Skypay

Technical Test 1: Banking Service

Summary

In this test, we would like to create the core functions of a banking system; deposit money, withdraw money, and print transactions.

When doing this test, think of the requirements in banking software systems. Some of them are listed below.

Instructions

Write a class named Account that implements the following public interface:

```
public interface AccountService
{
    void deposit(int amount)
    void withdraw(int amount)
    void printStatement()
}
```

Rules

- You cannot change the public interface of this class.

Desired Behaviour

Here’s the specification for an acceptance test that expresses the desired behaviour for this :

Given a client makes a deposit of 1000 on 10-01-2012
And a deposit of 2000 on 13-01-2012
And a withdrawal of 500 on 14-01-2012
When they print their bank statement
Then they would see

```
Date       || Amount || Balance
14/01/2012 || -500   || 2500
13/01/2012 || 2000   || 3000
10/01/2012 || 1000   || 1000
```

Technical Requirements

- Handle Exceptions whenever needed (invalid inputs,...)
- Think of performance issues and have your code be efficient (when needed).
- Do not hesitate to test your code well.
- Do not use repositories. Use ArrayLists and update them (ignore the case that they are destroyed at the end of the program, this is just for this test).
- We're using `int`s for the money amounts to keep the auxiliaries as simple as possible. In a real system, we would always use a datatype with guaranteed arbitrary precision.
- Don't worry about spacing and indentation in the statement output. (You could instruct your acceptance test to ignore whitespace if you wanted to).
- When in doubt, go for the simplest solution!
- If you ever need help or have any questions, please do not hesitate to reach out to us.

Good Luck