Программирование (Паскаль)

1. Введение

Что такое программирование?

Программирование — это создание программ для компьютеров. Этим занимаются **программисты**.

Чем занимаются **программисты:**

анализ задачи (выделение исходных данных, связей между ними, этапов решения задачи)	системные аналитики
разработка **алгоритмов**	алгоритмисты
написание и отладка **программ**	кодировщики
тестирование программ	тестировщики
написание **документации**	технические писатели

Направления в программировании

системный программист	операционные системы, утилиты, драйверы
прикладной программист	прикладные программы, в т.ч. для мобильных устройств
веб-программист	веб-сайты
программист баз данных	системы управления базами данных

Простейшая программа

название программы

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

комментарии внутри {}
не обрабатываются

? Что делает эта программа?

Вывод на экран

```
program Hello;
begin
  write('Привет!');
end.
```

оператор вывода

Оператор — это команда языка программирования.

~~`write('Привет', Вася!);`~~

? Что плохо?

```
write('Привет, Вася!');
```

вся строка в апострофах

Переход на новую строку

```
write('Привет, Вася!');
write('Привет, Петя!');
```

ожидание:

```
Привет, Вася!
Привет, Петя!
```

реальность:

```
Привет, Вася!Привет, Петя!
```

решение:

```
writeln('Привет, Вася!');
writeln('Привет, Петя!');
```

и перейти на новую строку

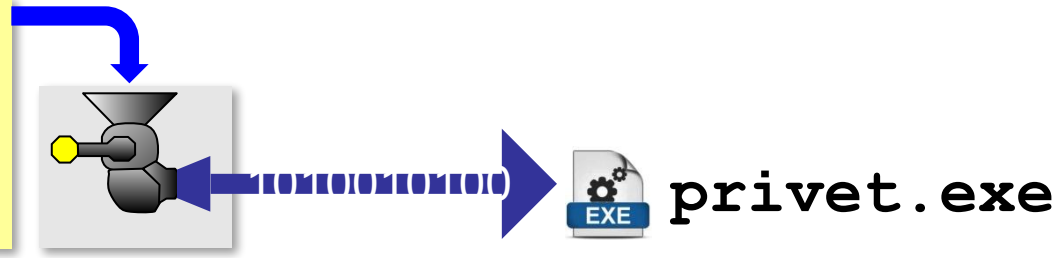
Системы программирования

Системы программирования — это средства для создания новых программ.

Транслятор — это программа, которая переводит тексты программ, написанных программистом, в машинные коды (команды процессора).

- **компилятор** — переводит всю программу в машинные коды, строит исполняемый файл (**.exe**)

```
program Hello;
begin
 write('Привет!');
end.
```



- **интерпретатор** — сам выполняет программу по частям (по одному оператору).

Системы программирования

Отладчик — это программа для поиска ошибок в других программах.

- **пошаговый режим** — выполнение программы по шагам (по одному оператору)
- **просмотр значений переменных** во время выполнения программы
- **точки останова** – операторы в программе, перед выполнением которых нужно остановиться.

Среда программирования (IDE):

- редактор текста программ
- транслятор
- отладчик

Программирование (Паскаль)

2. Линейные программы

Пример задачи

Задача. Ввести два числа и вычислить их сумму.

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

? Выполнится?

Псевдокод – алгоритм на русском языке с элементами языка программирования.

! Компьютер не может исполнить псевдокод!

Зачем нужны переменные?

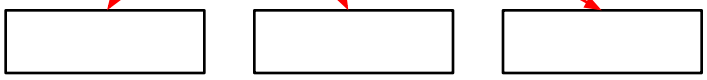
```
program Sum;
begin
   { ввести два числа }
   { вычислить их сумму }
   { вывести сумму на экран }
end.
```

Где запомнить?

Переменная — это величина, которая имеет имя, тип и значение. Значение переменной может изменяться во время выполнения программы.

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

объявление переменных



ячейки памяти

Имена переменных

Идентификатор — это имя программы или переменной.

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

заглавные и строчные буквы **НЕ различаются**

МОЖНО использовать

- латинские буквы (A-Z, a-z)
- цифры

Имя не может начинаться с цифры!

