Advanced Object Oriented Programming, DT4014, VT24
Assignment 2 (due **April 19**)

Wojciech Mostowski
Halmstad University, ITE

Introduction

You are expected to work as a team in the group you were assigned. Please submit your asnwers through the Blackboard submission system. Include the ZIP file with your code and the short PDF report that summarises your work and answers any "textual" questions that were asked.

The exercises are taken from chapters 5 and 6 in the course book. The site

http://bcs.wiley.com/he-bcs/Books?action=index&itemId=0471744875&bcsId=2561

has solutions to selected exercises. You have to choose **Browse by chapter** and check the **Student Solutions Manual** for the chapter you are interested in. Please also note that the PDF of the 3rd edition of the book that we have available for the course may have slightly different exercise numbering than the book website. In this exercise sheet we always refer to exercise numbers from the **newer book**.

The deadline for assignment 2 is **April 19**. The exercises marked with `(*)` must be passed in order to pass the assignment.

Exercises

1. `(*)` During Lecture 5 we discussed the class `Signal` to illustrate the patterns *Observer* and *Strategy*. Combine the two versions so that the signal is observed by the two observers we described and the sampler can be set to be either the default or the sinus samplers. Write a test program that shows the two views of the signal with the sinus sampler.

2. Define a class for a cosinus sampler and a test program that shows the two views of the signal with the cosinus sampler.

3. There is a solution to Exercise 5.1 in the course book. Download it and test it.

4. `(*)` (Exercise 5.2 in the course book).

5. (Exercise 5.9 in the course book).

6. (Exercise 6.18 in the course book).

7. (Exercise 6.20 in the course book).

8. **(*)** In assignment 1 you programmed an interface `Filter` and a method `filter`. In this exercise, use the *Template* method pattern instead to define an abstract class `Filter` with a public method `filter` (the template method) that calls the method `accept` (the hook method) that can be implemented in different ways in the different concrete classes. Write a test program by extending the class `Filter` and defining accept so that only strings of at most three characters are accepted.

The paper *Frameworks in CS1 – a Different Way of Introducing Event-driven Programming* by Henrik Bærbak Christensen and Michael E. Caspersen explains how to use application frameworks in an introductory programming course. You can download the paper from

https://users-cs.au.dk/mec/publications/conference/03--iticse2002.pdf

In the paper they first introduce the `ImagePresenter` framework and later the more general `Presenter` framework.

The paper was written in 2002 and it uses an old version of the GUI framework in Java (the AWT framework). Try to use the more modern Swing framework in your programs.

9. **(*)** Implement the `Presenter` framework. Write a short report for the assignment. What classes are part of the framework? What does the application programmer need to do to write a complete application?

10. **(*)** Use it to implement a *slide show*: use the *horizontal buttons* to show images in the drawing area and texts in the text area. The texts should be *memory pads* for talking to the presentation. Write a short report for the assignment. What classes did you define to specialize the framework for this application?

11. Use it to implement the fifteen puzzle:

    http://en.wikipedia.org/wiki/15_puzzle

    Write a short report for the assignment. What classes did you define to specialize the framework for this application?

12. **(*)** During Lecture 7 we implemented a couple of visitors for the `Tree` composite structure. Download the finished code from Blackboard and add another visitor to pretty print trees on the console. It is most convenient to use the `A`-parametric version of the *Visitor* pattern.