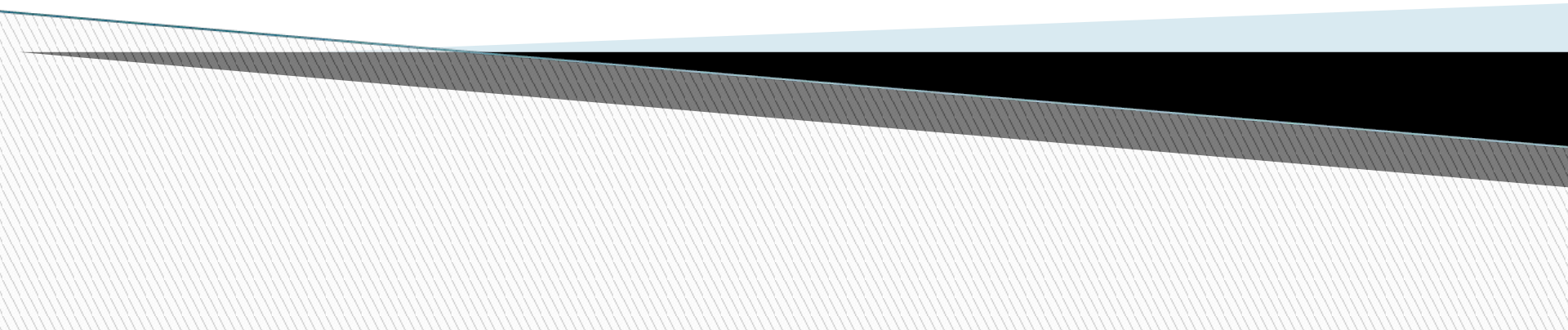


# Цикл For



- ▣ В большинстве задач, встречающихся на практике, необходимо производить многократное выполнение некоторого действия.
- ▣ **Циклический алгоритм** – алгоритм, при котором выполняется одинаковая последовательность действий.
  - цикл с **известным** числом шагов (арифметический)
  - цикл с **неизвестным** числом шагов (цикл с условием)- *итерационный*

**Задача.** Вывести на экран 5 раз слово «Привет».

---

```
program qq;  
begin  
  writeln ( ' Привет ' ) ;  
  writeln ( ' Привет ' ) ;  
  writeln ( ' Привет ' ) ;  
  writeln ( ' Привет ' ) ;  
  writeln ( ' Привет ' ) ;  
end.
```



Что плохо?

# Циклы

```
program qq;
begin
```

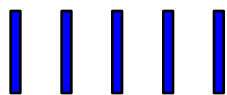
```
  { сделай 5 раз }
```

```
    writeln ( ' Привет ' ) ;
```

```
end.
```

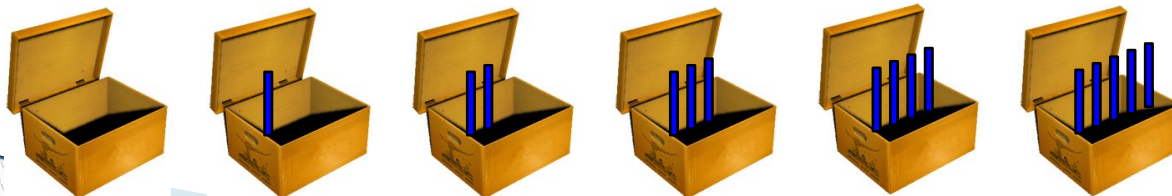


Как отсчитать ровно 5 раз?



