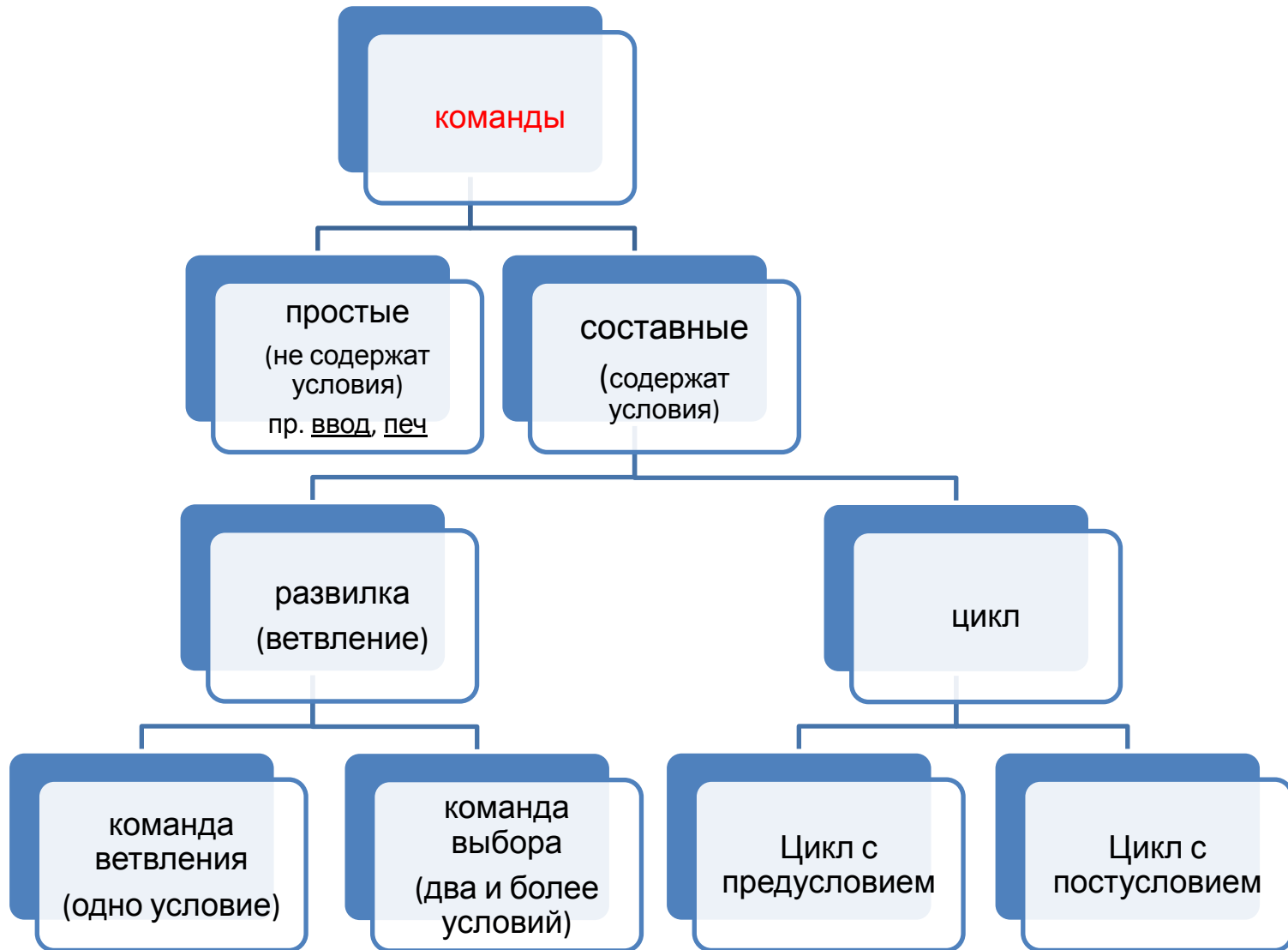
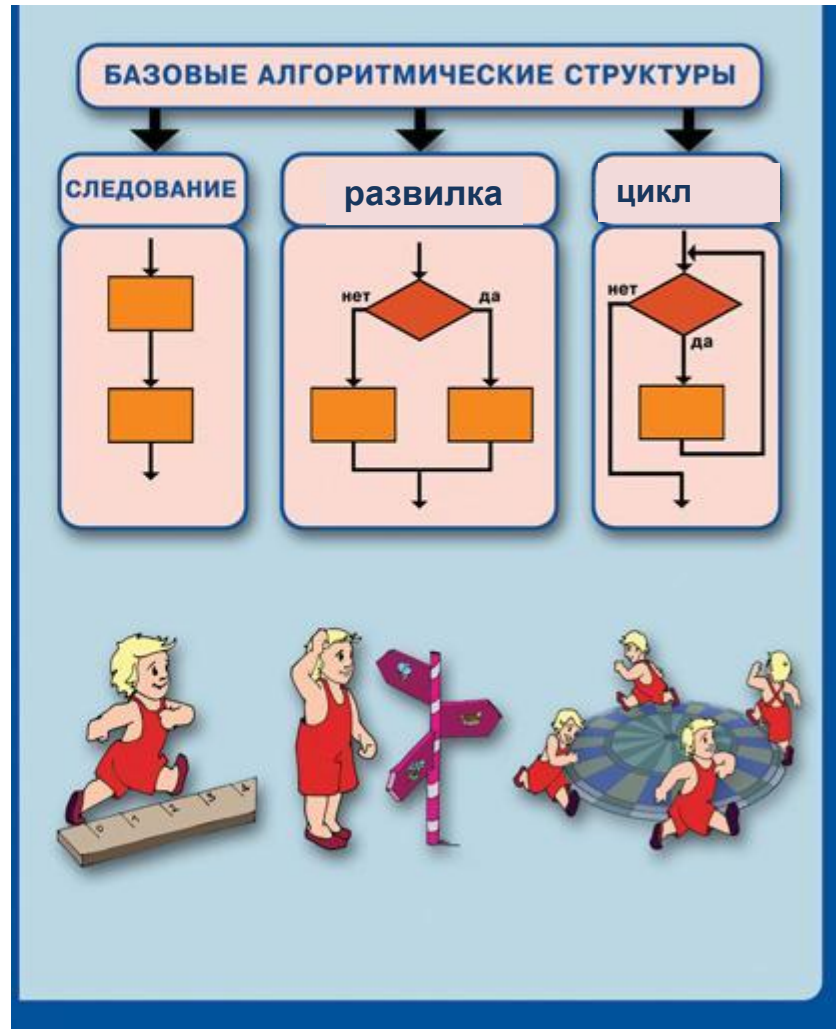