- знак подчеркивания _

НЕЛЬЗЯ использовать ~~скобки, знаки ", &, |, *, +, =, !, ? и др.~~

Какие имена правильные?

AXby R&B 4Wheel Вася “PesBarbos”
TU154 [QuQu] _ABBA A+B

Работа с переменными

Присваивание (запись значения)

```
a := 5;
```

оператор присваивания

$a \leftarrow 5$

```
a := 5;
a := 18;
```

? Что будет храниться в *a*?

Вывод на экран

```
write(a);
```

? В чём разница?

```
c:= 14;
write(c);
```

14

```
c:= 14;
write('c');
```

c

Работа с переменными

Изменение значения

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

увеличить на 1

```
i ← i + 1
```

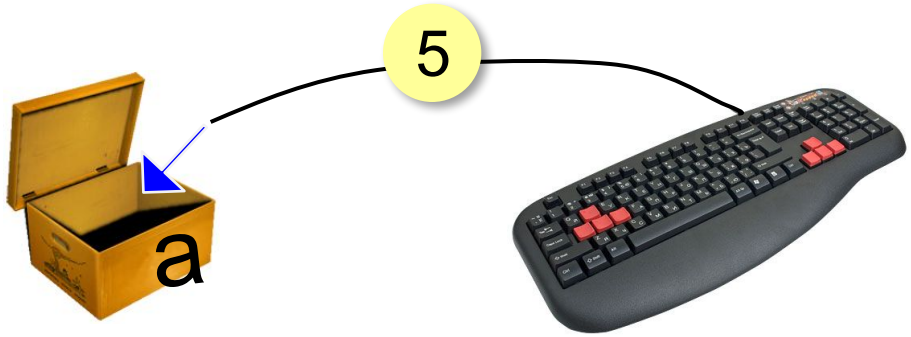
```
a:= 4;
b:= 7;
a:= a + 1;
b:= b + 1;
a:= a + b;
b:= b + a;
a:= a + 2;
b:= b + a;
```

a	b
4	
	7
5	
	8
13	
	21
15	
	36

Ввод с клавиатуры

Цель – изменить исходные данные, не меняя программу.

```
read(a);
```

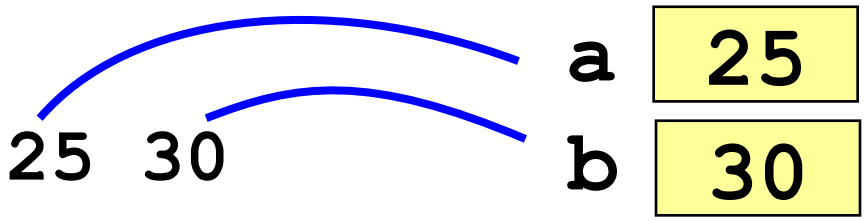


1. Программа ждет, пока пользователь введет значение и нажмет *Enter*.
2. Введенное значение записывается в переменную **a**.

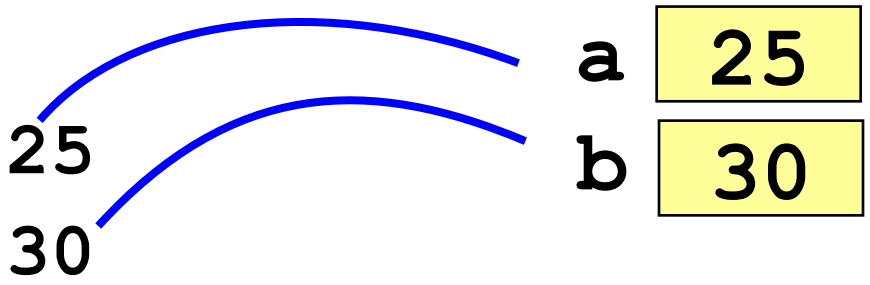
Ввод с клавиатуры

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

через пробел:



через *Enter*:



Программа сложения чисел

```
program Sum;
var a, b, c: integer;
begin
  read(a, b); { ввести два числа }
  c:= a + b;  { вычислить их сумму }
  write(c) { вывести сумму на экран }
end.
```

? Что плохо?

