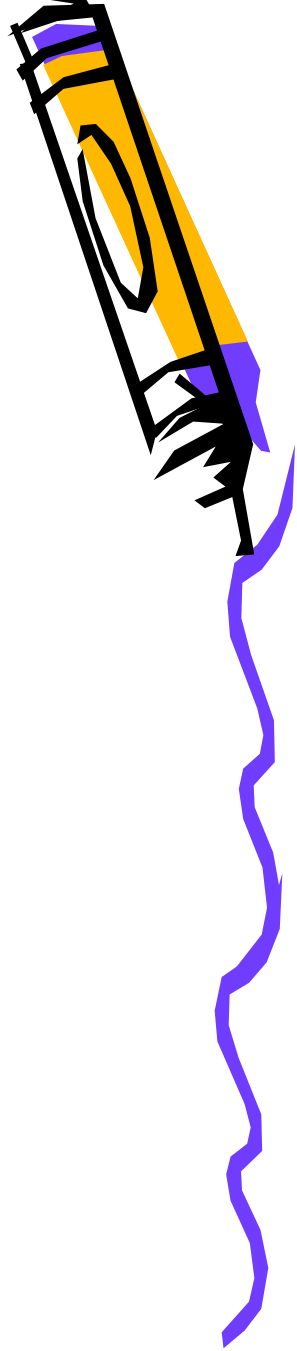


A large yellow diamond shape is centered on the page. A red line starts from the tip of a red and yellow marker at the top left and curves down to the top of the text. A blue line starts from the tip of a blue and yellow marker at the bottom right and curves up to the bottom of the text.

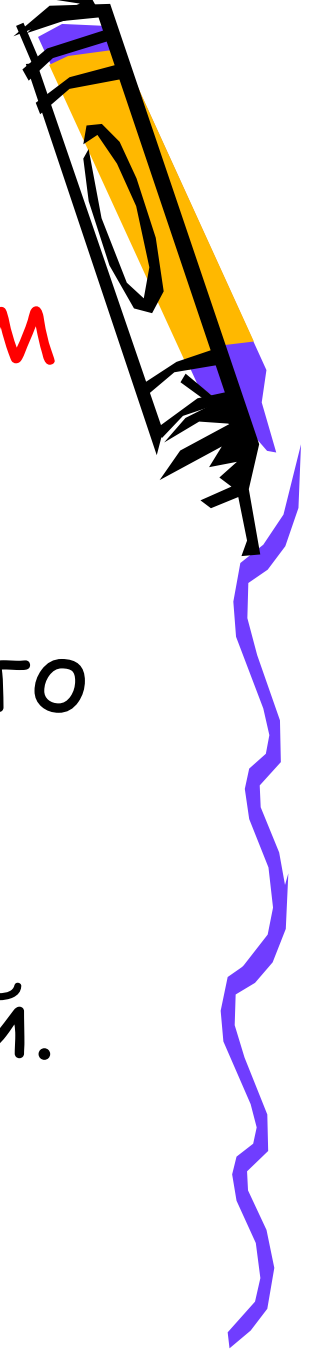
# Разветвляющиеся алгоритмы.

Кондрина А.В.  
учитель информатики и ИКТ

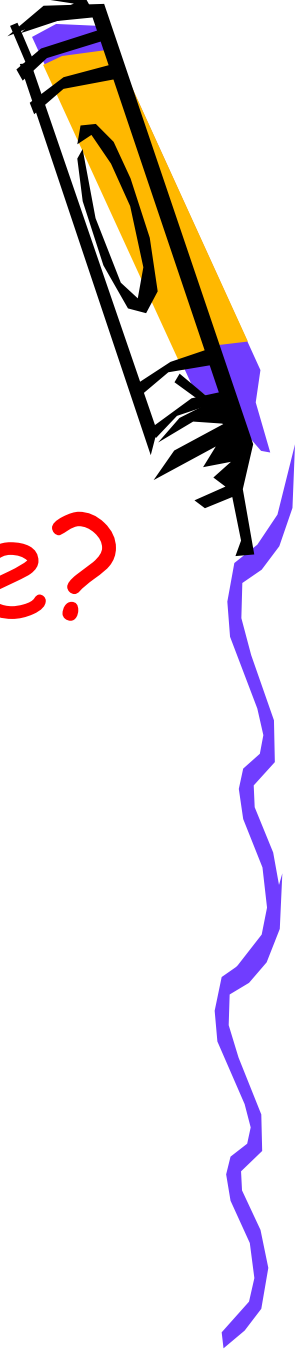
**Сформулируйте  
определение  
разветвляющегося  
алгоритма ?**



- **Разветвляющийся алгоритм**  
- алгоритм, в котором в зависимости от выполнения или не выполнения некоторого условия совершается либо одна, либо другая последовательность действий.

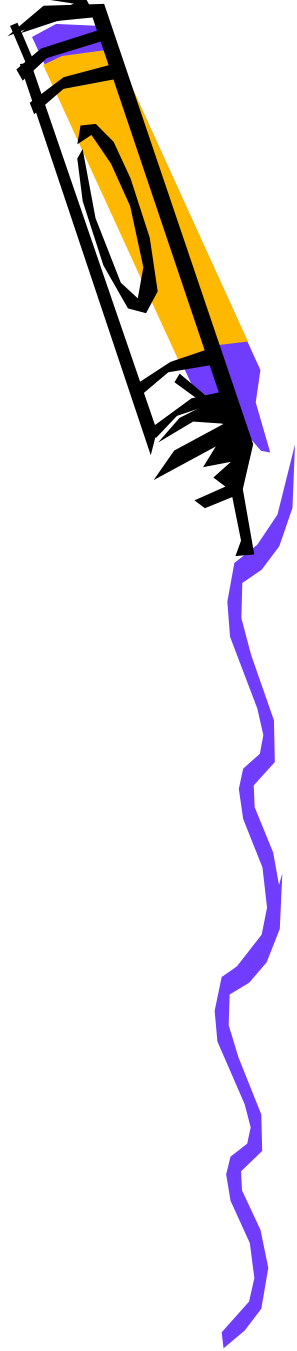


Какие формы  
ветвления вы знаете?

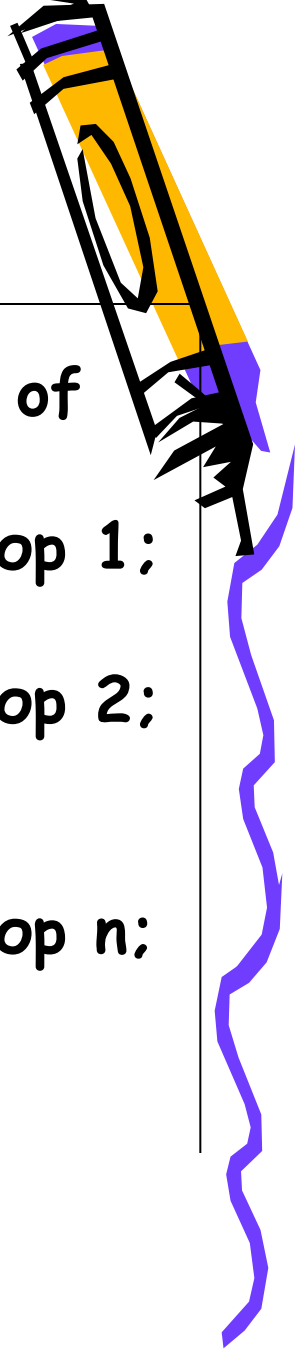


1) Полная;

2) Не полная.



