Lernjournal

## Woche 1

Heute war ein Einstieg in die Module 226a und 153. Für die LB2 Projektgruppe habe ich mich mit Dimitrios und Arman zusammengetan, da wir oft zusammenarbeiten. Dazu habe ich kurz zur Auffrischung einen Objektorientierten Taschenrechner umgesetzt. Dazu habe ich MySql Workbench installiert. Wir haben uns auch schon entschieden als Projekt einen Onlineshop umzusetzen.

## Woche 2

Heute hatten wir nochmal eine Einführung in Objektorientiertes Programmieren und die Unterschiede zur prozeduralen Programmierung. Hier habe ich vor allem gelernt, dass ich oft beides gemischt hatte und versuche dies nun zu vermeiden. Dazu habe ich die Generalisierung und Spezialisierung bei Datenbanken kennengelernt, was einfach zu verstehen war, da ich dies von Klassendiagrammen schon kenne. Der grösste Unterschied ist, dass man anstatt einfach nacheinander z.B etwas ausgibt, zwischen den Klassen hin und her geht und dort die Methoden implementiert sind. Dies steigert auch die Wiederverwendbarkeit.

## Woche 3

Heute habe ich neu gelernt, wie man Use Case Diagramme erstellt, Klassendiagramme habe ich schon gekannt doch konnte mein Wissen nochmal aufgefrischt. Zuerst hatte ich Probleme Rekursionen zu verstehen, doch inzwischen verstehe ich diese und weiss, wenn man eine brauchen kann. Non-Identifying und Identifying war in diesem Modul meiner Meinung nach das schwerste. Doch habe ich nun gelernt was der Unterschied ist und dass bei einer N zu M Beziehung automatisch eine Identifying Relationship erstellt wird. Das Stücklistenproblem kann ich nun auch identifizieren und weiss, wie man dieses mit einer N zu M Rekursion beheben kann. Das Stücklistenproblem entsteht bei einer Rekursion, wenn z.B ein Mitarbeiter 2 Chefs hat. Darum braucht man eine Zwischentabelle welche den Primary Key 2-mal beinhaltet, um beliebig viele Chefs zu erfassen.

## Woche 4

Heute hatten wir LB1, in der ich eine 5.7 geschrieben habe. Dazu haben wir Sequenzdiagramme kennengelernt, wobei ich Mühe hatte diese umzusetzen. Doch nachher habe ich das ganze verstanden als ich es selbst umgesetzt habe. Ich denke man braucht einfach Übung darin. Bei Modul 153 habe ich nicht viel neues gelernt, da wir heute nur das ERD angefangen haben. Sequenzdiagramme werden benutzt, um eine Visualisierung eines Prozesses zu haben. Z.B wer kommuniziert miteinander im Programm, wenn eine Bestellung ausgelöst wird, und was wird zurückgegeben. Dies dient zur Verständlichkeit für die anderen Entwickler im Team.

## Woche 5

Heute haben ich vorallem am Klassendiagramm gearbeitet mit Dimitrios, wo ich keine Probleme hatte. Dazu haben wir Junit tests angeschaut. Zuerst habe ich diese nicht verstanden, jedoch hat mir Dimitrios das ganze nachher im Projekt gezeigt und mich auch ein paar schreiben lassen, damit ich diese nun auch verstehe. In M153 hatten wir LB1 welche leider nicht so gut lief, ich war zwar vorbereitet aber hatte irgendwie ein Blackout bei dem Ganzen. Klassendiagramme sind wie Sequenzdiagramme zur Planung und Veranschaulichung eines Prozesses da. Damit jeder Entwickler im Team die Klassen und deren Funktionen kennt.

## Woche 6

Heute hatten wir als Repetition Javadoc welches ich schon gut kannte, aber hier nochmal auffrischen konnte. Dazu habe ich heute unser Use Case Diagramm noch angepasst, sowie unser Sequenzdiagramm. Dazu habe ich Transaktionen in SQL kennengelernt, ich wusste das sie existieren, aber nicht wie diese funktionieren. Darum fand ich dies sehr spannend und konnte lernen, wie das Ganze funktioniert. Javadoc ist eine Ergänzung zu den normalen Kommentaren, es erstellt sich automatisch, wenn man Kommentare dafür erstellt hat. Es Beschreibt Attribute, Methoden und sogar Klassen. Es gibt Parameter wie @Author mit denen man Authoren einer Klasse festlegen kann.

## Woche 7

Heute haben wir Collections und Java Interfaces angeschaut, welche ich schon kannte. Collections waren schwer zu verstehen, jedoch konnte mit Dimitrios diese gut erklären und ich verstehe diese nun. Dazu hatten wir angeschaut was NoSQL ist, dies war sehr spannend für mich da ich dies noch nicht kannte. Nun kenne ich diese und auch den Use Case von NoSQL DBs. Ich denke ich könnte eine NoSQL Datenbank nun umsetzen mit ein bisschen einarbeiten. Collections sind Sammlungen von Elementen/Daten und sind in vielen Formen vorhanden. Listen, Arraylisten oder sogar Treecollections existieren. Diese kann man mit einem Designpattern namens Iterator Pattern verbinden, welches durch Collections iteriert. Designpatterns sind einfach Lösungsvorschläge zu bestimmten Problemen in der Informatik. NoSQL Datenbanken sind nicht so wie SQL strukturiert. Man kann einfach Daten einlesen. Dabei wird die Konsistenz der Daten zwar schlecht. Jedoch bei grossen Datenbanken kann dieses viel schneller sein als SQL.

## Woche 8

Heute haben wir nurnoch an den Projekten gearbeitet. Ich habe das ERD nochmals überarbeitet so wie unsere Indexierungen und Löschkonzept. Dazu haben wir das Testscript ausgeführt und waren erfreut als alles so funktioniert hat, wie wir es wollten. Die Junit Tests funktionieren auch. Wir müssen nurnoch unser Sequenzdiagramm anpassen und wären sonst somit fertig.