# Lesson 7 – Conditional Statements

* Boolean Variables
* Relational Operators
* Logical Operators
* If Statement
* If-Else Statement
* If-Else-Else If Statement
* MAX Algorithms

What students should know

**4h**

## Logical Variables

So far, we have seen three types of variables integer, real, and string. As a type a logical variable is extremely simple because it can accept only two values: True – False which in fact internally on the computer are translated into states 1 and 0.

The declaration of a logic (from now on we will call it Boolean) becomes like the other simple variables we have known:

**Private** intDistance = 100, intTotalTravel = 0 As **Int**

**Private** blnFlag  **As**  **Boolean** = False

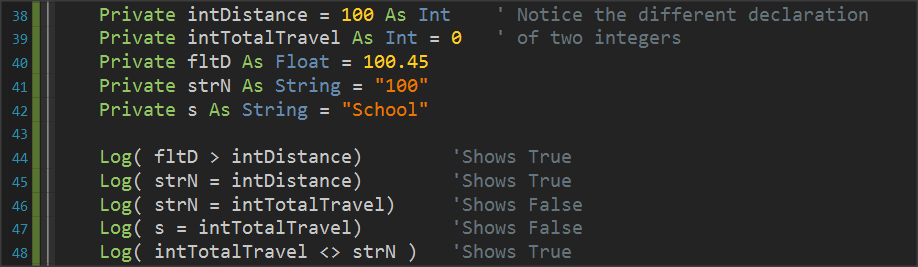
**Private** blnDone As **Boolean**

## Comparative Operators

Comparative operators are used to make comparisons between values used in programming languages. They are the known mathematical symbols of inequalities but when programming they are written in a slightly different way.

|  |  |  |
| --- | --- | --- |
| **Mathematical Symbol** | **B4X** | **Meaning** |
| = | = | Equality |
| ≤ | <= | Smaller or Equal |
| ≥ | >= | Greater or Equal |
| ≠ | <> | Different |
| < | < | Smaller |
| > | > | Larger |

Generally, to make a comparison you must compare variables or values of the same type. Eg. Integer values with integer values, real with real prices, etc. Also, in B4X you can compare numeric variables, such as integer variable and float, or Strings (strings) and numeric variables because internally the language converts strings into numbers.



Picture 12 Variable Comparisons

**Remember**

The result of a comparison is always a logical value of True or False.



## Logical Operators

Consider a sentence such as 'I am going to school now'.

Mathematician George Boole developed an algebra based on logical sentences.

In Mathematics and Mathematical Logic, Boolean Algebra is the algebra where the values of the variables can be either true or false. Usually represented by 1 and 0 respectively. Unlike elementary algebra where the values of variables are numbers and the main acts are addition and multiplication, in Boolean there are three main acts **And**, **OR** and **NOT**.

**Remember**

**AND** (conjunction), denoted x AND y, satisfies x AND y = 1 if x = y = 1, and xANDy = 0 otherwise.

**OR** (disjunction), denoted x OR y, satisfies x OR y = 0 if x = y = 0, and x OR y = 1 otherwise.

**NOT** (negation), denoted NOT x, satisfies Not x = 0 if x = 1 and Not x = 1 if x = 0.



Example:

**1st Sentence**: Today it rains. **2nd Sentence**: *I am going to school*.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Today it's raining.**  **(P1)** | ***I am going to school.***  **(P2)** | **P1 AND P2** | **P1 OR P2** | **NOT P1** |
| True | True | True | True | False |
| True | False | False | True | False |
| False | True | False | True | True |
| False | False | False | False | True |

From the table we observe that

• Comparing the two sentences using the logical AND evaluates to the value TRUE only when BOTH are true : It is raining (true) / I am going to school (true).

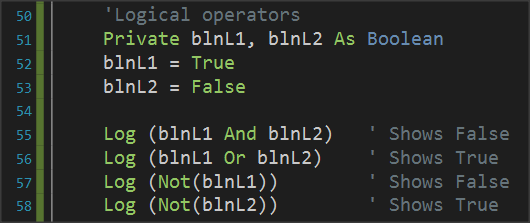
• Comparing the two sentences using the logical OR evaluates to the value TRUE when any one sentence is true : It is raining (true) / I am NOT going to school (false). Similarly : It is NOT raining (false) / I am going to school (true).

• The logical NOT reverses/converts true to false.

### Logical operators in programming

Logical operators are used in programming to create complex comparative expressions. This helps the developer optimize their code with fewer lines and simpler code.

In B4X logical variables are used as below.



Picture 13 Use of Logical Acts

Notice that logical operations must link logical variables or logical expressions as we will see later.