# Как записывается полная(неполная) форма ветвления на языке Паскаль?



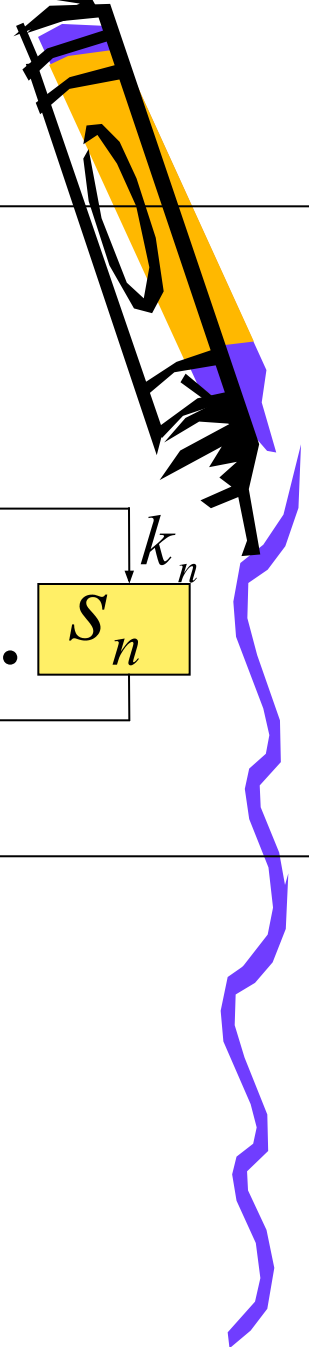
1) If <условие>  
then <серия команд 1 >  
else <серия команд 2 >;

2) If <условие>  
then <серия команд>;

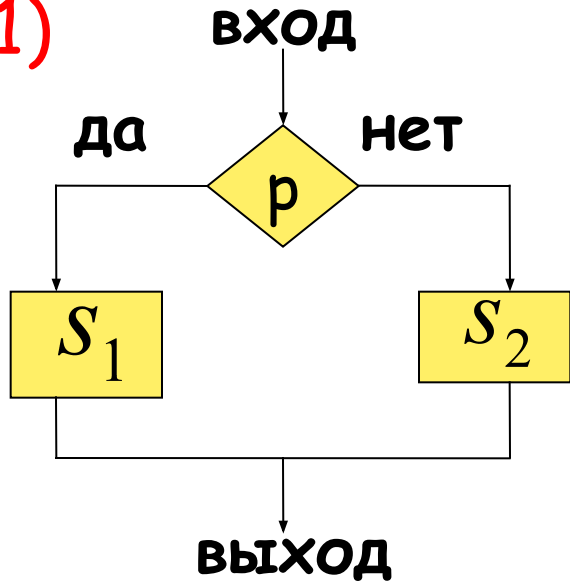
3) Case <выражение> of  
значение 1:  
выполняемый оператор 1;  
значение 2:  
выполняемый оператор 2;  
-----  
значение n:  
выполняемый оператор n;  
end;



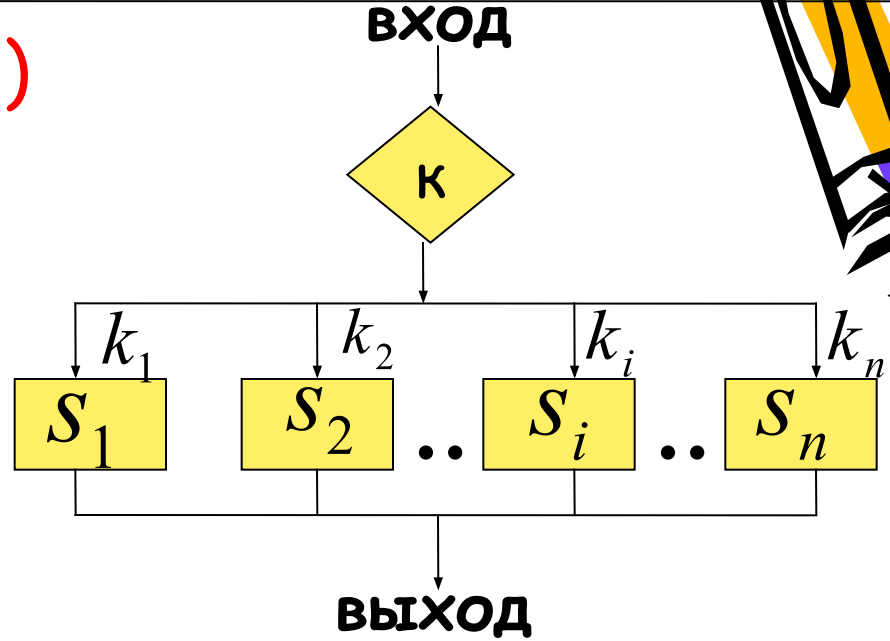
# Как выглядит блок-схема полного (неполного) ветвления?



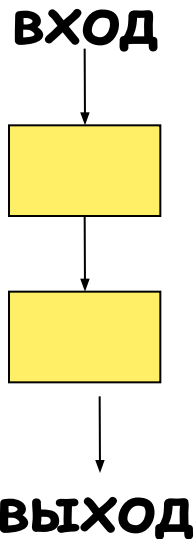
1)



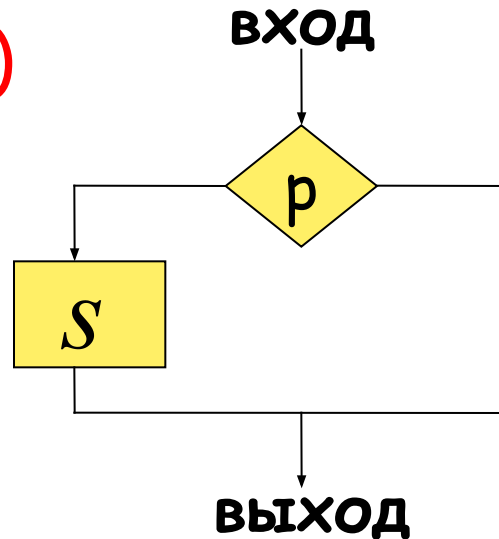
2)

