

SOIL - if statements and for loops

Consider the two new methods added to the banking model in the code excerpt below.

```
class Customer
  attributes
    name : String
    wealth : Integer
...
  openNewAccount(accType : String)
  begin
    declare acc : BankAccount;

    if accType = 'savings' then
      acc := new SavingsAccount
    else
      acc := new CurrentAccount
    end;

    acc.balance := 0;
    Write('Enter new account number: ');
    acc.accNo := ReadLine();
    insert (self, acc) into HasAccounts
  end

  showWealth() : Integer
  begin
    declare wealth : Integer;
    wealth := 0;
    for acc in self.accounts do
      wealth := wealth + acc.balance
    end;
    self.wealth := wealth;
    WriteLine('Wealth is ');
    --Write(wealth);
    result := wealth
  end
end
```

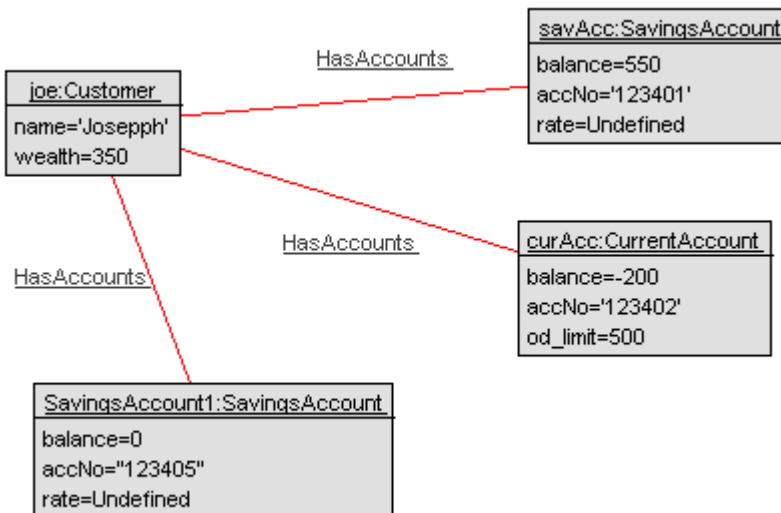
Load this modification of **bank.use** into USE and open **bank.soil** created previously to set up a few objects. Open the Object diagram view to see the objects.

```
use> open bank.soil
bank.soil> -- Script generated by USE 4.2.0
bank.soil>
bank.soil> !new Customer('joe')
bank.soil> !joe.name := 'Joseph'
bank.soil> !new SavingsAccount('savAcc')
bank.soil> !new CurrentAccount('curAcc')
bank.soil> !savAcc.accNo := '123401'
bank.soil> !curAcc.accNo := '123402'
bank.soil> !savAcc.balance := 550
bank.soil> !curAcc.balance := - 200
bank.soil> !curAcc.od_limit := 500
bank.soil> !insert (joe,savAcc) into HasAccounts
bank.soil> !insert (joe,curAcc) into HasAccounts
bank.soil>
```

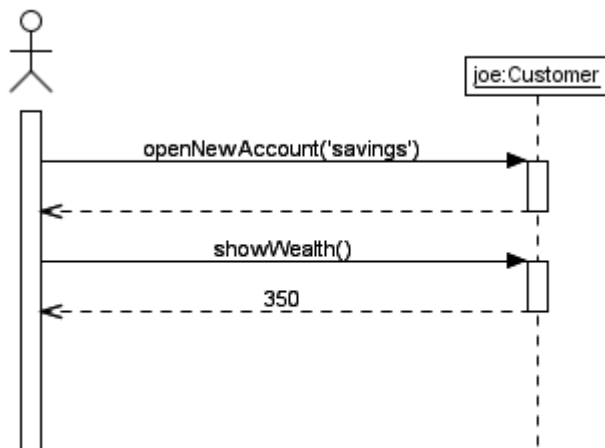
Then from the command window try:

```
bank.soil>
use> !joe.openNewAccount('savings')
Enter new account number: '123405'
use> !joe.showWealth()
Wealth is
use>
```

After this, object diagram should look like



Then open sequence diagram view.



