

## SESSION 6: MODULARIZATION (I)

---

### GOAL:

- To strengthen the concepts studied in theory about modular programming (organizing the code into methods).

### EXERCISE:

Implement a modularised program containing methods to perform each of the following tasks:

- Provide an integer in the range  $[LB, UB]$  read from the keyboard. The values of  $LB$  and  $UB$  will be the input parameters of the method and will meet the restriction  $LB \leq UB$ . The return value will be the integer read from keyboard, once it has been assured that it is in the range  $[LB, UB]$ .
- Provide a positive integer to be read from the keyboard. The return value will be the integer read from keyboard, once it has been assured that it is a positive one.
- Print on the output the content of a vector (one-dimensional array) of Boolean values passed as a parameter. The vector to be shown will be the only parameter in the method.
- Create a matrix (two-dimensional array) of  $N$  rows and  $M$  columns of Boolean values, initialized with the value  $B$  for each of its elements.  $N$  and  $M$  are positive values and  $B$  is a logical value, all of them passed as input parameters to the method. The method shall return the created array.
- Print on the output the content of a matrix (two-dimensional array) of Boolean values passed as a parameter. The matrix to be shown will be the only parameter in the method.
- Search for the amount of elements with a given value  $B$  in a matrix of Boolean values. Both the array and the logical value  $B$  will be input parameters to the method. The method will return the number of elements in the array that have that  $B$  value.

When executing the program, a menu will be displayed showing an option for each of the methods. The code to show the menu and to control which option the user selects can be included in an additional method in the program.