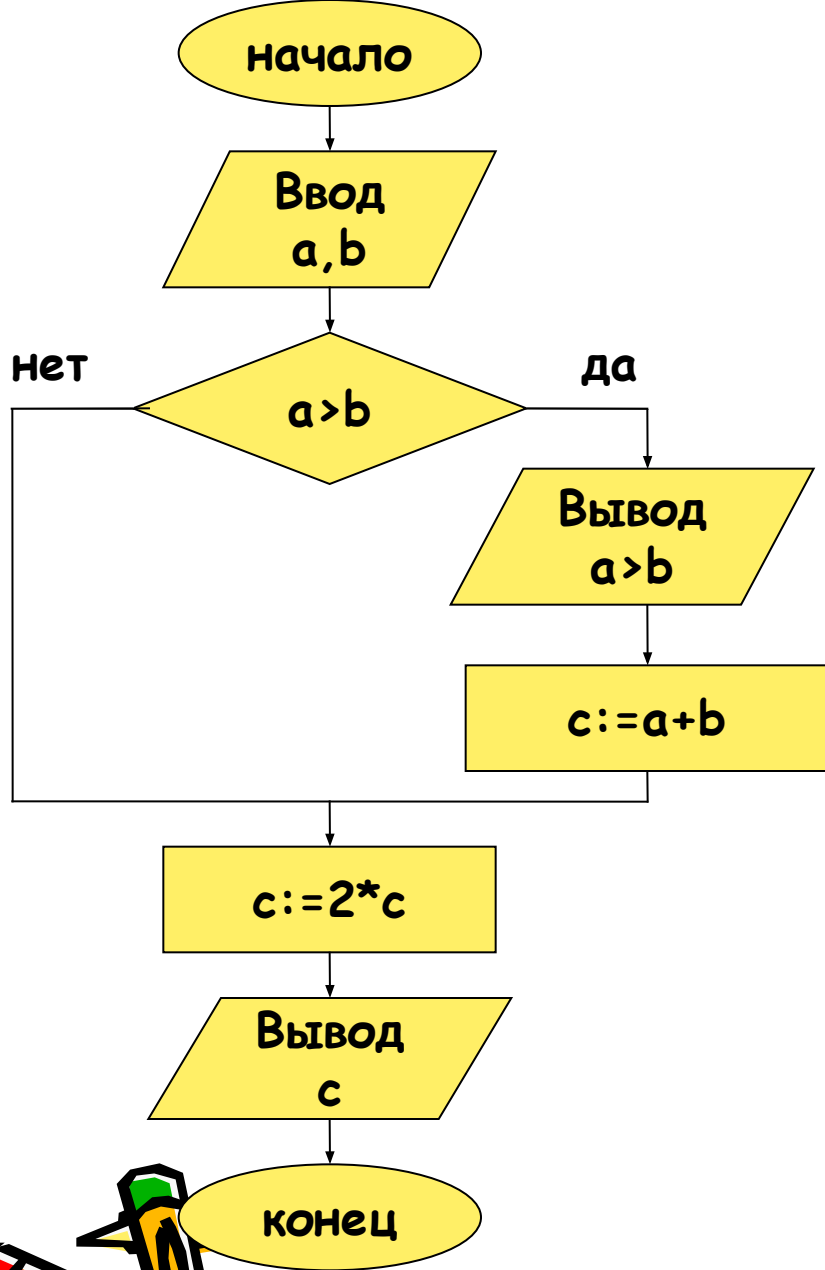


3)



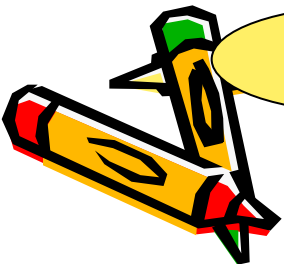
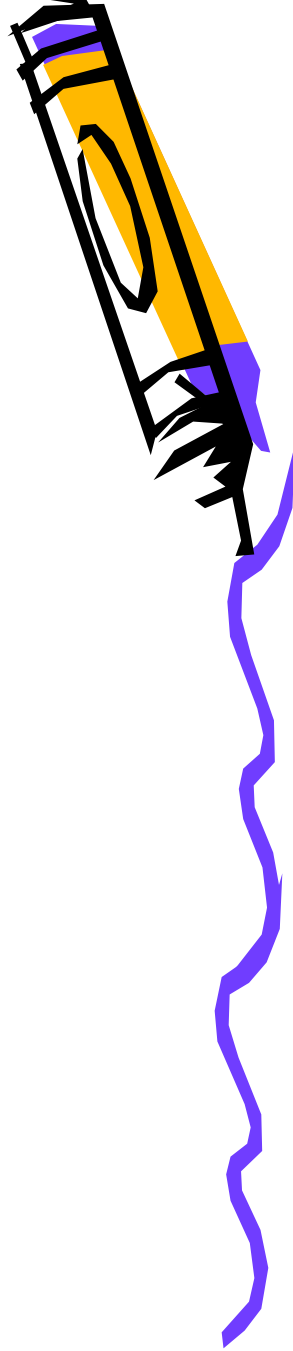
4)





Program sum;  
Var a, b, c:integer;  
Begin

  readln(a,b);  
  if a>b then





## Составной оператор

-это группа операторов,  
отделенных друг от друга точкой с  
запятой и ограниченная Begin...End.

### Пример:

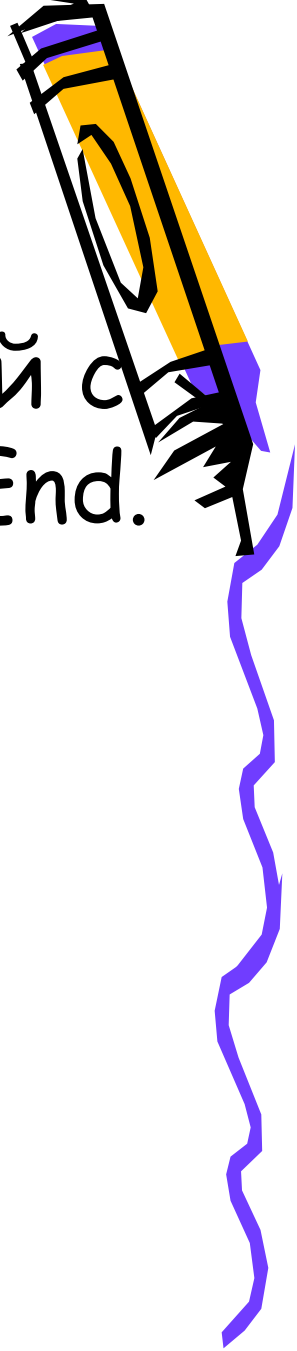
...

