11/25/2017

Steven Barry Elliot Rourke Scott Pirrie Ciaran Service Mark Baxter

CS313 Testing Strategy Documentation

# Test Drivers

As a group we have completed task A with six test drivers each for completing the tasks listed on My Place. The drivers thoroughly test our code under different inputs and test whether the code meets our desired and expected output. We have tested the concurrent system under certain conditions, and used the different types of accounts. Our synchronization uses the lock object mechanism to prevent two or more threads accessing the same data at the same time, preventing race conditions. We also have a condition in the withdrawal method, if the user tries to withdraw an amount that will put the account overdrawn the condition will give the user the chance to wait until the account has money deposited by another user before completing the withdrawal transaction. Our drivers test:

* 2 account holders trying to check the balance simultaneously
* One account holder tries to check the balance while the other is withdrawing money
* 2 account holders are simultaneously withdrawing and then checking the balance
* An employee completes a transfer whilst the bank account holder is withdrawing
* Condition mechanism that waits for the balance to grow
* 2 employees are simultaneously trying to edit an account holder’s details
* Transfer after depositing in the same account

# Expected vs Actual Output of Drivers

Test Driver 1 Check Balance whilst action is being taken expected output:

Balance at start of thread before first action: 0

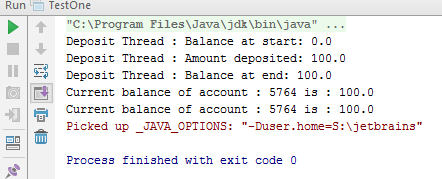
Deposit 100 into the account, balance: 100

Same customer views the balance: 100

Different customer views the balance, the balance should be: 100

3 Threads in this driver, deposit, customer 1 balance, customer 2 balance.

Actual Result:



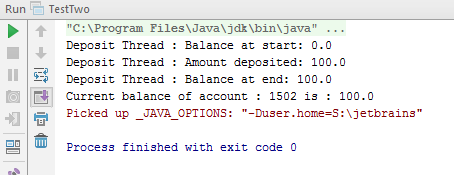
Test Driver 2

Balance at start of thread before first action: 0

Deposit 100 into the account, balance: 100

Deposit thread views the balance: 100

Actual Output:



Test Driver 3:

Check withdraw, deposit, and then view the balance simultaneously. Use 6 threads to complete the driver.

2 customers share the same account.

Check balance at start of threads being run, balance is: 0

User1 Deposit thread deposits 100 into the account, checks balance, balance is: 100

User2 Deposit thread deposits another 100 into the account, checks balance, balance now is: 200

User1 is trying to withdraw 100 from the account, then checks the balance, balance now is: 100

User2 is trying to withdraw 100 from the account, then checks the balance, balance now is: 0

User1 checks the balance of the account, balance is: 0

User2 checks the balance of the account, balance is: 0

Expected outcome 2:

Check balance at start of threads being run, balance is: 0

User2 Deposit thread deposits 100 into the account, checks balance, balance is: 100

User1 Deposit thread deposits another 100 into the account, checks balance, balance now is: 200

User2 is trying to withdraw 100 from the account, then checks the balance, balance now is: 100

User1 is trying to withdraw 100 from the account, then checks the balance, balance now is: 0

User2 checks the balance of the account, balance is: 0

User1 checks the balance of the account, balance is: 0

Expected outcome 3:

Check balance at start of threads being run, balance is: 0

User2 Deposit thread deposits 100 into the account, checks balance, balance is: 100

User1 Deposit thread deposits another 100 into the account, checks balance, balance now is: 200

User1 is trying to withdraw 100 from the account, then checks the balance, balance now is: 100

User2 is trying to withdraw 100 from the account, then checks the balance, balance now is: 0

User2 checks the balance of the account, balance is: 0

User1 checks the balance of the account, balance is: 0

Expected outcome 4:

Check balance at start of threads being run, balance is: 0

User1 Deposit thread deposits 100 into the account, checks balance, balance is: 100

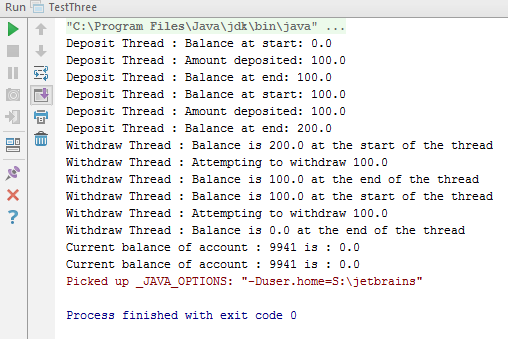
User2 Deposit thread deposits another 100 into the account, checks balance, balance now is: 200

User2 is trying to withdraw 100 from the account, then checks the balance, balance now is: 100

User1 is trying to withdraw 100 from the account, then checks the balance, balance now is: 0

User1 checks the balance of the account, balance is: 0

User2 checks the balance of the account, balance is: 0

Actual Output:  


Test Driver 4, deposit then withdraw whilst an employee does a standing order in and out, this also test the overdraft feature as the account will be put into negative value.

