

Re-examination Report (Individual)

Reflect on the main changes and updates made to your project since your last submission.

1. Changes made since the last submission:

For each epic feature, describe below **which classes and methods were changed**. **Also describe the change** (e.g., changed to a do/while loop, removed inheritance, introduced Exception handling via try/catch, etc.)

- Epic feature 2:
 - Changed Classes / Methods: Item class
 - Description of changes: The implementation of encapsulation was change in the Item class, since it was wrong. The access modifier Private was added to the variables when they are declared. The method for truncation decimals was fixed and moved to the Façade class.
- Epic feature 3:
 - Changed Classes / Methods: Was implemented from scratch
 - Description of changes: The composition was introduced here, so that Item is a composite and Review is a component. Item is created with an empty list of reviews, the method for adding reviews into the list also was added to an item behavior.
- Epic feature 4:
 - Changed Classes / Methods: Was implemented from scratch
 - Description of changes: Here I have the composition and encapsulation as well, but in slightly different manner the Façade is a composite for a transaction, also Transaction was implemented as an immutable object in order no one can change it later. There are no setters methods either for amount, id, purchasePrice as well as key word Final was added when declared variables.

Note: If you are submitting the project for the first time (or from scratch), explain, **for each implemented Epic Feature**, which **OOP concepts** you used.

2. Object-oriented design:

Based on our knowledge of object-oriented design, what relevant concepts or methods did you apply? For each concept, explain which classes or methods you applied that concept.

I have implemented Encapsulation and Composition so far. Item, Review and Transaction classes have encapsulation. That means their variables are declared with usage of the access modifier PRIVATE so that they are not accessible from other classes without special methods so-called getters and setters that have PUBLIC modifier. Since an ID of an item cannot be changed we can only retrieve it but not change so I have not done the setId method according to the project specification. If I want to change a value of a variable that are protected by encapsulation I can do it through set-methods. In Epic Feature 4 the transaction object has to be implemented as an immutable object.

For the Review object the Composition were made since the Item could contain many reviews and those are stored in the list. I chose for that ArrayList because it has indices and allows duplication since the reviews could be similar. Through the item we are able to access the size of the list, get all reviews (getReviews) as well as get specific review knowing its index.

Polimorphism, Inheritance and probably abstraction should be used in the next Epic Features.○

I know there many weaknesses in the code and it still hardly readable. There are many methods that I don't like myself and would change if I had more time, but now I restricted in time by two other courses and if I fail again I am going to make the code more readable and consistent.