Истинность составного высказывания

В информатике, кроме обычных математических операций (+, -, \*, /) существует отдельный тип операций, которые называются логическими операциями. Результатом выполнения данной операции является значение булевого (логического) типа данных.

Булев тип (boolean, bool) может принимать один из двух вариантов значений: истинное (true) или false (ложь).

Истинное значение, обозначается как true (в некоторых языках программирования - True), получается тогда, когда утверждение верно или, еще говорят, истинно. Например, выражение 5 > 3 имеет значение true.

Ложное значение, обозначается как false (в некоторых языках программирования - False), получается тогда, когда утверждение неверно или, еще говорят, ложно. Например, выражение 5 < 3 имеет значение false.

Значения true и false обычно используются в операторах условия (if...else...), либо в циклах с условием (while…, repeat...).

True, может быть представлен в виде 1,

False в виде 0.

Из примеров выше видно, что необходимо необходимо предварительно составить выражение, чтобы вычислить его истинность или ложность. Обычно, выражения составляются с помощью операторов больше (>), меньше (<), больше равно (>=), меньше равно (<=), проверки на равенство (==), проверки на неравенство или “не равно” (!= или <> в паскале).

При создании выражения данные операторы всегда возвращают булев тип, например:

|  |  |
| --- | --- |
| Выражение | Результат вычислений |
| 3 == 3 | true |
| 3 != 3 | false |
| 5 == 3 | false |
| 5 != 3 | true |
| 5 > 3 | true |
| 5 < 3 | false |
| 3 > 5 | false |
| 3 < 5 | true |
| 5 >= 3 | true |
| 5 <= 3 | false |
| 3 >= 5 | false |
| 3 <= 5 | true |
| 5 >= 5 | true |

Выражение может быть составлено только из одинаковых типов данных. Но, это не значит, что мы можем работать только с целыми числами, сравнивать можно вещественные числа, символы, строки, массивы, например:

“Привет” == “привет” вернет false.

Составные выражение

Рассмотренные ранее выражения являются простыми, чтобы составить сложное выражение, необходимо простые выражения связать между собой.

Логических связок существует несколько:

* and (в некоторых языках программирования - &&) - логическое И;
* or (в некоторых языках программирования - ||) - логическое ИЛИ.
* not (в некоторых языках программирования - !) - логическое отрицание - НЕ.
* xor - исключающее ИЛИ.

Оператор И

Логический оператор И имеет вид a and b, где a и b - некоторые выражения. Оператор проверяет булево значение выражений справа и слева и возвращает соответствующее булево значение.

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| true | true | true |
| true | false | false |
| false | true | false |
| false | false | false |

Оператор and называют так же “оператором логического умножения”. Действительно, давайте подставим 1 вместо true и 0 вместо false в таблицу выше.

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

1 \* 1 = 1

1 \* 0 = 0

0 \* 1 = 0

0 \* 0 = 0

Как мы видим, оператор and истинен тогда и только тогда, когда оба выражения истинны, во все остальных случаях он вернет ложь.

Пример:

5>3 and 4 < 6

5 > 3 - вернет true,

4 < 6 - вернет true

true and true - вернет true

Оператор ИЛИ

Логический оператор ИЛИ имеет вид a or b, где a и b - некоторые выражения. Оператор проверяет булево значение выражений справа и слева и возвращает соответствующее булево значение.

|  |  |  |
| --- | --- | --- |
| a | b | a or b |
| true | true | true |
| true | false | true |
| false | true | true |
| false | false | false |

Оператор or называют так же “оператором логического сложения”. Действительно, давайте подставим 1 вместо true и 0 вместо false в таблицу выше.

|  |  |  |
| --- | --- | --- |
| a | b | a and b |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

1 + 1 = 1

1 + 0 = 0

0 + 1 = 0

0 + 0 = 0

Как мы видим, оператор or ложен тогда и только тогда, когда оба его выражения ложны, во всех остальных случаях он вернет true.

Пример:

5 > 4 or 5 > 6

5 > 4 - true

5 > 6 - false

true or false - true.

Оператор НЕ

Оператор НЕ, в отличии от описанных выше операторов, выполняется над одним выражением и записывается как not a.

|  |  |
| --- | --- |
| a | not a |
| true | false |
| false | true |

Оператор not еще называют логическим отрицанием или инвертором значений.

|  |  |
| --- | --- |
| a | not a |
| 1 | 0 |
| 0 | 1 |

Пример:

not 5 > 4

5 > 4 - true

not true - false

Оператор исключающее ИЛИ

Логический оператор исключающее ИЛИ имеет вид a xor b, где a и b - некоторые выражения. Оператор проверяет булево значение выражений справа и слева и возвращает соответствующее булево значение.

|  |  |  |
| --- | --- | --- |
| a | b | a xor b |
| true | true | false |
| true | false | true |
| false | true | true |
| false | false | false |

|  |  |  |
| --- | --- | --- |
| a | b | a xor b |
| 1 | 1 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

Как видно из таблиц, оператор xor истинен тогда и только тогда, когда a и b имеют противоположные значения, во всех остальных случаях он ложен.

Пример:

5 > 4 xor 4 > 6

5 > 4 - true

4 > 6 - false

true xor false - false

Порядок проведения вычислений

Операторы and, or, not, xor можно комбинировать, составляя более сложные логческие цепочки.

1. Для начала необходимо вычислить значения a и b.
2. Если есть скобки, то действие первым выполняется в них.
3. Далее необходимо выполнить отрицание.
4. Потом необходимо выполнить логическое умножение.
5. Потом необходимо выполнить логическое сложение.

Например:

(5 > 4) and (not (6 < 4))

1. 5 > 4 - true
2. 6 < 4 - false
3. not false - true
4. true and true - true
5. Ответ - true.

Так же от скобок можно избавиться, т.к. мы знаем, что оператор not выполнится раньше оператора and:

(5 > 4) and not (6 < 4)

Выражение not (6 < 4) мы так же можем упростить, раскрыв скобки и заменив знак выражения на противоположный, то есть < поменять на >:

not (6 < 4) = 6 >= 4

Неравенство стало нестрогим, т.к. изначальное неравенство было строгим и “выкалывало” 6 из множества чисел, противный данному неравенству знак должен “включать” 6.

Таким образом, останется:

(5 > 4) and (6 >= 4)

5 > 4 - true

6 >= 4 - true

true and true - true

Ответ: true

Как мы видим, результат от наших действий не изменился.

Ниже представлена таблица противоположных знаков:

|  |  |
| --- | --- |
| Знак | Противоположный |
| > | <= |
| < | >= |
| >= | < |
| <= | > |
| == | != |
| != | == |

Итого

Логические выражения являются одними из самых распространенных в программировании, т.к. позволяют создавать ветвления в программах, в зависимости от значений, делая приложения более гибкими.

Логические выражения могут быть простыми или составными.

Простые логические выражения состоят из двух значений и оператора сравнения:

< - меньше, > - больше,<= меньше равно, >= - больше равно,== - равенства, != неравенства. Например, 4 < 5.

Составные выражения состоят из простых, связанных между собой логическими операторами and, or, not, xor.

Оператор and (логическое И) - логическое умножение, истинен тогда, когда два значения истины.

Оператор or (логическое ИЛИ) - логическое сложение, ложен тогда, когда два значения ложны.

Оператор not (логическое НЕ) - логическое отрицание, меняет значение выражения на противоположное.

Оператор XOR - исключающее ИЛИ, истинен тогда, когда выражения имеют разные значения.