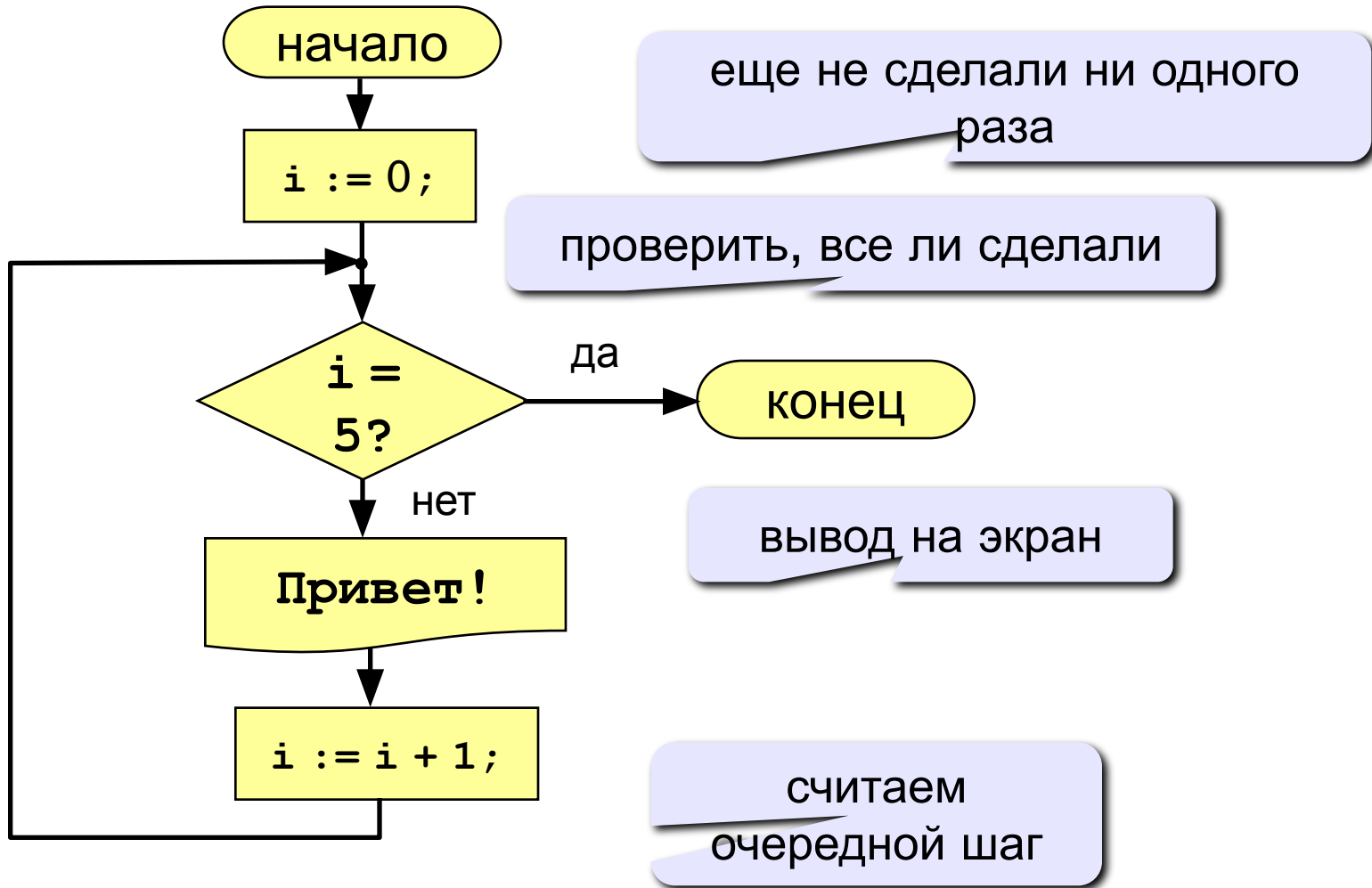
Как запоминать, сколько раз

уже сделали?



```
i := i + 1;
```

# Алгоритм



# Цикл с переменной

---

## Увеличение переменной на 1:

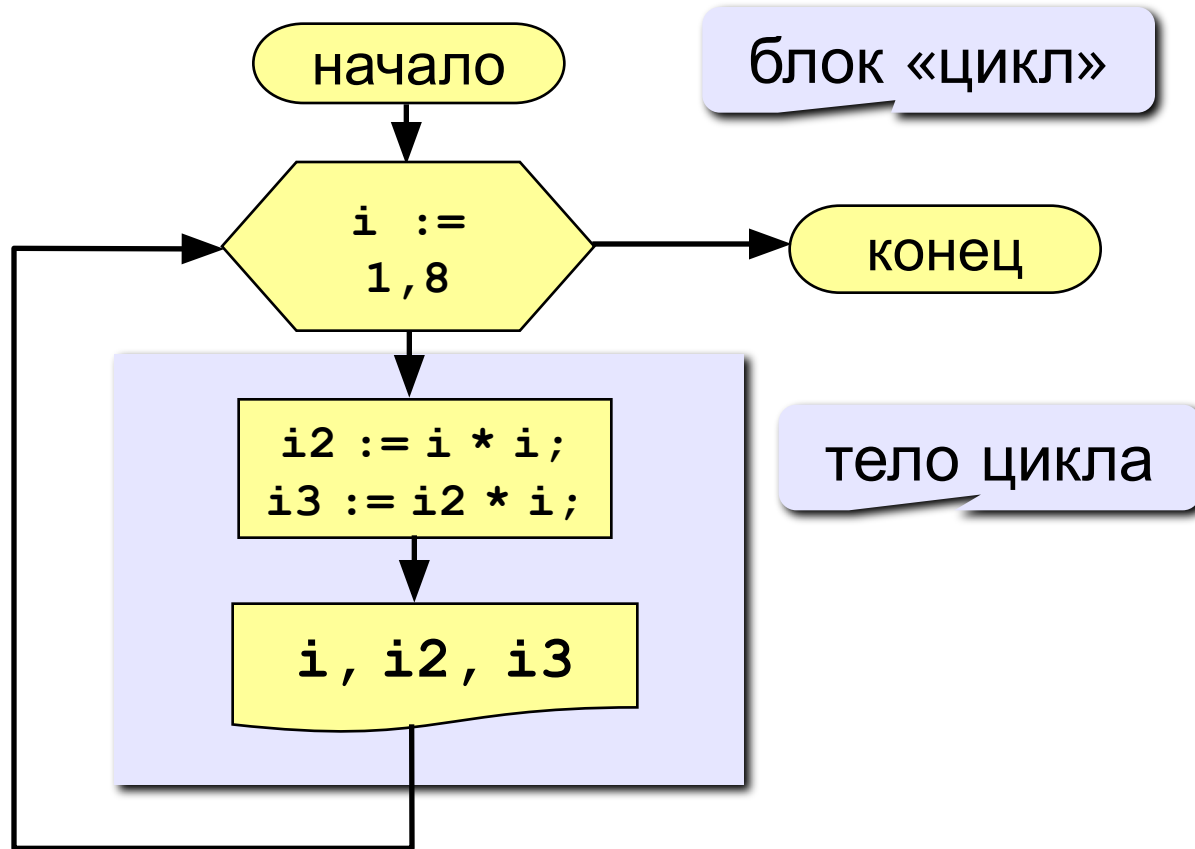
```
for <переменная> := <начальное значение> to <конечное значение>
do begin
    {тело цикла}
end;
```

Цикл for вызывает *операторы*, находящиеся в теле цикла, по одному разу для каждого значения в диапазоне от *начального значения* до *конечного значения*.

## Уменьшение переменной на 1:

```
for <переменная> := <начальное значение> downto <конечное значение>
do begin
    {тело цикла}
end;
```

# Алгоритм (с блоком «цикл»)



# Цикл с переменной

---

## Особенности:

- переменная цикла может быть только целой (**integer**)
- шаг изменения переменной цикла всегда равен 1 (**to**) или -1 (**downto**)
- если в теле цикла только один оператор, слова **begin** и **end** можно не писать:

```
for i:=1 to 8 do  
    writeln('Привет');
```

