IT-Universitetet i København

Analysis Design and Software Architecture

Assignment 03

Group 38

Jonas Lindvig	jonli@itu.dk
Daniel Lundin Lind	dlli@itu.dk
Mads Ljungberg	malj@itu.dk

September 29, 2021

C#

Repository link: `https://github.com/surtshow/Assignment_3`

Solve the following questions with extension methods (one-liners):

1. Flatten the numbers in xs.

```
public static IEnumerable<T> Flatten<T> (this IEnumerable<T>[] items)
   => items.SelectMany(x => x).ToList<T>();
```

2. Select numbers in ys which are divisible by 7 and greater than 42.

```
public static IEnumerable<int> Filter<T>(this IEnumerable<int> items)
    => items.Where<int>(item => item % 7 == 0 && item > 42);
```

3. Select numbers in ys which are leap years.

```
public static IEnumerable<int> LeapYearFilter<T>(this IEnumerable<int> items)
    => items.Where<int>(item => item % 400 == 0
    || item % 100 != 0 && item % 4 == 0);
```

Implement the following anonymous functions using the built-in delegates and lambda expressions:

1. A method which takes a string and prints the content in reverse order (by character)

```
public Action<string> ReverseString = s
    => Console.WriteLine(new string(s.Reverse().ToArray()));
```

2. A method which takes two decimals and returns the product.

```
public Converter<(double,double),double> Product = c => c.Item1 * c.Item2;
```

3. A method which takes a whole number and a string and returns true if they are numerically equal. Note that the string " 0042" should return true if the number is 42.

```
public Predicate<(string,int)> NumEqual = num => int.Parse(num.Item1) == num.Item2;
```

Software Engineering

Exercise 1

Research what a Kanban board is. Possibly starting from here, acquire sufficient application domain knowledge to write two as-is scenarios that represent the main usage of a physical Kanban board within a software engineering team. Note: make sure to cite your references in the submission document.

Scenario name	searchFunctionReadyForTesting
Participating actor instances	adam: SoftwareEngineer bob: SoftwareTestEngineer
Flow of events	1. Adam is finished with his work on the search functionality of the main program. 2. Adam moves his "Search functionality" sticky note from the "In progress" column of the kanban board to the "QA" column. 3. Bob goes and looks at the kanban board and sees that the "Search function" sticky note is in the "QA" column and can now begin testing the software.

Scenario name	wipLimitReached
Participating actor instances	adam, clara: SoftwareEngineer
Flow of events	1. Clara has finished a task an goes to look at the kanban boards "To do" column for a new task to begin working on. 2. Clara finds a task "Filter options" that she can begin working on. Clara checks if the in "In progress" column has enough space for her to begin this task. 3. Clara sees on the kanban board that the "In Progress" column has reached its limit on work-in-progress. She leaves the "Filter option" sticky note in the "To do" column. 4. Clara looks at the kanban board and finds a task in the "In progress" column, that Adam is working on. She joins him to help empty the "In Progress" column.

Sources for Kanban Borad:

- Rehkopf, M. (n.d.). What is a KANBAN BOARD? Atlassian. Retrieved September 29, 2021, from https://www.atlassian.com/agile/kanban/boards.
- Kanban Boardthe #1 way to improve the efficiency of your team. Kanban Tool. (n.d.). Retrieved September 29, 2021, from https://kanbantool.com/kanban-board.

Exercise 2

As a design activity (i.e., before writing the code): draw a class diagram of the objects representing your analysis of the application domain related to Exercise 1. The purpose of the diagram should be to *sketch* the main relationships between the entities and their multiplicity. To challenge yourselves, consider that often teams use the boards to also highlight task allocation (i.e., who is responsible for/will complete/has taken a card) and that nowadays pair programming or other group practices are used to implement code. How would you model this feature?

