

The purpose of this lab is to practice working with interfaces, abstract classes, defining and using customized exceptions and obligatory methods.

1. Setup

Please download `Lab7.zip` that is attached to this description.

- Open `eclipse`.
- Click on *File* and select *Import*.
- Choose *Existing Projects into Workspace* and click *Next*.
- Click on *Select Archive File* and then *Browse*. Find `Lab7.zip` and click *Finish*.
- Please make sure that you do not already have a project called `EECs2030_Lab7`, otherwise, eclipse cannot import it for you.

You should see two files, one is called `Account.java` and one `AccountTester.java`.

Please note that `AccountTester.java` shows several compiler errors. This is expected as the code in `Account.java` has not been implemented yet.

2. JavaDoc generation

The javaDoc has been written for you only for the superclass. For the subclass, which you are going to write from scratch, you need to write the javaDoc yourself.

When you finished, all you need to do is to generate it as an HTML file to make it easier for navigation. For this, right click on `Account.java` -> `export` -> `javaDoc` -> *Next*. It will ask you for the location in which you want to store the documentation. Enter the path and then click *Finish*.

If you look at the location in which you stored the documentation, you will see there is a file called `index.html`. Clicking on this file, shows the documentation of the project in your browser.

3. Programming Task

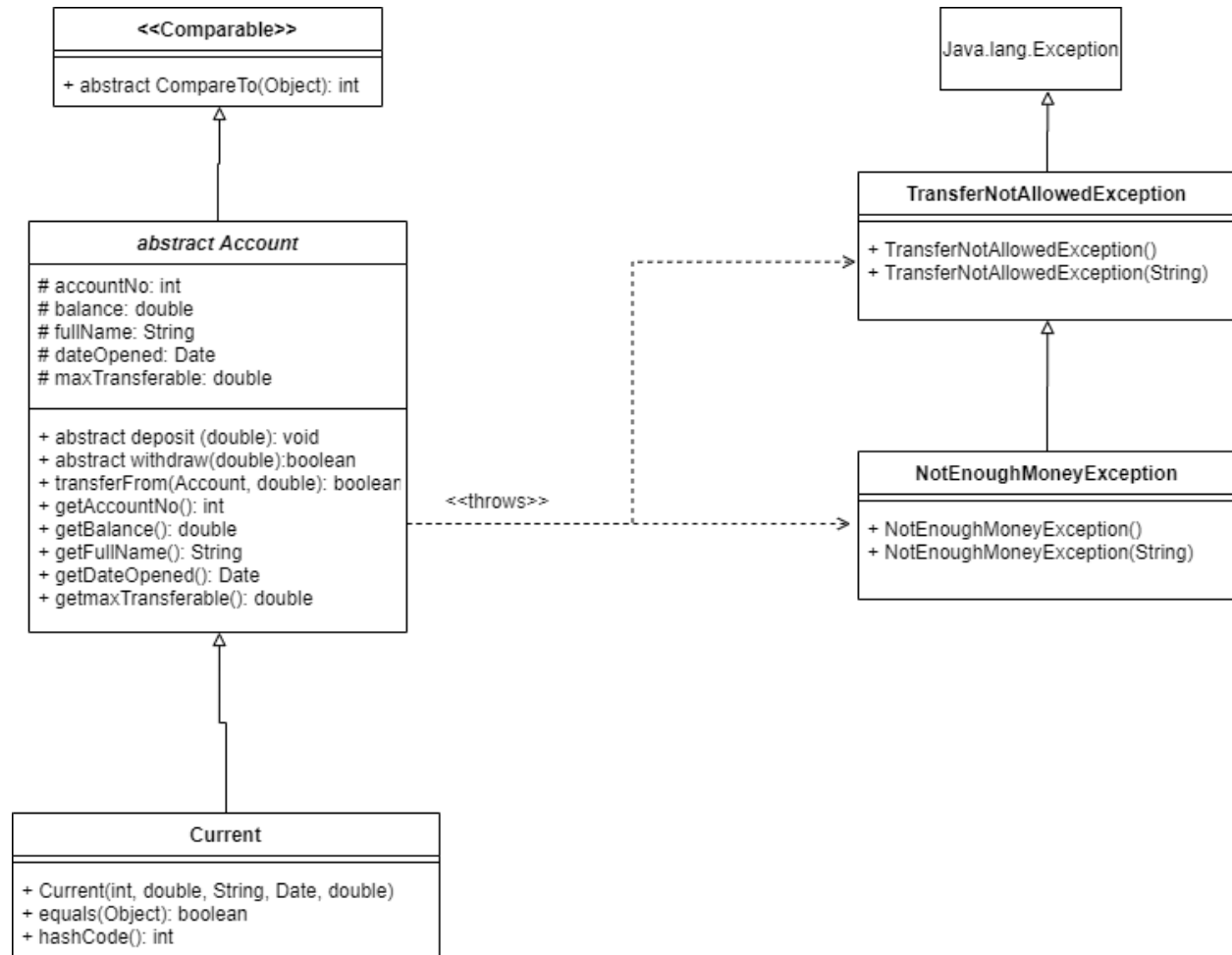
A bank account can have many types, including current, saving, investment and so on. In this lab, we are going to work with current accounts only. Your

job for this lab is to look at the UML below and implement the code that conforms with this UML.

An account can be deposited or withdrawn by some amounts of money. This is done by two methods called `deposit` and `withdraw` respectively.

It is not possible to withdraw more money than the `balance` of the account. If a client tries to do so, an exception called `NotEnoughMoneyException` is thrown. This is a user-defined exception.

Also, it is not possible to withdraw money more than what `maxTransferable` shows. If a client tries to do so an exception called `TransferNotAllowedException` is thrown. This is also a user-defined exception, which is the base class (i.e., superclass) of `NotEnoughMoneyException`.



It is possible that some amount of money is withdrawn from the account and deposited to another. This is done by `transferFrom` method. This method may throw two exceptions if either more than the `balance` or maximum amount allowed for the transfer, is withdrawn.

Your job is to implement all the classes and the methods that you see in the UML.

4. Submit

You only submit one file that is called `Account.java` via eClass by clicking on the lab link.

You do not need to submit the tester or HTML files.