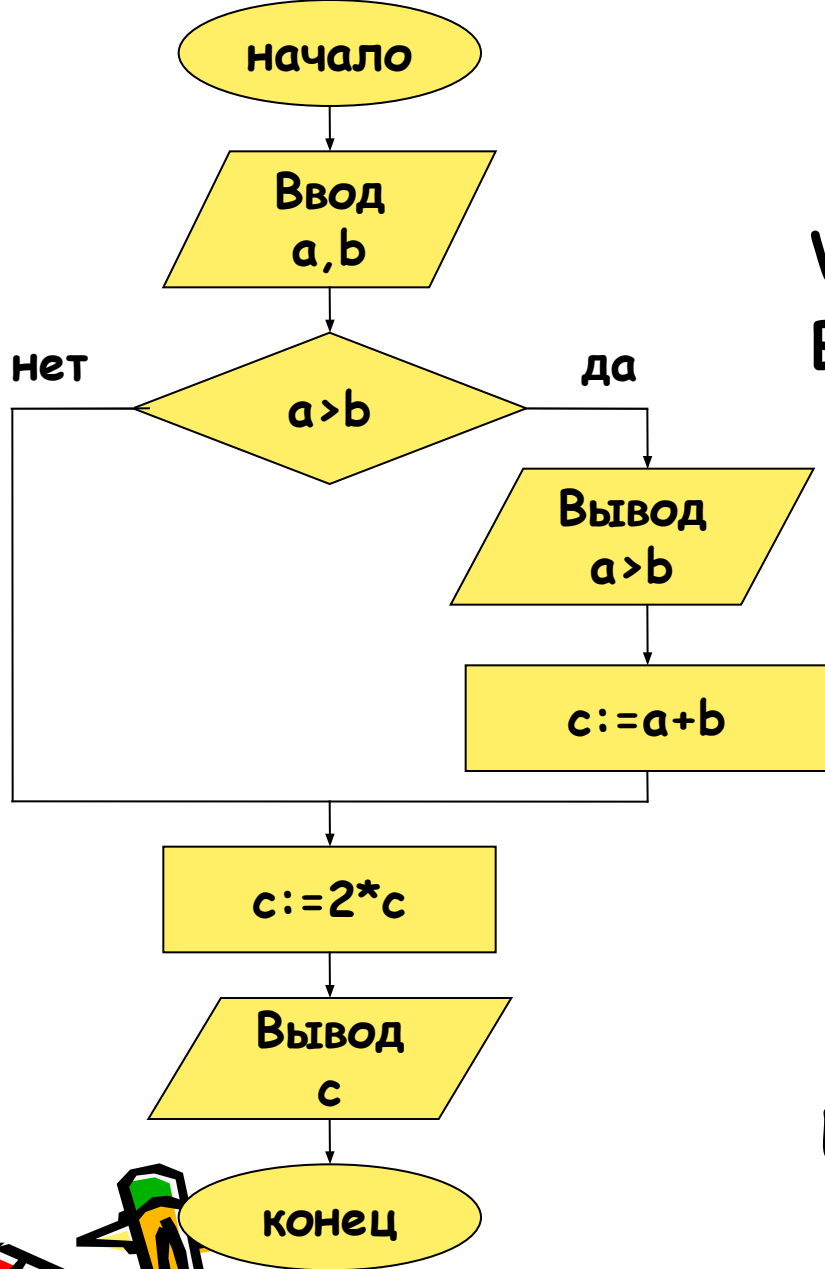
Begin {начало}

...

} {группа операторов}

End; {конец}





Program sum;  
Var a, b, c:integer;  
Begin

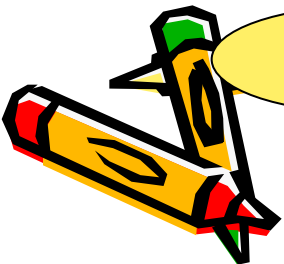
```
  readln(a,b);  
  if a > b then  
  begin
```

```
    writeln('a > b');  
    c := a + b;
```

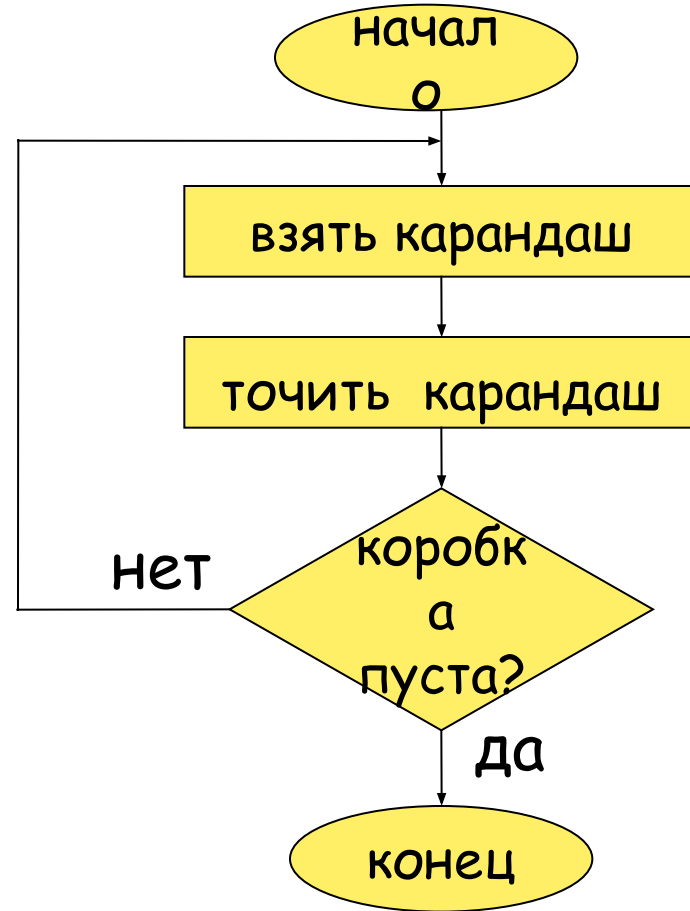
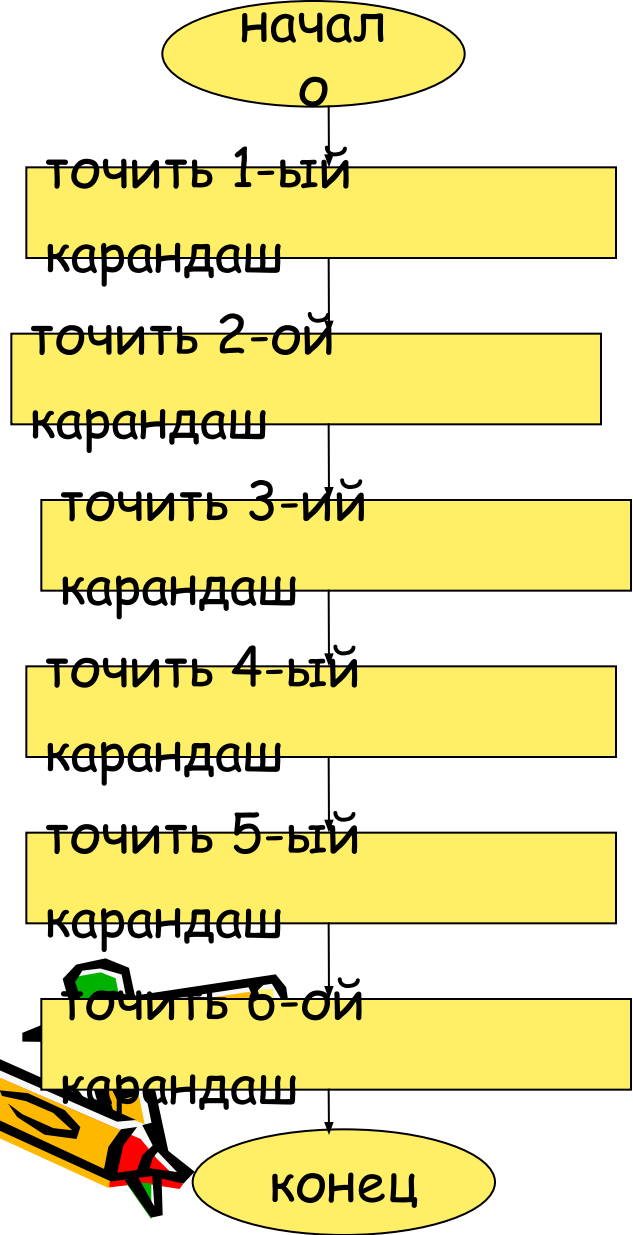
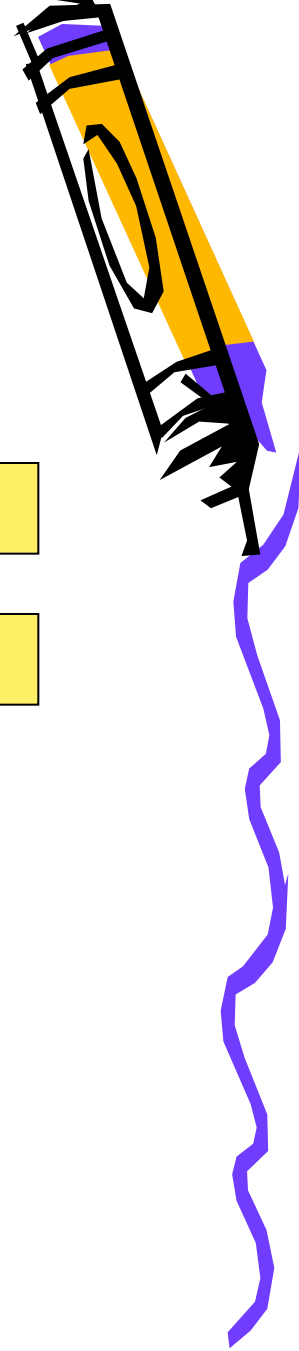
} группа операторов

```
  end;  
  c := 2 * c;  
  writeln(c);
```

End.

