Deethya Makonahalli
dgm92

TESTING FOR ACCOUNT CLASS

This is a test for whether the constructors of the account class run

```
>import Account
> Account accountNumber = new Account("Hundred")
> Account balanceLimit = new Account("Hundred",2000)
```

The next is to see whether the values are actually stored by testing the getter methods of both account number and balance limit

```
> accountNumber.getAccountNumber()
"Hundred"
> Account balance = new Account("Hundred",2000)
> balanceLimit.getBalanceLimit()
2000
```

The following test evaluates whether the set method of balance works properly which was evaluated by setting balance and then getting balance

```
> balance.setBalance(50.5)
> balance.getBalance()
50.5
```

The next test evaluates whether the charge number will increment balance by the input value by applying the method and then getting balance

```
> balance.charge(5)
> balance.getBalance()
55.5
```

The next test evaluates whether the credit number will decreasing balance by the input value by applying the method and then getting balance

```
> balance.credit(15)
> balance.getBalance()
40.5
```

The next test evaluates whether the balance limit can be changed using the set balance method and then calling getbalance.

```
balanceLimit.setBalanceLimit(4000)
> balanceLimit.getBalanceLimit()
4000
```

TESTING FOR LIBRARYACCOUNT CLASS

The first test evaluates whether the 2 constructors for the LibraryAccount class run

```
> import LibraryAccount
> LibraryAccount accountNumber = new LibraryAccount("Ten")
> LibraryAccount accountNumber = new LibraryAccount("Ten", 300,
3.5, 2.3)
> LibraryAccount bookFine = new LibraryAccount("Ten", 300, 3.5,
2.3)
> LibraryAccount reserveFine = new LibraryAccount("Ten", 300,
3.5, 2.3)
> LibraryAccount balanceLimit = new LibraryAccount("Ten", 300,
3.5, 2.3)
```

The efficacy of the constructor and the getters is then tested using the following code in the interactions.

```
> bookFine.getBookFine()
3.4
> reserveFine.getReserveFine()
2.3
> accountNumber.getAccountNumber()
"Ten"
>balanceLimit.getBalanceLimit()
"300"
```

The set methods of the book fine and reserve fine were then tested by setting it to a different value and then getting to evaluate whether the right value was stored

```
> bookFine.setBookFine(4.5)
> reserveFine.setReserveFine(1.2)
> bookFine.getBookFine()
4.5
> reserveFine.getReserveFine()
1.2
```

The methods to increment and decrement the overdue books was then tested to see whether the code runs and if the appropriate value was stored, which was evaluated by getting the overdue books

```
> LibraryAccount overdueBooks = new LibraryAccount("Ten", 300,
3.5, 2.3)
> overdueBooks.incrementOverdueBooks()
> overdueBooks.getOverdueBooks()
1
> overdueBooks.decrementOverdueBooks()
> overdueBooks.getOverdueBooks()
-1
```

The methods to increment and decrement the overdue reserve was then tested to see whether the code runs and if the appropriate value was stored, which was evaluated by getting the overdue reserve

```
> LibraryAccount overdueReserve = new LibraryAccount("Ten", 300,
3.5, 2.3)
> overdueReserve.incrementOverdueReserve()
> overdueReserve.getNumberOverdueReserve()
1
> overdueReserve.decrementOverdueReserve()
> overdueReserve.getNumberOverdueReserve()
```

```
0
```

This code tests whether the canBorrow method returns an accurate boolean value by setting the balance less than the balance limit such that a “true” value indicates functioning code

```
> LibraryAccount balance = new LibraryAccount("Ten", 300, 3.5,
2.3)
> balance.setBalance(30.5)
> balance.canBorrow()
True
```

This code tests whether the endofDay method runs and whether when you apply getBalance the adjusted value is returned

```
>balance.endOfDay()
> balance.getBalance()
30.5
```

This test indicates that the stored value was not adjusted hence there is a flaw in the code that did not allow for an appropriate change.