Expected Output:

Balance at start of thread: 0

Thread one deposits 99, balance is now: 99

Thread 5 checks balance, balance is: 99

Thread two deposits 100, balance is now: 199

Employee completes standing order of: 101

Balance is now: 300

Thread 4 withdrawing 300, balance is now: 0

Thread 4 withdrawing 102, balance is now: -102

Employee completes standing order out of: 200

Thread 6 checks balance, balance is: -302

Expected Output 2:

Balance at start of thread: 0

Thread one deposits 99, balance is now: 99

Thread two deposits 100, balance is now: 199

Employee completes standing order of: 101

Balance is now: 300

Thread 4 withdrawing 300, balance is now: 0

Thread 4 withdrawing 102, balance is now: -102

Employee completes standing order out of: 200

Thread 5 checks balance, balance is: -302

Thread 6 checks balance, balance is: -302

Expected Output 3:

Balance at start of thread: 0

Thread two deposits 100, balance is now: 100

Thread one deposits 99, balance is now: 199

Employee completes standing order of: 101

Balance is now: 300

Thread 4 withdrawing 300, balance is now: 0

Thread 4 withdrawing 102, balance is now: -102

Employee completes standing order out of: 200

Thread 5 checks balance, balance is: -302

Thread 6 checks balance, balance is: -302

Expected Output 4:

Balance at start of thread: 0

Thread one deposits 99, balance is now: 99

Employee completes standing order in of: 101

Thread two deposits 100, balance is now: 300

Balance is now: 300

Thread 4 withdrawing 300, balance is now: 0

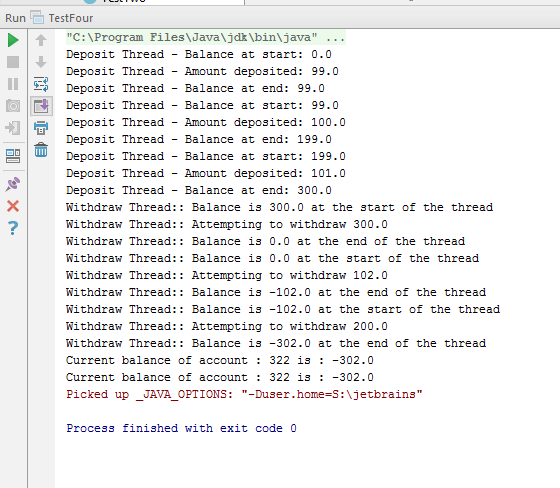
Thread 4 withdrawing 102, balance is now: -102

Employee completes standing order out of: 200

Thread 5 checks balance, balance is: -302

Thread 6 checks balance, balance is: -302

Actual Output:



Test Driver 5, attempting to withdraw 100 with a balance of 0, the system waits till the balance condition grows by letting the account handle a deposit then the withdrawal gets made:

Expected Output:

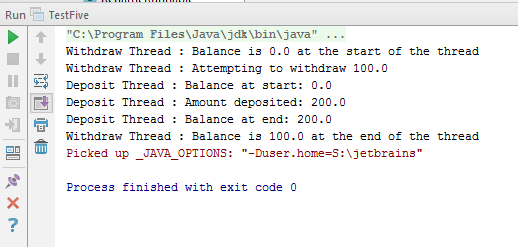
Balance at start of the thread: 0

Thread one attempts to withdraw 100, transaction waits for account to deposit, balance still: 0

Thread two completes a deposit of 200, balance now: 200

Withdrawal of 100 Transaction completes and the balance is now: 100

Actual Result



Test driver 6, changing a name from AA0001 to BB0001 then to CC0001 by starting two threads:

Expected Output:

Name Initially: AA0001

Name changed from: AA0001 to CC0001

New Customer Id: CC0001

Name changed from: CC0001to BB0001

New Customer Id: BB0001

Expected Output 2:

Name Initially: AA0001

Name changed from: AA0001 to BB0001

New Customer Id: BB0001

Name changed from: BB0001to CC0001

New Customer Id: CC0001

Expected Output 3:

Name Initially: AA0001

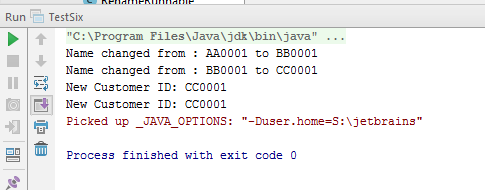
Name changed from: AA0001 to BB0001

Name changed from: BB0001to CC0001

New Customer Id: CC0001

New Customer Id: CC0001

Actual Result:



# Mutation Testing (Adding Bugs) and Description of Outcome

<<to be added >>

# Search and Filter Task B Demo (Screen Shots)

<<to be added >>

# Description of Thread Groups and Threads in Task B

<<to be added >>