# Виды команд



Любой алгоритм можно составить, используя 3 базовые алгоритмические структуры: следование, развилку(ветвление) и цикл.



## Виды алгоритмов

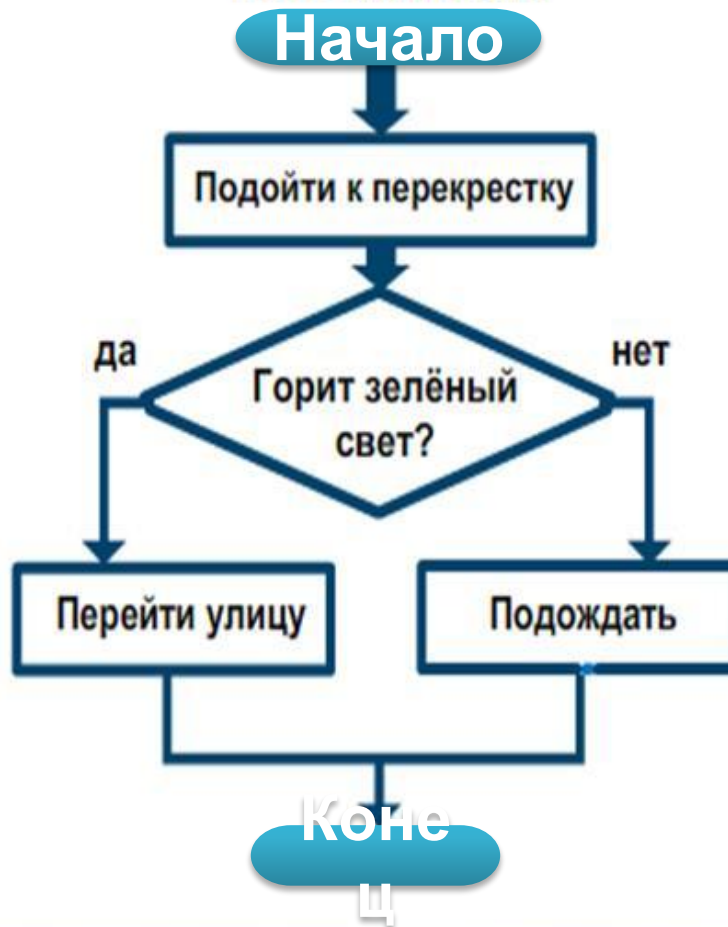


# Типы алгоритмов:

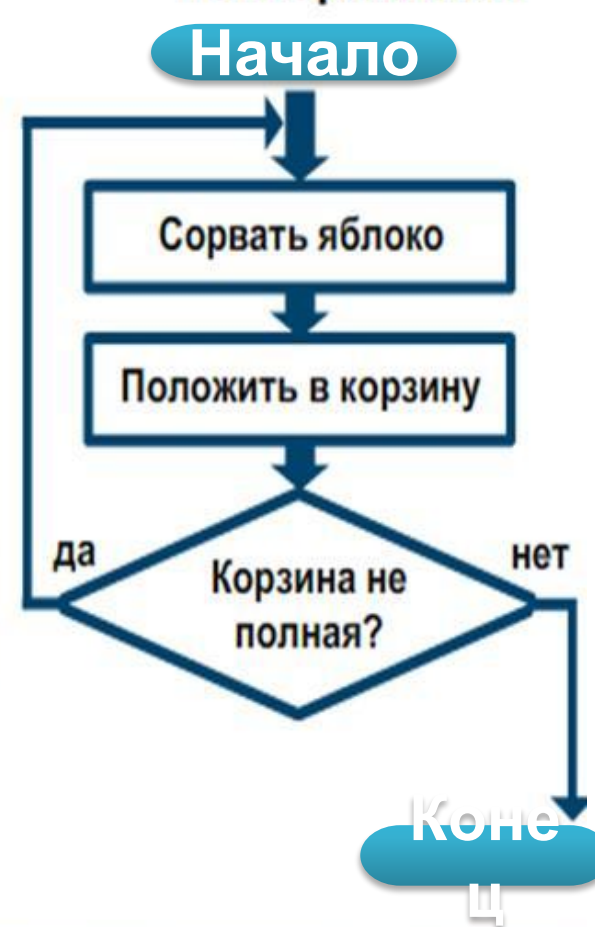
линейные  
алгоритмы



алгоритмы с  
ветвлениями



алгоритмы с  
повторениями



# Линейный алгоритм



Линейный алгоритм – это алгоритм, состоящий из простых