ожидание:

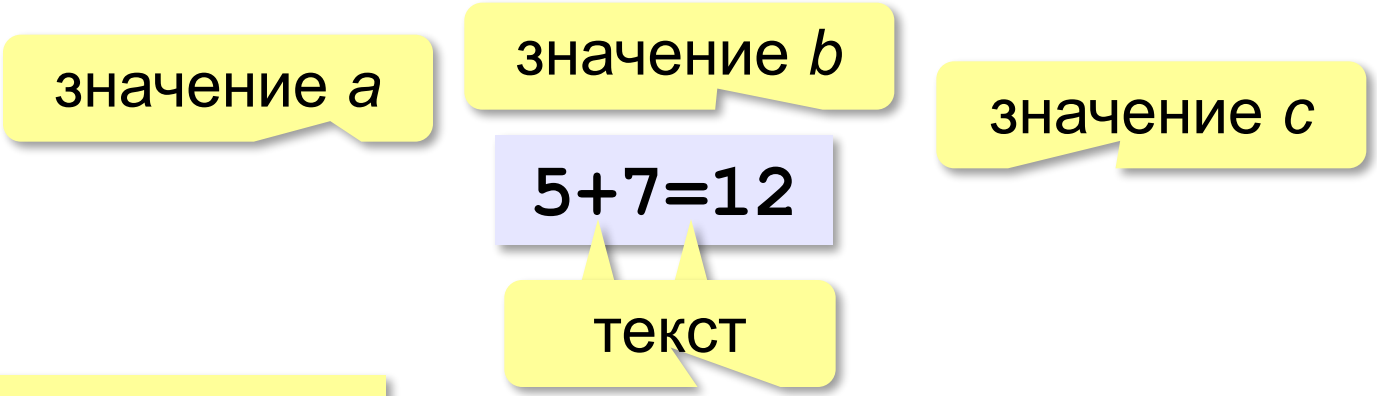
```
Введите два числа: 5 7
5+7=12
```

реальность:

```
5 7
12
```

? Как улучшить диалог?

write(данных с текстом




```
write(a);
write('+');
write(b);
write('=');
write(c);
```

→

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

Программа сложения чисел

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

Как переделать для 3-х чисел?

Арифметические выражения

$$a \leftarrow \frac{c+b-1}{2} \cdot d$$

Линейная запись (в одну строку):

```
a:=(c+b-1)/2*d;
```

Операции: **+ –**

***** – умножение

/ – деление

****** – возведение в степень ($x^2 \rightarrow$ **x**2**)

```
var x, a, b: integer;
read(a, b);
x:= a / b;
```

? Что плохо?

```
var x: real;
```

Порядок выполнения операций

1) действия в скобках
2) возведение в степень
3) умножение и деление, слева направо
4) сложение и вычитание, слева направо

```
     6    5      2    1      3    4
a := c + (1 - 2 * b) / 2 * d;
```

Частное и остаток

div – деление нацело (остаток отбрасывается)

mod – остаток от деления

175 сек = 2 мин 55 сек

Как получить 2 и 55?

```
var t, m, s: integer;
t:= 175;
m:= t div 60;
s:= t mod 60;
```

Частное и остаток

Что получится?

```
n:= 123
d:= n div 10;
k:= n mod 10;
```

При делении на 10 нацело отбрасывается последняя цифра числа.

Остаток от деления на 10 – это последняя цифра числа.

Форматный вывод

```
var a, b, c: integer;
a:=1; b:=2; c:=3;
write(a, b, c);
```

→ 123

```
write(a,' ',b,' ',c);
```

→ 1 2 3

```
write(a, b:3, c:5);
```

→ 1␣␣2␣␣␣␣3

3 5

количество знаков на вывод числа

? Сколько знаков для вывода *a*?

Форматный вывод

```
var x: real;
x:=12.34567891234;
write(x);
```

вариант:

```
12.345679
```

6 по умолчанию

```
write(x:10:3);
```

```
____12.346
```

всего на число

в дробной части

3

10

```
write(x:8:2);
```

```
___12.34
```

```
write(x:2:2);
```

```
12.34
```

```
write(x:0:1);
```

```
12.3
```

