Hi,

I submitted the interview test program today. Sorry for the delay, I did not get a chance to do it during the week. So I started doing it over the weekend, but very soon realized that there are some serious flaws in the interest calculations. The old code was not at all considering the transaction dates during interest rate calculations. For example, if there are deposits made on January 1st and July 1st, when we calculate interests with a rate of 10% simple interest at the end of the year (say, Dec 31st), we should apply 10% on Jan 1st deposit and 5% on the July 1st deposit as July 1st deposit was in the bank only for 6 months (half of the year). The old code was incorrectly giving 10% to both deposits, which is wrong. I modified the interest calculation functions to apply this pro-rata based interest rate based on the transaction history. Note that the calculations get even more complicated with daily accruals, interest rate slabs and 10-day withdrawal rules. I finished all the new features (transfers, 10-day withdrawal rule, and daily accruals) and tried my best to unit-test them with the results verified from the online interest calculation website (<http://www.pine-grove.com/online-calculators/compound-interest-calculator.htm>). To be honest, this was definitely not a 2 hour exercise:-).

Here are the modifications I made, at a high level

- Transaction date based Interest calculations (see InterestCalculator class and different Account sub classes). InterestCalculatorTest has several unit tests

- Made Account base class (with IAccount interface) and added CheckingAccount, SavingsAccount and MaxSavingsAccount sub classes to separate interest calculations based on the account. We could add more account specific logic later. Used AccountFactory to create different accounts.

- Made all the classes thread -safe

- Maintain current balance in Account class so that we do not need to iterate over all transactions just to find the current balance. Also, add balance to Transaction to know the balance when the transaction was made. This will be very handy in interest calculations.

- Add account ID to Account class so that we can use account IDs for transfer functions. Ex: transfer from account id 1 to account id 2. Note that we could think of using account type instead of id, but customers may open multiple accounts with the same account type (2 checking accounts).

- Added several validations (withdraws without sufficient funds, adding same customer again, etc)

- Fixed amount formatting issues in the statement generation

- Fixed singleton thread safety issues in DateProvider. Also, unit-testing may fail with Now due to timing issues. so used total milliseconds since epoch difference.

- Added documentation wherever applicable.

I have not gotten a chance to test some corner cases, but this is the best I could do given the time I have. Have a great week ahead.

I’m not able to push the changes to GITHub so I’m attaching the code in this mail. I’m still trying to push the changes to GIT Hub too.

Thanks,

Kamala

[201 706 1107](tel:201%20706%201107)