- если конечное значение меньше начального, цикл (**to**) не выполняется ни разу (проверка условия в начале цикла, цикл с предусловием)



# Цикл с переменной

---

## Особенности:

- в теле цикла не разрешается изменять переменную цикла (почему?)
- при изменении начального и конечного значения внутри цикла количество шагов не изменится:

```
n := 8;  
for i:=1 to n do begin  
  writeln('Привет');  
  n := n + 1;  
end;
```

нет  
зацикливания

# Цикл с переменной

## Особенности:

- после выполнения цикла **во многих системах** устанавливается первое значение переменной цикла, при котором нарушено условие:

```
for i:=1 to 8  
  writeln('Привет');  
  writeln('i=', i);
```

i=9

**НЕ ДОКУМЕНТИРОВАНО**

```
for i:=8 downto 1 do  
  writeln('Привет');  
  writeln('i=', i);
```

i=0

# Сколько раз выполняется цикл?

```
a := 1;  
for i:=1 to 3 do a := a+1;
```

~~a = 4~~

```
a := 1;  
for i:=3 to 1 do a := a+1;
```

~~a = 1~~

```
a := 1;  
for i:=1 downto 3 do a := a+1;
```

~~a = 1~~

```
a := 1;  
for i:=3 downto 1 do a := a+1;
```

~~a = 4~~