команд. В линейном алгоритме порядок выполнения команд жестко задан.

Старт



Начало

Действие 1

...

Действие n

Конец



Финиш



# Линейный алгоритм

## Переход улицы



Начало

Подойти к  
переходу

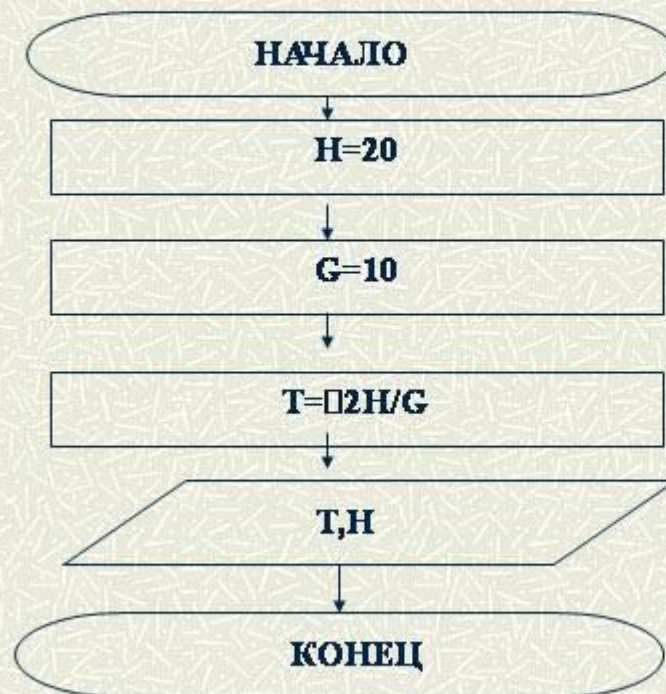
Дождаться зелёного света

Перейти улицу

Конец



# Линейный алгоритм (следование)



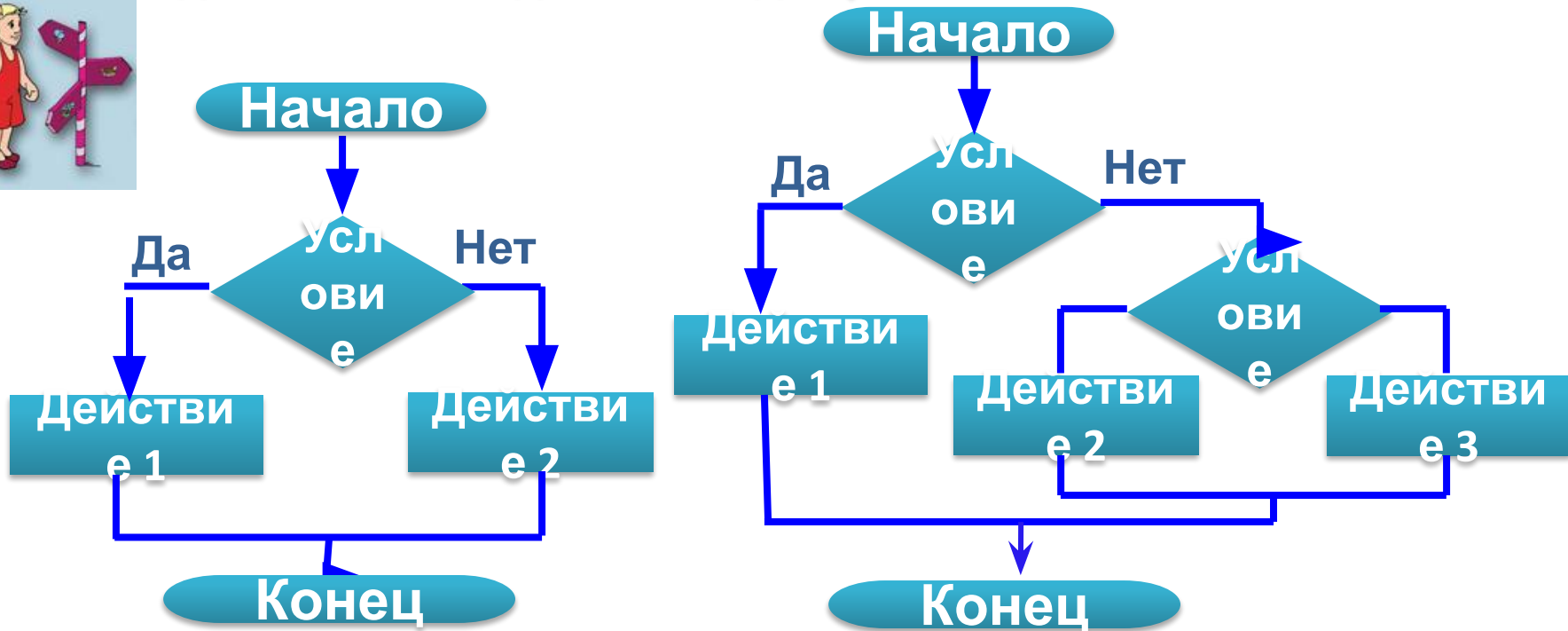
# Разветвляющийся алгоритм или алгоритм с ветвлением