### Examples of evaluation of logical sentences.

The following variables are given with their values:

**Private** intA = 10, intB = 20, = 30 **As** **int**

**Private** strName1 = “George”, strName2 = “Georgia”  **As**  **String**

**Private** blnA = True, blnB = False blnC = False **As** **Boolean**

Calculate the value of the following logical expressions.

1.

|  |  |  |
| --- | --- | --- |
| blnA | **AND** | blnB |
| True |  | False |
|  | **False** |  |
|  |  |  |

2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Inta | **>** | intC | **AND** | blnA |  |
| 10 |  | 30 |  | True |  |
|  | False |  |  |  |  |
|  |  |  | **False** |  |  |

3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| intA + intB | **>=** | intC | **AND** | (blnA | **OR** | blnB) |
| 10 + 20 |  | 30 |  | True |  | False |
|  | True |  |  |  | True |  |
|  |  |  | **True** |  |  |  |

4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| intA + intB | **>=** | intC | **OR** | (blnA | **AND** | blnB) |
| 10 + 20 |  | 30 |  | True |  | False |
|  | True |  |  |  | False |  |
|  |  |  | **True** |  |  |  |

5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| strName1 | **=** | “George” | **OR** | strName2 | **=** | “John” |
| George |  | George |  | Georgia |  | John |
|  | True |  |  |  | False |  |
|  |  |  | **True** |  |  |  |

## If statement

Often as in life we ask questions so in programming there is a need for the developer to ask questions to check values or change the continuity of the program in different directions.

The **if statement** is used to make the corresponding questions:

Its basic form is as follows:

**If** ( condition ) **Then**

Commands

**End If**

Where condition appears in this example it would be replaced by a comparative or logical expression as studied above.

The meaning is: If condition is **TRUE** execute the commands between Then and End If

Examples:

**Private** intA = 10, intB = 20 **As Int**

**Private** fltA **As Float**

**If** intA > 0 **Then**

**Log**(intA & " is positive Number")

**End If**

**If** intA > 10 **Or** intB > 10  **Then**

**Log**("One or both numbers are greater than 10")

**End If**

**If** intA **Mod** 2 = 0 **Then**

**Log**(intA & " is Even number")

**End If**

## If – Else

The if-Else statement adds the ability to **if-else** to execute code if its condition is not true.

Its basic form is as follows:

**If** ( condition ) **Then**

Execute this code

**Else**

Execute this code

**End If**

The meaning is: If condition is **TRUE** execute the commands between Then and Else otherwise Execute the commands between Else and End If.

Examples:

**Private** intA = 10, intB = 20 **As**  **Int**

**Private** fltA **As Float**

**If** intA > 0 **Then**

**Log**(intA & " is possitive Number")

**Else**

**Log**(intA & " is not possitive Number")

**End If**

**If** intA > 10 **Or** intB > 10 **Then**

**Log**("One or both numbers are greater than 10")

**Else**

**Log**("None of the two numbers are greater than 10")

**End If**

**If** intA **Mod** 2 = 0 **Then**

**Log**(intA & " is Even number")

**Else**

**Log**(intA & " is Odd number")

**End If**

## If – else - else if

Multiple **if** further extends the functionality of an **if** statement by adding more than 1 control to the structure.

How to write:

**If** ( condition1 ) **Then**

Commands

**Else If** (condition2 )  **Then**

Commands

**Else If (** condition3 ) **Then**

Commands

**Else If (** condition4 ) **Then**

Commands

**...**

**Else**

Commands

**End If**

The function of the multiple **if** is summarized as follows:

1. The first condition evaluated, and if it is true, then the code it contains runs and the if is completed.
2. If the first **if** is false then the second is evaluated and if it is true it executes the code it contains and completed.
3. Other checks are always performed when the previous ones are false.
4. If none of the checks are true then the **else** statement executed.

**Example 3**

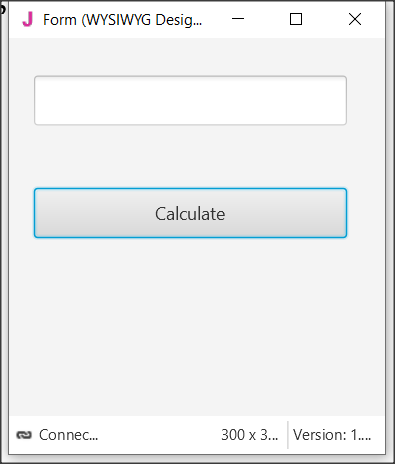
A fast-food chain has these meals:

|  |  |
| --- | --- |
| Meal | Price |
| Burger | 5$ |
| Pizza | 3$ |
| Hot Dog | 1,5$ |

Create a program that:

Reads the meal the customer wants. Prints the cost of the meal. Input example: "Hot Dog", Output: "Hot Dog 1,50$"

