SE116 – LAB#8 2023-2024 SPRING

Aim: Understanding the exception concept and the use of exception handling mechanisms.

Write a program to simulate basic arithmetic operations. Define an `ArithmeticOperation` class that contains two integer variables as its data members: `operand1` and `operand2`. Design a constructor that accepts these two members as its parameters to set the values of them. You also need to define setter/getter methods for the member variables.

Design four different methods for calculations as `add()`, `subtract()`, `multiply()`, `divide()`, for performing addition, subtraction, multiplication and division of two numbers, respectively.

For addition and subtraction operations, given numbers should be non-negative. If any negative number is entered, the program must throw an exception in respective methods. Therefore you need to design an exception handler of type `ArithmeticException` in `add()` and `subtract()` methods respectively, to check whether any number is negative or not.

For division and multiplication operations, neither of the numbers should be zero. If zero is entered for any number, the program must throw an exception in respective methods. Therefore you need to design an exception handler of type `ArithmeticException` in `multiply()` and `divide()` methods respectively, to check whether any number is zero or not.

Create a `main` method in a `Test` class, and construct four different objects from `ArithmeticOperation` class. Perform the operations of addition for the first, subtraction for the second, multiplication for the third and division for the fourth object. If a non-integer value is provided as input, the program must throw an exception of type `NumberFormatException` and display a message that informs the user about the wrong input. In such an exceptional case, the program should let the user re-enter a new input until its type is valid.