

# ПАСКАЛЬ

*Pascal*



Pascal ABC  
Ярлык  
1 КБ

*АВС*

Джордж Буль:

Логика

Для принятие программной средой решения, а не выполнение просто заданной программы используется ещё один тип переменных - **ЛОГИЧЕСКИЙ - BOOLEAN**

```
Program z ;  
Var H : integer ;  
Bool : Boolean;  
Begin  
.....  
End.
```

Использует два вида **БУЛЕВСКИХ ПЕРЕМЕННЫХ**:

**TRUE** – истина, когда условие выполняется

*Например:  $4 > 3$*

**FALSE** – ложь, когда условие не выполняется

*Например:  $2 > 3$*

# Булевские переменные в программе (присвоение)

```
program z66 ;
```

```
var X : integer ;
```

```
    bool : boolean ;
```

```
begin
```

```
X := 4;
```

```
    writeln ('X = ', X) ;
```

```
    bool := X > 3 ; {это утверждение истинно}
```

```
    writeln ('если X > 3, то ', bool) ;
```

```
    bool := X < 3 ; {это утверждение ложно}
```

```
    writeln ('если X < 3, то ', bool) ;
```

```
end.
```

**Только целые числа  
типа INTEGER**



```
X = 4
если X > 3, то True
если X < 3, то False
```

# Булевские переменные в программе (ручной ввод)



•Program1.pas

```
program z66 ;  
var X : integer ;  
bool : boolean ;  
begin  
readln (X) ;  
writeln ('X = ', X) ;  
bool := X > 3 ; {это утверждение истинно}  
writeln ('если X > 3, то ', bool) ;  
bool := X < 3 ; {это утверждение ложно}  
writeln ('если X < 3, то ', bool) ;  
end.
```

```
X = 15  
если X > 3, то True  
если X < 3, то False
```

z66

# Запись операций отношения и сравнения в Pascal

| Операция отношения | Запись в программе |
|--------------------|--------------------|
| Меньше             | <                  |
| Меньше или равно   | <=                 |
| Больше             | >                  |
| Больше или равно   | >=                 |
| Равно              | =                  |
| Не равно           | <>                 |

# Логические (булевы) операции

## 1. Логическое умножение (КОНЪЮНКЦИЯ)

Конъюнкция (логическое умножение)

определяется союзом «И» (в программе **AND**)

Например:  $X < 7$  **и**  $X > 3$

Согласно булевой алгебре, два логических выражения связанные между собой частицей «И» являются истинным только когда оба выражения истинны (истина – 1, ложь – 0)

| Операнд 1 | Операция   | Операнд 2 | Результат        |
|-----------|------------|-----------|------------------|
| TRUE (1)  | <b>AND</b> | TRUE (1)  | <b>TRUE (1)</b>  |
| TRUE (1)  | <b>AND</b> | FALSE (0) | <b>FALSE (0)</b> |
| FALSE (0) | <b>AND</b> | TRUE (1)  | <b>FALSE (0)</b> |
| FALSE (0) | <b>AND</b> | FALSE (0) | <b>FALSE (0)</b> |

# Логические (булевы) операции

## 2. Логическое сложение (ДИЗЬЮНКЦИЯ)

Дизъюнкция (логическое сложение) определяется союзом «**ИЛИ**» (в программе **OR**)

Например:  $X > 100$  **или**  $X < 10$

Согласно булевой алгебре, два логических выражения связанные между собой частицей «ИЛИ» являются ложными только когда оба выражения ложны (истина – 1, ложь – 0)

| Операнд 1 | Операция  | Операнд 2 | Результат        |
|-----------|-----------|-----------|------------------|
| TRUE (1)  | <b>OR</b> | TRUE (1)  | <b>TRUE (1)</b>  |
| TRUE (1)  | <b>OR</b> | FALSE (0) | <b>TRUE (1)</b>  |
| FALSE (0) | <b>OR</b> | TRUE (1)  | <b>TRUE (1)</b>  |
| FALSE (0) | <b>OR</b> | FALSE (0) | <b>FALSE (0)</b> |

# Логические (булевы) операции

## 3. Логическое исключающее сложение

Логическое исключающее сложение определяется

парой «**ЛИБО - ЛИБО**» (в программе **XOR**)