**Разветвляющийся алгоритм** – это алгоритм, состоящий из команд «развилка», т.е. из команд ветвления и/или команд выбора

**Команда ветвления**- это команда, содержащая одно условие, в зависимости от которого выполняется та или иная последовательность действий один раз.

**Команда выбора**- то команда, содержащая несколько условия, в зависимости от которых выполняется та или иная последовательность действий один раз.

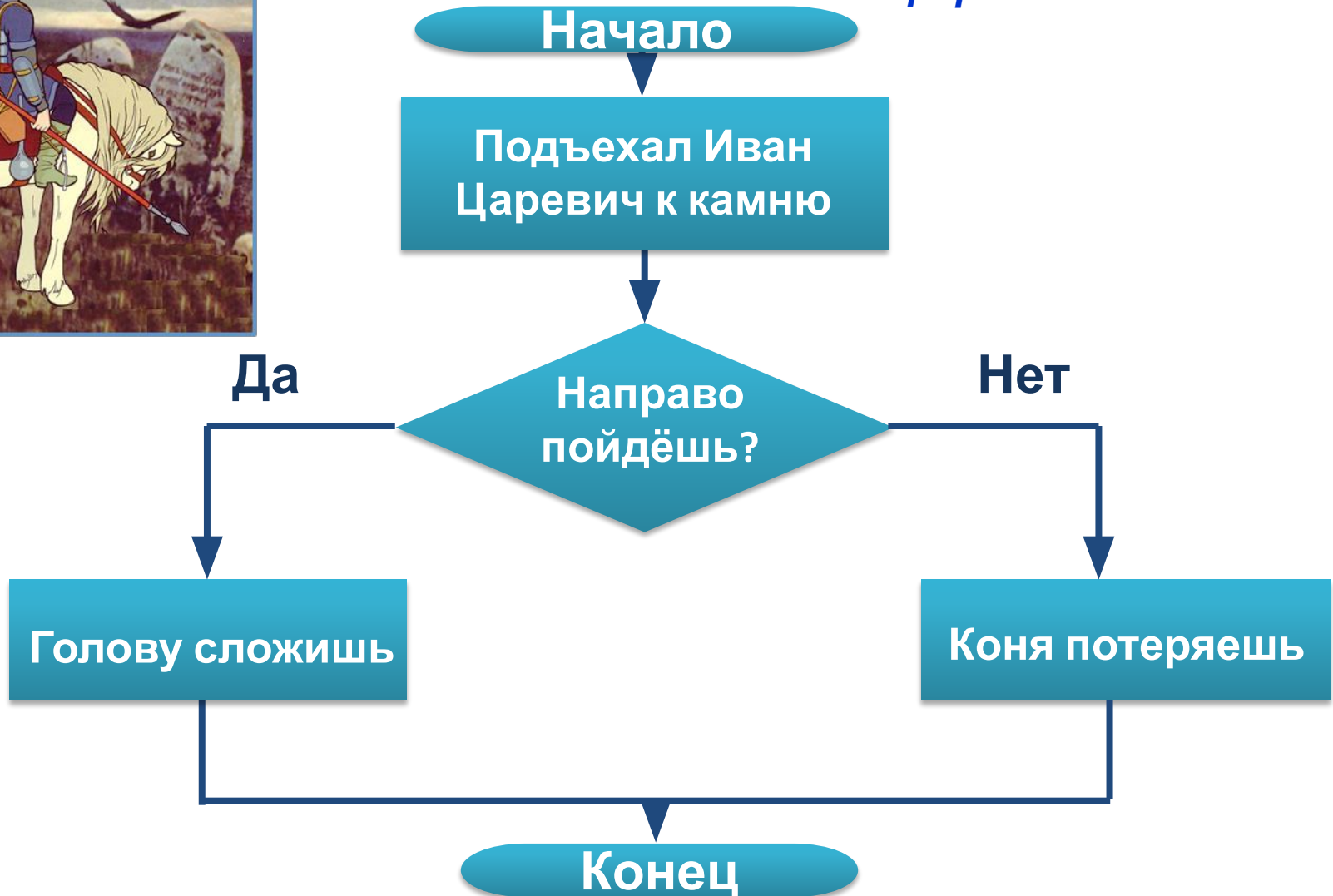




# Разветвляющийся алгоритм



*Команда ветвления в полной форме*



# Разветвляющийся алгоритм

*Команда ветвления в неполной форме*



