

EXERCISE 06

PROCEDURES & FUNCTIONS PART 2

Purpose of classes

Implementation of simple procedures and functions, exception handling.

Materials to be prepared

Materials placed on the course website: Lecture02 and Lecture03.

Tasks

Write procedures and functions described below.

1.	Procedure divide_1 is parameter less and divides the value entered in the selected cell by the number provided by the user. The number should be taken from the user from the InputBox function.
2.	Split the divide_1 procedure into two procedures and one function: • procedure read_values_1 has two parameters: dividend and divisor (in the first task dividend was taken from the active cell and divisor was taken from the input box displayed to the user), • function quotient_1 has two parameters (dividend and divisor) and performs the operation of dividing the first parameter by the second, • procedure divide_2 works in the same way as the divide_1, but it uses procedure read_values_1 and function quotient_1.
3	Procedure divide_3 is a modified, error-proofed, version of divide_1 . If the value in a spreadsheet cell or the value provided by the user is incorrect (they cannot be assigned to variables because they are not numbers), the following message should be displayed: <i>"The entered numerical values are incorrect or the division operation failed."</i>
4	Procedure divide_4 is a modified version of procedure divide_3 . It displays three error messages: • <i>No number was entered in the active cell on the worksheet</i> • <i>No number specified in the input box</i> • <i>Cannot perform division (division by 0)</i> Procedure should terminate after first error
5	Procedure divide_5 is a modified version of procedure divide_4 . In case of problems with: • reading the value from the active cell, asks the user (input box) to provide a new value, saves it to the spreadsheet and returns to the normal execution, • reading the value by which the active cell will be divided, displays an error message and returns to normal execution. Therefore, the procedure does not interrupt its operation if the active cell contains incorrect value or the user puts incorrect value, and it endlessly asks the user to provide correct values (tip: consider using the Resume and Resume Next statements).



6	Procedure read_values_2 is a modified version of procedure read_values_1. It displays two error messages: • <i>No number was entered in the active cell on the worksheet</i> • <i>No number specified in the input box</i> Procedure should terminate after first error
7	Function quotient_2 is a modified version of function quotient_1. It displays error message: • <i>Cannot perform division (division by 0)</i>
8	Procedure divide_6 works in the same way as the divide_1, but it uses procedure read_values_2 and function quotient_2. Check the behavior of the procedure in case of errors.
9	In order to eliminate the disadvantages of the solution from the previous task, write new versions of procedures and function: • procedure read_values_3 and function quotient_3 should raise an error after displaying the message, • procedure divide_7 could catch this error, preventing unnecessary calculations. <i>Hint:</i> To raise an error, use the Raise method of the Err object, the Raise method has one parameter, which is the number of the error being raised. The user should define his own errors using the structure: vbObjectError + □□□, where vbObjectError is the system constant and □□□ is the user error number.