```

Цикл «ДЛЯ»



Алгоритм выполнения цикла «ДЛЯ»

1. Параметру цикла присваивается начальное значение.

2. Параметр цикла сравнивается с конечным значением; если параметр цикла не превышает конечное значение, то выполняется тело цикла, увеличивается значение параметра цикла на шаг и снова осуществляется проверка параметра цикла; если же параметр цикла превышает конечное значение, то выполнение цикла заканчивается.

Пример цикла «ДЛЯ»

Алгоритм переправы через реку воинского отряда из пяти человек. Солдаты могут воспользоваться помощью двух мальчиков — хозяев небольшой лодки, в которой может переправиться или один солдат, или два мальчика.

алг переправа
нач
нц для i **от** 1 **до** 5
- Два мальчика переправляются на противоположный берег.
- Один мальчик высаживается на берег, другой плывет обратно.
- Солдат переправляется через реку.
- Мальчик возвращается на исходную позицию.

кц
кон

Законспектировал? Тогда все Ok!

Мультики тут и там,
подробнее здесь…

Для забывчивых ☺!

В тетради конспекты слайдов

2, 4, 5, 9, 11, 14 и 17

В начало