

## Теория для Занятия №13.

### Логический тип данных

**Логический тип** – это простой стандартный порядковый тип, предназначенный для хранения логической переменной, которая принимает одно из значений: **True** (истина) или **False** (ложь).

Описание переменных логического типа:

Var

Flag,t,p,q : Boolean;

#### Операции, определенные над логическими переменными

1. Операция присваивания  
t:=false; p:=t; q:=5<x; flag:=succ(t);
2. Операция вывода.  
Write (t); WriteLn(q); Write (5>2)

**Замечание:** Величины логического типа можно присваивать, выводить, но нельзя вводить процедурой *read*.

#### 3. Логические (булевские) операции

Операция	Пояснение
and	Логическое умножение (конъюнкция)
or	Логическое сложение (дизъюнкция)
not	Логическое отрицание (не)
xor	Логическое деление

Приведем определения логических операций.

**Отрицание.** Если логическая величина С является отрицанием логического выражения А, то С истинно, если А ложно, и ложно, если А истинно.

**Логическое умножение.** Если А и В истинны, то С также истинно. Если же хотя бы одно из них ложно, то С также ложно.

**Логическое сложение.** Если А и В ложны, то С также ложно. Если же хотя бы одно из логических выражений А и В истинно, то С также истинно.

**Исключающее или** (иногда эту операцию еще называют *все что угодно, но не все сразу*). Это логическая операция, устанавливающая соответствие между логическими выражениями А и В и логической величиной С следующим образом: С ложно, если А и В либо одновременно истинны, либо одновременно ложны.

Упомянутые ранее определения логических операций можно также описать в виде *таблиц истинности*.

*Таблица. Таблица истинности для всех логических операций*

A	B	not A	A and B	A or B	A xor B
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

Примеры логических выражений:

(a and b) or (b and not c)

(a xor b) and ( a or c) xor (not h)

*Логическое выражение* — это такое выражение, которое может принимать только два значения: истина (true) и ложь (false). В языке Паскаль логическими выражениями могут быть:

- специальные логические переменные (тип **Boolean**);
- выражения, содержащие сравнения (например,  $s < > h$ );
- сложные выражения, содержащие выражения двух описанных ранее типов, соединяемых знаками логических операций и скобками.

Еще несколько примеров сложных логических выражений:

$(a < > b)$  and (f or (h<5))

f xor (5=6\*i)

$(s < > 6)$  or (g>8)

### **Замечания:**

1. Логические выражения представляют собой мощный математический аппарат, позволяющий проверять несколько элементарных условий в одном операторе. В их отсутствие для использования условного оператора пришлось бы осуществлять проверку сложного условия как проверку нескольких элементарных условий, для каждого из которых потребовался бы собственный оператор условия. Например:

```
if (y<7) and (h>5) then h:=y;
```

Этот же оператор, без применения логических операций, можно записать так:

```
if (y<7) then if (h>5) then h:=y;
```

Видно, что вторая запись длиннее и сложнее. Точно так же более просто организуются (с применением логических операций) и циклы по условию.

Еще одна очень серьезная выгода связана с наличием в языке Паскаль специального типа данных (логического типа). Этот тип данных позволяет создавать и вычислять логические выражения в операторе присваивания аналогично тому, как это делается с арифметическими выражениями. Например выражение:

```
if t=6 then y:=true else y:=false;
```

вполне можно заменить следующей более простой конструкцией:

```
y:=t=6;
```

2. Логические операции, операции отношения и арифметические операции часто встречаются в одном выражении. При этом отношения, стоящие слева и справа от знака логической операции должны быть заключены в скобки, так как логические операции имеют более высокий приоритет.

1. Действия в скобках
2. NOT
3. AND, умножение (\*), div, mod, деление (/)
4. OR, XOR, сложение (+), вычитание (-),
5. <, >, =, <>. <=, >=

Например:

A or B and not (A or B)

4. Определены следующие функции:

Функция	Аргумент	Результат	Назначение
<b>Succ(x)</b>	логический	логический	Succ(false)=true, Succ(true) не определена
<b>Pred(x)</b>	логический	логический	Pred(true)=false, Pred(false) не определена
<b>Odd(x)</b>	целый	логический	Проверяет нечетность значения <b>x</b> . <b>odd (x) = true</b> , если <b>x</b> нечетно;
<b>Ord(x)</b>	Порядковый (логический)	целый	Возвращает порядковый номер, соответствующий значению <b>x</b> . (Ord(False)=0, Ord(True)=1)

## Практическая часть

1. Наберите следующую программу и выясните что делают операторы Shl и Shr? Подставьте вместо числа 2 и 1 другие целые числа.

```
begin
writeln(2 Shr 1);
writeln(2 Shl 1);
end.
```

2. Наберите следующую программу и выясните, как работают лог.операторы с целыми числами(\$ff –число в 16-тиричной системе счисления) и почему получаются такие результаты? Убедитесь что  $-256 \text{ and } 256=0$   $-256 \text{ or } 256=-1$  и  $-256 \text{ xor } 256=-1$

```
begin
writeln(1365 and 2730);
writeln(1365 or 2730);
writeln(1365 xor 2730);
writeln(1365 and $ff);
writeln(1365 and $ff0);
writeln(1365 and $ff00);
end.
```

**Замечание:** Переведите эти числа в двоичную систему счисления, может это вам поможет понять, как работают лог.операторы с числами.

- 3.** Составьте программу, которая по введенным двум числам определяет, попадает ли точка с такими координатами внутрь окружности с радиусом 2 и центром в начале координат? (программу прислать!)