# Lab 2 - Object-Oriented Software Development / ... with Design Patterns TOMK18 / TOUK18 - H22

Domain Model, Design Principles, Design Model

**last presentation opportunity: Thursday, 22 September 2022**
**submission deadline: Friday, 23 September 2022, 23:59**

Jönköping University, School of Engineering
Tuesday, 06 September 2022

## General instructions

In this lab we deal with the processes that occur after the first step of requirements gathering: further analysis with a domain model, then creation of a design model.

## Part A: Domain model

In this part you shall create a domain model. Relevant background for this task is lecture 2 and chapter 9 in *Applying UML and Patterns* (3rd edition).

Read now carefully, and possibly multiple times, the following domain description.

## 1 Domain description (part A)

Our domain is teaching at University. Our focus lies on students and teachers, and on the different tools and systems they use to support and organize their interaction. It follows a detailed description of this domain (inspired by the real-world, but simplified and cut off for the purpose of this lab).

On the one hand we have users such as students, teachers, administrators, and on the other hand we have a number of systems that interact with one another and that the users interact with. PingPong is a system supporting the processes of teaching and learning, and communication between students and teachers. Teachers upload on PingPong material such as study guides, exercises, lab assignment instructions, lecture material. Students retrieve this material from PingPong. Teachers create on PingPong assignments for the students. Students submit their solutions for assignments on PingPong, where teachers retrieve submissions and give feedback via PingPong. There are two exam systems for digital exams: either PingPong as well, or Inspera. Teachers create digital exams, students can sit digital exams and submit their solutions via the exam system. Teachers can retrieve students' solutions for exams and give feedback via the exam system. Grades for exams and other examination elements are entered by the teachers into the system Ladok. This is only possible, however, if administrators have made a course and its examination

elements available for the teachers in the Ladok system. The administrators take the information about a course's examination elements form the system Kursinfo. It is the teachers who specify courses and their examination elements by creating course syllabi (kurs plan) and uploading this information to Kursinfo. A schedule group prepares the schedules for courses in the system Kronox. Teachers can also edit the schedule in Kronox. A mobile application JU-App is used by students and teachers. JU-App extracts schedule information from Kronox and information about grades and credits from Ladok in order to present it to the user in a readable manner. Students use the system Studentwebben (Current student) in order to view the schedule (extracted from Kronox), information about grades and credits (extracted from Ladok) and to get the course syllabi (extracted from Kursinfo).

# 2 Your task (part A)

Create a domain model. In order to facilitate the work for you and your lab assistant, we recommend you follow the following incremental steps.

1. Do a noun analysis: identify potential conceptual classes. Decide which concepts should be present in your model.

2. Decide which conceptual classes should be modeled as a class, and which should just be attributes of classes. A rule of thumb is to model users and systems as classes (they have a behavior), and other entities as attributes.

3. Draw the classes. Add attributes where you deem it reasonable. Add associations, with names, reading directions and multiplicities.

4. Identify the core (most essential part) of the system: where should one potentially start designing if one was to implement the systems (or one of them)? Justify your answer.

# 3 Hints (part A)

You may need to go back in the steps and revise. Try to get your domain model as accurate as possible (fitting the text description). But do not forget the purpose of a domain model: understanding the domain. Do not overmodel or get lost in details that do not contribute to understanding the domain (at this stage, we do **not** model with the purpose of designing or implementing).

# Part B: Design model

Now we turn to the design phase. The relevant background for this task is found in lectures 2,3,4 and the corresponding readings.

Creating a (good) design model is a complex task and requires a lot of experience. Since we do not have this experience yet, we defer the creation of an actual design (model) to later labs. Here we just want to get more familiar with design models and with (detailed) UML-class diagramming. For this purpose we reverse the process: you are given a code base and shall reverse engineer the design model from it.

## 4 Given code base (part B)

Download the archive file *Lab02_chess.zip* and get familiar with the code. Either import it into Eclipse (*File->Import...->General->Existing Projects into Workspace, Select archive file*) or just unzip it and read the .java files with any text editor. You can also run the chess game by running main.java. You will experience an unfinished chess game started by Peter Larsson-Green. Several rules/moves are already implemented. However, your task is not to continue programming on the chess game. Just get familiar with the different classes and how they work together in order to achieve certain functionality.

## 5 Your task (part B)

Create a design model matching the given code base, in form of a UML-class diagram.

## 6 Hints (part B)

- Start drawing (all) the classes in boxes without any attributes or methods, and model (all) relationships between them.
- Rearrange the classes in order to reduce crossings of relationships.
- Add attributes and methods, with visibility modifiers and parameters.

## 7 Deliverables and presentation (A + B)

You shall deliver *one* pdf document that contains for part A your domain model and the results of steps 1 and 4, and for part B your design model. You need to orally present your solution to your lab assistant. Be prepared to answer any question regarding your solutions.

If you want to draw by hand, this is accepted, as long as it is **well-readable** and submitted within the pdf (scann it).