Например: **либо**  $X > 5$  **либо**  $X < 0$

Согласно булевой алгебре, два логических выражения связанные между собой частицей «**ЛИБО-ЛИБО**» являются истинными только когда одно из выражений истинно (истина - 1, ложь - 0)

| Операнд 1 | Операция   | Операнд 2 | Результат        |
|-----------|------------|-----------|------------------|
| TRUE (1)  | <b>XOR</b> | TRUE (1)  | <b>FALSE (0)</b> |
| TRUE (1)  | <b>XOR</b> | FALSE (0) | <b>TRUE (1)</b>  |
| FALSE (0) | <b>XOR</b> | TRUE (1)  | <b>TRUE (1)</b>  |
| FALSE (0) | <b>XOR</b> | FALSE (0) | <b>FALSE (0)</b> |



# Логические (булевы) операции

## 4. Логическое отрицание (ИНВЕРСИЯ)

Инверсия (логическое отрицание) определяется частицей «**НЕ**» (в программе **NOT**)

Например: **не** (X > 100)

Согласно булевой алгебре, результат операции противоположен отрицаемому утверждению (истина - 1, ложь - 0)

| Операнд   | Операция   | Результат        |
|-----------|------------|------------------|
| TRUE (1)  | <b>NOT</b> | <b>FALSE (0)</b> |
| FALSE (0) | <b>NOT</b> | <b>TRUE (1)</b>  |

# Приоритет логических операций:

| Приоритет<br>(последовательность) | Логическая операция |
|-----------------------------------|---------------------|
| 1. (самый высокий)                | <b>NOT</b>          |
| 2.                                | <b>AND</b>          |
| 3. (самый низкий)                 | <b>OR, XOR</b>      |

Например: **не А или не В и С**

1) **НЕ (А)**

2) **НЕ (В)**

3) **(не В) И (С)**

4) **(не А) ИЛИ (не В и С)**

Приоритет любой операции сравнения (<, >, =, пр.) ниже, чем у логической операции. Поэтому каждое сравнение в программе надо брать в скобки **2 < X < 4** записывают: **(2 < X) AND (X < 4)**

# Логические операции в программе

```
program z67 ;
var   X : integer ;
      Operand1, Operand2, RESULTAT : boolean ;
begin
  readln (X);
      writeln ('X = ', X) ;
  Operand1 := X > 3 ;
      writeln ('Operand1 = ', Operand1) ;
  Operand2 := X < 3 ;
      writeln ('Operand2 ', Operand2) ;
  RESULTAT := Operand1 and Operand2;
      writeln ('логическое умножение (И)= ', RESULTAT) ;
  RESULTAT := Operand1 or Operand2;
      writeln ('логическое сложение (ИЛИ)= ', RESULTAT) ;
  RESULTAT := Operand1 xor Operand2;
      writeln ('логическое исключающее сложение (ЛИБО-ЛИБО)= ', RESULTAT) ;
  RESULTAT := not Operand1;
      writeln ('логическое отрицание (НЕ)= ', RESULTAT) ;
end.
```

```
4
X = 4
Operand1 = True
Operand2 False
логическое умножение (И)= False
логическое сложение (ИЛИ)= True
логическое исключающее сложение (ЛИБО-ЛИБО)= True
логическое отрицание (НЕ)= False
```

Program z68 ; {составить выражения, определяющие,  
является ли студент первокурсником, получающим стипендию}

Var Price : Boolean ; {наличие стипендии у студента}

Kurs1: Boolean ; {является ли студент первокурсником}

Rezultat : Boolean ; {результат}

Begin

Kurs1 := True ; {пусть студент будет первокурсником}

Price := True ; {пусть студент получает стипендию }

Rezultat := Price and Kurs1 ;

WriteIn ('Студент первокурсник со стипендией? - ',  
Rezultat ) ;

Price := False ; {пусть студент не получает стипендию }

Rezultat := Price and Kurs1 ;

WriteIn ('Студент первокурсник со стипендией? - ',  
Rezultat ) ;

End.

z68

```
Студент первокурсник со стипендией? - True  
Студент первокурсник со стипендией? - False
```