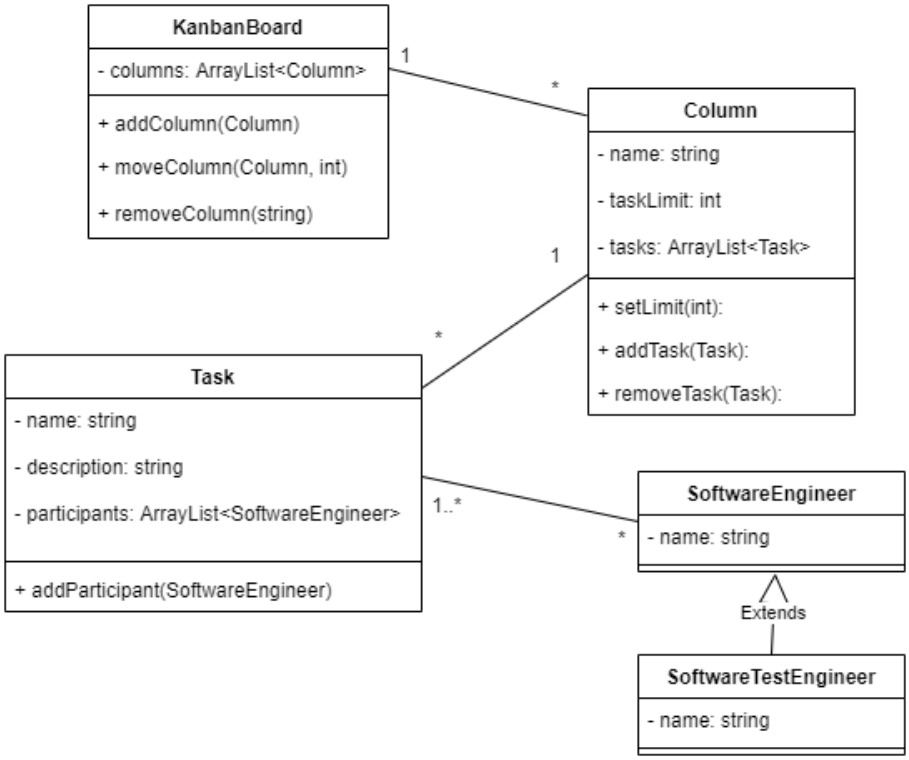

Figure 1: Kanban board class diagram

Exercise 3

1. The acronym FURPS+ stands for: ***FUNCTIONALITY***; ***USABILITY***; ***RELIABILITY***; ***PERFORMANCE***; ***SUPPORTABILITY***; + ***ADDITIONAL NEEDS***.

2. The requirements engineering process is an iterative process that includes requirements ***ELICITATION***, ***SPECIFICATION***, and ***VALIDATION***.

3. Software engineering is a collection of ***TECHNIQUES***, ***METHODOLOGIES***, and ***TOOLS*** that help with the production of a ***HIGH QUALITY*** software system developed ***WITH A GIVEN BUDGET*** before a ***DEADLINE*** while change occurs.

4. Requirements need to be complete, ***CONSISTENT***, ***CLEAR***, and ***CORRECT***.

5. Important properties of requirements are realism, ***VERIFYABILITY***, and ***TRACEABILITY***.

6. The output of the ***REQUIREMENTS ELICITATION*** activity is the ***ANALYSIS MODEL***, which include both the non-functional requirements and the functional model.

Exercise 4

What level of details should UML models have?
UML Models should be clear concise and contain enough information such that there exists only one interpretation of that given model.
What is the difference between structure diagrams and behavior diagrams in UML? Provide two examples per category.

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts. [1]
See figure: 5 For an example on a class diagram. Second example is a package diagram.

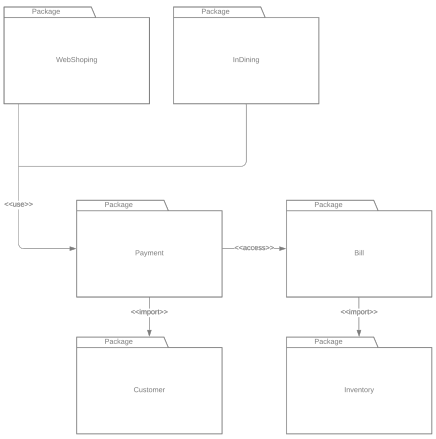

Figure 2: Package Diagram Example

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time. [1]
See figure: 4 for an example of a sequence diagram. Second example is an Activity diagram.