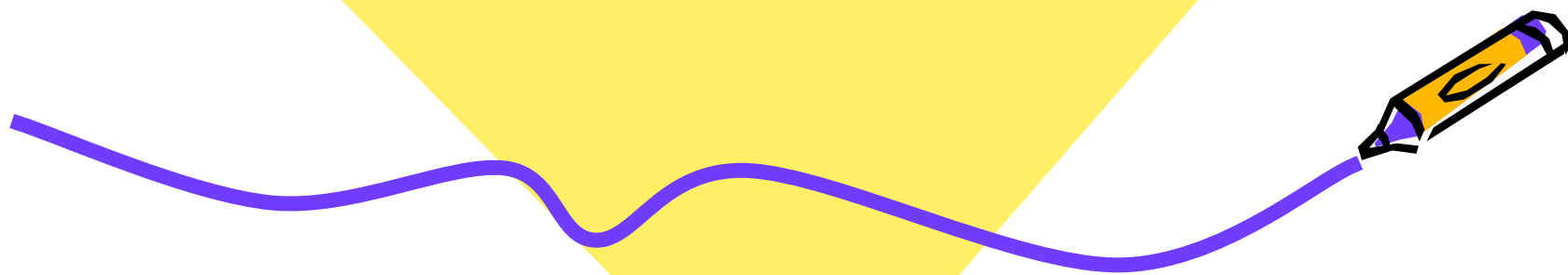


**Пример 1:** имеется коробка карандашей, надо поточить все карандаши в этой коробке.





# Циклические алгоритмы

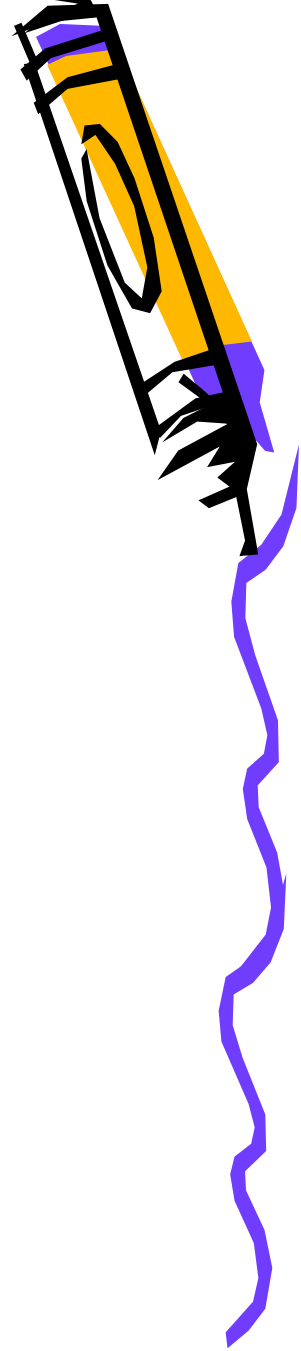


- Алгоритм, предусматривающий многократное повторение одного и того же действия над новыми данными, называется **циклическим**.

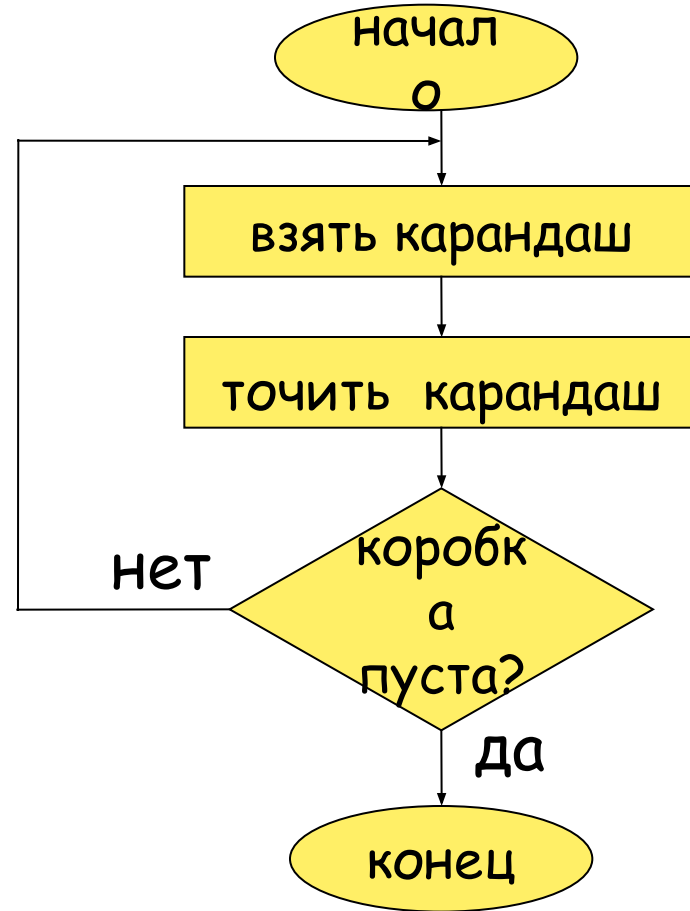
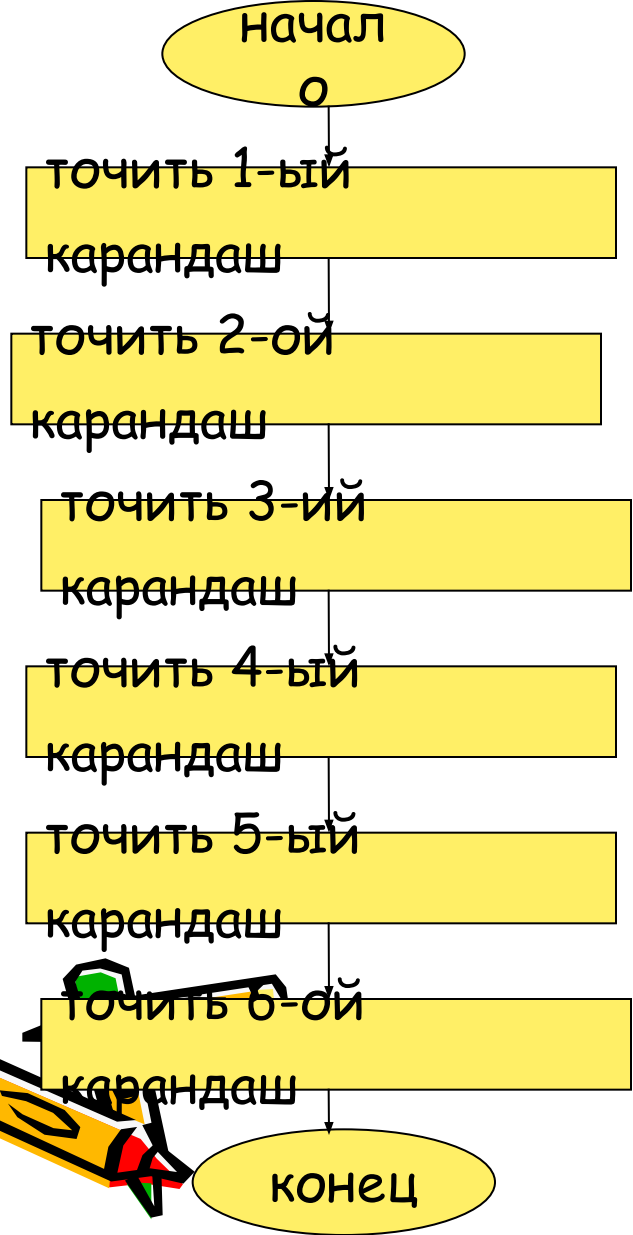
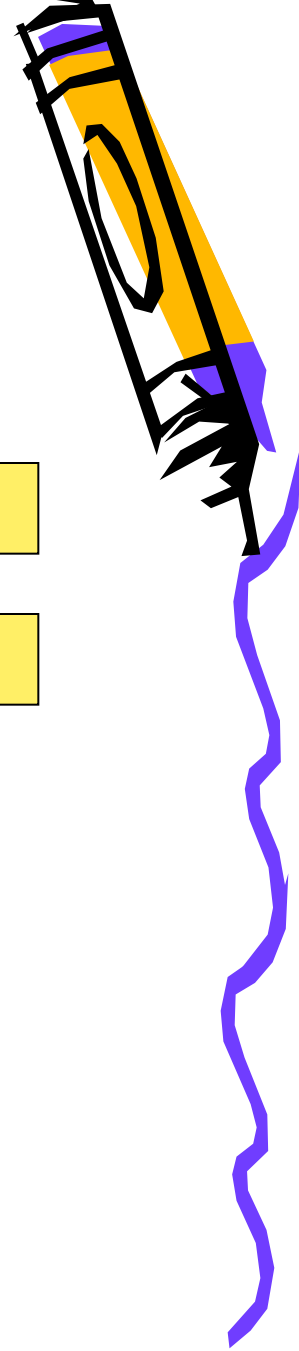