Solution



Picture The app screen

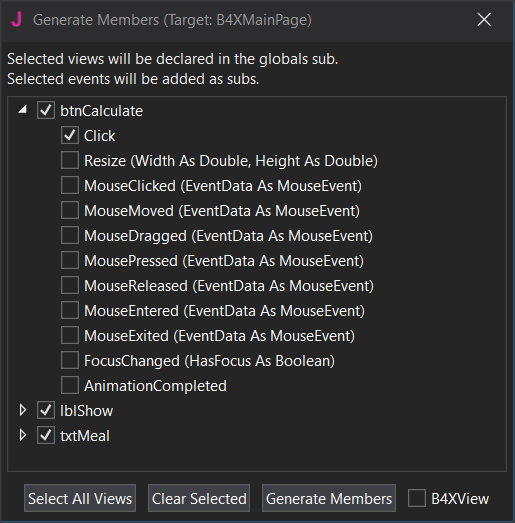
**Step 1**

Start a new project and give dimensions 300 x 300.

**Step 2**

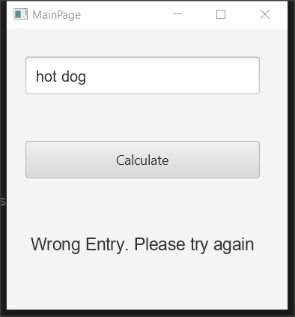
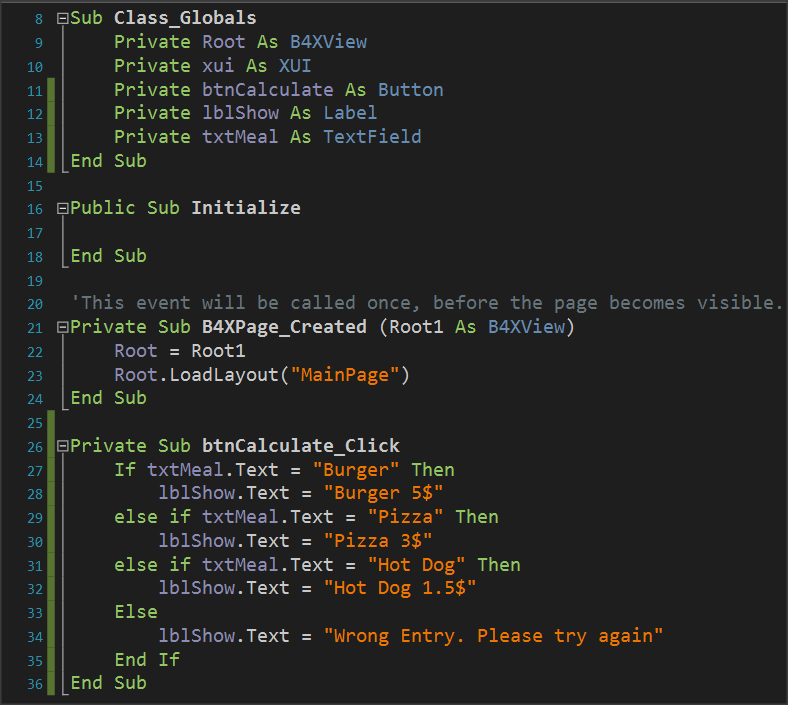
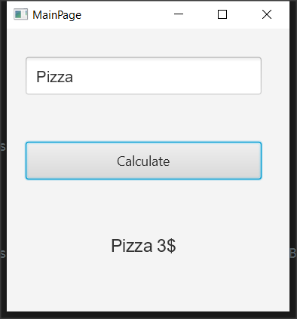
In the designer design the app screen

**Step 3**

****Enter txtMetal , btnCalculate, lblShow , and btnCalculate\_Click.

**Step 4**

The code you need to write is for btnCalculate\_Click



Note that the code considers capitals to be different from small letters. So Hot Dog is different from Hot Dog and HOT DOG and a user will have to input the correctly formated meal-name. Consider this when coding your program.

## Algorithms with if

Find Maximum Number

Read 3 integers and find the largest in three different ways

Method 3 – Max Algorithm

**Max = inta**

**If** intB > Max  **then**

MAX = intB

**End If**

**If** intC > Max  **then**

MAX = intC

**End If**

**Log(**Max**)**

Method 2 – Nested If

**If** Inta > intB **then**

**If** Inta > intC **then**

**Log**(Inta)

**End If**

**End If**

**If** IntB > IntA  **then**

**If** IntB > intC  **then**

**Log**(IntB)

**End If**

**End**

**If** IntC > IntA  **then**

**If** IntC > IntA  **then**

**Log**(IntC)

**End If**

**End If**

Method 1 – Simple If Statement

**If** Inta > intB  **AND** Inta > intC  **then**

**Log**(Inta)

**End If**

**If** intB > Inta  **AND** intB > intC  **then**

**Log**(intB)

**End If**

**If** intC > Inta  **AND** IntC > intB  **then**

**Log**(intC)

**End If**

## Exercises

1. Turn logical expressions into code.
   1. ‘A’ belongs to space [-5, 6).
   2. ‘A’ is less than 3 or more than 15.
   3. ‘A’ is equal to ‘B’ and ‘C’.
   4. ‘A’ does not have a value of 3.
   5. ‘A’ is less than 2 or ‘B’ is greater than 78.
   6. ‘A’ and ‘B’ true and ‘C’ false.
   7. the ‘A’ true and one of the ‘B’, ‘C’ are true.

