

Программирование на языке Паскаль

Тема 1. Введение

Алгоритм

Алгоритм – это четко определенный план действий для исполнителя.

Свойства алгоритма

- **дискретность**: состоит из отдельных шагов (команд)
- **понятность**: должен включать только команды, известные исполнителю (входящие в СКИ)
- **определенность**: при одинаковых исходных данных всегда выдает один и тот же результат
- **конечность**: заканчивается за конечное число шагов
- **массовость**: может применяться многократно при различных исходных данных
- **корректность**: дает верное решение при любых допустимых исходных данных

Программа

Программа – это

- алгоритм, записанный на каком-либо языке программирования
- набор команд для компьютера

Команда – это описание действий, которые должен выполнить компьютер.

- откуда взять исходные данные?
- что нужно с ними сделать?

Оператор – это команда языка программирования высокого уровня.

1970 – язык Паскаль (Н. Вирт)

Простейшая программа

название программы

```
program qq;  
begin { начало программы }  
end.  { конец программы }
```

комментарии в фигурных скобках
не обрабатываются



Что делает эта программа?

Вывод текста на экран

```
program qq;  
begin  
▶ write('2+');  
▶ writeln('2=?'); { на новую строку}  
▶ writeln('Ответ: 4');  
end.
```

Протокол:

2+

Ответ: 4

Задания

«4»: Вывести на экран текст "лесенкой"

Вася

пошел

гулять

«5»: Вывести на экран рисунок из букв

```
Ж
ЖЖЖ
ЖЖЖЖЖ
ЖЖЖЖЖЖЖ
НН НН
ZZZZZ
```

Переменные

Задача. Ввести с клавиатуры два числа и найти их сумму.

Протокол:

Введите два целых числа

компьютер

25 30

пользователь

25+30=55

компьютер считает сам!

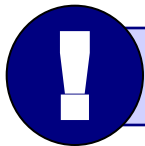


1. Как ввести числа в память?
2. Где хранить введенные числа?
3. Как вычислить?
4. Как вывести результат?

Программа

```
program qq;  
begin  
  { ввести два числа }  
  { вычислить их сумму }  
  { вывести сумму на экран }  
end.
```

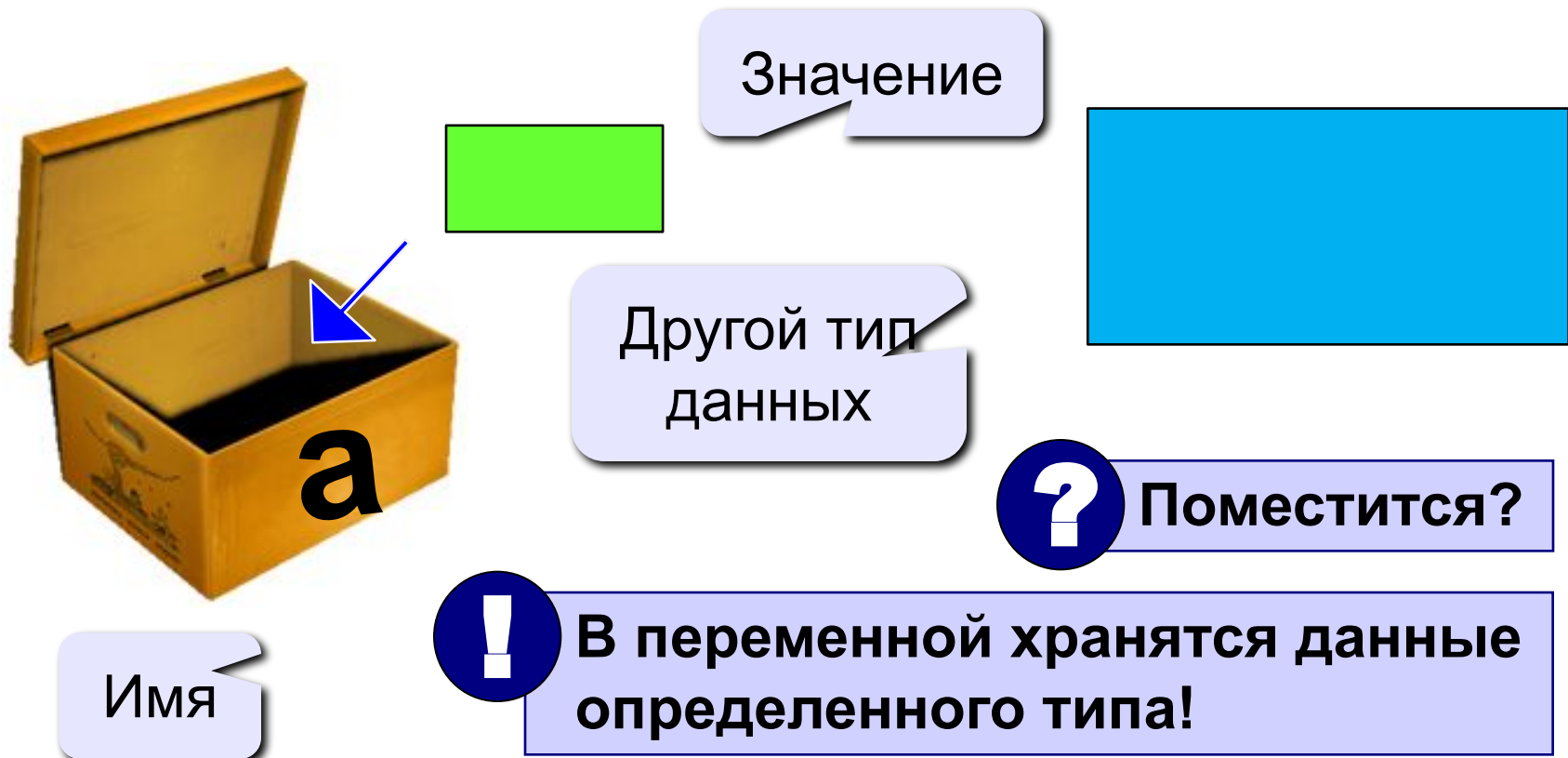
Псевдокод: алгоритм на русском языке с элементами Паскаля.



Компьютер не может исполнить псевдокод!

Переменные

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.



Имена переменных

В именах **МОЖНО** использовать

- латинские буквы (A-Z)

заглавные и строчные буквы не различаются

- цифры

имя не может начинаться с цифры

- знак подчеркивания _

В именах **НЕЛЬЗЯ** использовать

- русские буквы
- пробелы
- скобки, знаки +, =, !, ? и др.

Какие имена правильные??

AXby R&B 4Wheel Вася “PesBarbos”

TU154 [QuQu] _ABBA A+B

Переменные

Типы переменных:

- integer { целая }
- real { вещественная }
- и другие...

Выделение
места в памяти

Объявление переменных:

variable – переменная

тип – целые

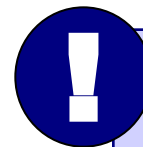
```
var a, b, c: integer;
```

СПИСОК ИМЕН
переменных

Как записать значение в переменную?