# Разветвляющийся алгоритм

*Команды ветвления в полной и неполной форме*

**ЕСЛИ** хочешь быть здоров,  
**ТО** закаляйся,  
**ИНАЧЕ** валяйся на диване  
**ВСЁ**



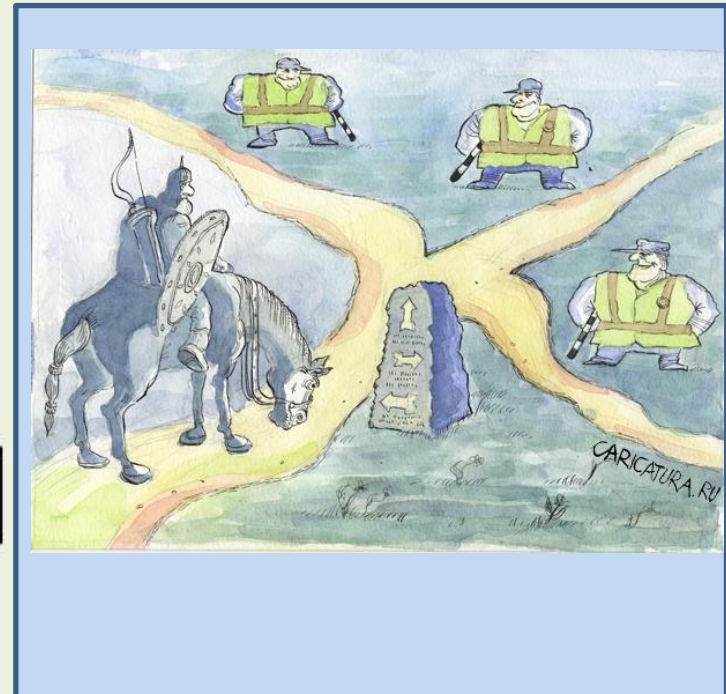
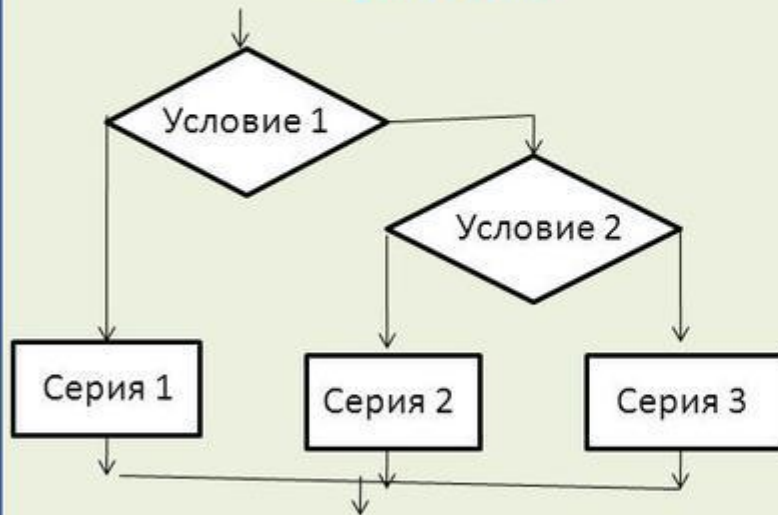
**ЕСЛИ** назвался груздем,  
**ТО** полезай в кузов  
**ВСЁ**

# Разветвляющийся алгоритм

## Развилка: команда выбора

В алгоритмической структуре **«выбор»** выполняется одна из нескольких последовательностей команд при истинности соответствующего условия

Блок - схема

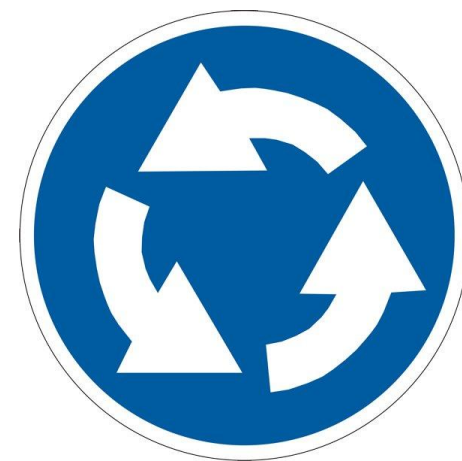
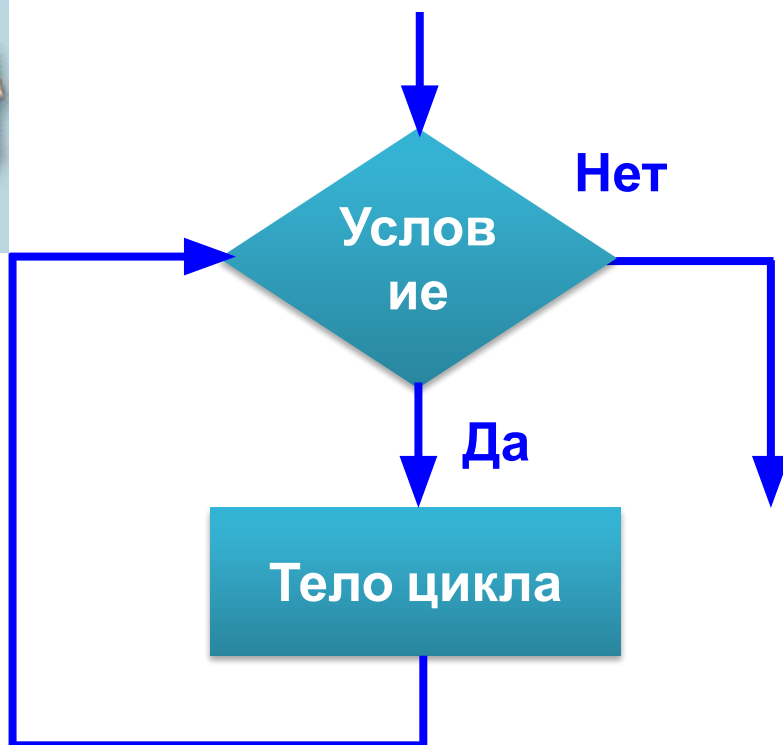


