

1 курс

**ПЛАН – КОНСПЕКТ**  
проведения занятия по дисциплине «Информатика»

**Раздел 4. «Основы алгоритмизации и программирования.»**

**Тема 4.1: «Общие принципы построения базовых  
алгоритмических структур в среде программирования.»**

**часть 5**

Подготовил: преподаватель  
В.Н. Борисов

**Вопросы занятия:**

1. Логический тип данных Turbo Pascal 7.0.
2. Правила записи логических выражений.

**Время проведения занятия – 2 часа.**

**Первый вопрос: Логический тип данных.**

## Логические типы

- Основной логический тип данных - **boolean**. Величины этого типа занимают в памяти 1 байт и могут принимать два значения: true (истина) или false (ложь). Внутреннее представление значения false — 0 (ноль), значения true — 1.
- Для совместимости с другими языками определены типы **ByteBool**, **WordBool** и **LongBool** длиной 1, 2 и 4 байта соответственно. Истинным в них считается любое отличное от нуля значение.
- К величинам логического типа применяются *логические операции* **and**, **or**, **xor** и **not** и *операции отношения*.

# Логический тип и логические выражения (BOOLEAN)



- Для логических переменных определены операции: NOT, AND, OR, XOR. В QBasic определены еще две логических операции: логическое следование IMP и эквивалентность EQW.
- Логический тип, как и целые типы, относятся к порядковым типам. Порядковый тип - это тип данных, для которого определены отношения порядка, то есть для любого элемента можно определить последующий и предыдущий элемент.



---

## **Логический тип**

(Boolean, ByteBool, WordBool, LongBool)

представлен двумя значениями: (истина) и (ложь).

Он широко применяется в логических выражениях и выражениях отношения.

Над данными булевского типа определены операции отношения: =, >, <, <>, <=, >=, вырабатывающие результат логического типа

Типы ByteBool, WordBool, LongBool являются нововведением Turbo Pascal 7.0, они были введены для обеспечения совместимости создаваемых программ в Windows.

---

**BOOLEAN** – данные логического типа

В языке *Pascal* имеются две логические константы: **TRUE** (истина) и **FALSE** (ложь). Логическая переменная принимает одно из этих значений.

**Допустимые логические операции:**

**AND** – логическое умножение

**OR** – логическое сложение

**NOT** - отрицание

**XOR** – исключающее «или»

## Логические операции not, and, or

В языке Паскаль имеются логические операции, применяемые к переменным логического типа. Это операции not, and, or и xor.

Значения операндов		Результат операции		
X	Y	not X	X and Y	X or Y
False	False	True	False	False
False	True	True	False	True
True	False	False	False	True
True	True	False	True	True

---

Второй вопрос: Правила записи логических выражений.

## Выражения

- *Выражение* — правило вычисления значения.
- В выражении участвуют *операнды*, объединенные знаками операций.
- Операндами выражения могут быть константы, переменные и вызовы функций.
- Операции выполняются в соответствии с *приоритетами*.
- Для изменения порядка выполнения операций используются *круглые скобки*.
- Результатом выражения является значение определенного типа. Этот тип определяется типами входящих в него операндов.
- Величины, участвующие в выражении, должны быть *совместимых типов*.

$t + \sin(x)/2 * x$   
результат имеет  
вещественный тип

$a \leq b + 2$   
результат имеет  
логический тип

$(x > 0) \text{ and } (y < 0)$   
результат имеет  
логический тип

# Логический тип и логические выражения (BOOLEAN)



Порядок выполнения операций определяется скобками. Например, в логическом выражении `A OR B AND NOT (A OR B)` сначала выполняется заключенная в скобки операция `OR`, а затем операции `NOT`, `AND`, `OR`.

В языке Turbo Pascal 7.0 нет возможности ввода логических данных с помощью оператора `read`. Однако предусмотрен вывод значений переменных логического типа с помощью оператора `write`. В этом случае для идентификаторов `FALSE` и `TRUE` автоматически отводится по 6 позиций: две — перед словом `TRUE` и одна — перед `FALSE`.

