CS 4472A

Software Specification, Testing,

and Quality Assurance

Assignment 1

Black Box Testing

Introduction

In this assignment you will apply three of the foundational black box techniques we have seen in the class. The techniques will be applied on the ATM system, the code of which has been uploaded to OWL.

The focal points of your assignment will be the *withdrawal*, the *deposit*, and the *transfer* operations.

Each of these operations has their own business logic for calculating banking fees for the given operation.

The business logic specification for calculating the fees for each of the given operations is provided below:

Withdrawal

- If the client is a *student* and…
  - If the transaction is performed during the *weekend*, then there is ***no fee***
  - Otherwise, the fee is ***0.1%*** of the amount withdrawn.
- If the client is *not a student* and…
  - If the *balance* of the account is less than *$1,000*, then the fee is ***0.2%*** of the amount withdrawn.
  - If, however, the *balance* is *$1,000*, or more, but less than *$10,000*, the fee is ***0.1%*** of the amount withdrawn.
  - If the balance is more than *$10,000*, then there is ***no fee***.

Deposit

- If the client is a *student* and…
  - If the amount withdrawn is more than *$100* and…
    - If the *balance* of the account is more than *$1,000*, then the fee is ***1%*** of the amount withdrawn.
    - Otherwise, the fee is ***0.5%*** of the amount withdrawn.
  - If, however, the amount withdrawn is *$100* or less and…
    - If the *balance* of the account is more than *$5,000*, then the fee is ***0.5%*** of the amount withdrawn.
    - Otherwise, there is ***no fee***.
- If the client is *not a student* and…
  - If the amount withdrawn is more than *$500* and…
    - If the *balance* of the account is more than *$5,000*, then the fee is ***1%*** of the amount withdrawn.
    - Otherwise, the fee is ***0.5%*** of the amount withdrawn.
  - If, however, the amount withdrawn is *$500* or less and…
    - If the *balance* of the account is more than *$10,000*, then the fee is ***0.5%*** of the amount withdrawn.
    - Otherwise, there is ***no fee***.

Transfer

- If the client is a *student* and…
  - If the amount withdrawn is less than *$100* and…
    - If the *balance* of the account the money is coming from is less than *$1,000* and…
      - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.1%*** of the amount withdrawn.
      - Otherwise, the fee is ***0.05%*** of the amount withdrawn.
    - If, however, the *balance* of the account the money is coming from is *$1,000*, or more and…
      - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.5%*** of the amount withdrawn.
      - Otherwise, the fee is ***0.25%*** of the amount withdrawn.
  - If the amount withdrawn is *$100*, or more, and…
    - If the *balance* of the account the money is coming from is less than *$1,000* and…
      - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.05%*** of the amount withdrawn.
      - Otherwise, the fee is ***0.025%*** of the amount withdrawn.
    - If, however, the *balance* of the account the money is coming from is *$1,000*, or more and…
      - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.25%*** of the amount withdrawn.
      - Otherwise, the fee is ***0.125%*** of the amount withdrawn.

- If the client is a *not student* and…
    - If the amount withdrawn is less than *$100* and…
        - If the *balance* of the account the money is coming from is less than *$1,000* and…
            - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.2%*** of the amount withdrawn.
            - Otherwise, the fee is ***0.1%*** of the amount withdrawn.
        - If, however, the *balance* of the account the money is coming from is *$1,000*, or more and…
            - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***1%*** of the amount withdrawn.
            - Otherwise, the fee is ***0.5%*** of the amount withdrawn.
    - If the amount withdrawn is *$100*, or more, and…
        - If the *balance* of the account the money is coming from is less than *$1,000* and…
            - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.1%*** of the amount withdrawn.
            - Otherwise, the fee is ***0.05%*** of the amount withdrawn.
        - If, however, the *balance* of the account the money is coming from is *$1,000*, or more and…
            - If the *balance* of the account the money is going to is less than *$1,000*, then the fee is ***0.5%*** of the amount withdrawn.
            - Otherwise, the fee is ***0.25%*** of the amount withdrawn.

ATM Session

When it comes to this part, you will only test the *withdrawal* transaction (*transaction choice 1*). Failure to input the right kind of data results in the following outcomes:

- Invalid amount -> throws InvalidAmountException
    - Valid amount for withdrawal -> product of 20 and 50
- Invalid PIN format -> throws InvalidPINException
    - PIN should be 4 digits

The Testing Targets

You will perform black box testing on the following classes

1. **atm.Session.java**: The focus here is to check for PIN format, and valid amount chosen for a *withdrawal* transaction(*transaction choice 1*) as per the business logic presented above for **ATM Session**. The technique to use for this test is *Robust Worst Case Boundary Value Analysis* under the single fault assumption principle. In the program provided a withdrawal is limited to the available balance of the account and the daily limit. For this assignment you must create an account with a daily limit of $1000 and an available balance of $5,000. The daily limit of $1000 is your upper boundary in this case.
2. **bank.FeesCalculator.java:** The focus here is to check for the correct calculation of fees while performing withdrawal, deposit, and transfer.

- **For withdrawal:** Use *Robust Worst Case Boundary Value Analysis* to generate your test cases.
- **For deposit**: Use *Weak Robust Equivalence Class Analysis* to generate your test cases.
- **For transfer**: Use *Decision Table Analysis* to generate your test cases.

What to Run

For each of the testing targets (**atm.Sesssion.java** and **bank.FeesCalculator.java**, run the different test cases you have generated under the different analysis techniques as JUnit tests. For example, for atm.Sesssion.java you will have one class containing the tests for the PIN format and the withdrawal amount. Similarly, for the Bank.FeesCalculator.java you will have one class for each of the three transactions, one for *withdrawal*, one for *deposit*, and one for *transfer*.

What to Deliver

For each of the JUnit tests you have run, deliver:

1. The test cases you have created. This part of the deliverable will have two sections. In the first section you will need to comment on the process and the assumptions you have used to calculate the test cases, using the analysis technique for the corresponding case (e.g. Robust Worst Case Boundary Value Analysis for PIN format –see above *The Testing Targets*). In the second section you will have the tables with the test cases.
2. The test results. This part of the deliverable will provide a table indicating the test case and whether the test passed or failed.
3. The archive with the system's code you have been provided along with the JUnit code with which you run your tests.

Deadline

Upload the three parts of the deliverable as one archived file as an OWL submission by October 25 midnight.