Оператор
присваивания

```
a := 5;
```

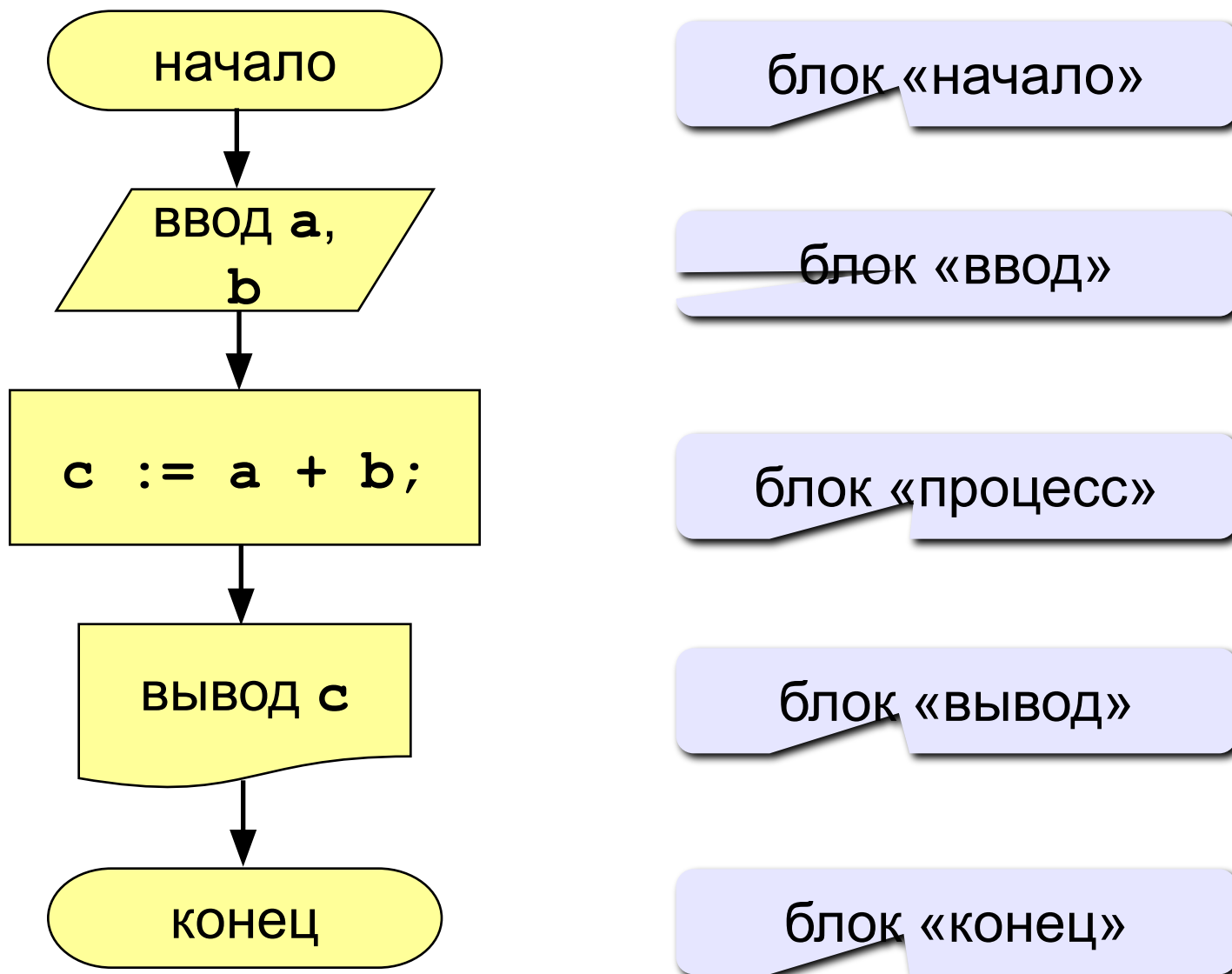


При записи нового
значения старое
стирается!

Оператор – это команда языка программирования (инструкция).

Оператор присваивания – это команда для записи нового значения в переменную.

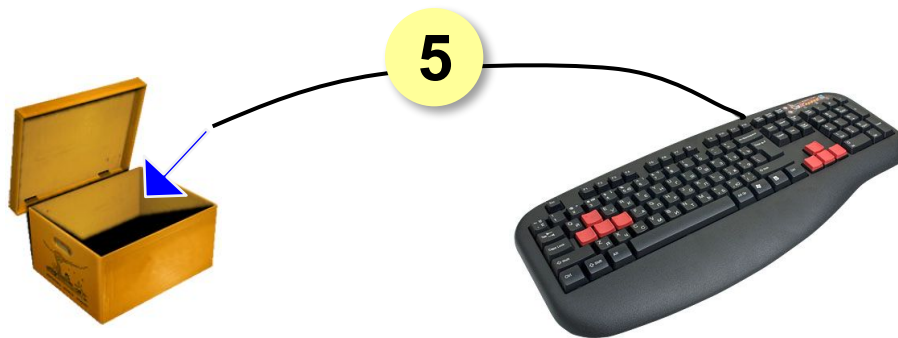
Блок-схема линейного алгоритма



Как ввести значение с клавиатуры

Оператор
ввода

```
read ( a );
```



1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную *a*.

Ввод значений двух переменных

```
read ( a, b );
```

Ввод значений двух переменных (через пробел или *Enter*).

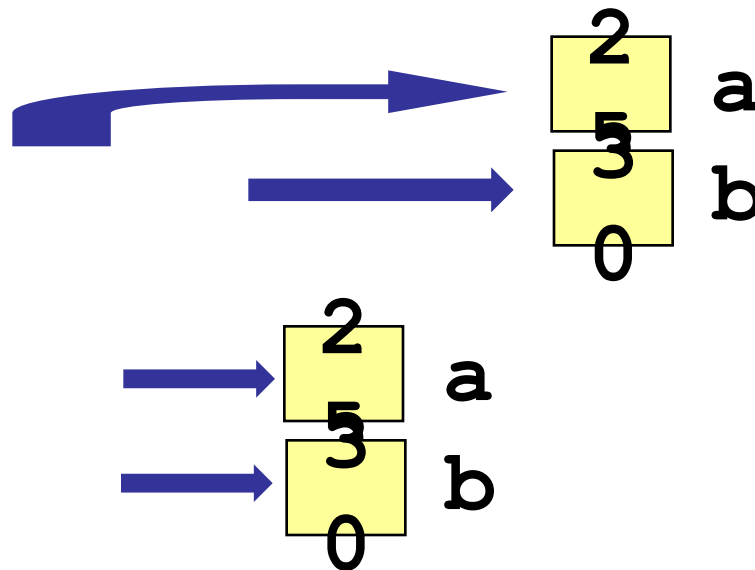
через пробел:

25 30

через *Enter*:

25

30



Оператор вывода

`write(a);` { вывод значения
переменной a }

`writeln(a);` { вывод значения
переменной a и **переход
на новую строку** }

`writeln('Привет!');` { вывод текста }

`writeln('Ответ: ', c);`

{вывод текста и значения переменной c}

`writeln (a, '+', b, '=', c);`

Сложение двух чисел

Задача. Ввести два целых числа и вывести на экран их сумму.

