CST3110 – Lab 7
Contracts for Java : Preconditions

1 Background

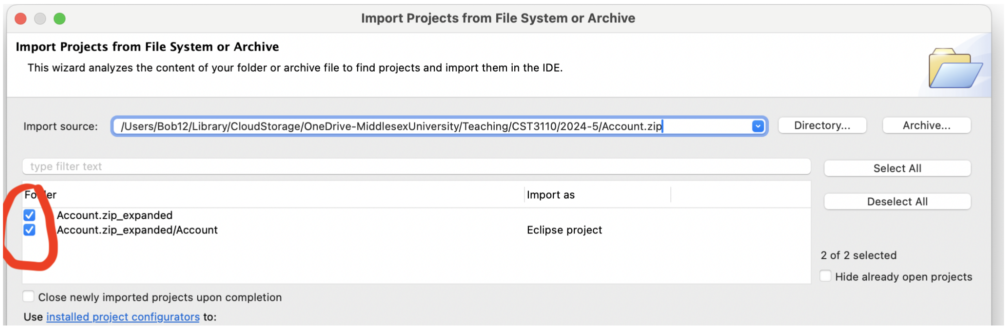
Class `Account` is an implementation of a bank account. Instances of `Account` contain a single variable, `balance`, to record amount of money in the account. Methods allow the balance to be altered by withdrawing or depositing funds, and making a transfer from one account to another.

Tests, defined in `AccountTest.java`, check the pre-condition part of contracts by running tests that falsify preconditions. When a pre-condition is checked and evaluates to **`false`**, a `PreconditionError` exception is thrown, and the test cases expect this with `assertThrows`.

2 Exercise

1. Download the project `Account.zip` in Week 7 from My Learning.
2. Import the project in your Eclipse environment.

When importing the proejct, Eclipse trues to create two projects, one nested inside the other[1]. In the import process, after the `Account.zip` archive has been chosen, Eclipse offers two project folders, both checked by default:



Import Projects from File System or Archive

Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: /Users/Bob12/Library/CloudStorage/OneDrive-MiddlesexUniversity/Teaching/CST3110/2024-5/Account.zip

Folder	Import as
☑ Account.zip_expanded	
☑ Account.zip_expanded/Account	Eclipse project

2 of 2 selected

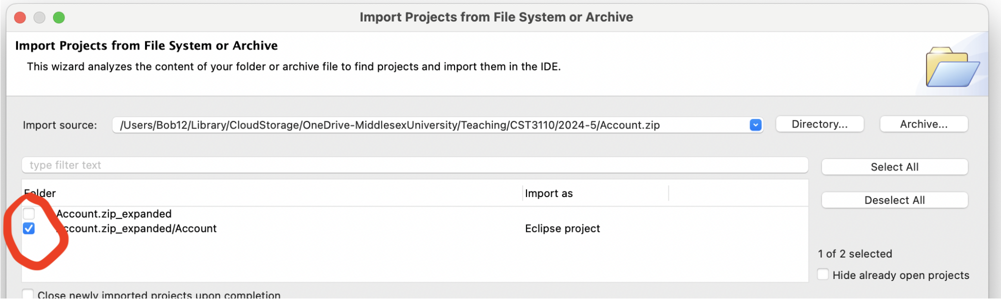
□ Hide already open projects

□ Close newly imported projects upon completion

Use installed project configurators to:

To ensure that Eclipse creates a simpler import with a single project, and without nesting, **unselect the first item in the list** of project folders:

[1] No idea why!



Import Projects from File System or Archive

This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.

Import source: /Users/Bob12/Library/CloudStorage/OneDrive-MiddlesexUniversity/Teaching/CST3110/2024-5/Account.zip

type filter text

Folder	Import as
□ Account.zip_expanded	
☑ Account.zip_expanded/Account	Eclipse project

1 of 2 selected

□ Hide already open projects

□ Close newly imported projects upon completion

3. Run the tests in `AccountTest`. Make sure that all tests fail. They should fail as the code is incomplete and no contracts have been specified.
4. There is a constructor and a number of methods in class `Account`. For each method you need to

   - Read the comments related to the method. These specify the requirements.
   - Write the *most general (i.e., weakest) precondition* using `@Requires`.
   - Implement the method.

   The above tasks, when completed, should ensure all tests pass.
5. **Keep this Eclipse project, you will need this again later.**

3 Challenge questions

Modify the code as follows.

1. Add a private integer variable called `overdraft`. This must always be a positive or zero number.
2. Modify the constructor so that it takes two arguments, the new one providing an initial value for `overdraft`.
3. Modify the methods so that the balance can become negative but never less than `-overdraft`.
4. Rewrite the preconditions and tests accordingly.