

# Seminar 3

Object-Oriented Design, IV 1350

Lucas Larsson

[lulars@kth.se](mailto:lulars@kth.se)

09-06-2021

# Contents

1 Introduction	3
2 Method	4
3 Result	4
4 Discussion	5

# 1 Introduction

This seminar is the third seminar in a series of seminars that aims to give a practical experience of well-established theories within object-oriented programming with the focus in this seminar on implementing the design in code.

The task at hand in this seminar is to implement the designed program in seminar 2 in code.

## ***Collaborations:***

*Dennis Hadzialic*

*Jabez Kung Otieno*

Other students during “*Handlednings Pass*”.

Other students in Discord Server.

## 2 Method

I worked on this assignment in small many iterations, the work flow I used was implementing on system operation at a time, after designing the operation for seminar 2 I implemented it directly in code.

Many extra details were needed in order to maintain encapsulation, high cohesion and low coupling. Such a Example is adding a package to contain the DTOs in the model package, changing the visibility of methods from public to private, and even at sometimes changing the whole design in order to get the program to work.

## 3 Results

The results of the code implementation is on this [Git-repository](#).

```
A new Sale has started.
Entering itemID: 123456, quantity: 1.
Description: MacBook Pro Ultra Max
Price: 69.0 $
Running total: 69.0 $

Entering itemID: 123457, quantity: 3.
Description: Ipad Pro Ultra Super Duper
Price: 420.0 $
Running total: 1329.0 $

Entering itemID: 1234600, quantity: 5.
Description: Regular milk
Price: 20.0 $
Running total: 1429.0 $

Cashier concludes sale.
Total is: 1773.249999731779 $
Customer pays: 2000.0 $
  logging sale information in Accounting system
Total Amount in Register is 1773.249999731779$
Printing Receipt . . .
Receipt{
Sale start at: 2021-06-09T22:52:23.875295
items=[Product{price=69.0, itemDescription='MacBook Pro Ultra Max', VAT=0.25, soldQuantity=1}, Product{price=420.0, itemDescription='Ipad Pro Ultra Super Duper', VA
Total amount= 1429.0 $ + 344.2499997317791 $ VAT = 1773.249999731779 $
Amount Paid by customer= 2000.0 $
Change= 226.7500002682209 $ }
Process finished with exit code 0
|
```

Figure3:1 A sample of the output of the program

The figure 3:1 shows a mock sample of the programs output, showing the specified items and the Receipt containing all the information required including list of the items, price, VAT, total payment and change back.

## 4 Discussion

This discussion is built as a series of answers to the questions from the assessment criteria for seminar 3 provided on the course web page.

The code is easy to understand since all variables have descriptive names and all methods have comments that describe what they do, the methods are even held as little as possible, to avoid confusion. The MVC and layer patterns are followed, all classes are divided according to their functionality. Junit framework is used for the tests.

There is three test classes, Inventory system, Item and View.

No static methods are used, primitive data types are used in some cases where it is not necessary to use objects.

There is no methods with long parameter lists.