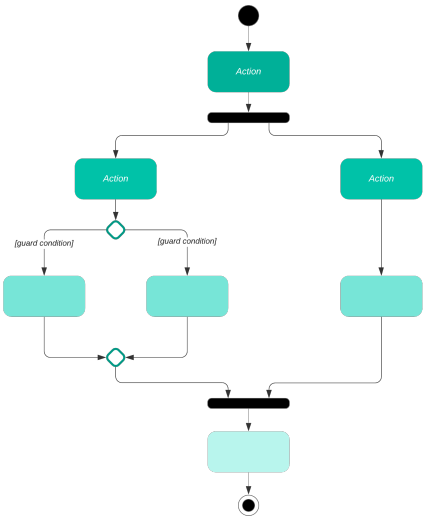

Figure 3: Activity Diagram Example

Exercise 5

Consider a file system with a graphical user interface, such as Macintosh's Finder, Microsoft's Windows Explorer, or Linux's KDE. The following objects were identified from a use case describing how to copy a file from a floppy disk to a hard disk: **File**, **Icon**, **TrashCan**, **Folder**, **Disk**, **Pointer**. Specify which are entity objects, which are boundary objects, and which are control (interactor) objects.

1. Entities: File, Icon
2. Interactors: TrashCan, Folder, Disk. ?
3. Boundary: Pointer.

Exercise 6

Assuming the same file system as before, consider a scenario consisting of selecting a file on a floppy, dragging it to Folder and releasing the mouse. Identify and define one control (interactor) object associated with this scenario.

1. MouseController.

Exercise 7

Arrange the objects listed in Exercises SE.5-6 horizontally on a sequence diagram, the boundary objects to the left, then the control (interactor) object

you identified, and finally, the entity objects. Draw the sequence of interactions resulting from dropping the file into a folder. For now, ignore the exceptional cases.

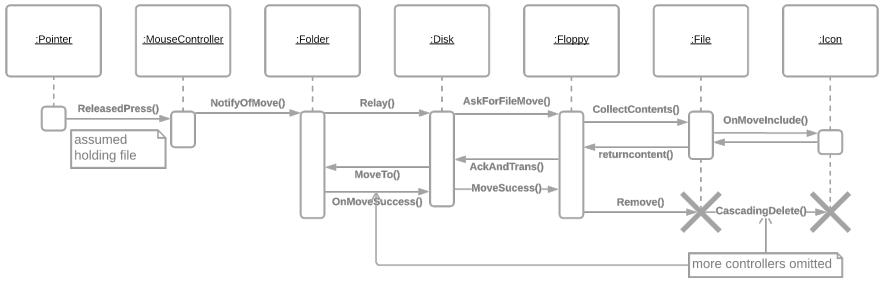

Figure 4: Seqence Diagram

Exercise 8

From the sequence diagram Figure 2-34, draw the corresponding class diagram. Hint: Start with the participating objects in the sequence diagram.

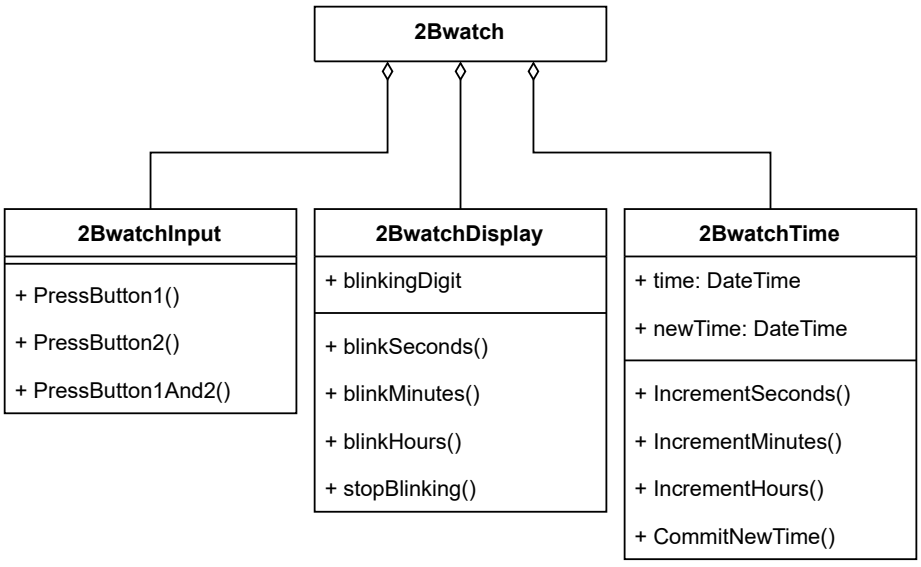

Figure 5: 2Bwatch class diagram

References

[1] uml-diagrams.org. *UML 2.4 Diagrams*. URL: https://www.uml-diagrams.org/uml-24-diagrams.html. (accessed: 29.09.2021).