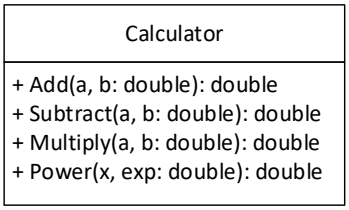
Lab Exercise: Hand-testing class Calculator

In this exercise, you will implement a class `Calculator` that can perform some very simple arithmetic operations. Then, you will test the class as well as you can by means of the methods tools you are used to use (i.e., hand-testing from Main()).

NOTE: We are going to use the class `Calculator` in the next lab exercise, so be sure to finish this exercise before class.

Exercise 1:
Create a C# Console application (.Net Core) project. This will also create a solution, which will contain the project, and other projects, if you add some later on. Keep only related projects in any one solution, e.g. the classes, apps and tests for one exercise or hand-in.

Implement the class `Calculator` according to the UML class diagram given below.



- `Add(a, b)` shall return the sum of a and b, `a + b`
- `Subtract(a, b)` shall return the difference of a and b, `a - b`
- `Multiply(a, b)` shall return the product of a and b, `a * b`
- `Power(x, exp)` shall return x raised to the expth power, x^{exp}

Exercise 2:
Test the class `Calculator` as well as you can.

The easiest way to do this is to add some code to the Main method for the application, which will create an instance of your class that you can then test.

As you do, consider the following:

1. How are you going to test the different operations?
   - Which different *tests* do you need for each operation?
   - What are the expected outputs of the tests?
   - How are you going to run the tests so that you can check that each test passes, i.e. that the output of the operation under test is as expected?

2. How well does your test method *scale*?
   - If you introduce many more tests or many more operations in `Calculator`, is your test method still practically usable?

3. What could make your life easier?
   - Considering the above (especially 2.), what could make your life easier? What would you require of a tool to help you?