минимально возможное

Научный формат чисел

```
var x: real;
x:=123456789;
write(x);
```

→ `1.234568e+008`

$1{,}234568 \cdot 10^{8}$

```
var x: real;
x:=0.0000123456789;
write(x);
```

→ `1.234568e-005`

$1{,}234568 \cdot 10^{-5}$

количество знаков может отличаться

Операции с вещественными числами

trunc – целая часть числа (дробная часть отбрасывается)

round – округление к ближайшему целому

frac – дробная часть

```
x:= 1.6;
write(trunc(x));
```
→ 1

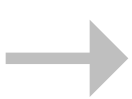
```
write(round(x));
```
→ 2

```
write(frac(x));
```
→ 0.6

Операции с вещественными числами

sqrt – квадратный корень

```
x:= 2.25;
write(sqrt(x));
```



```
1.5
```

Операции с вещественными числами

1/3 = 0,33333…

бесконечно много знаков

Большинство вещественных чисел хранятся в памяти компьютера с ошибкой!

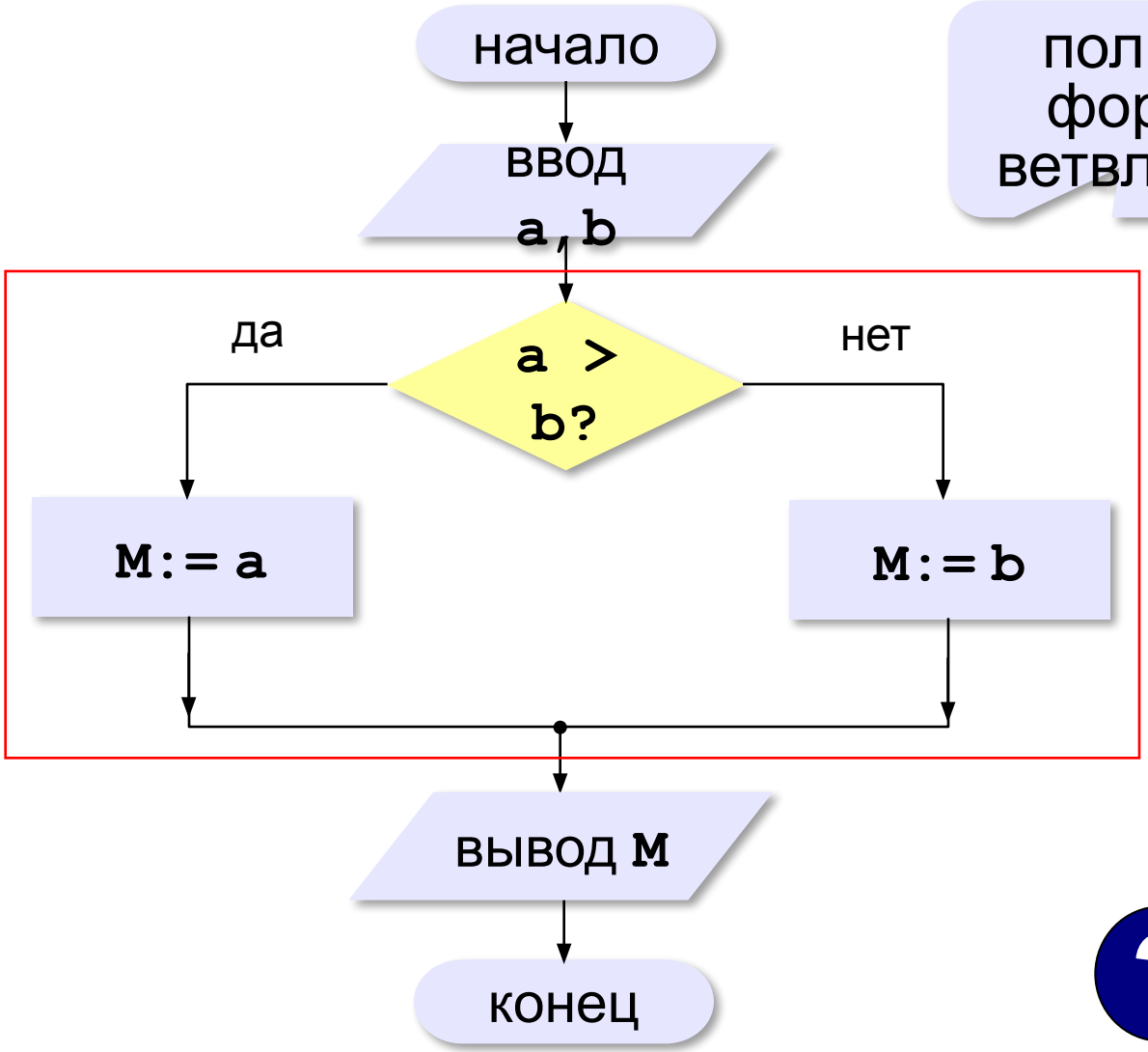
```
var x, y, z: real;
x:= 1/2;
y:= 1/3;
z:= 5/6;  { 5/6=1/2+1/3 }
write(x+y-z);
```

```
-1.110223e-016
```

