

# Программирование на языке Паскаль

## Ветвления



# Разветвляющиеся алгоритмы

---

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

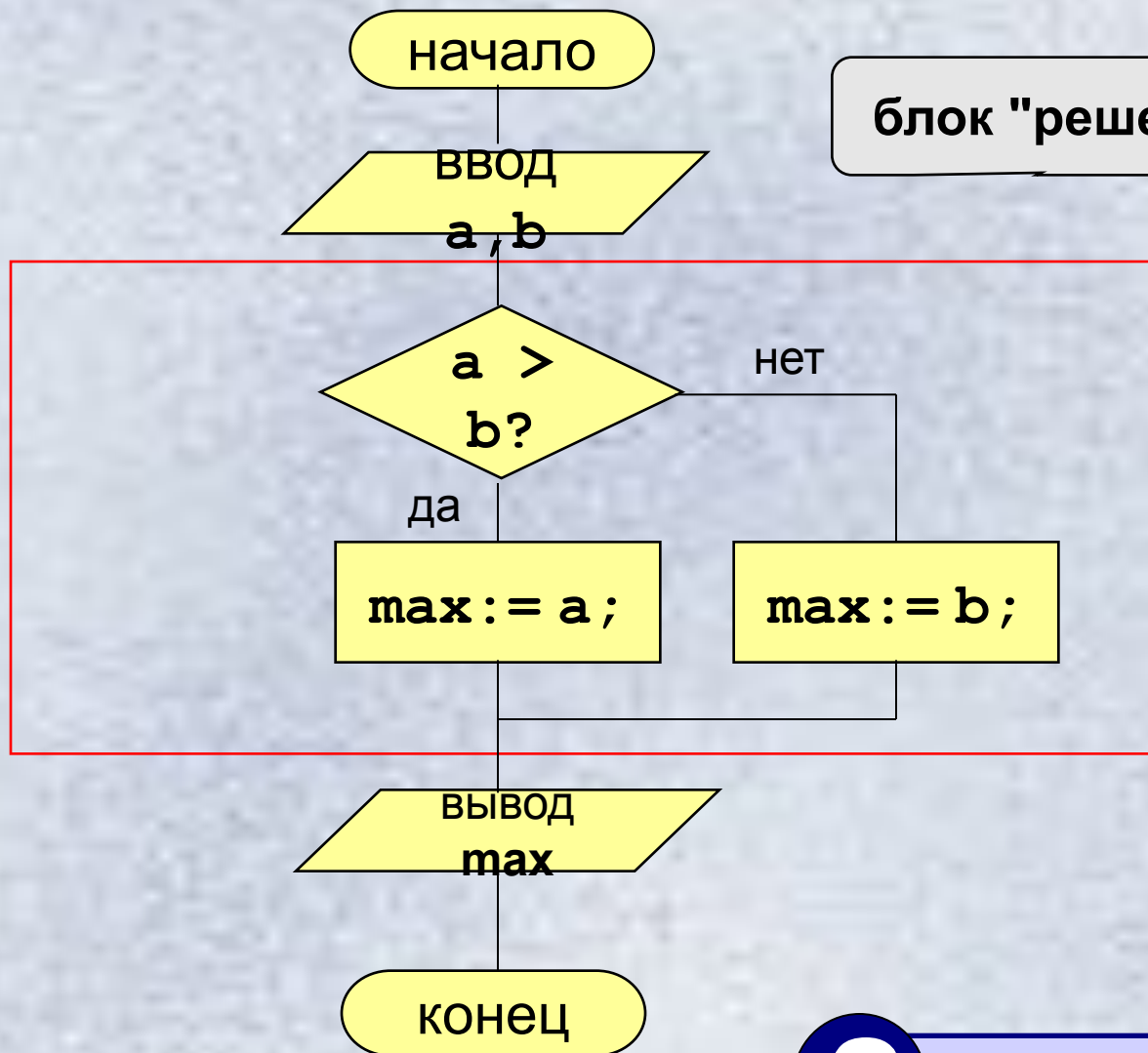
**Идея решения:** надо вывести на экран первое число, если оно больше второго, или второе, если оно больше первого.