Простейшее решение:

```
program qq;  
var a, b, c:  
integer;  
begin  
  read ( a, b );  
  c := a + b;  
  writeln ( c );  
end.
```



Что плохо?

Полное решение

```
program qq;  
var a, b, c: integer;  
begin  
  writeln('Введите два целых числа');  
  read ( a, b );  
  c := a + b;  
  writeln ( a, '+', b, '=', c );  
end.
```

Протокол:

КОМПЬЮТЕР

Введите два целых числа

25 30

ПОЛЬЗОВАТЕЛЬ

25+30=55

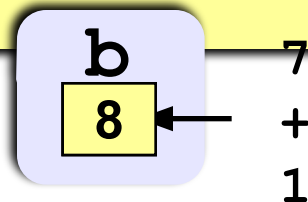
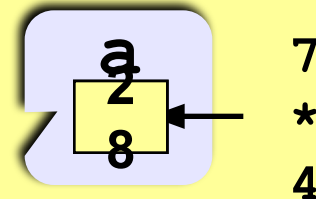
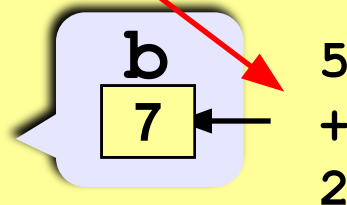
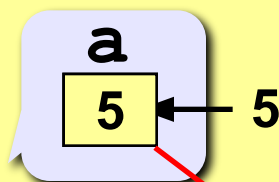
Как изменить значение переменной?

Пример:

```

program qq;
var a, b: integer;
begin
  a := 5;
  b := a + 2;
  a := (a + 2) * (b - 3);
  b := b + 1;
end.

```



Арифметические операции

+ сложение **-** вычитание

***** умножение **/** деление

div деление нацело (остаток отбрасывается)

mod остаток от деления

```
var a, b: integer;  
begin  
    a := 7*3 - 4;  
    a := a * 5;  
    b := a div 10;  
    a := a mod 10;  
end.
```

Какие операторы неправильные?

```
program qq;  
var a, b: integer;  
    x, y: real;  
begin  
    a := 5;  
    10 := x;  
    y := 7,8;  
    b := 2.5;  
    x := 2*(a + y);  
    a := b + x;  
end.
```

имя переменной должно
быть слева от знака :=

целая и дробная часть
отделяются **точкой**

нельзя записывать
вещественное значение в
целую переменную

Порядок выполнения операций

- 1) вычисление выражений в скобках
- 2) умножение, деление, **div**, **mod** слева направо
- 3) сложение и вычитание слева направо

1 2 4 5 3 6

```
z := (5*a+c) / a * (b-c) / b;
```

$$x = \frac{5c^2 - d(a+b)}{(c+d)(d-2a)}$$

$$z = \frac{5a+c}{ab} (b-c)$$

2 3 5 4 1 10 6 9 8 7

```
x := (5*c*c - d*(a+b)) / ((c+d) * (d-2*a));
```


Вывод целых чисел

```
program qq;  
var a, b: integer;  
begin  
  a := 15;  
  b := 45;  
  writeln ( a, b );  
  writeln ( a:4, b:4 );  
end.
```

1545

15 45

СИМВОЛОВ
на ЧИСЛО

Вывод вещественных чисел

```

program qq;
var x: real;
begin
  x := 12.345678;
  writeln ( x );
  writeln ( x:10 );
  writeln ( x:7:2 );
end.

```

ВСЕГО
СИМВОЛОВ

1,234568·10¹

1.234568E+001

1.23E+001

12.35

ВСЕГО
СИМВОЛОВ

В дробной
части

Задания

«4»: Ввести три числа, найти их сумму и произведение.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

«5»: Ввести три числа, найти их сумму, произведение и среднее арифметическое.

Пример:

Введите три числа:

4 5 7

$$4+5+7=16$$

$$4*5*7=140$$

$$(4+5+7) / 3=5.33$$

Программирование на языке Паскаль

Тема 2. Ветвления

Разветвляющиеся алгоритмы

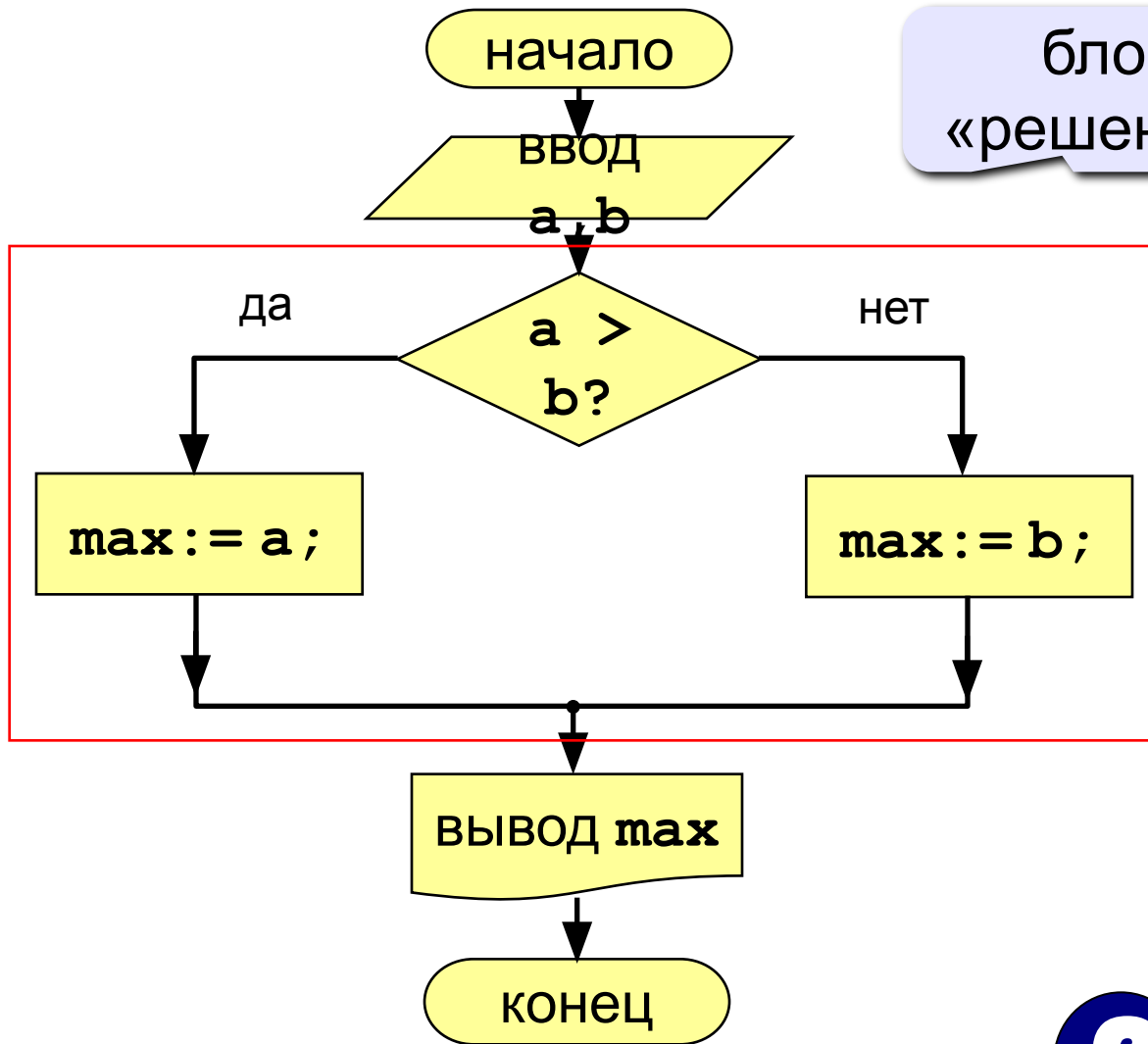
Задача. Ввести два целых числа и вывести на экран наибольшее из них.