## Различают три вида циклов:

- Цикл с постусловием;
- Цикл с предусловием;
- Цикл с параметром.



**Пример 1:** имеется коробка карандашей, надо поточить все карандаши в этой коробке.

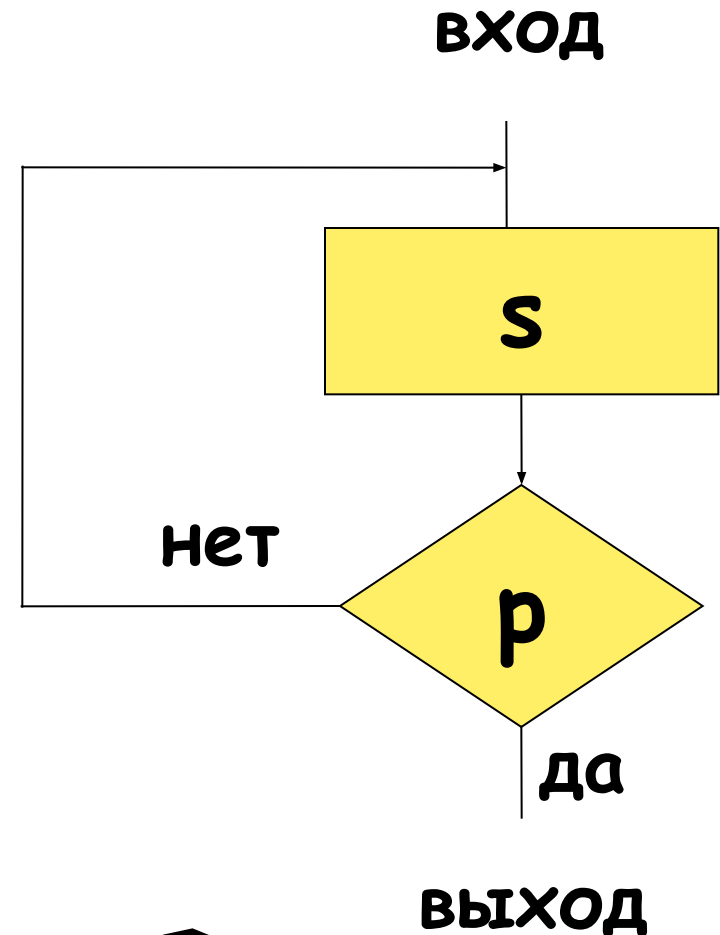


# Цикл с постусловием



## Особенности:

- 1) тело цикла расположено до проверки условия
- 2) цикл выполняется хотя бы 1 раз





# Цикл с постусловием ( цикл - до)

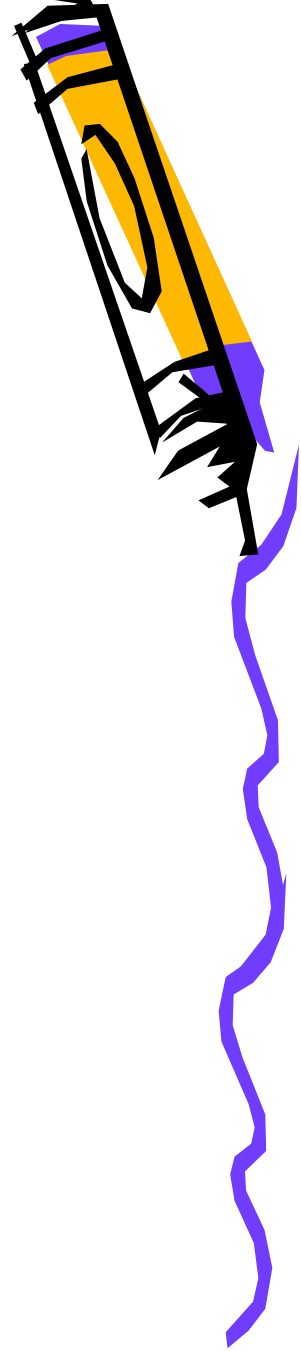
Repeat

<оператор 1>;

