Advanced SWC – Mandatory Assignment

The assignment involves creating two classes **ObsInt** and **ObsIntExt**, to which several requirements are stated (see below). The solution **ObservableInt** – which can be found on *GitHub* in the folder **OOProg/ASWC/MandAssign** – should be used for solving the assignment, since the solution con­tains unit tests for the new classes.

# Requirements

You are supposed to implement a class **ObsInt** (short for *observable int*), which contains a single **int** property, but also makes it possible for clients to be notified about changes to the value stored in a specific **ObsInt** object. Below is an example of how **ObsInt** is supposed to be used:

**// Create an ObsInt object with the internal value initially set to 17.**

**ObsInt oiA = new ObsInt(17);**

**// This client wishes to be notified when the value in oiA changes.**

**ObserverA obsA = new ObserverA();**

**// Client now subscribes to changes in oiA.**

**oiA.ValueChanged += obsA.MethodA;**

**// Method MethodA on obsA will now be invoked.**

**oiA.Value = 35;**

In addition to this requirement, the below requirements must also be implemented:

1. Constructor for **ObsInt** must take one argument, with a default value of 0 (zero).
2. **ObsInt** must contain a property named **Value** of type **int**. Both the **get**- and **set**-part of the property must be implemented.
3. The operators +, - and \* must be overloaded. All three operators must work when applied to two **ObsInt** objects, but also when applied to one **ObsInt** object and one integer value.
4. It must be possible to implicitly convert an **ObsInt** object to an **int** value, and vice versa (Hint: look up in­for­ma­tion about the **implicit** operator).
5. **ToString** should return the value itself (as a **string**, of course).
6. It must be possible to call two methods **Total** and **Smallest** on a collection of **ObsInt** ob­jects which implements **IEnumerable<ObsInt>**. This should be implemented as exten­sion met­hods in a class named **ObsIntExt**. Both methods must return an **int**. **Total** must return the sum of the values in the **ObsInt** objects, while **Smallest** must return the smallest value in the **ObsInt** objects.

Note that the solution includes the project **ObservableIntUnitTest**. This project contains unit tests for both **ObsInt** and **ObsIntExt**. Your implementation of **ObsInt** and **ObsIntExt** should cause all unit tests to pass! The unit tests are initially commented out; uncomment the unit tests as you progress in your implementation of the classes.

# Work on the assignment

The intention is that you should be able to solve the entire assignment with an effort equivalent to one full “school day”, i.e. within 5-6 hours of work (some of you will probably be able to finish the assignment faster than this). You can work on the assignment whenever you wish, but note that the *ASWC* classes in **Week 17** – i.e. April 24th – will be reserved to working on the assignment. It is thus up to you to plan when to work on the assignment.

# Handing in your solution to the assignment

Send your solution directly to my e-mail ([perl@zealand.dk](mailto:perl@zealand.dk)), **not later than April 26th, kl. 13.00**. Send your solution as a **.zip** file, which only contains your final version of two files **ObsInt.cs** and **ObsIntExt.cs**. Do not send the entire solution folder! Once I have evaluated your solution, I will send the result of the evaluation to you by replying to the mail you sent me.

The deadline set above is a hard deadline. Remember that you must hand in a solution – and the solution has to be approved by me – in order to be able to attend the final 4th semester exam!