# BDSA-Assignment1

kalw, anti, fret

September 2021

## 1 Generics

For the first method, 'T' must be of type 'IComparable¡T¿', where as for the second method, 'T' must be of type 'U, and 'U' must be of type 'IComparable¡U¿'. This means that the second method has one additional type constraint on U.

## 2 Iterators Regular Expressions

Follow the link to our GitHub:
`https://github.itu.dk/anti/BDSA-Assignment-01.git`

## 3 Software Engineering

### 3.1 Exercise 1

*What is meant by "knowledge acquisition is not sequential"? Provide a concrete example of knowledge acquisition that illustrates this.*

**Answer:**

It is not linear, in the sense that, through out the time span of a project or getting to know a system, we acquire new knowledge. This new knowledge can then invalidate our previous knowledge/understanding.

If we compare knowledge acquisition to a known concept such as the waterfall model where the process is much more sequential, it turns out that the classical waterfall model does not provide a great understanding of the main issues during the development. When working with the waterfall model, the process is already planned out and new initiatives won't be included, but working with iterative processes, the development becomes open to new ideas and solutions.

## 3.2 Exercise 2

*Specify which of the following decisions were made during requirements or system design:*

- *1. "The ticket distributor is composed of a user interface subsystem, a subsystem for computing tariff, and a network subsystem managing communication with the central computer."*
- *2. "The ticket distributor will use PowerPC processor chips."*
- *3. "The ticket distributor provides the traveler with an on-line help."*

**Answer:**

- These are decisions about the *design* of the software in the system
- This is a decision about *requirements* to the hardware in the system
- This is a decision about the *design* of the software in the system

### 3.3 Exercise 3

*In the following description, explain when the term account is used as an application domain concept and when as a solution domain concept:*
*"Assume you are developing an online system for managing bank accounts for mobile customers. A major design issue is how to provide access to the accounts when the customer cannot establish an online connection. One proposal is that accounts are made available on the mobile computer, even if the server is not up. In this case, the accounts show the amounts from the last connected session."*

- 1. *"Assume you are developing an online system for managing bank* ***accounts*** *for mobile customers"*

  Application domain - It describes a crucial part of the problem description

- 2. *A major design issue is how to provide access to the* ***accounts*** *when the customer cannot establish an online connection.*

  Application domain - The word "account" is in the context of the issue that should be solved.

- 3. *One proposal is that* ***accounts*** *are made available on the mobile computer, even if the server is not up.*

  Solution domain - Proposal of a solution to the problem

- 4.*In this case, the* ***accounts*** *show the amounts from the last connected session.*

  Solution domain - Consequence of implementation of mentioned solution

## 3.4 Exercise 4

*A passenger aircraft is composed of several millions of individual parts and requires thousands of persons to assemble. A four-lane highway bridge is another example of complexity. The first version of Word for Windows, a word processor released by Microsoft in November 1989, required 55 person-years, resulted into 249,000 lines of source code, and was delivered 4 years late. Aircraft and highway bridges are usually delivered on time and below budget, whereas software is often not. Discuss what are, in your opinion, the differences between developing an aircraft, a bridge, and a word processor, which would cause this situation.*

**Answer:**

Software can be very complex and is often difficult to describe in terms of "problem" and "solution". The development journey of a rather complicated construction such as a plane or a skyscraper might be long and tedious process, but it follows a rather sequential development.
Software development often(almost always) needs plenty of iterations and reconsidering regarding design, testing and implementation. This demanding development requires a lot more man hours and is often overlooked in other business areas.

## 3.5 Exercise 5

*Specify which of these statements are functional requirements and which are nonfunctional requirements:*

- *"The TicketDistributor must enable a traveler to buy weekly passes."*
- *"The TicketDistributor must be written in Java."*
- *"The TicketDistributor must be easy to use."*
- *"The TicketDistributor must always be available."*
- *"The TicketDistributor must provide a phone number to call when it fails."*

**Answer:**

- Functional
- Non-functional
- Non-functional
- Non-functional
- Functional

### 3.6 Exercise 6

*What is the purpose of modeling?*

**Answer:** The purpose of modeling is to provide a conceptual representation of the system in order to answer questions about the design of a solution. Modeling can be used to visualize the structure of a system by using standards like UML which ensures greater understandings throughout the whole development team.