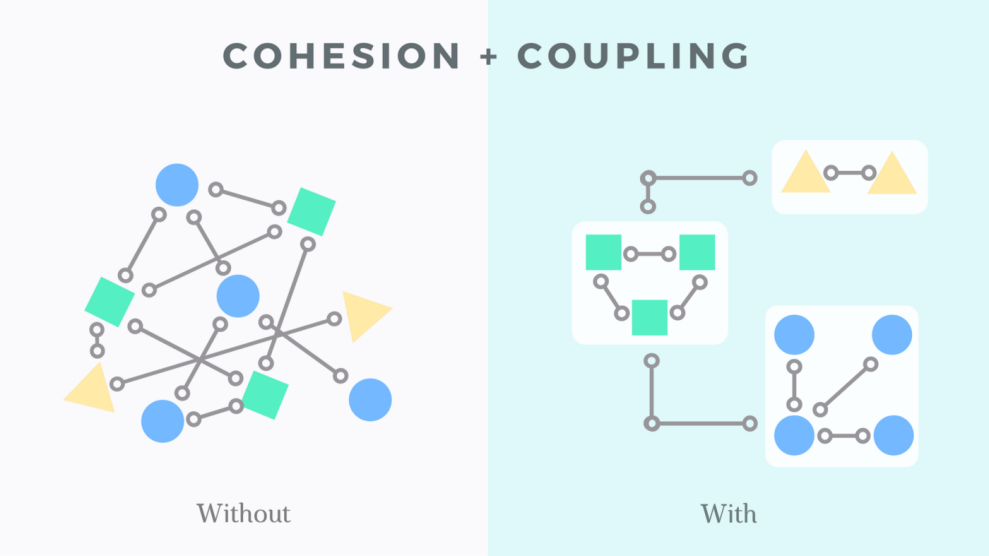
CYDEO

Coupling

Date: 2nd September 2022

Coupling



- In this lab, you are responsible to implement loose coupling structure.
- As you can see in the image everything is connected each other and it is very hard to break the relations.
- We want you to understand how you will manage to decrease dependencies.

Problem

- In this project there are 2 classes **CustomerBalance** and **GiftCardBalance**. Under the loose package you are going to see a **BalanceService**. If you look inside of the class, You will realise that both **customerBalance** and **giftCardBalance** objects are coming from constructor classes and if you would like to add another balance (assume that xxxBalance) you need to create/update bunch of method.

- On the other hand, assume that your company doesn't provide gift card balance anymore. If you would like to delete this business from your code this will be a problem, because your application is strongly dependent gift card balance business. We want you to change dependency schema to loose coupling.

Solution

- As you can see in the application, there are 2 additional classes under low package:**Balance** and **BalanceManager**. What is expected from you is to redesign the project using these 2 classes.

- **Balance** is an abstract class. You need to make sure that other balance classes inherit from abstract Balance class. **Balance** class has some missing parameters you need to add them

- In **BalanceManager** you will see checkout method. This method needs to accept all inherited balance subclasses and amount. Inside the method you need to compare **Balance** is enough to complete the transaction.