Modularity

Exercise 1.

“What is the value of the variable result at the end of the program when the variable nums refers to the array [1, 2] accepted by the routine *avg*? What about when the array is empty? Why?”

Answer. The value of the variable result is 1.0. An “int” value is divided by 2. Therefore, the decimal part is lost. The variable “sum” could be declared as “float” or “double” type to preserve the decimal part.

If the array is empty, there is no local variable “result” (the variable “result” is not initialized). Instead, an “EmptyArray” exception is raised.

“Modify the routine avg so that it only accepts arrays composed of non-negative numbers (x >= 0). If the array contains negative numbers, the routine should inform the client of all the indices where there are negative numbers in the array. The routine avg only informs the client, it does not print anything itself. Using this information, the client could do various things, but here wants to print a message "The X-th number Y in your array is invalid" for each case of a negative number. Here X would be the array index (starting from value 1) and Y the value at that index.”

Answer. The asked modification is done by introducing a new exception type “NegArray”. In the subprogram “avg” all negative numbers and their indices are written to two String type variables. These variables are attributes of the “NegArray” exception. The client (main program “main”) reads these values from the “NegArray” type exception object and stores them to two ArrayList type objects. Then, the elements of these lists are printed according to the demands of the exercise. The name of the attached program is “Modularity_e1.java”.

Modified version of the program uses ArrayList variables instead of String variables to convey the information to the client, “Modularity_e1_2.java”.