**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 begi
  b :=n a;
end;
```

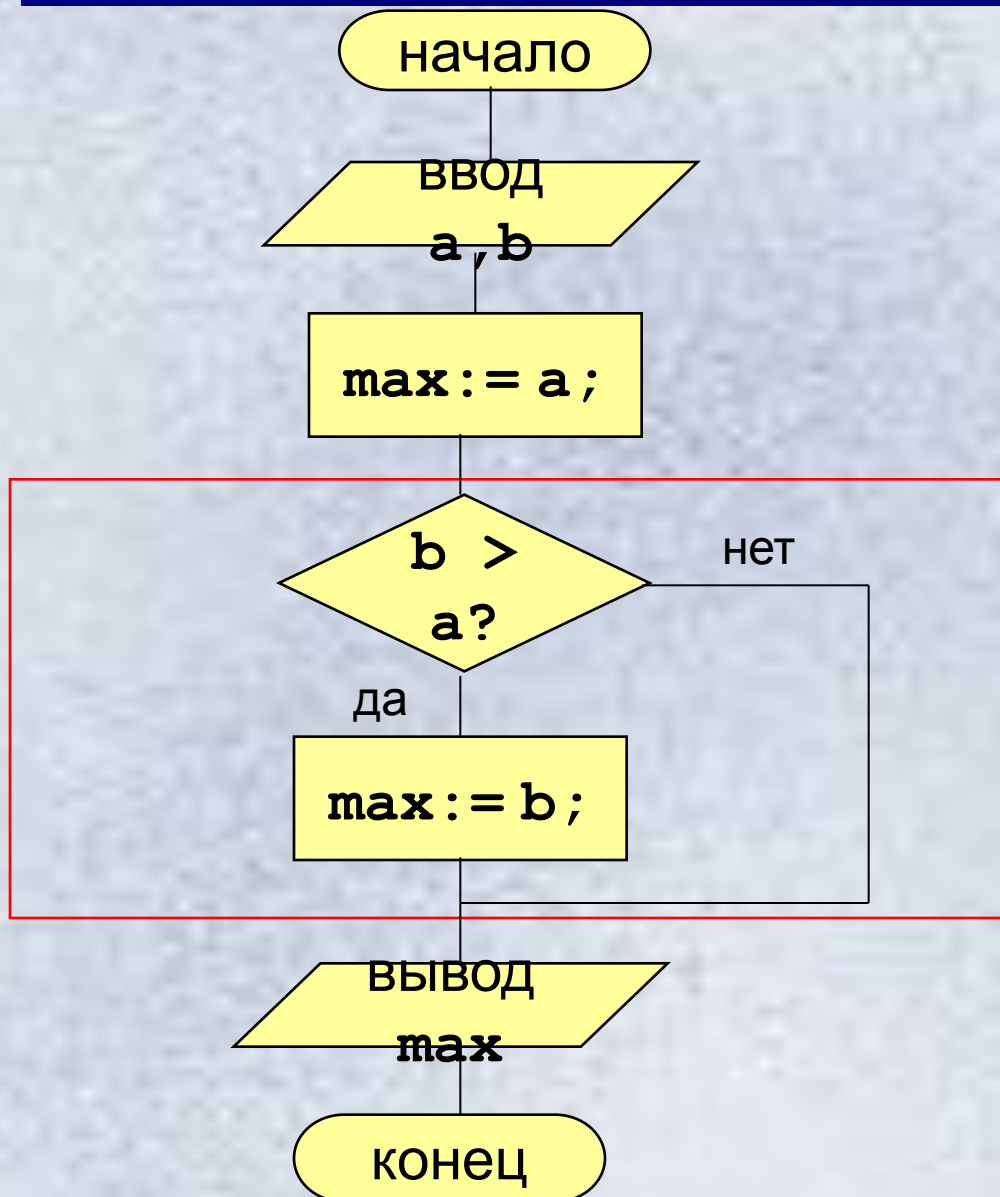
```
if a > b then begin
  a := b; end
else begin
  b := a;
end;
```

```
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 then  
    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