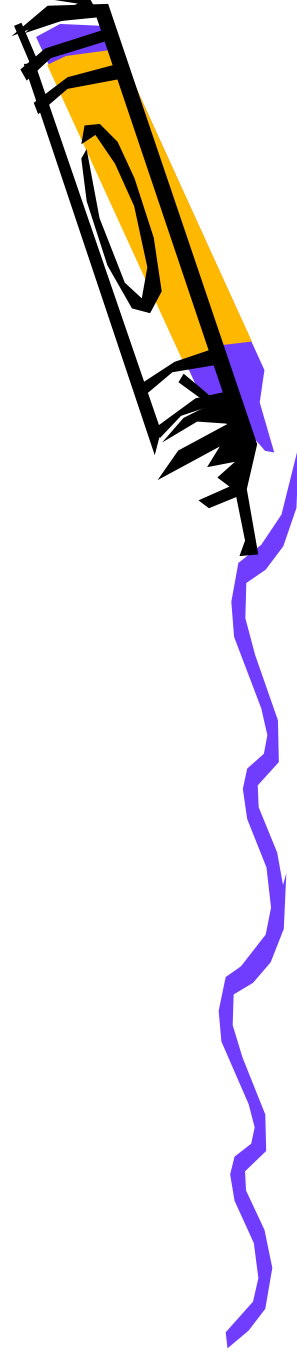
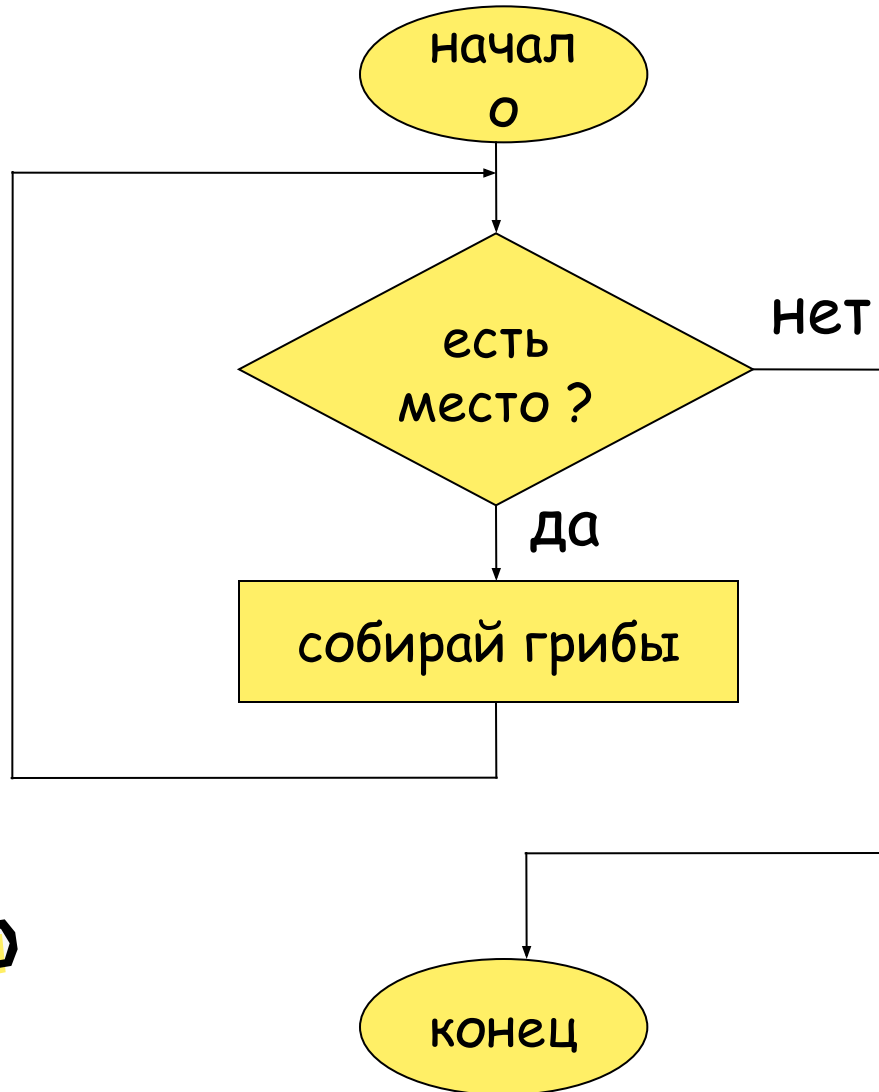
<оператор 2>;

.....

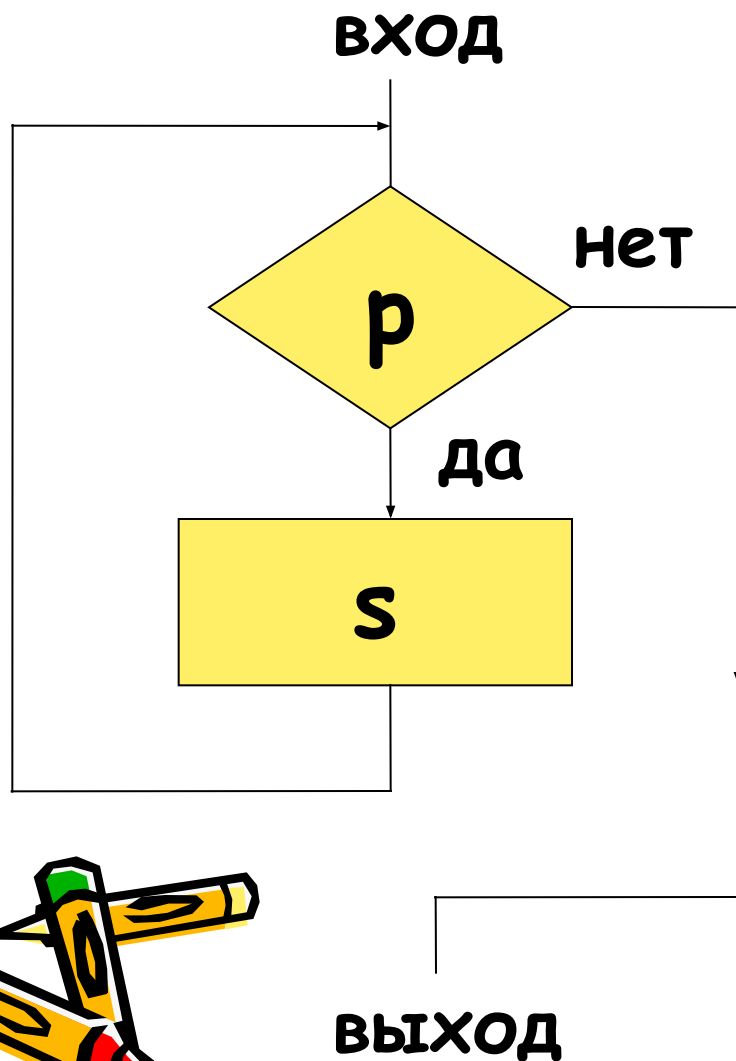
Until <условие>;



Пример 2: собирай грибы, пока в корзине есть место.



# Цикл с предусловием



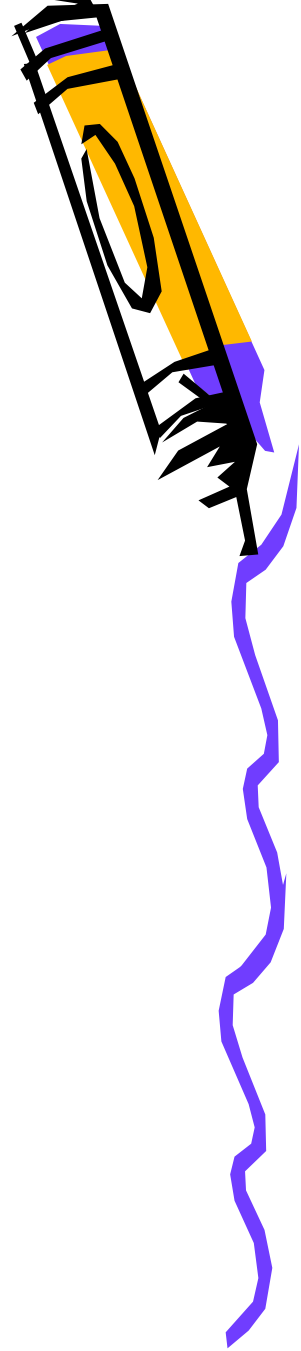
## Особенности:

- 1) тело цикла расположено после проверки условия
- 2) Цикл может ни разу не выполниться