TESTING FOR CREDITACCOUNT CLASS

The first test evaluates whether the constructor for Creditaccount class runs

```
> CreditAccount balance = new CreditAccount("hundred", 0.25)
> accountNumber.getAccountNumber()
"Hundred"
> CreditAccount interestRate = new CreditAccount("hundred",
0.25)
> interestRate.getInterestRate()
0.25
```

This code tests the setters for balance and interest rate

```
> balance.setBalance(30)
> balance.getBalance()
30
```

```
> interestRate.setInterestRate(0.4)
> interestRate.getInterestRate()
0.4
```

This code tests the setters and getters for monthlyPayment

```
> CreditAccount monthlyPayment = new CreditAccount("hundred",
0.25)
> monthlyPayment.setMonthlyPayment(55.6)
> monthlyPayment.getMonthlyPayment()
55.6
```

This code tests the setters and getters for amountPaidThisMonth

```
> CreditAccount amountPaidThisMonth = new
CreditAccount("hundred", 0.25)
> amountPaidThisMonth.setAmountPaidThisMonth(60.5)
> amountPaidThisMonth.getAmountPaidThisMonth()
60.5
```

This code tests whether the endOfMonth method works by applying it and then getting balance. Two scenarios were tested to test the if condition coded in the method which changes the set balance and sets amountPaidThisMonth top zero

```
> balance.endOfMonth()
> balance.getBalance()
0.53125

> monthlyPayment.setMonthlyPayment(3)
> balance.endOfMonth()
> balance.getBalance()
0.011067708333333334
> amountPaidThisMonth.getAmountPaidThisMonth()
0
```

This code tests whether the credit method reduces the value of amountPaidThisMonth and balance by applying and then using the getters and setters as well as accounting for the condition of the method

```
> amountPaidThisMonth.credit(4.5)
> amountPaidThisMonth.getAmountPaidThisMonth()
56.0
> balance.credit(4.5)
> balance.getBalance()
25.5
```

TESTING FOR STUDENTTACCOUNT CLASS

The first test tests the constructor of the student account class by using the getter and setters

```
> import StudentAccount
> StudentAccount accountNumber = new StudentAccount("ten",
"deethya")
> accountNumber.getAccountNumber()
"ten"
> name.getName()
"Deethya"
```

This tests the setter for the account holder name by applying and checking using the get method

```
> StudentAccount name = new StudentAccount("ten", "deethya")
> name.setName("Makonahalli")
> name.getName()
"Makonahalli"
```

This tests the setter for the address name by applying and checking using the get method

```
> StudentAccount address = new StudentAccount("ten", "deethya")
> address.setAddress("green")
> address.getAddress()
"green"
```

This code tests the setter and getter for the library account by setting it to a value of type library and getting it.

```
> import LibraryAccount
>StudentAccount library = new StudentAccount("ten", "deethya")

> LibraryAccount l1 = new LibraryAccount("Ten")
> l1.setBalance(20)
> library.setLibraryAccount(l1)
> library.getLibraryAccount().getBalance()
20
```

This code tests the setter and getter for the tuition account by setting it to a value of type library and getting it.

```
> import CreditAccount
> StudentAccount tuitionAccount = new StudentAccount("ten",
"deethya")
> CreditAccount c1  = new CreditAccount("hundred", 0.25)
> c1.setBalance(40)
> tuitionAccount.setTuitionAccount(c1)
> tuitionAccount.getTuitionAccount().getBalance()
40
```

This code tests the setter and getter for the Dinning account by setting it to a value of type library and getting it.

```
> import CreditAccount
> CreditAccount d1  = new CreditAccount("hundred", 0.25)
> StudentAccount diningAccount = new StudentAccount("ten",
"deethya")
>d1.setBalance(10)
> diningAccount.setDinningAccount(d1)
> diningAccount.getDiningAccount().getBalance()
10
```

The get balance was tested through this statement in each accounts' test

```
> library.getLibraryAccount().getBalance()
20.0
> tuitionAccount.getTuitionAccount().getBalance()
40.0
```

The charge method was then tested accounting for each condition

```
> tuitionAccount.charge(40)
> tuitionAccount.getTuitionAccount().getBalance()
50
> tuitionAccount.charge(4)
> balance.getBalance()
-26
```

The credit method was then tested accounting for each condition

```
> tuitionAccount.getTuitionAccount().credit(3)
> tuitionAccount.getTuitionAccount().getBalance()
37.0
> library.getLibraryAccount().credit(8)
> library.getLibraryAccount().getBalance()
12.0
```

This appears to be the credit method of the credit account and hence the override failed. Additionally a "`NullPointerException`" was obtained when the credit method was attempted in other ways for instance `tuitionAccount.credit(3)`, hence during the test I may have failed to initialize certain variables within the function of the method.