## 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.

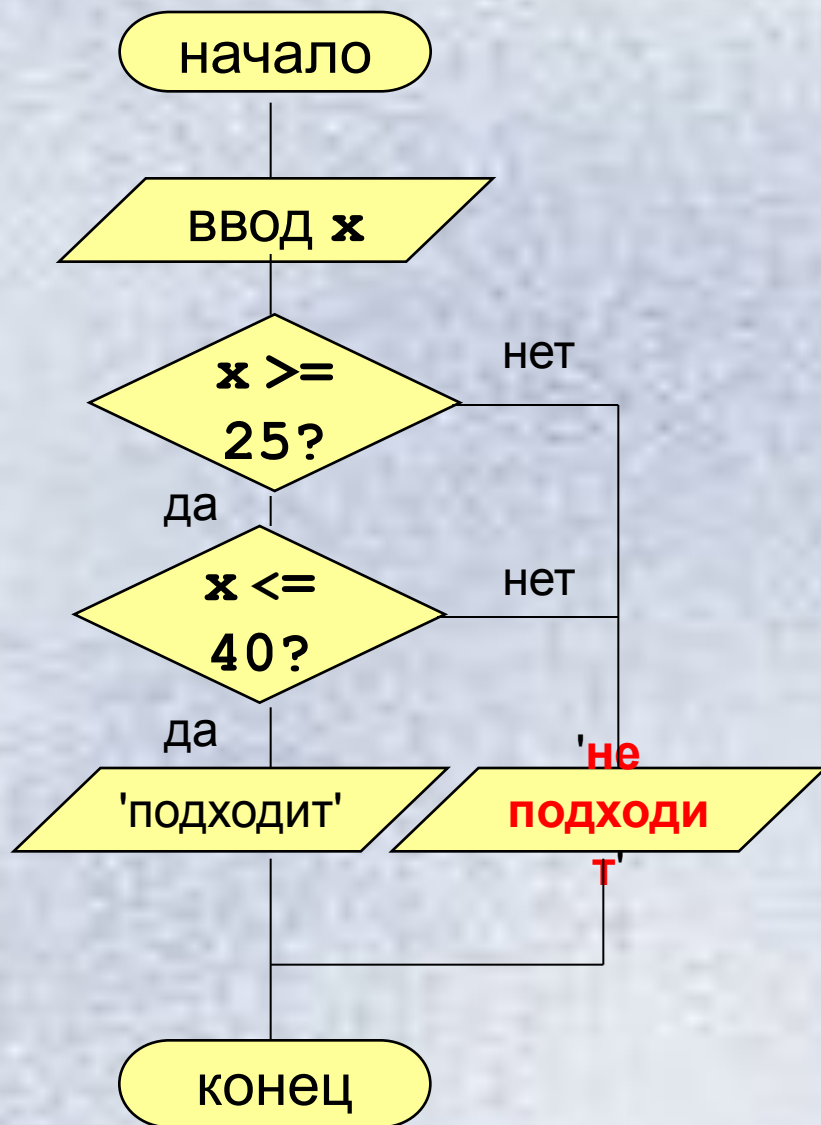


**Можно ли решить известными методами?**



# Вариант 1. Алгоритм

---



# Вариант 1. Программа

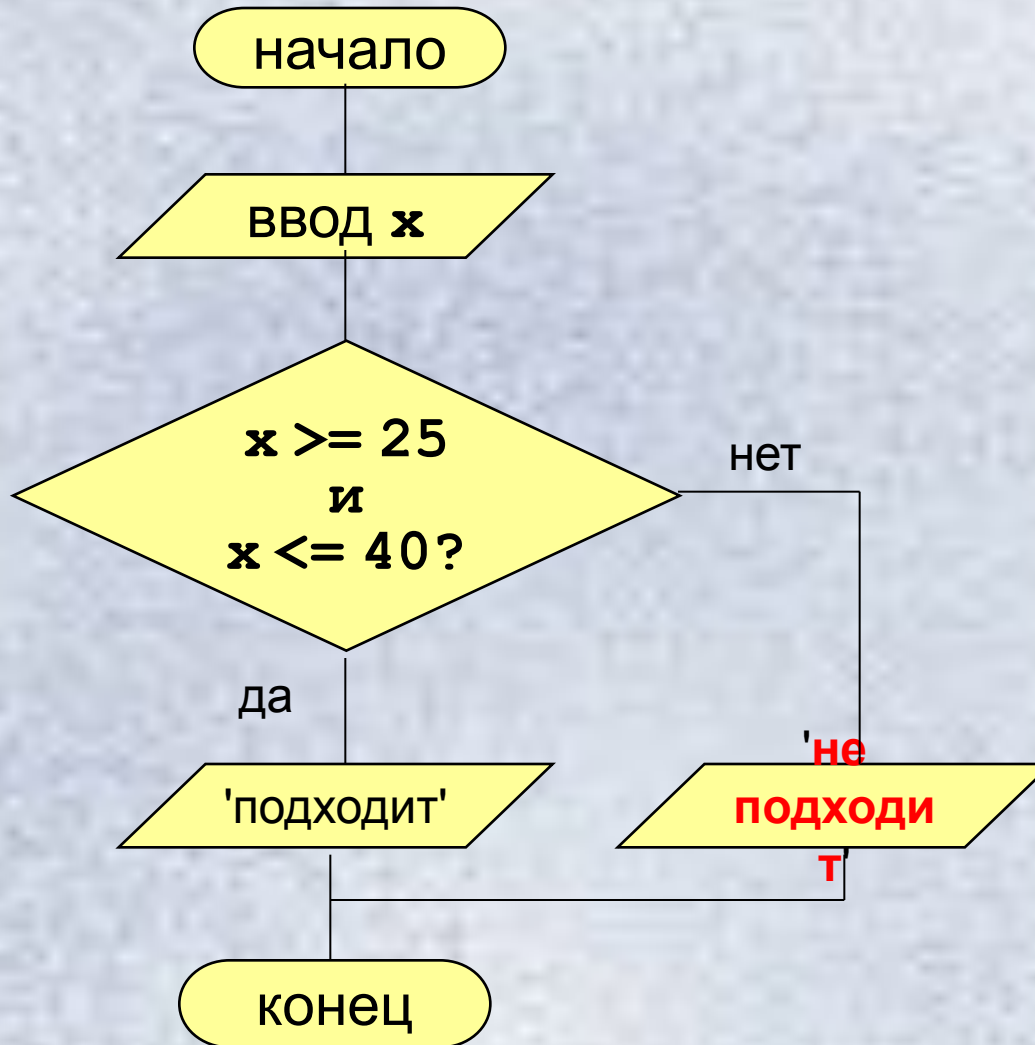
---

```
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

---

**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

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

**Особенность** – каждое из простых условий обязательно заключать в скобки.

## Пример

```
      4      1      6      2      5      3  
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