Идея решения: надо вывести на экран первое число, если оно больше второго, или второе, если оно больше первого.

Особенность: действия исполнителя зависят от некоторых условий (*если ... иначе ...*).

Алгоритмы, в которых последовательность шагов зависит от выполнения некоторых условий, называются **разветвляющимися.**

Вариант 1. Блок-схема



блок
«решение»

полная
форма
ветвления



Если $a = b$?

Вариант 1. Программа

```
program qq;  
var a, b, max: integer;  
begin  
  writeln('Введите два целых числа');  
  read ( a, b );  
  if a > b then begin  
    max := a;  
  end  
  else begin  
    max := b;  
  end;  
  writeln ('Наибольшее число ', max);  
end.
```

полная форма
условного
оператора

Условный оператор

```
if <условие> then begin
    {что делать, если условие верно}
end
else begin
    {что делать, если условие неверно}
end;
```

Особенности:

- перед ***else*** **НЕ** ставится точка с запятой
- вторая часть (***else*** ...) может отсутствовать (неполная форма)
- если в блоке один оператор, можно убрать слова ***begin*** и ***end***

Что неправильно?

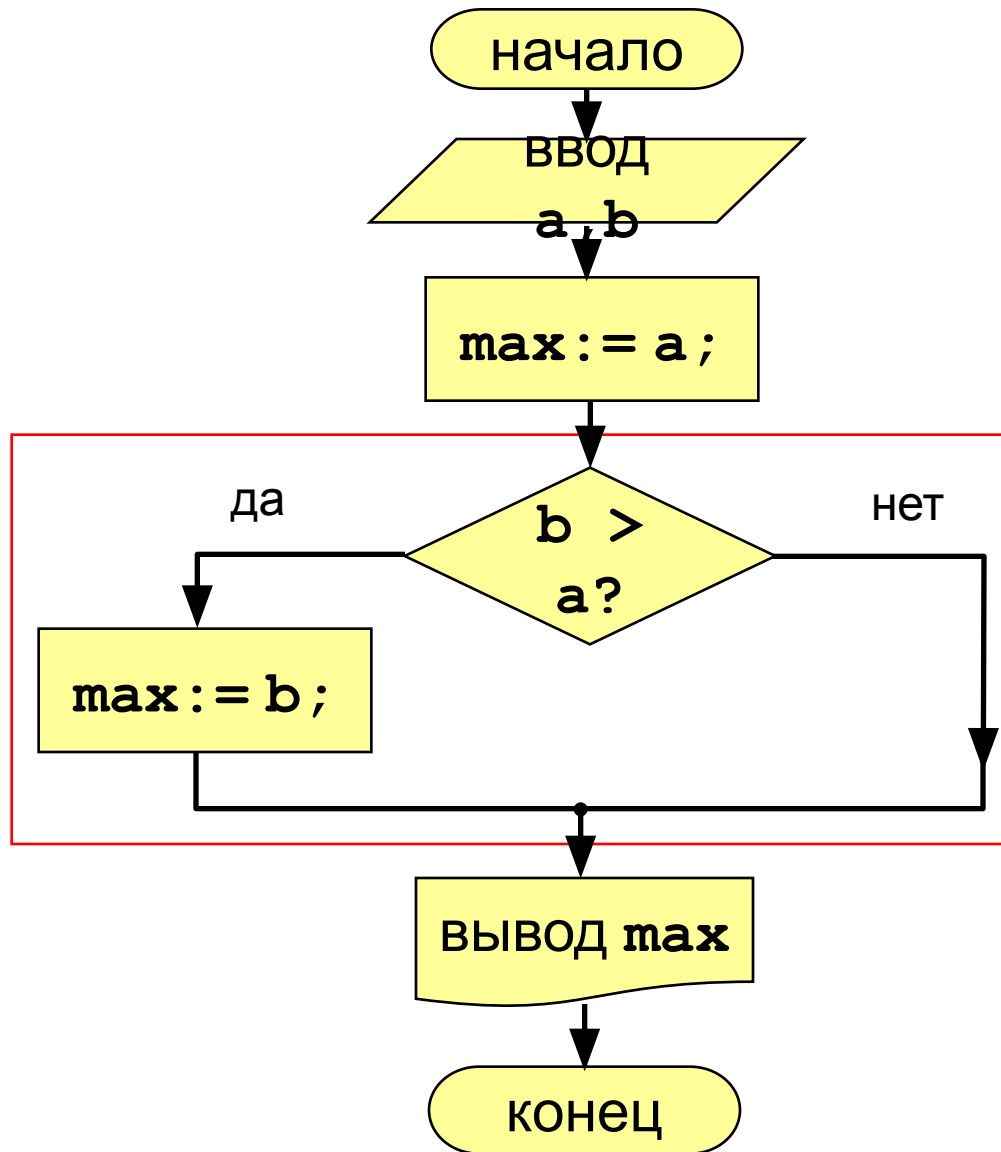
```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b; end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```

```
if a > b then begin
  a := b;
end
else begin
  b := a;
end;
```


Вариант 2. Блок-схема



неполная
форма
ветвления

Вариант 2. Программа

```
program qq;  
var a, b, max: integer;  
begin  
  writeln('Введите два целых числа');  
  read ( a, b );  
  max := a;  
  if b > a then  
    max := b;  
  writeln ('Наибольшее число ', max);  
end.
```

неполная
форма
условного
оператора

Вариант 2Б. Программа

```
program qq;  
var a, b, max: integer;  
begin  
  writeln('Введите два целых числа');  
  read ( a, b );  
  max := b;  
  if a > b then  
    max := a;  
  writeln ('Наибольшее число ', max);  
end.
```

Что неправильно?

```
if a > b the  
  a := b  
else b := a;
```

```
if a > b then begin  
  a := b;  
end  
else b := a;
```

```
if a > b then  
  a := b  
else b := a;
```

```
if b >= a then  
  b := a;
```

Задания

«4»: Ввести три числа и найти наибольшее из них.

Пример:

Введите три числа:

4 15 9

Наибольшее число 15

«5»: Ввести пять чисел и найти наибольшее из них.

Пример:

Введите пять чисел:

4 15 9 56 4

Наибольшее число 56

Программирование на языке Паскаль

Тема 3. Сложные условия

Сложные условия

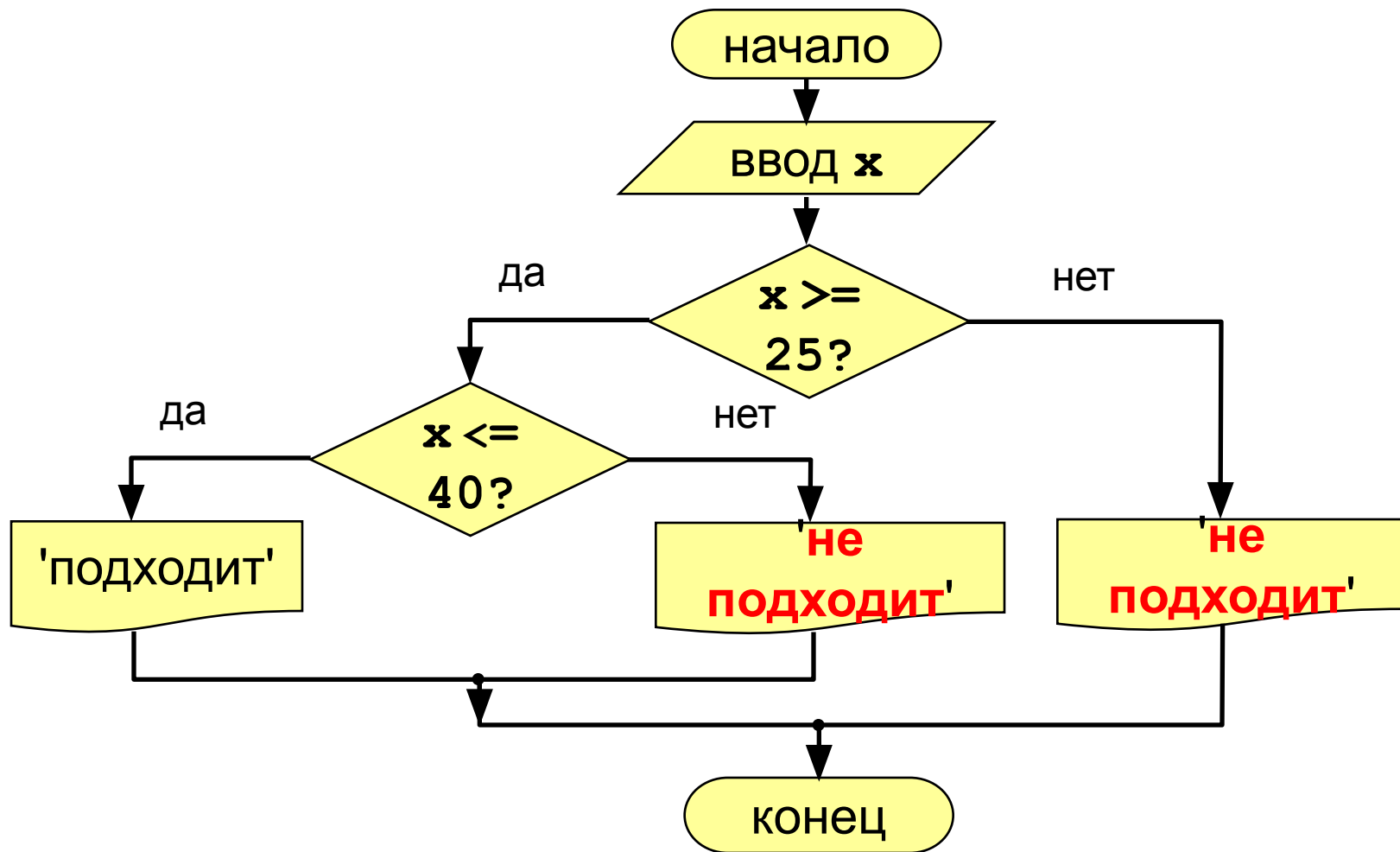
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.



Можно ли решить известными методами?

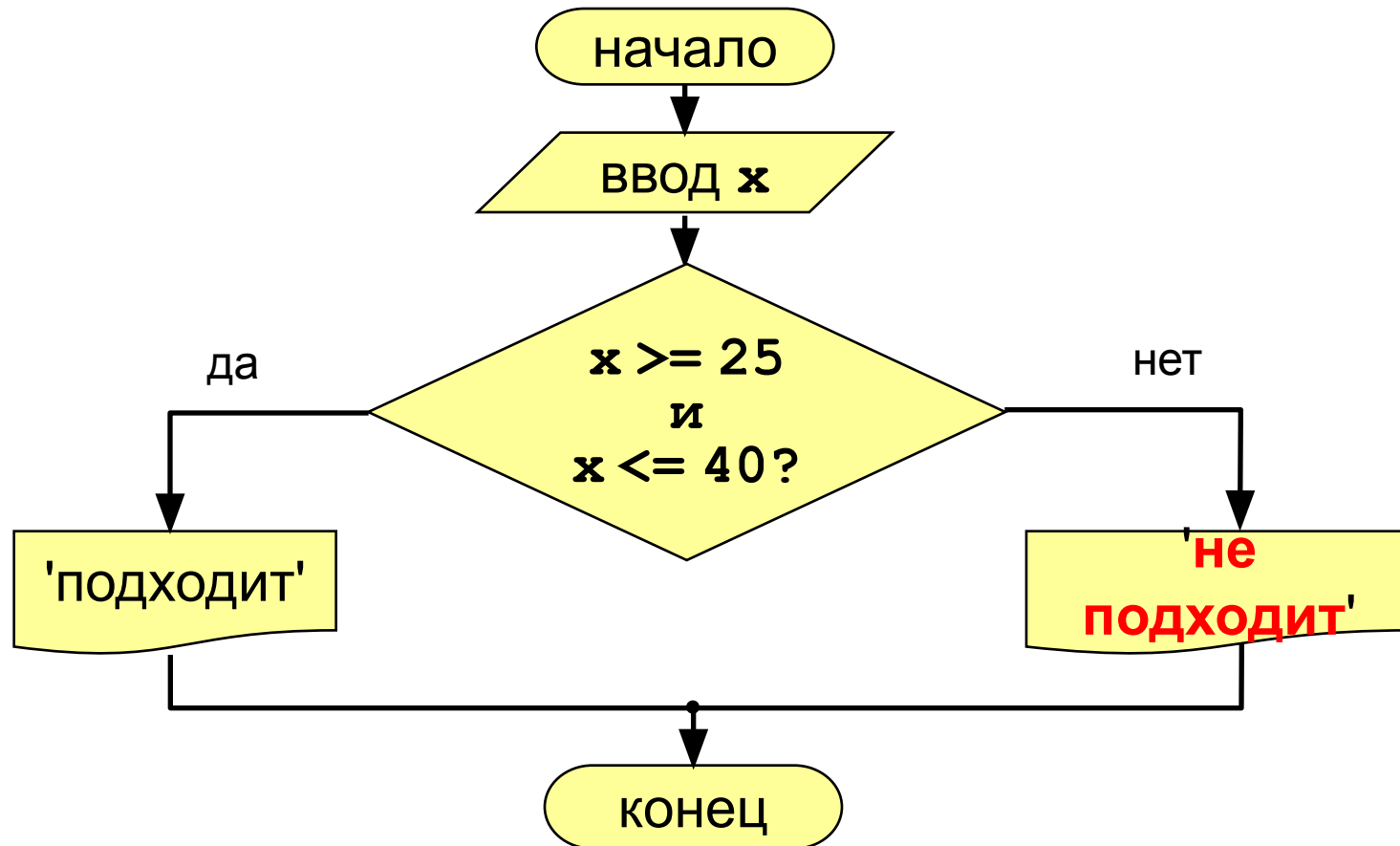
Вариант 1. Алгоритм



Вариант 1. Программа

```
program qq;  
var x: integer;  
begin  
  writeln('Введите возраст');  
  read ( x );  
  if x >= 25 then  
    if x <= 40 then  
      writeln ('Подходит')  
    else writeln ('Не подходит')  
  else  
    writeln ('Не подходит');  
end.
```

Вариант 2. Алгоритм



Вариант 2. Программа

```
program qq;  
var x: integer;  
begin  
  writeln('Введите возраст');  
  read ( x );  
  if (x >= 25) and (x <= 40) then  
    writeln ('Подходит')  
  else writeln ('Не подходит')  
end.
```

СЛОЖНОЕ
УСЛОВИЕ

Сложные условия

Простые условия (отношения)

равно

< <= > >= = <>

не равно

Сложное условие – это условие, состоящее из нескольких простых условий (отношений), связанных с помощью **логических операций**:

- **not** – НЕ (отрицание, инверсия)
- **and** – И (логическое умножение, конъюнкция, одновременное выполнение условий)
- **or** – ИЛИ (логическое сложение, дизъюнкция, выполнение хотя бы одного из условий)
- **xor** – исключающее ИЛИ (выполнение только одного из двух условий, но не обоих)

Сложные условия

Порядок выполнения (приоритет = старшинство)

- выражения в скобках
- `not`
- `and`
- `or`, `xor`
- `<`, `<=`, `>`, `>=`, `=`, `<>`

Особенность – каждое из простых условий обязательно заключать в скобки.

Пример

```
      4      1      6      2      5  
if not (a > b) or (c <> d) and (b <> a)  
then begin  
    ...  
end
```

Сложные условия

Истинно или ложно при $a := 2; b := 3; c := 4;$

`not (a > b)`

True

`(a < b) and (b < c)`

True

`not (a >= b) or (c = d)`

True

`(a < c) or (b < c) and (b < a)`

True

`(a < b) xor not (b > c)`

FALSE

Для каких значений **x** истинны условия:

`(x < 6) and (x < 10)`

`(x < 6) and (x > 10)`

`(x > 6) and (x < 10)`

`(x > 6) and (x > 10)`

`(x < 6) or (x < 10)`

`(x < 6) or (x > 10)`

`(x > 6) or (x < 10)`

`(x > 6) or (x > 10)`

$(-\infty; 6)$	$x < 6$
\emptyset	
$(6; 10)$	
$(10; \infty)$	$x > 10$
$(-\infty; 10)$	$x < 10$
$(-\infty; 6) \cup (10; \infty)$	
$(-\infty; \infty)$	
$(6; \infty)$	$x > 6$

Задания

«4»: Ввести номер месяца и вывести название времени года.

Пример:

Введите номер месяца:

4

весна

«5»: Ввести возраст человека (от 1 до 150 лет) и вывести его вместе с последующим словом «год», «года» или «лет».

Пример:

Введите возраст:

24

Вам 24 года

Введите возраст:

57

Вам 57 лет

Программирование на языке Паскаль

Тема 4. Циклы

Циклы

Цикл – это многократное выполнение одинаковой последовательности действий.

- цикл с **известным** числом шагов
- цикл с **неизвестным** числом шагов (цикл с условием)

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

Особенность: одинаковые действия выполняются 5 раз.



Можно ли решить известными методами?

Циклы

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



Что плохо?

Циклы

```
program qq;  
begin
```

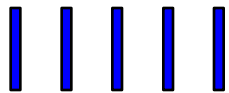
```
  { сделай 5 раз }
```

```
    writeln( 'Привет' );
```

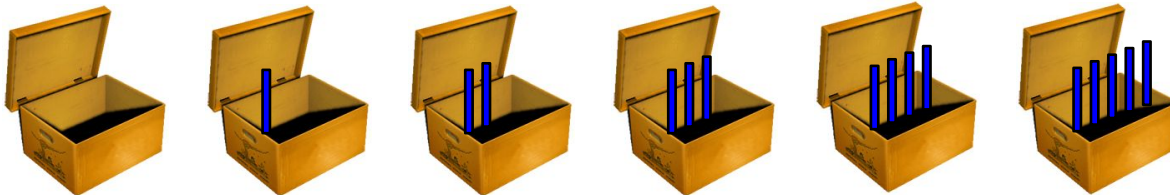
```
end.
```



Как отсчитать ровно 5 раз?

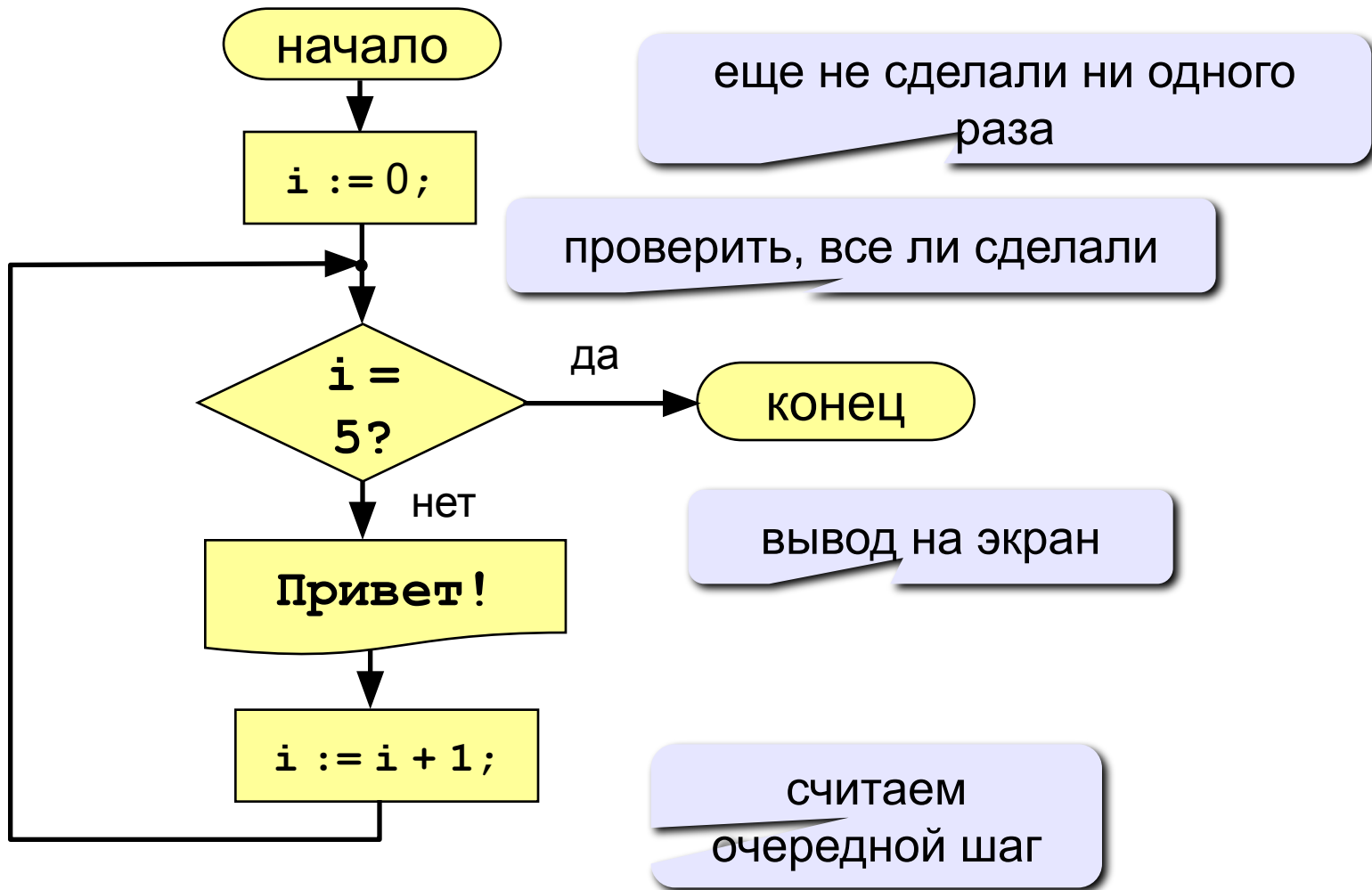


Как запоминать, сколько раз уже сделали?



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

Алгоритм



Циклы

```
program qq;  
var i: integer;  
begin  
  for i:=1 to 5 do  
    writeln('Привет');  
end.
```

«Для всех i от 1 до 5
делай ...»

Если в цикле более одного оператора:

```
for i:=1 to 5 do begin  
  write('Привет');  
  writeln(', Вася!');  
end;
```



Что получится?

Циклы

Задача. Вывести на экран квадраты и кубы целых чисел от 1 до 8 (от **a** до **b**).

Особенность: одинаковые действия выполняются 8 раз.



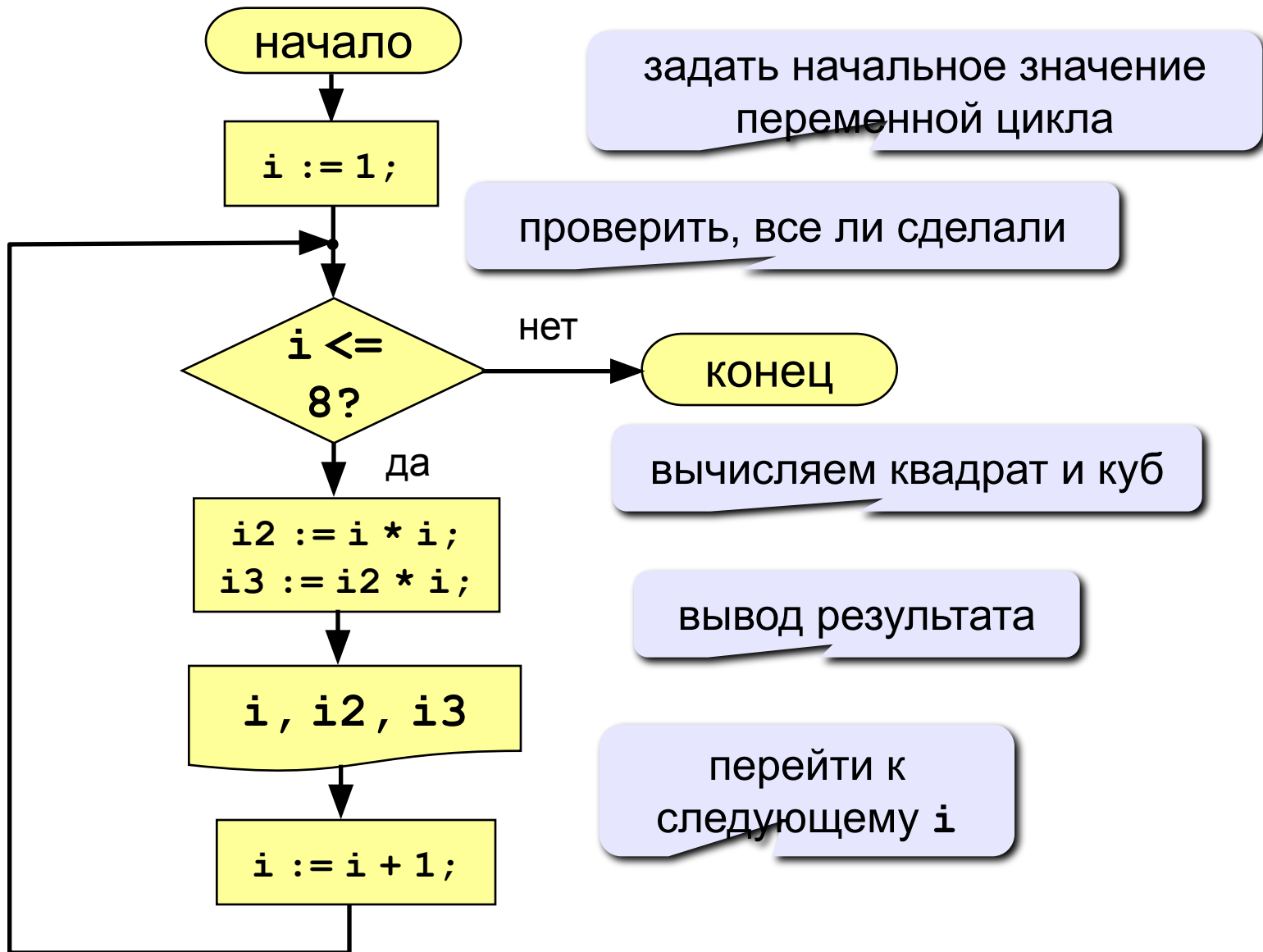
Можно ли решить известными методами?

```
i := 1;      { очередное число }  
i2 := i*i;   { его квадрат }  
i3 := i2*i;  { куб }  
writeln(i:4, i2:4, i3:4);  
i := 2;  
...
```

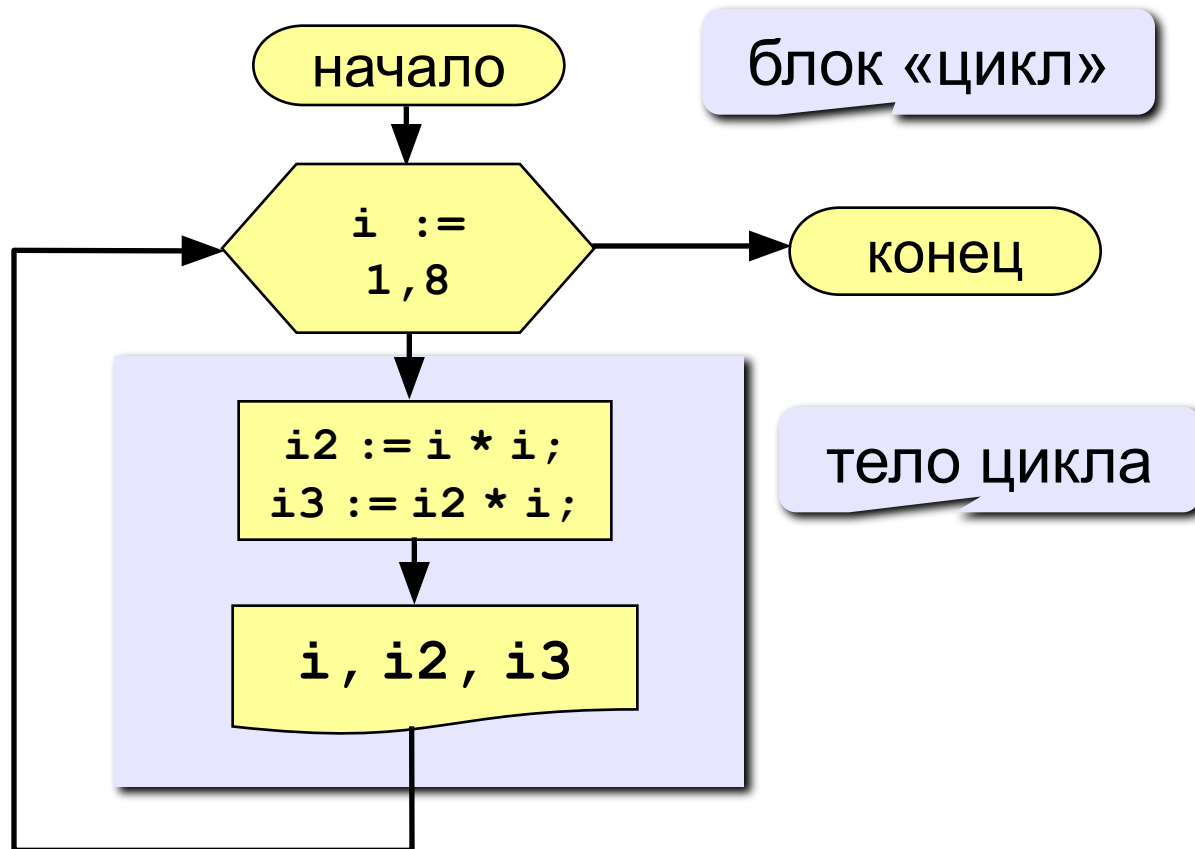


А если начальное и конечное значения вводятся с клавиатуры?

Алгоритм



Алгоритм (с блоком «цикл»)



Программа

```
program qq;  
var i, i2, i3: integer;  
begin  
    for i:=1 to 8 do begin  
        i2 := i*i;  
        i3 := i2*i;  
        writeln(i:4, i2:4, i3:4);  
    end;  
end.
```

переменная
цикла

начальное значение

конечное значение

Цикл с уменьшением переменной

Задача. Вывести на экран квадраты и кубы целых чисел от 8 до 1 (в обратном порядке).

Особенность: переменная цикла должна уменьшаться.

Решение:

```
for i:=8 downto 1 do begin
  i2 := i*i;
  i3 := i2*i;
  writeln(i:4, i2:4, i3:4);
end;
```

Цикл с переменной

Увеличение переменной на 1:

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

Уменьшение переменной на 1:

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

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
защипливания

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~~

Как изменить шаг?

Задача. Вывести на экран квадраты и кубы нечётных целых чисел от 1 до 9.

Особенность: переменная цикла должна увеличиваться на 2.

Проблема: в Паскале шаг может быть 1 или -1.

Решение:

```
for i:=1 to 9 do begin
  if i mod 2 = 1 then begin
    i2 := i*i;
    i3 := i2*i;
    writeln(i:4, i2:4, i3:4);
  end;
end;
```

выполняется
только для
нечётных *i*



Что плохо?

Как изменить шаг? – II

Идея: Надо вывести всего 5 чисел, переменная **k** изменяется от 1 до 5. Начальное значение **i** равно 1, с каждым шагом цикла **i** увеличивается на 2.

Решение:

```
i := 1;
```

```
for k:=1 to 5 do begin
```

```
  i2 := i*i;
```

```
  i3 := i2*i;
```

```
  writeln(i:4, i2:4, i3:4);
```

```
  i := i + 2;
```

```
end;
```

Как изменить шаг? – III

Идея: Надо вывести всего 5 чисел, переменная **k** изменяется от 1 до 5. **Зная k, надо рассчитать i.**

k	1	2	3	4	5
i	1	3	5	7	9

$$i = 2k - 1$$

Решение:

```
for k:=1 to 5 do begin
  i := 2*k - 1;
  i2 := i*i;
  i3 := i2*i;
  writeln(i:4, i2:4, i3:4);
end;
```

Задания

«4»: Ввести a и b и вывести квадраты и кубы чисел от a до b .

Пример:

Введите границы интервала:

4 6

4 16 64

5 25 125

6 36 216

«5»: Вывести квадраты и кубы 10 чисел следующей последовательности: 1, 2, 4, 7, 11, 16, ...

Пример:

1 1 1

2 4 8

4 16 64

...

46 2116 97336