Withdraw Test Report

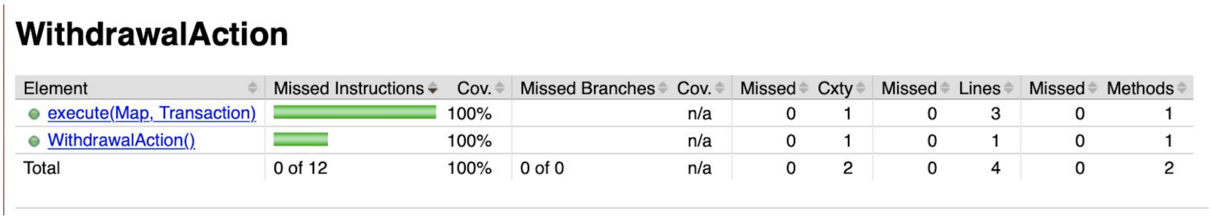
Source Listing

- execute
    - Action.assertExist
    - Action.apply
        - Account.applyTransaction
            - Account.balanceAdjust

Test Cases

Method of white-box testing: **Statement Coverage Testing**

Total Coverage:



WithdrawalAction

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
● execute(Map, Transaction)	🟩	100%		n/a	0	1	0	3	0	1
● WithdrawalAction()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 12	100%	0 of 0	n/a	0	2	0	4	0	2

Testcases:

withdrawTest	The transaction has a valid account listed as the source account and the value is set for all the funds in the account. This test should cover all the statements in the WithdrawAction.execute function, the 2nd statement of Account.applyTransaction, and the final statement of Account.adjustBalance
withdrawNothing	The transaction has a valid account listed as the source account but the value is set to 0. This test should cover all the statements in the WithdrawAction.execute function, the 2nd statement of Account.applyTransaction, and the final statement of Account.adjustBalance
withdrawNegative	The transaction has a valid account listed as the source account but the value is negative. This test should cover all the statements in the WithdrawAction.execute function, the 1st statement of Account.applyTransaction, and nothing else.
insufficientFounds	The transaction has a valid account listed as the source account but the value exceeds the funds in the account. This test should cover all the statements in the WithdrawAction.execute function, the 2nd statement of Account.applyTransaction, and the first statement of Account.adjustBalance
nonexistentAccount	The transaction has an invalid account listed as the source account. This test should cover only the first statement in the

	WithdrawalAction.execute the function and will cover the statements that throw the exception for the invalid source account

Test Input

Transaction Summary File, Line Input					Master Accounts File, Line Input
withdrawTest					
WDR	1000000	10000	0000000	***	1000000
withdrawNothing					
WDR	1000000	00000	0000000	***	1000000
insufficientFounds					
WDR	1000000	99999	0000000	***	1000000
nonexistentAccount					
WDR	9000000	10000	0000000	***	1000000

Test Report

Test Case	If pass before fix	Problam found	Fix	If pass after fix
withdrawTest	passed			passed
withdrawNothing	passed			passed
insufficientFounds	failed	An exception was thrown after the account balance had been modified. Expected behaviour was that balance should not be changed on exception	The error condition check was moved before the balance was assigned to the new value	passed
nonexistentAccount	passed			passed