Программирование (Паскаль)

3. Ветвления

Выбор наибольшего из двух чисел



полная форма ветвления

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

? Если a = b?

Вариант 1. Программа

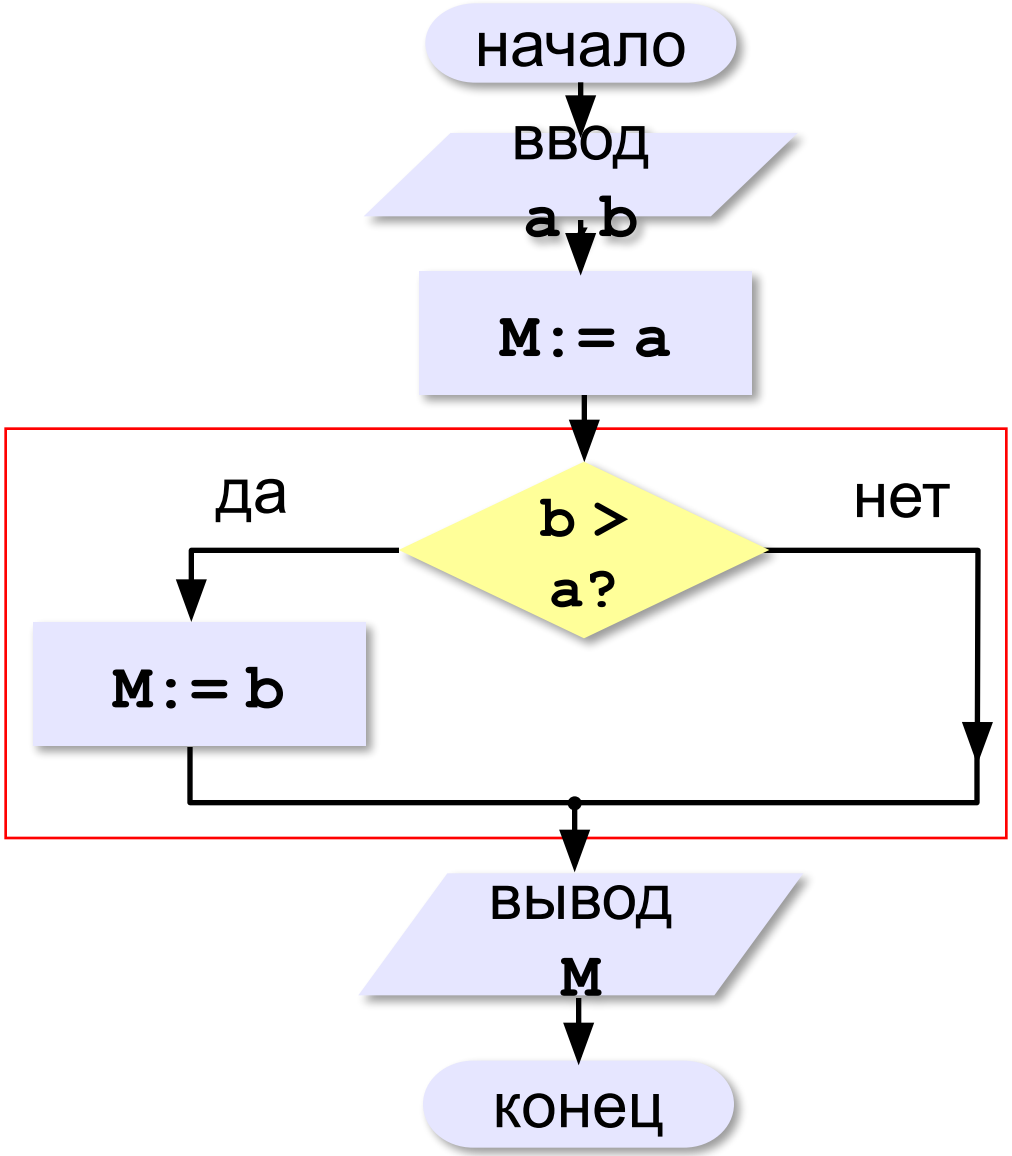
```
program Maximum;
var a, b, M: integer;
begin
  writeln('Введите два целых числа');
  read(a, b);
  if a > b then
    M:= a
  else
    M:= b;
  writeln('Наибольшее число ', M);
end.
```

