Assignment 01

Magnus Dalsgaard Larsen (daml), Mie Milvang-Jensen (milv) and Lotte Juul Damgaard (loda)

September 15, 2021

GitHub repository

Link: https://github.com/BlueMango10/bdsa-assignment1

C#

Generics

Compare the following two methods:

```
int GreaterCount<T, U>(IEnumerable<T> items, T x)
where T : IComparable<T>

int GreaterCount<T, U>(IEnumerable<T> items, T x)
where T : U
where U : IComparable<U>
```

Both methods returns the amount of elements in `items` which are *greater than* `x`.

Explain in your own words what the type constraints mean for both methods.

1. The first method has no constraints on `U`, but `T` has to implement `IComparable<T>`.
2. The second method needs `U` to implement `IComparable<U>` and `T` needs to derive from *or* be the same type as `U`.

Software Engineering

Exercise 1

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

Gaining knowledge is not sequential, it is much more fluent in the sense that new information can "overwrite" old information.

Example: As a kid, you might have been told that Santa is real. Later, you find out he is not. This new information means that the old information is no longer valid.

Exercise 2

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

- "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."

  This is system design, since it describes how the system is build from subsystems.

- "The ticket distributor will use PowerPC processor chips."

  This is also system design, since it is a hardware specification, which is first decided in the system design process.

- "The ticket distributor provides the traveler with an on-line help."

  This is a requirement for the program, because it is a concrete function the program should implement.

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."

Application domain: First two
Solution domain: Last two

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.

There are several factors resulting in this. One element is the lack of visibility during the process, since it is not a physical product. Therefore, it is harder to know how far along the project is.

Another factor is that it is a fairly new field. Asserting time and budget can be difficult, when there isn't much experience to learn from.

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."

  Functional

- "The TicketDistributor must be written in Java."

  Nonfunctional

- "The TicketDistributor must be easy to use."

  Nonfunctional

- "The TicketDistributor must always be available."

  Nonfunctional

- "The TicketDistributor must provide a phone number to call when it fails."

  Functional

Exercise 6

What is the purpose of modeling?

Modeling helps visualize software, which can be difficult to understand and agree on without it. It also functions as documentation for the plans and ideas for the product.