Query Operations

Here instead of code like

```
wealth := wealth + acc.balance
```

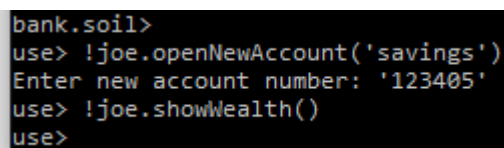
in showWealth(), which accesses a bank account object attribute value directly, we use a query method **getBalance()** which will simply return the balance. This is better OO code design. So USE code will in class BankAccount now look like:

```
getBalance() : Integer
begin
    result := self.balance;
end
```

and in class Customer

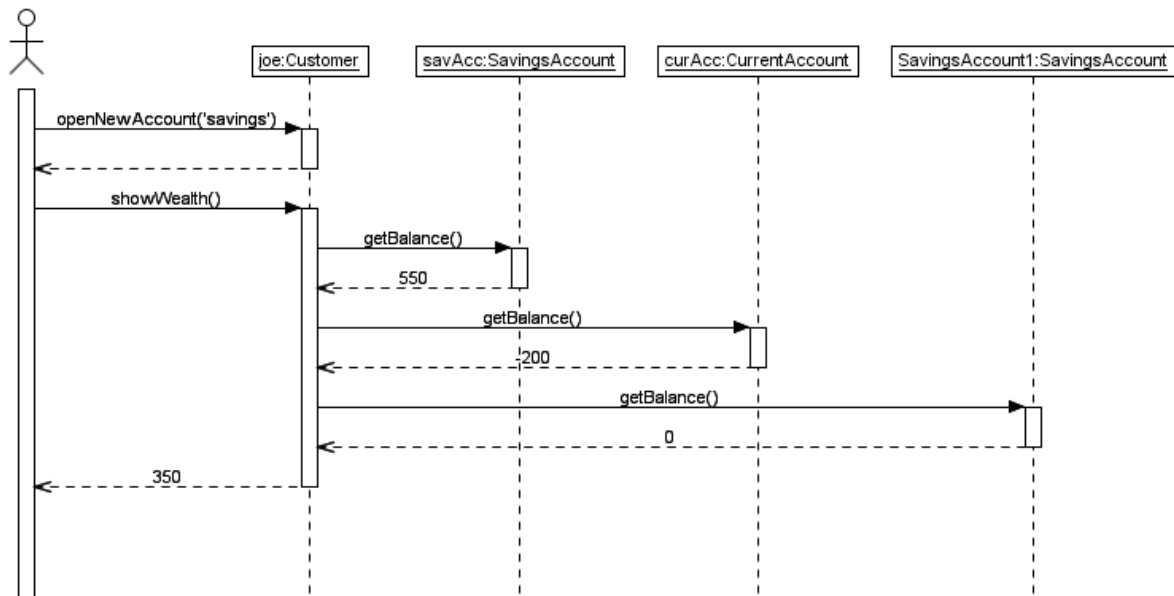
```
showWealth() : Integer
begin
    declare wealth : Integer, b : Integer;
    wealth := 0;
    for acc in self.accounts do
        b := acc.getBalance();
        wealth := wealth + b
    end;
    self.wealth := wealth;
    result := wealth
end
```

If you now open the SOIL file and run as before the command



```
bank.soil>
use> !joe.openNewAccount('savings')
Enter new account number: '123405'
use> !joe.showWealth()
use>
```

the sequence diagram view will now look like:



USE seems to only show method invocations in sequence diagram when they are non-query methods. A method is non-query when it has an assignment or invokes another non-query method.

For example, change the `getBalance()` operation from

to

```
getBalance() : Integer = balance
```

which is a simple query method.

Then rerun the previous command and look at the sequence diagram. Now would look like:

