# Программирование на языке Паскаль

## Тема 2. Ветвления

# Разветвляющиеся алгоритмы

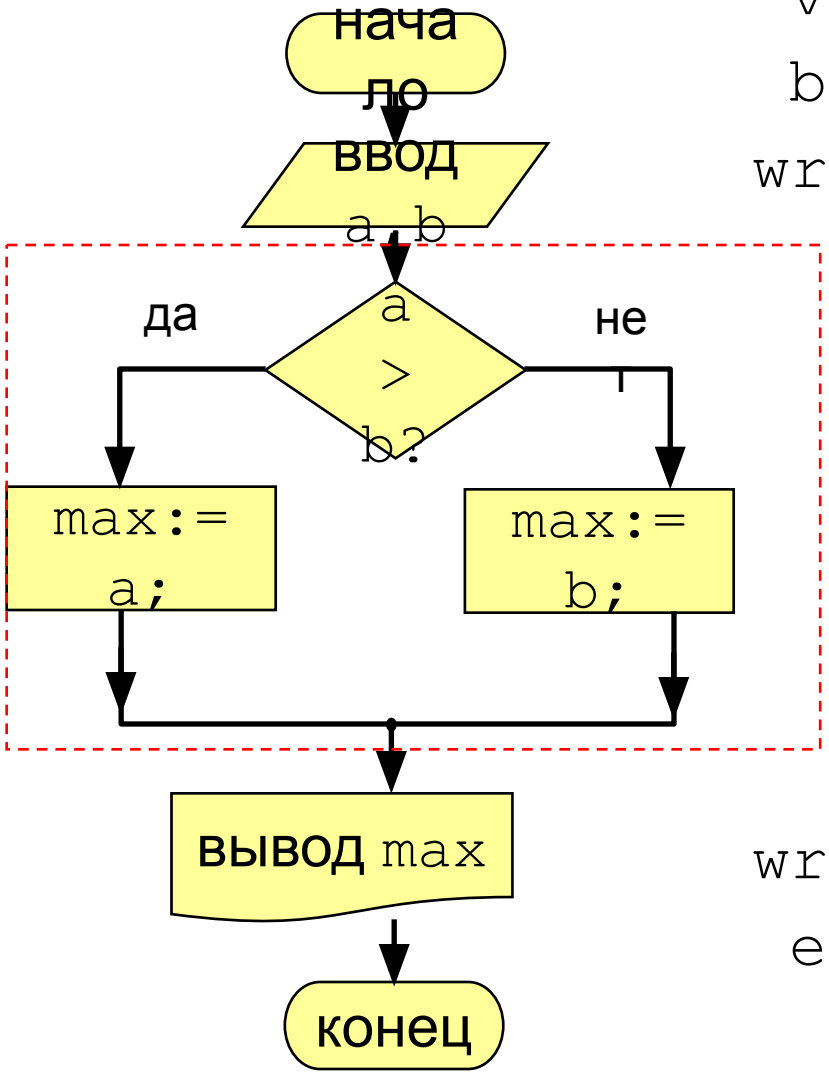
Задача. Ввести два целых числа и вывести на экран наибольшее из них.

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 … то… иначе …*).

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

# Блок-схема




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

? Если a = b?

# 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*

# 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

# Что неправильно?

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

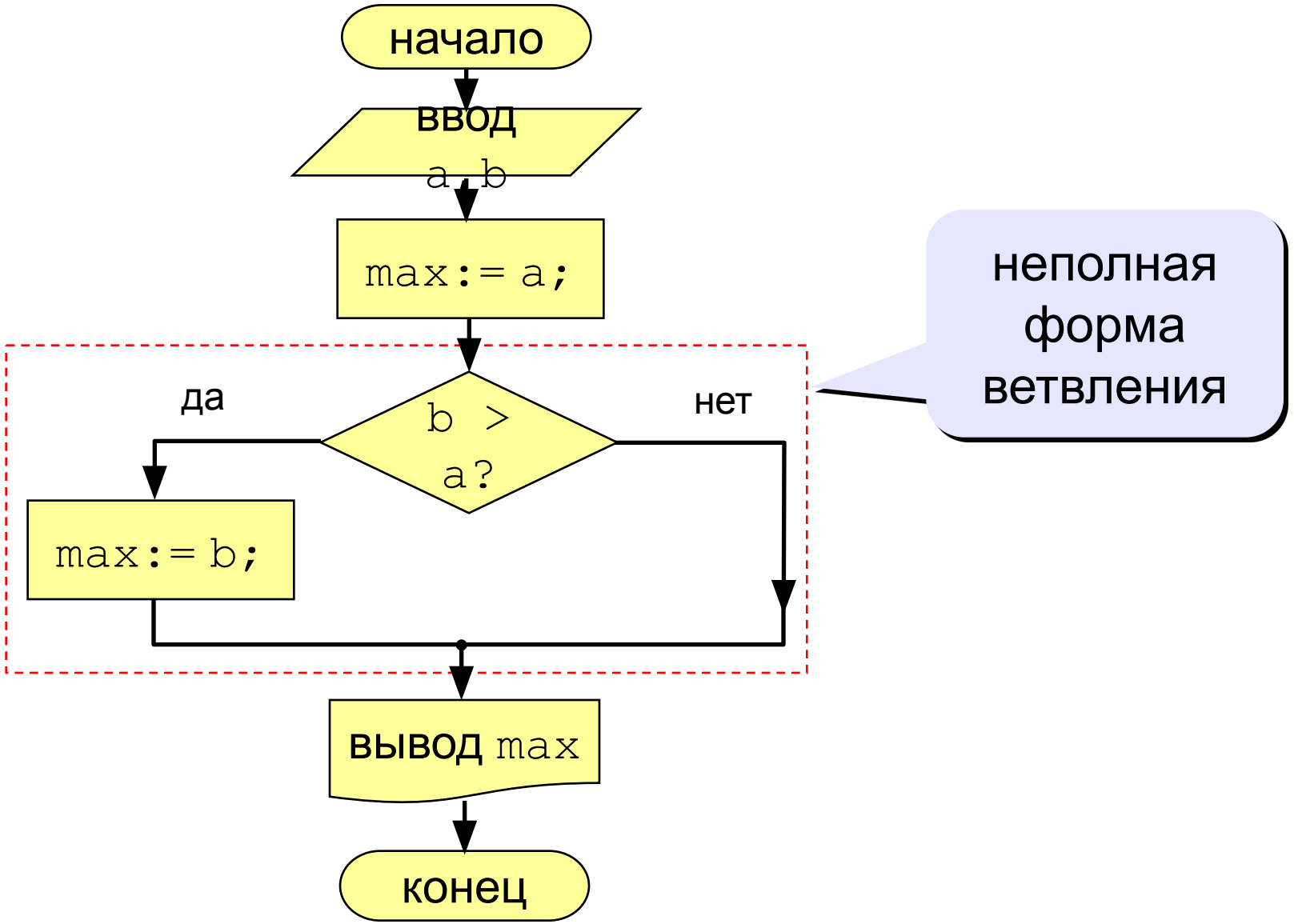
```
if a > b then begin
   a := b;
end
else begin
   b := a;
end;
```

```
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;
```

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

# Вариант 2. Блок-схема



# 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.
```

# Задания

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

Пример:

```
Введите три числа:
4   15   9
Наибольшее число 15
```

# Программирование на языке Паскаль

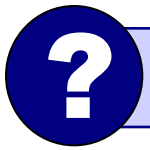
## Тема 3. Сложные условия

© К.Ю. Поляков, 2006-2009

# Сложные условия

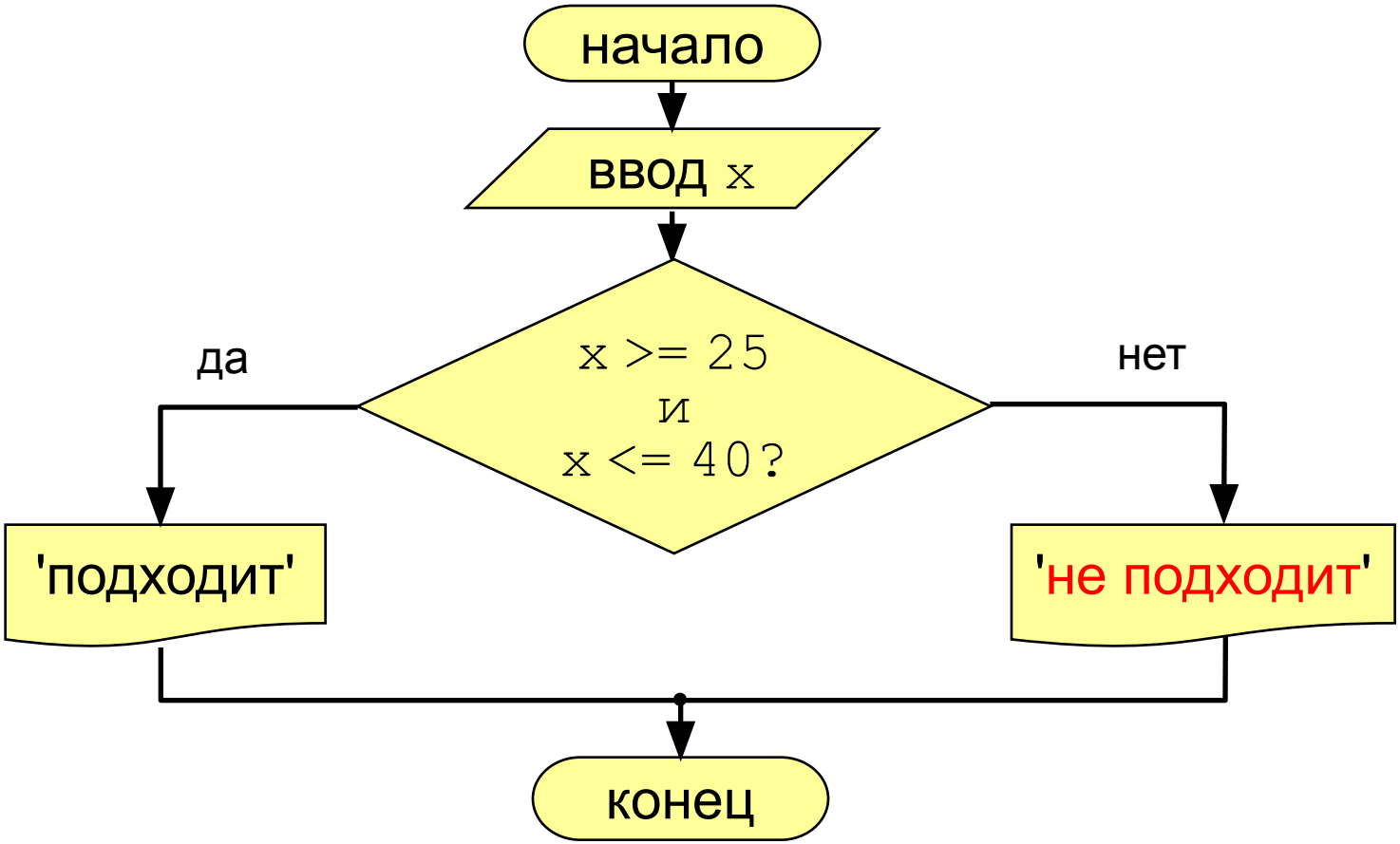
Задача. Фирма набирает сотрудников от 25 до 40 лет включительно. Ввести возраст человека и определить, подходит ли он фирме (вывести ответ «подходит» или «не подходит»).

Особенность: надо проверить, выполняются ли два условия одновременно.




Можно ли решить известными методами?

# Вариант 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

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)

Простые условия (отношения)

`<` `<=` `>` `>=` `=` `<>`

равно

не равно

# 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)
```

| | |
|---|---|
| (−∞; 6) | x < 6 |
| ∅ | |
| (6; 10) | |
| (10; ∞) | x > 10 |
| (−∞; 10) | x < 10 |
| (−∞; 6) ∪ (10;∞) | |
| (−∞; ∞) | |
| (6; ∞) | x > 6 |

# Задания

«4»: Ввести с клавиатуры три числа A,B,C и определить, принадлежит ли число C отрезку [ A ; B].

Пример:

```
Введите три числа A,B,C:
4   15   9
Число 9 принадлежит отрезку!
```

«5»:Ввести коэффициенты квадратного уравнения A,B,C и решить уравнение $Ax^2 + Bx + C = 0$