**Seminar 3**

**Object-Oriented Design, IV1350**

**Adrian Kastrati,** [**kastrati@kth.se**](mailto:kastrati@kth.se)

**2022-05-04**

**Contents**

**1 Introduction x**

**2 Method x**

**3 Result x**

**4 Discussion x**

1. **Introduction**

The seminar task is to implement a design for program in the object-oriented language Java. Discussions of the task with Deni Persson, Jesper Munkeby and William Eriksson.

1. **Method**

The method for the implementation was writing code for the each of the classes with their attributes and methods from the design. During the process some of the classes and methods were altered since they could not work as intended in the design. I Tried to write code that had descriptive names for functions and objects/datatypes. The methods were written with keeping in mind that they only do one thing and not affect other instances of classes. The writing of the test controlled two of the classes and their methods, except simple getters and setter methods. The test were written to see how they behave when given the wrong or correct input.

1. **Result**

En bild som visar text

Automatiskt genererad beskrivning

[**https://github.com/adriankastrati/IV1350**](https://github.com/adriankastrati/IV1350)

The resulting code works and has items created in the inventory database in an array and item identifiers in the view to be able to search for different preset items. The Controller connects the different layers and handles the information sent between them through data transfer objects or simple primitive datatypes. The tests were for a CustomerPayment object and Sale object where they have different setters and other methods tested. The testing takes use of other instances of classes and primitive datatypes which shows that it works for the implementation and when it does not.

1. **Discussion**

The finishing code has a function that I now realize should have been split up in 2 different methods since it does take use of code that other methods could use if the program would scale or mature. Primitive data is used where it should and I tried to implement as many classes as possible to refactor the program, there are still some classes as, as Receipt that have more primitive data than the other classes, but I did that in the mindset of the data transfer object being sent across layers and it may help with understanding what it holds directly.

One more issue with the Receipt is that there are different arrays that have indexes conncected to one another, this should have been done with an DTO object instead. Another fault that could exist is that I did not use private attributes for my data transfer object (SaleDTO, ReceiptDTO, ItemDTO) since they are to hold data only and not other methods. That could be my misunderstanding of how DTOs are to be implemented but it is an easy change to correct.

The tests work and exist but one of them with different layers should have been tested further to see how it acts with different objects since it depends on other methods to do anything of importance. But since it is a test of the class Sale the test is not based on other classes methods the test should be for them.