1. What is the logical result (true or false) of performing the following operations if the following variables have the values below?

A = 10, B = 2, C = -4, D = 9 and E = 1

* 1. (A>B) OR (D=10)
  2. (D >= B) AND (E <> C)
  3. NOT (E<=C) OR (D<=C)
  4. NOT ((B<=C) AND (D<2))
  5. NOT (NOT (B<=E) OR NOT (C<=B))
  6. ((E<=A) and (E>=C)) AND NOT (C>=A)
  7. NOT (NOT (A >= 2) AND (C <>9))

1. A fast-food chain has these meals:

| **Meal** | **Price** |
| --- | --- |
| Burger | 5$ |
| Pizza | 3$ |
| Hot Dog | 1,5$ |

Create a program that:

Reads the meal the customer wants and second how many items of this meal ordered.

Prints the cost of the meal.

Input example: "Hot Dog", 2

Output: " 2 x Hot Dog 3$"

1. You have consumed X amount of Mbps on Wikipedia and Y amount of Mbps on memes. The cost of visiting Wikipedia is 0,10$ per Mb and the cost for watching memes is 0,05$ per Mb. If total consumption is more than 100$ print "Too much consumption". If watching meme consumption is greater than reading Wikipedia consumption print "WOW MANY MEMES", "SUCH LOL"(in new line). Create a program that:
   1. Reads X (Wikipedia Mb consumption) and Y(watching meme Mb consumption)
   2. Calculates the total consumption.
   3. If total consumption greater than 100$ print proper message If watching meme consumption is greater than reading Wikipedia articles print appropriate messages
2. An internet cafe has 2 ways of charging. If the user is a member pays 2$/hour, Else the user pays 5$. Find if someone is a member or not and calculate the price based on how many hours the user spent. If the user is a member the tax is 10% else the tax is 20%. Create a program that:
   1. Reads how many hours the user spent.
   2. Check if is a member.
   3. Add the proper tax fee.
   4. Print the total amount the user must pay Output: "The user is a member stayed 2 hours for 2$/hour plus the 10% the total amount is 4.4$"
3. You want to buy something from Amazon. The seller charges different prices for shipping cost based on location. For US it's 5$ for Europe it's 7$ for Canada it's 3$ for other places it's 9$. Create a program that:
   1. Reads the cost of the product.
   2. Reads your location.
   3. Print the amount of money you must pay.
   4. Output: "You have to pay 23$, 20$ for the product and 3$ for shipping cost".
4. A company sells a product for 0.70 € a piece if up to 200 pieces are ordered and for 0.50 € a piece if more than 200 pieces are ordered. Read the number of pieces ordered and calculate their value.
5. A cell phone company has the following billing policy.

|  | **Fixed cost 25$** |
| --- | --- |
| Call duration(in seconds) | Charge($/per second) |
| 1-500 | 0,01 |
| 501-800 | 0,008 |
| 801+ | 0,005 |

Create a program that:

* 1. Reads how many seconds was the calls duration.
  2. Calculates the monthly bill for the subscriber.
  3. Prints the total amount.
  4. Output: "total amount: 48$"
  5. Notice that the charge for the first 500 seconds is 0,01$ then for the next 501 to 800 seconds it's 0,008 and then it's 0,005$

1. In the qualifying races in the long jump at the Olympic Games, an athlete makes 3 initial attempts and if he has a performance of more than 7.50 meters, then he is entitled to continue and make another 3 more attempts. Read the first 3 attempts of an athlete and print a message whether he is entitled to continue or not and, if he is entitled to, find and and print the best effort of the athlete.

**You can use method Visible = True or False to hide or show labels, textFields and Buttons.**

1. In a municipality there are parking spaces for a short period of time. Parking charges are staggered, as shown in the table below:

|  |  |
| --- | --- |
| **Parking time** | **Charge per hour** |
| Up to 1 hour 3.50 € | |
| The next 2 hours | 8.00 € |
| The next 2 hours | 12.00€ |
| Over 5 hours | 15.00 € |

Make a program that reads the duration someone left his car in parking and calculates the total cost.