Object-oriented Design
Seminar 2 – 2019-04-11
Daniel Westerlund
My Answers

Alexander Palm

• Is it easy to understand the design?
I feel that the design is easy to understand.

• Are the MVC and Layer patterns used correctly? Are there enough layers? Enough packages? Enough classes? Does the controller handle any logic (it should not)?
The layers are used correctly, and it is a reasonable number of classes. Classes customer registry and external accounting system is missing.

•Is there view-related code in the model (there should not be any)?
No.

• Is there low coupling, high cohesion and good encapsulation?
The coupling seams ok.
Cohesion can get some improvements, for example get quantity in item registry is not relevant, they have two classes for item DTO which we think is nor needed. Some public methods could be package private instead.

• Is the use of low coupling, high cohesion and encapsulation explained in the report?
Yes.

• Are parameters, return values and types correctly specified? Is it clear from where values come?
Some parameter can clerer
Parameter values type can be better for example identifier can be string and their method addItem doesn't return anything. Price after discount is not returned.

• Are objects used instead of primitive data as much as possible?
Yes.

• Are there static methods or attributes? If so, is that a mistake?
None.

• Objects shall be kept intact as much as possible. Is that the case, or do objects hand out their attributes?
It seems like the object keep as much data as possible for it self.

• Are Java naming conventions followed?
Yes, I cant not find any wrong with it.

• Is the method and result explained in the report? Is there a discussion? Is the discussion relevant?
yes

Juan Diego Martinez

• Is it easy to understand the design?
Yes, it was easy to follow.

• Are the MVC and Layer patterns used correctly? Are there enough layers? Enough packages? Enough classes? Does the controller handle any logic (it should not)?
The layers are used correctly, and it is a reasonable number of classes. Classes customer registry and external accounting system is missing.

•Is there view-related code in the model (there should not be any)?
No.

• Is there low coupling, high cohesion and good encapsulation?
Coupling seams ok.
Cohesion is ok, but some methods could be removed to get better cohesion.
They have high encapsulation which is good.

• Is the use of low coupling, high cohesion and encapsulation explained in the report?
yes

• Are parameters, return values and types correctly specifed? Is it clear from where values come?
Yes.

• Are objects used instead of primitive data as much as possible?
Yes

• Are there static methods or attributes? If so, is that a mistake?
Main should be static but it is not.

• Objects shall be kept intact as much as possible. Is that the case, or do objects hand out their attributes?
Yes.

• Are Java naming conventions followed?
Yes.

• Is the method and result explained in the report? Is there a discussion? Is the discussion relevant?
Yes.

Erik Hanstad

• Is it easy to understand the design?
The class diagrams pictures is a bit hard to follow.

• Are the MVC and Layer patterns used correctly? Are there enough layers? Enough packages? Enough classes? Does the controller handle any logic (it should not)?
Yes, both MVC and layer patterns are are used correctly. Layers/classes seams like reasonable amount.

•Is there view-related code in the model (there should not be any)?
No.

• Is there low coupling, high cohesion and good encapsulation?
We could not see the coupling from diagrams, it has high cohesion only printer is not registry. And high encapsulation.

• Is the use of low coupling, high cohesion and encapsulation explained in the report?
Nej

• Are parameters, return values and types correctly specifed? Is it clear from where values come?
Yes

• Are objects used instead of primitive data as much as possible?
Some primitive data could be “hide” in objects.

• Are there static methods or attributes? If so, is that a mistake?
No.

• Objects shall be kept intact as much as possible. Is that the case, or do objects hand out their attributes?
They have a DTO which does not have any attributes and getter methods

• Are Java naming conventions followed?
Yes

• Is the method and result explained in the report? Is there a discussion? Is the discussion relevant?
They are explained, but they could be explained a bit more.