

L#3

# Основы алгоритмизации и программирования.

## Введение

Course Basics of Programming Semester 1, FIIT

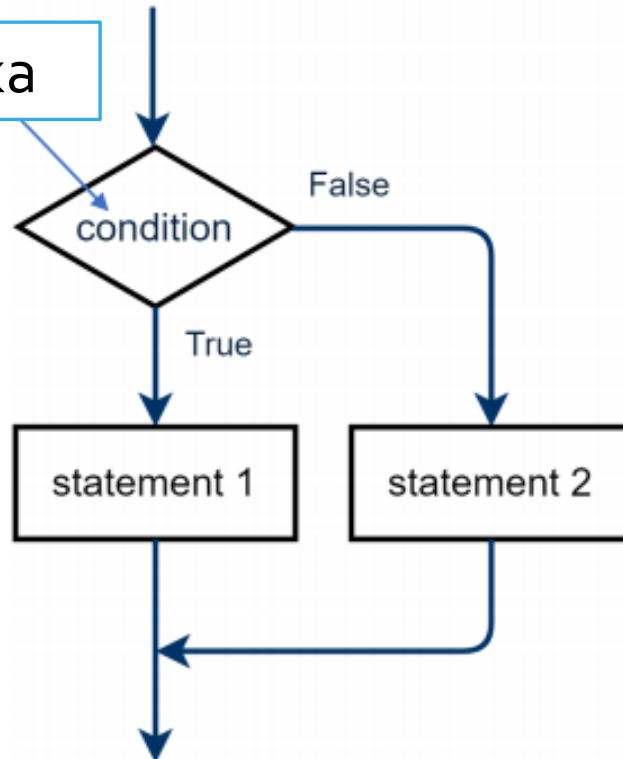
Mayer Svetlana Fyodorovna

# Условный оператор

Полная форма

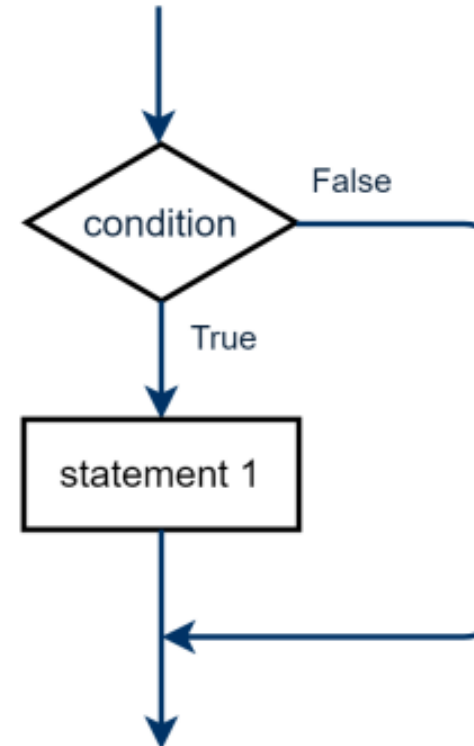
```
if Mark > 2 then  
    Print('OK')  
else Print('Bad mark');
```

проверка



Неполная форма

```
if Mark > 2 then  
    Print('OK');
```



# Примеры

- Значение переменной четное или нечетное?

```
if x mod 2 <> 0 then  
    Print('Odd')  
else Print('Even');
```

- Является ли число одноразрядным?

```
if n < 10 then  
    Print('One-Digit number');
```

# Логические выражения

```
if Mark > 2 then  
  Print('OK')
```

Выражение `Mark > 2` это вопрос

**Mark > 2 ?**

Ответом на вопрос будет значение **True** или **False**.

Мы говорим, выражение **Mark > 2** имеет логический тип.

Такой тип имеет только два значения **True** и **False**

- Можно вывести логическое выражение:

```
var mark := 2;  
Print($"is it true that mark >2? {mark > 2}") // output false
```

# Логический тип

- Переменная логического типа определяется, как **Boolean**:

```
var Mark := ReadInteger('Enter Mark:');  
var B: boolean;  
B := Mark > 2;  
if B then  
    Print('OK')
```

# Булевские операции: and, or, not

```
var A, B: boolean;  
A := True;  
B := False;  
Print (A and B);  
Print (A or B);  
Print (not A);
```

- **A and B** равно *True* если **A = True** и **B = True** одновременно. В остальных случаях **A and B = False**
- **A or B** равно *False* если **A = False** и **B = False**. В остальных случаях **A or B = True**
- **not A** принимает противоположное значение: **not A = True** если **A is False**

# Таблицы истинности

Truth tables give us a values of boolean operations  
A and B, A or B, not A  
when A and B take all possible values

| <b>A</b> | <b>B</b> | <b>A and B</b> | <b>A or B</b> |
|----------|----------|----------------|---------------|
| True     | True     | True           | True          |
| True     | False    | False          | True          |
| False    | True     | False          | True          |
| False    | False    | False          | False         |

| <b>A</b> | <b>not A</b> |
|----------|--------------|
| True     | False        |
| False    | True         |

