

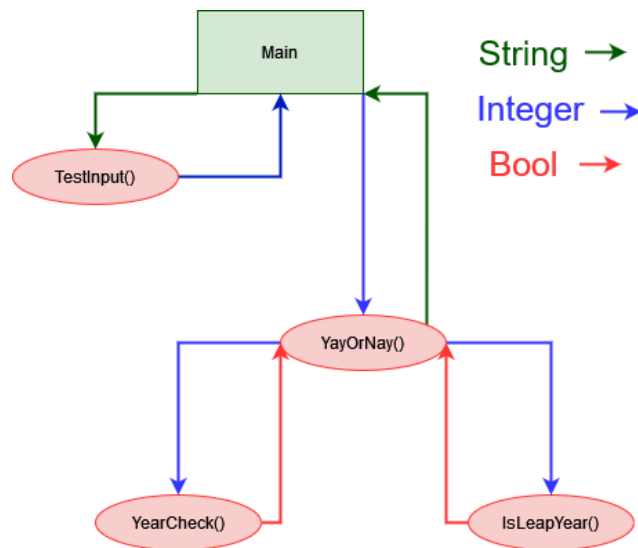
# BDSA Assignment 0

Viggo Starcke  
Software Development, IT University of Copenhagen

9th of September 2022

## A leap year algorithm

This is the project derived of a test-driven development of a C# console application project using Xunit. The simple function of this application is to determine whether an input is a leap year or not, handling errors and exceptions along the way.



UML diagram of the application

My leap year algorithm prompts the user to input a year earlier than the gregorian calendar reform when run.

To start off this input is stored in a *string* in the `Main()` to avoid errors.

This *string* is then parsed on to the `TestInput()`-method which returns the user input as an integral value, if it is convertible. If it is not convertible then the returned value is -1. (In the rare case where the user input is "-1" the `TestInput()`-method will handle this).

The output is stored in an *int* in the Main()-method, which is parsed on to the YayOrNay()-method. Here the input is first run against the YearCheck()-method, which simply returns a *bool* (*boolean*) value (is the year greater or equal to 1582 or is it not). If this *bool* comes back *false* a correctional error message in *string*-format is returned to the Main() which then prints it. If the *bool* however comes back *true* an if statement determines whether "yay" or "nay" should be returned in *string*-format. This is done by parsing the *int* from Main() to the IsLeapYear()-method which runs through all the conditions a year has to pass in order to be considered a leap year in relation to the gregorian calendar. Finally printing "yay", or "nay".