полная форма условного оператора

!

Перед **else** не ставится точка с запятой!

Выбор наибольшего из двух чисел-2



неполная форма ветвления

Вариант 2. Программа

```
program Maximum2;
var a, b, M: integer;
begin
  writeln('Введите два целых числа');
  read(a,b);
  M:=a;
  if b > a then
    M:=b;
  writeln('Наибольшее число ', M);
end.
```

неполная форма условного оператора

Примеры

Поиск минимального:

```
if a < b then
  M:= a;
if b < a then
  M:= b;
```

? Что плохо?

? Когда работает неверно?

```
if a < b then
 begin
   c:= a;
   a:= b;
   b:= c
 end;
```

? Что делает эта программа?

составной оператор

! Перед end можно не ставить точку с запятой!

В других языках программирования

Python:

```
if a < b:
   c = a
   a = b
   b = c
```

C:

```
if (a < b) {
   c = a;
   a = b;
   b = c;
   }
```

Вложенные условные операторы

Задача. В переменной **a** записан возраст Антона, а в переменной **b** – возраст Бориса. Определить, кто из них старше.

? Сколько вариантов ответа?

```
if a = b then
    writeln('Одного возраста')
  else
    if a > b then
      writeln('Андрей старше' )
    else
      writeln('Борис старше' );
```

вложенный условный оператор

else относится к ближайшему **if**

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:

возраст ≥ 25 **возраст ≤ 40**

Можно ли решить известными методами?

Плохое решение

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

вложенный условный оператор

Хорошее решение (операция «И»)

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

сложное условие

! Каждое условие – в скобки!

Примеры

Задача. Вывести 'Да', если число в переменной a – двузначное.

```
if (10 <= a) and (a <= 99) then
  write('Да');
```

Задача. Вывести 'Да', если число в переменной a – двузначное и делится на 7.

```
if (10 <= a) and (a <= 99)
             and (a mod 7 = 0) then
  write('Да');
```

Сложные условия

Задача. Самолёт летает по понедельникам и четвергам. Ввести номер дня недели и определить, летает ли в этот день самолёт.

Особенность: надо проверить, выполняется ли **одно из двух** условий:

день = 1 **день = 4**

```
if (d = 1) or (d = 4) then
    write('Летает')
  else
    write('Не летает');
```

сложное условие

Ещё пример

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'). Использовать «ИЛИ».

```
if (x < 25) or (x > 40) then
  write('Не подходит!')
else
  write('Подходит.');
```

Простые и сложные условия

Простые условия (отношения)

равно

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

не равно

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

- **И** – одновременное выполнение условий

```
x >= 25 and x <= 40
```

- **ИЛИ** – выполнение хотя бы одного из условий

```
x <= 25 or x >= 40
```

- **НЕ** – отрицание, обратное условие

```
not (x > 25)    ⇔    x <= 25
```

Порядок выполнения операций

- выражения в скобках
- НЕ (not)
- И (and)
- ИЛИ (or), исключающее ИЛИ (xor)

```
    4     1     6     2       5     3
if not(a > 2) or (c <> 5) and (b < a) then
   ...
```

Сложные условия

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

```
not (a > b)
```
Да

```
(a < b) and (b < c)
```
Да

```
(a > c) or (b > c)
```
Нет

```
(a < b) and (b > c)
```
Нет

```
(a > c) and (b > d)
```
Нет

```
not(a >= b) or (c = d)
```
Да

```
(a >= b) or not (c < b)
```
Да

```
(a > c) or (b > c) or (b > a)
```
Да