# Основные операторы системы:

|  |                             |
|--|-----------------------------|
| Имя программы                                  | <b>program</b>              |
| Начало и Конец                                 | <b>Begin</b> и <b>End.</b>  |
| Переменные                                     | <b>VAR</b>                  |
| Целое число                                    | <b>Integer;</b>             |
| Вещественное число                             | <b>Real;</b>                |
| Вывод на экран                                 | <b>Write ('x = ', x);</b>   |
| Вывод на экран с новой строки                  | <b>Writeln ('x = ', x);</b> |
| Модуль числа                                   | <b>Abs (x);</b>             |
| Возведение в квадрат                           | <b>Sqr (x);</b>             |
| Квадратный корень числа                        | <b>Sqrt (x);</b>            |
| Синус числа                                    | <b>Sin (x);</b>             |
| Косинус числа                                  | <b>Cos (x);</b>             |
| Арктангенс числа                               | <b>Arctan (x);</b>          |
| Логарифм числа                                 | <b>Ln (x);</b>              |
| Возведение числа <b>E</b> в степень <b>X</b>   | <b>Exp (x);</b>             |
| Вычисление числа <b>Пи</b>                     | <b>Pi;</b>                  |
| Отбрасывает дробную часть числа                | <b>Trunc (x);</b>           |
| Округление дробного числа до ближайшего целого | <b>Round (x);</b>           |



# ЗАДАНИЕ:

Z66 – Z68) Набрать задачи Z66, Z67, Z68 по образцу в тетради и сохранить под этими названиями в своей папке

Z69) Определите в программе четыре логических переменных, которые содержат следующую информацию о людях:

**Married** – «истина», если человек женат (замужем)

**Blond** – «истина», если человек светловолосый

**Male** – «истина», если человек – мужчина

**Employed** – «истина», если человек работает

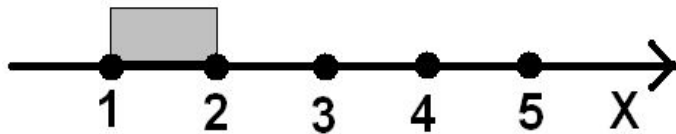
Составьте логические выражения, с помощью которых можно определить, является ли человек:

- 1) Замужней женщиной
- 2) Неженатым мужчиной
- 3) Незамужней блондинкой
- 4) Безработной незамужней женщиной
- 5) Либо неженатым, либо безработным, либо и тем и другим.

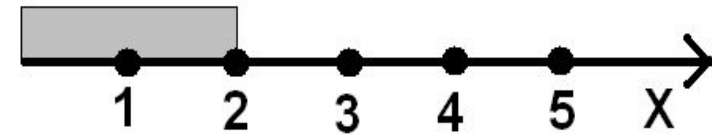
# ЗАДАНИЕ:

Z70 – Z75) Запишите на языке Паскаль выражение, которое истинно, если переменная  $X$  принадлежит заштрихованной области:

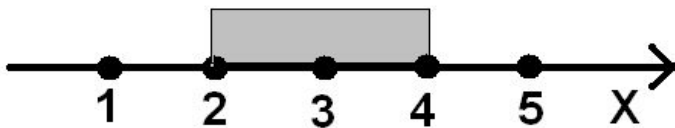
Z70)



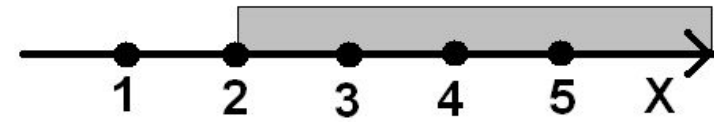
Z73)



Z71)



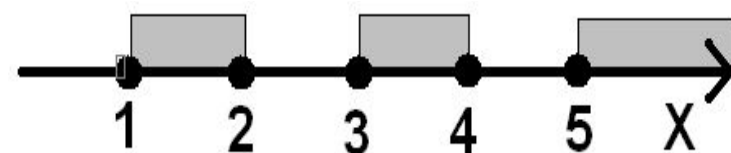
Z74)



Z72)



Z75)





# Литература:

- 
1. М. Э. Абрамян. Programming Taskbook. Электронный задачник по программированию. Версия 4.6./ Ростов-на-Дону - 2007 г.
  2. Ушаков Д.М., Юркова Т.А. Паскаль для школьников. СПб.: Питер, 2010. — 256 с.