# Циклический алгоритм или алгоритм с повторением



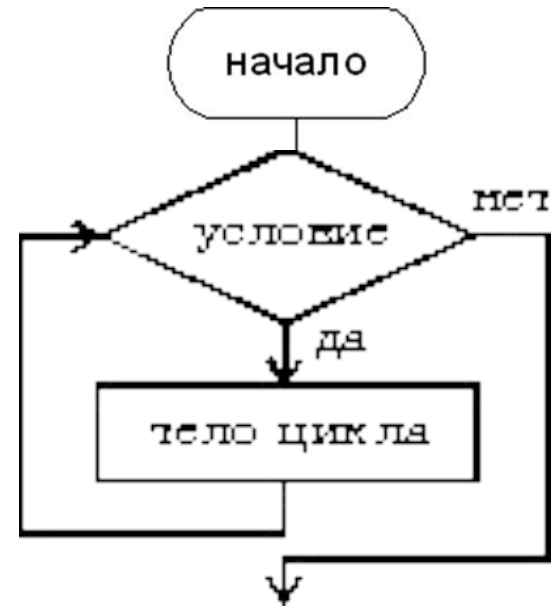
**Циклический алгоритм** –это алгоритм, состоящий из циклов.

**Цикл**- это команда, содержащая условие, в зависимости от которого определенная группа команд может выполняться несколько раз.



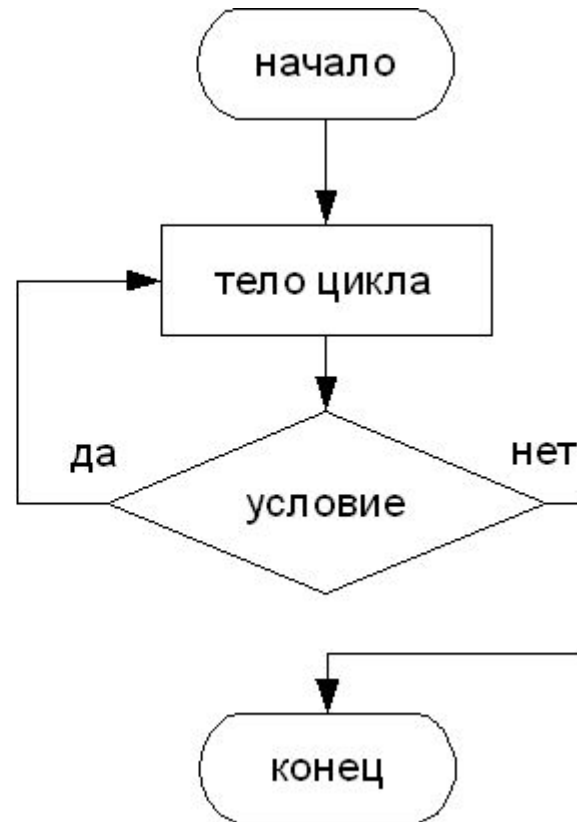
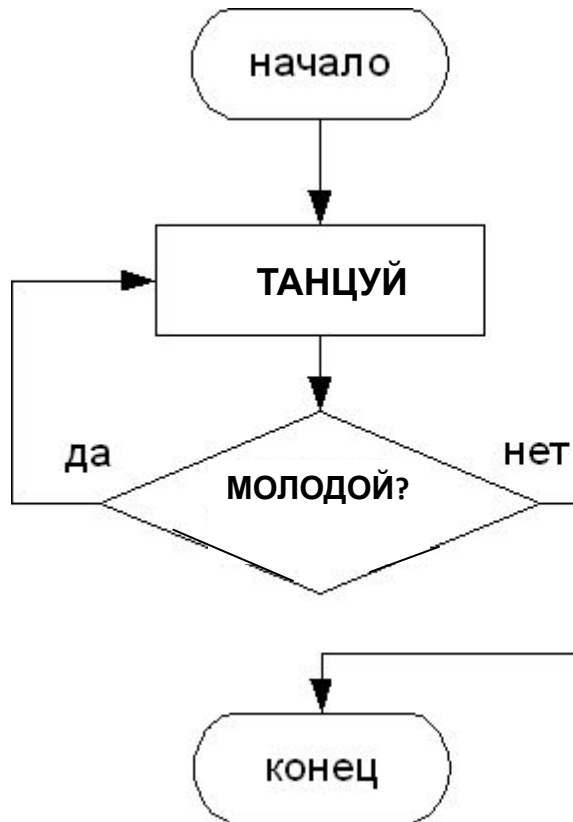
# Циклические алгоритмы