# Примеры

```
var x := ReadReal;  
var B := (x > 3) and (x < 5);
```

$x \in (3, 5)$

```
var x := ReadInteger;  
if (x = 1) or (x = 3) or (x = 5) then  
  Print('x=1 or 3 or 5');
```

```
var x := ReadInteger;  
if not (x > 2) then  
  Print('x<=2');
```



# Вложенные условия If

**Задание:**

**Дано:** Point  $(x,y)$  on a coordinate plane,  
 $x \neq 0, y \neq 0$ .

**Найти:** номер координатной плоскости

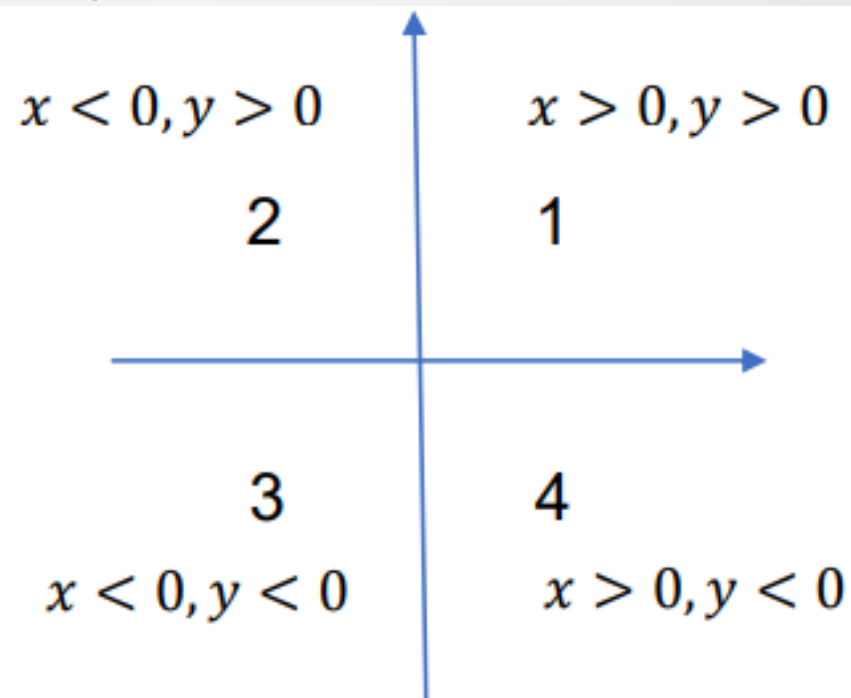
**Code:**

```
var (x,y) := ReadInteger2;  
var quarter: integer;  
Assert((x<>0) and (y<>0));  
if x>0 then
```

```
  if y>0 then  
    quarter := 1  
  else quarter := 4;
```

```
else
```

```
  if y>0 then  
    quarter := 2  
  else quarter := 3;
```



# Цепочка условий If

**Задание:** даны порядковые номера сезонов года

**Найти:** вывести название сезона по номеру

**Code:**

```
var Season := ReadInteger('Enter Season (1..4):');  
if Season=1 then  
    Print('Winter')  
else if Season=2 then  
    Print('Spring')  
else if Season=3 then  
    Print('Summer')  
else if Season=4 then  
    Print('Autumn')  
else Print('Wrong Season')
```

# Логические (boolean) операции в условном операторе **if**

- Условия могут состоять из логических операций: **not**, **or**, **and**. Если условие состоит из нескольких выражений, то каждое отдельное условие должно заключаться в скобки.
- Пример:

```
if (x < 20) or (y > 18)
then begin
  // if body
end
```

```
var a: = 5;
if (not (a < 4)) and (7 > 5) then // ← True
begin
  // if body
end
```

Если одно или оба условия = **True** тогда будет выполняться тело условного оператора.



# Оператор выбора

- Мы можем использовать оператор выбора, когда есть множественный выбор (много условий для одной переменной).

```
var Season := ReadInteger('Enter Season (1..4):');  
case Season of  
  1: Print('Winter');  
  2: Print('Spring');  
  3: Print('Summer');  
  4: Print('Autumn');  
  else Print('Wrong Season');  
end;
```

case **switch** expression

labels

# Case: другой пример

## Example. Translator

```
var Word := ReadString;  
var Translation: string;  
case Word of  
  'dog': Translation := 'собака';  
  'use': Translation := 'использовать';  
  'find': Translation := 'находить';  
  else Translation := 'не знаю 😊';  
end;
```

# Case: использование диапазона

**Задание:** даны порядковые номера сезонов года. Вывести название по номеру

```
var Month := ReadInteger;  
Assert((Month>=1) and (Month<=12));  
var Season: string;  
case Month of  
  3..5: Season := 'Spring';  
  6..8: Season := 'Summer';  
  9..11: Season := 'Autumn';  
  12,1,2: Season := 'Winter';  
end;
```

Ranges & enumerations must **not overlap**

enumeration

range



Q & A