# Логический тип и логические выражения (BOOLEAN)



Логические операции, операции отношения и арифметические операции часто встречаются в одном выражении. Причем отношения, стоящие слева и справа от знака логической операции, должны быть заключены в скобки, поскольку логические операции имеют более высокий приоритет. Вообще, в логическом выражении принят следующий приоритет операций:

NOT

AND, \*, DIV, MOD

OR, XOR, +, -

операции сравнения.



## Логические операции

С помощью логических операторов можно формировать сложные логические выражения. Логические операторы часто применяются по отношению к простым логическим выражениям.

В языке программирования Pascal предусмотрены следующие логические операции:

- **Конъюнкция (логическое умножение, пересечение) - and.** Выражение  $a$  and  $b$  дает значение true только в том случае, если  $a$  и  $b$  имеют значение true. Во всех остальных случаях значения выражения  $a$  and  $b$  дает false.
  - true and true = true
  - true and false = false
  - false and true = false
  - false and false = false

- **Дизъюнкция (логическое сложение, объединение) – or.** Выражение  $a$  or  $b$  дает значение `false` только в том случае, если  $a$  и  $b$  имеют значение `false`. Во всех остальных случаях результат – `true`.

- `true or true = true`

- `true or false = true`

- `false or true = true`

`false or false = false`

- **Отрицание (инверсия) – not.** Выражение `not a` имеет значение, противоположное значению  $a$ .

- `not true = false`

`not false = true`

- **Исключающее ИЛИ – xor.** Выражение  $a$  xor  $b$  дает значение `true` только в том случае, когда только один из операндов имеет значение `true`.

- `true xor true = false`

- `true xor false = true`

- `false xor true = true`

`false xor false = false`

Последовательность выполнения логических операторов: `not`, `and`, `or`.

В языке Паскаль сначала выполняются логические операторы (`and`, `or`, `xor`, `not`), а уже потом операторы отношений (`>`, `>=`, `<`, `<=`, `<>`, `=`), поэтому не нужно забывать расставлять скобки в сложных логических выражениях.

```

C:\FPC\2.4.0\bin\logic2.pas
var
  k, l, m, n: integer;
  b1, b2, b3, b4, b5, b6: boolean;

begin
  write ( 'Введите четыре числа: ' );
  readln (k, l, m, n);

  b1 := (k < l) and (k < n);
  b2 := (k < l) or (k < n);
  b3 := (k = 0) or (l = 0) or (m = 0) or (n = 0);
  b4 := not ( k +_l > m + n);

  writeln ( 'Первое меньше второго и четвертого. ', b1);
  writeln ( 'Первое меньше второго или четвертого. ', b2);
  writeln ( 'Одно из чисел равно нулю. ', b3);
  writeln ( 'Сумма первой пары не больше суммы второй ', b4);

readln
end.
12:20

```

```

Running "c:\fpc\2.4.0\bin\logic2.exe "
Введите четыре числа: 4 7 6 0
Первое меньше второго и четвертого. FALSE
Первое меньше второго или четвертого. TRUE
Одно из чисел равно нулю. TRUE
Сумма первой пары не больше суммы второй FALSE

Running "c:\fpc\2.4.0\bin\logic2.exe "
Введите четыре числа: 2 6 8 15
Первое меньше второго и четвертого. TRUE
Первое меньше второго или четвертого. TRUE
Одно из чисел равно нулю. FALSE
Сумма первой пары не больше суммы второй TRUE

```

Сложные булевы выражения могут не обрабатываться до конца, если продолжение вычислений не изменит результат. Если булево выражение в обязательном порядке нужно обрабатывать до конца, то это обеспечивается включением директивы компиляции {B+}.

## Стандартные функции, возвращающие булевы значения

- **odd(x) = true**, если  $x$  нечетный ( $x$  целый);
- **eoln(x) = true**, если встретился конец строки текстового файла  $x$ ;

- **eof(x) = true**, если встретился конец файла *x*.

В остальных случаях эти функции принимают значение **false**.