Цикл с предусловием



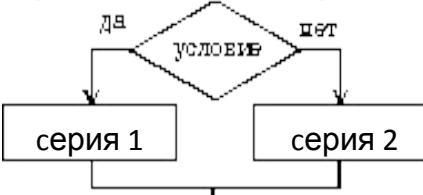
# Циклические алгоритмы

Цикл с постусловием



# Любой алгоритм можно составить, используя 3 базовые алгоритмические структуры:

## следование, развилку(ветвление) и цикл.

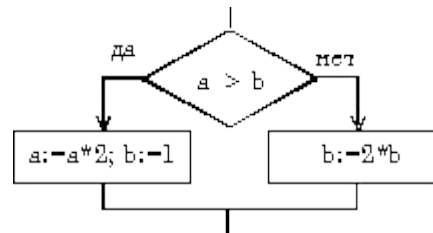
<p><b>Базовая структура "следование".</b></p> <p>Образуется последовательностью действий, следующих одно за другим</p>	<p><b>Базовая структура "развилка (ветвление)".</b></p> <p>Обеспечивает в зависимости от результата проверки условия (да или нет) выбор одного из альтернативных путей работы алгоритма. Каждый из путей ведет к общему выходу, так что работа алгоритма будет продолжаться независимо от того, какой путь будет выбран.</p>	<p><b>Базовая структура "цикл".</b></p> <p>Обеспечивает многократное выполнение некоторой совокупности действий</p>	
<p><b>Следование:</b> условий нет</p> 	<p><b>Команда ветвления:</b> содержит одно условие, в зависимости от которого выполняется та или иная серия команд один раз</p> 	<p><b>Команда выбора: (вид2)</b> содержит несколько условий</p> 	<p><b>Цикл:</b> содержит одно условие, в зависимости от которого группа команд может выполняться несколько раз</p>  <p><b>Цикл с предусловием</b></p> <p><b>Цикл с постусловием</b></p>
	<p><b>Команда выбора (вид1):</b> содержит несколько условий</p> 		



**Команда ветвления**- это команда, содержащая в себе условие, в зависимости от которого выполняется та или иная последовательность действий

**Команда ветвления (полная форма):** если—то—иначе—всё

**если**  $a > b$   
**то**  $a := 2*a; b := 1$   
**иначе**  $b := 2*b$   
**все**

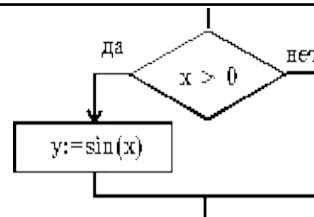


```
IF A>B THEN  
  begin  
    a :=2*a;  
    b := 1;  
  end  
ELSE b := 2*b;
```

**PS.1.**Если команд несколько, они заключаются в операторские скобки  
begin.... end  
**2.**Перед ELSE ; НЕ СТАВИТСЯ

**Команда ветвления(неполная форма):** : если—то—всё

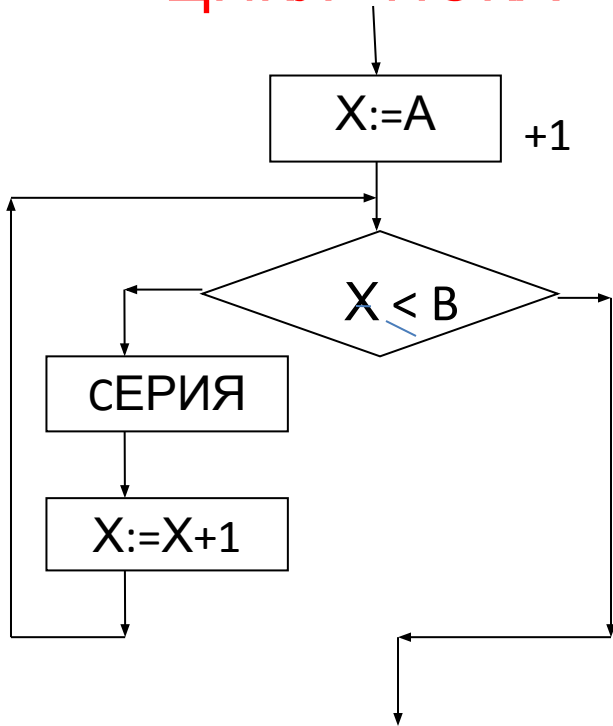
**если**  $x > 0$   
**то**  $y := \sin(x)$   
**все**



```
IF x > 0 THEN y := sin(x);
```

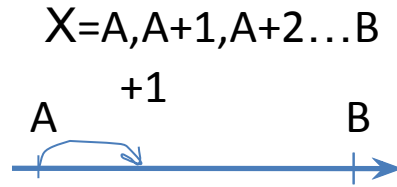
# ЦИКЛЫ С ПРЕДУСЛОВИЕМ

ЦИКЛ «ПОКА»