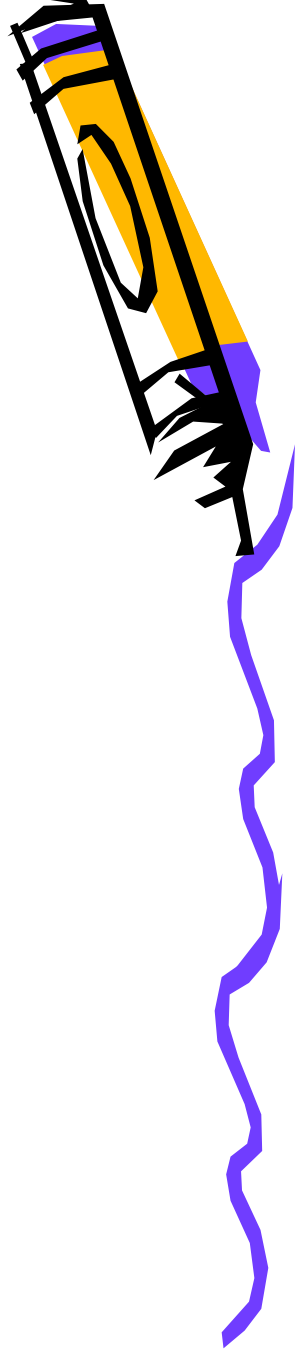
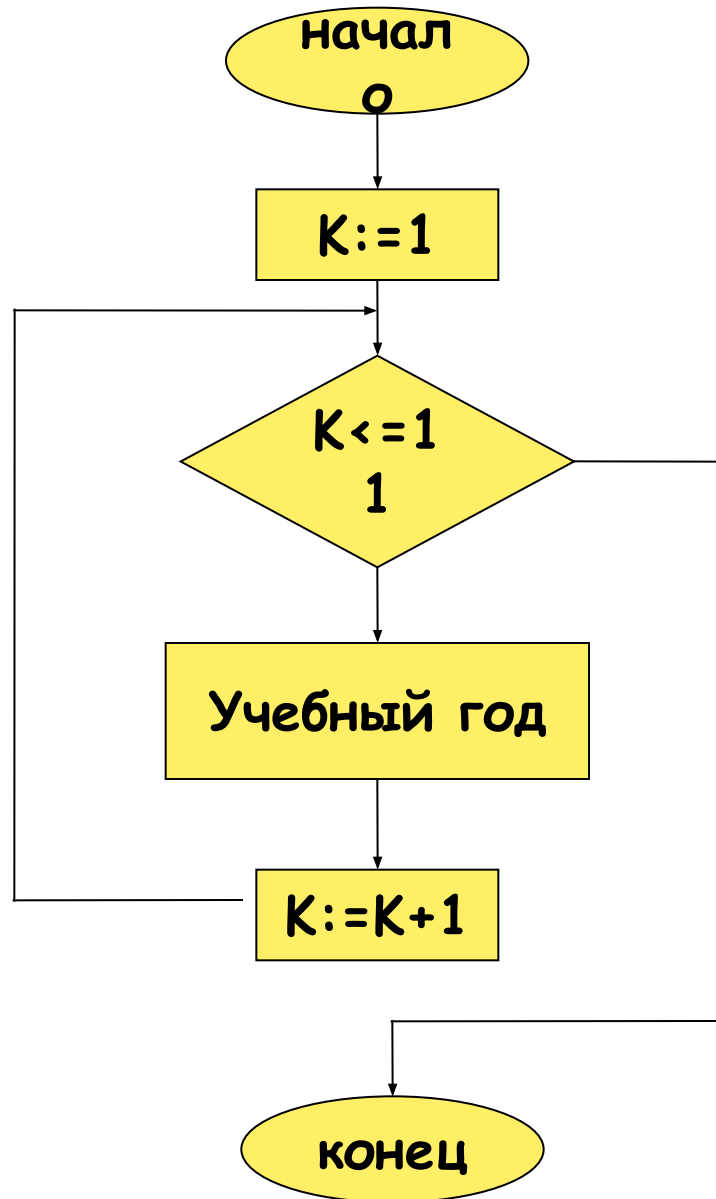


# Цикл с предусловием (цикл - пока)

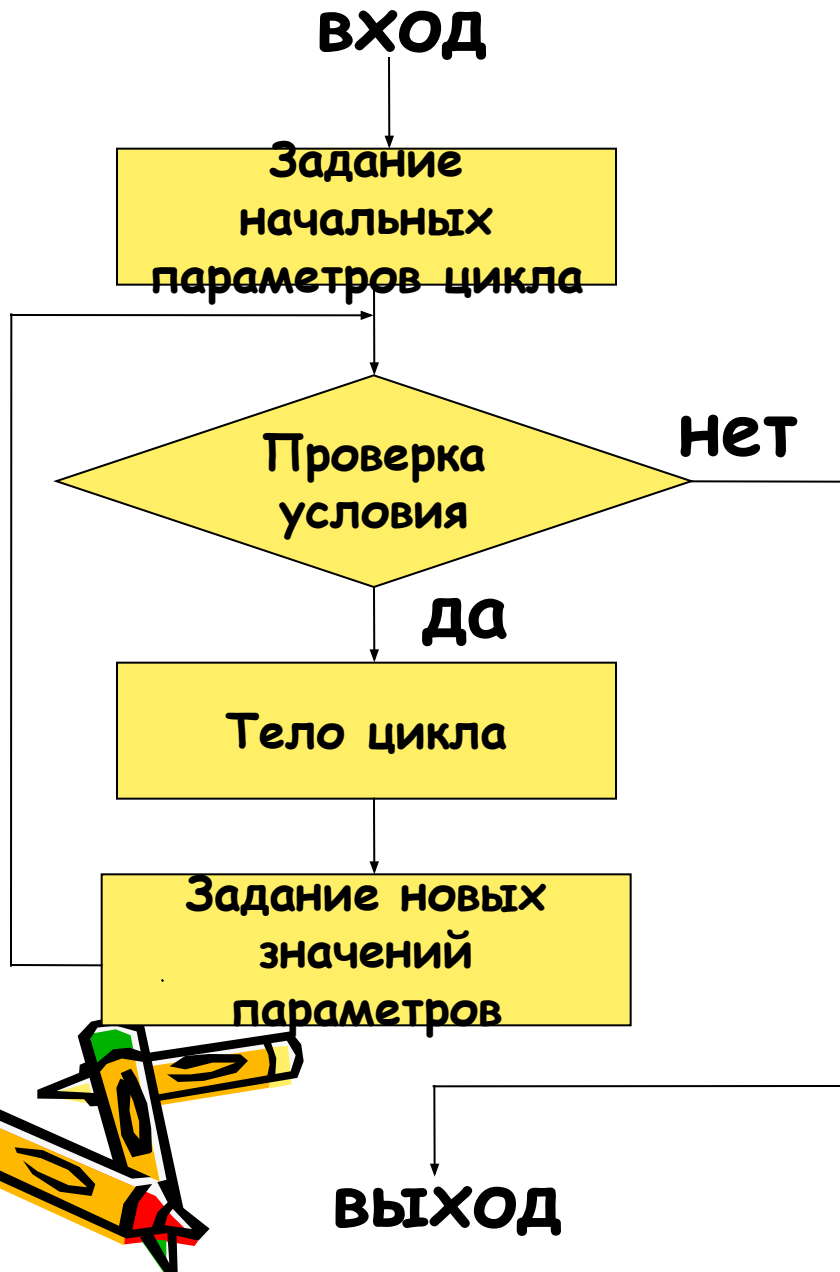
`While <условие> do <оператор>`



Пример 3: описать процесс обучения в школе.



# ЦИКЛ С ПАРАМЕТРОМ



## Особенности:

- 1) число повторений цикла известно заранее или может быть вычислено
- 2) параметр только целого типа (integer)



# ЦИКЛ С ПАРАМЕТРОМ

параметр

значение 1

значение 2

For  $i:=1$  **to**  $n$  do <оператор>;

**to** автоматически увеличивает значение параметра на 1

Если <значение 1> **>** <значения 2>, то

For  $i:=n$  **downto** 1 do <оператор>;

**downto** автоматически уменьшает значение параметра на 1