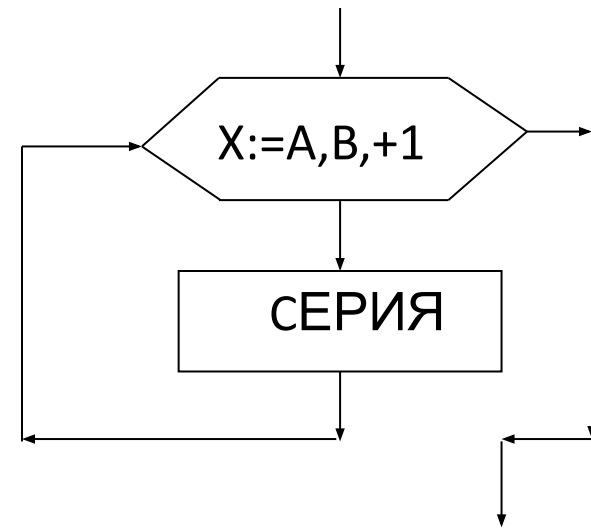


```
X:=A;  
WHILE X<= B DO  
  BEGIN  
    СЕРИЯ;  
    X:=X+1;  
  END;
```

ШАГ=+1



ЦИКЛ  
«ДЛЯ»



На Паскале

FOR X:=A TO B DO

ОДИН  
 ОПЕРАТОР;  
DO

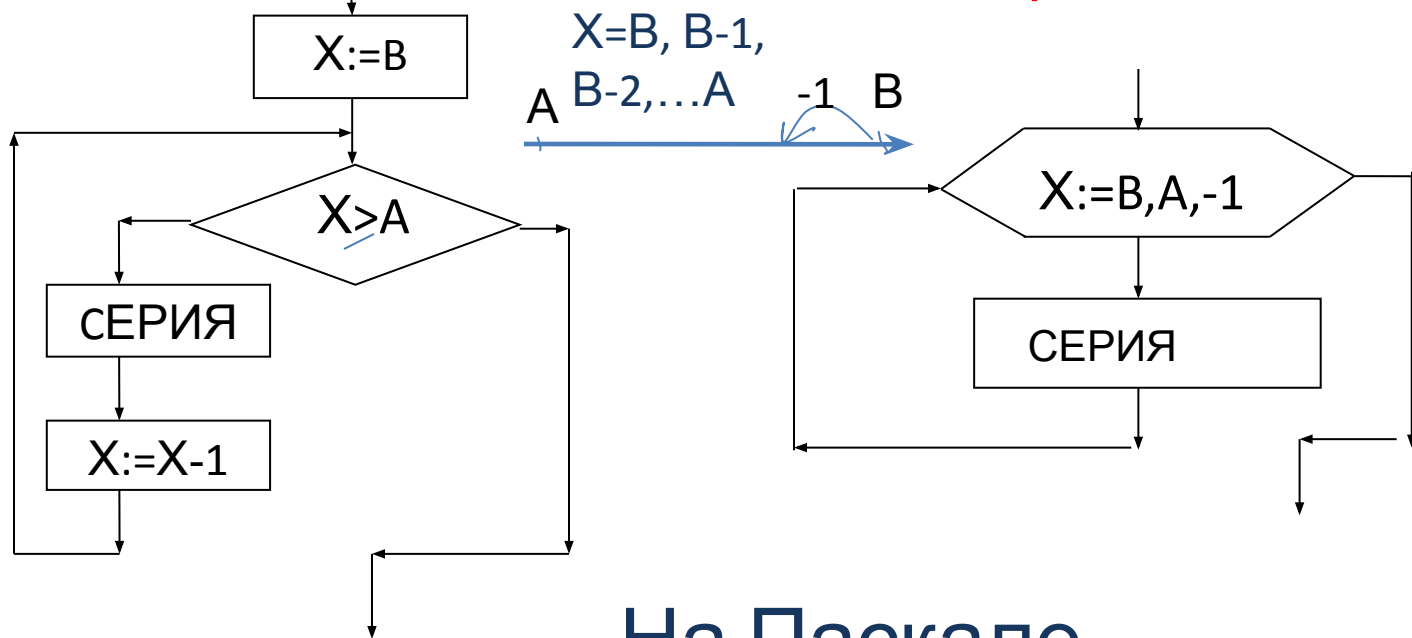
# ЦИКЛЫ С

## ПРЕДУСЛОВИЕМ

ЦИКЛ «ПОКА»

ШАГ = -1

ЦИКЛ «ДЛЯ»



На Паскале

```
X:=B;  
WHILE X>= A DO  
  BEGIN  
    СЕРИЯ;  
    X:=X-1;  
  END;
```

FOR X:=B DOWNT0 A DO

ОДИН ОПЕРАТОР;

# ЦИКЛ С ПОСТУСЛОВИЕМ

ЦИКЛ «ПОВТОРЯТЬ ДО ТЕХ ПОР ПОКА НЕ»



На Паскале

```
Repeat  
серия  
Until условие;
```

## Виды циклических алгоритмов

Цикл с  
предусловием



Цикл типа *Пока*

Цикл с  
постусловием



Цикл типа *до*

Цикл с  
параметром



Цикл типа