1. Diseño de pruebas (Collections - Stack)

Clase	Método	Escenario	Entrada	Salida
TestStack	popTest1()	Setup1()	-----	Null, because the pop doesn't exist
TestStack	popTest2()	Setup1()	-----	True, because the size of the Stack is one.
TestStack	popTest3()	Setup1()	-----	True, because the value is equals.
TestStack	pushTest1()	Setup1()	-----	Equals, because the size of the Stack is one.
TestStack	pushTest2()	Setup1()	-----	True, because the size of the Stack is two.
TestStack	pushTest3()	Setup1()	-----	True, because the size of the Stack is two.
TestStack	peekTest1()	Setup1()	-----	True, because the peak exists, but is null.
TestStack	peekTest2()	Setup1()	-----	True, because the peak doesn't exist.
TestStack	peekTest3()	Setup1()	-----	True, because the String is equals.
TestStack	isEmptyTest1()	Setup1()	-----	True, because the Stack is empty.
TestStack	isEmptyTest2()	Setup1()	-----	False, because the Stack doesn't empty.
TestStack	isEmptyTest3()	Setup1()	-----	False, because the Stack doesn't empty.

2. Diseño de pruebas (Collections - Queue)

Clase	Método	Escenario	Entrada	Salida
TestQueue	offerTest1()	setup1()	-----	True, because the size of the queue is one.
TestQueue	offerTest2()	setup1()	-----	True, because the size of the queue is two.
TestQueue	offerTest3()	setup1()	-----	True, because the size of the queue is four.
TestQueue	pollTest1()	setup1()	-----	Null, because the poll of the queue doesn't exist.
TestQueue	pollTest2()	setup1()	-----	True, because the poll of the queue exists.
TestQueue	pollTest3()	setup1()	-----	True, because the poll is equal.
TestQueue	peekTest1()	setup1()	-----	Null, because the peak of the queue doesn't exist.
TestQueue	peekTest2()	setup1()	-----	True, because the peek exists, but is null.
TestQueue	peekTest3()	setup1()	-----	True, because the peek is equal.
TestQueue	isEmptyTest1()	setup1()	-----	True, because the queue is empty.
TestQueue	isEmptyTest2()	setup1()	-----	False, because the queue it's no empty.
TestQueue	isEmptyTest3()	setup1()	-----	False, because the queue it's no empty.

3. Diseño de pruebas (Collections - HashTable)

Clase	Método	Escenario	Entrada	Salida
TestHashTable	insertTest1()	setup1()	-----	True, because the size is one.
TestHashTable	insertTest2()	setup1()	-----	True, because the size is five.
TestHashTable	insertTest3()	setup1()	-----	True,the size is three thousand.
TestHashTable	deleteTest1()	setup1()	-----	Null, deletes an element that doesn't exist.
TestHashTable	deleteTest2()	setup1()	-----	True, deletes one element.
TestHashTable	deleteTest3()	setup1()	-----	True,the size is two.
TestHashTable	searchTest1()	setup1()	-----	Null, because it searches for an element that doesn't exist.
TestHashTable	searchTest2()	setup1()	-----	Not Null, because it searches for an element that does exist.
TestHashTable	searchTest3()	setup1()	-----	True, because the value is 7568.
TestHashTable	isEmptyTest1()	setup1()	-----	Equals, because the Hash Table is empty.
TestHashTable	isEmptyTest2()	setup1()	-----	False, because the Hash Table is not empty.
TestHashTable	isEmptyTest3()	setup1()	-----	False, because the Hash Table is not empty.

4. Diseño de pruebas (Collections - Priority Queue)

Clase	Método	Escenario	Entrada	Salida
TesPriorityQueue	maxHeapInsertTest1()	setup1()	-----	True, the size of the priority queue is one.
TesPriorityQueue	maxHeapInsertTest2()	setup1()	-----	True, the size of the priority queue is zero.
TesPriorityQueue	maxHeapInsertTest3()	setup1()	-----	True the priority queue is empty..
TesPriorityQueue	heapExtractMaxTest1()	setup1()	-----	True, the heap max is null.
TesPriorityQueue	heapExtractMaxTest2()	setup1()	-----	True, the heap max exists.
TesPriorityQueue	heapExtractMaxTest3()	setup1()	-----	True, the heap max is null.
TesPriorityQueue	maxHeapifyTest1()	setup1()	-----	Null, max heapify is correct.
TesPriorityQueue	maxHeapifyTest2()	setup1()	-----	Equals, the max heapify is correct.
TesPriorityQueue	maxHeapifyTest3()	setup1()	-----	True, the max heapify is equal.
TesPriorityQueue	isEmptyTest1()	setup1()	-----	Equals, empty is equals.
TesPriorityQueue	isEmptyTest1()	setup1()	-----	False, the priority queue is not empty.
TesPriorityQueue	isEmptyTest1()	setup1()	-----	True, the size is one.

5. Diseño de pruebas (Model - Active Client)

Clase	Método	Escenario	Entrada	Salida
TestActiveClient	testPayCreditCard1()	setup1()	-----	False, because the object doesn't exist.
TestActiveClient	testPayCreditCard2()	setup2()	-----	False, because the money it's not correct.
TestActiveClient	testPayCreditCard3()	setup1()	-----	False, because the object doesn't exist.
TestActiveClient	testRetrieveCredit1()	setup1()	-----	False, because the object doesn't exist.
TestActiveClient	testRetrieveCredit2()	setup2()	-----	False, because the money it's not correct.
TestActiveClient	testRetrieveCredit3()	setup3()	-----	False, because the money it's not correct.
TestActiveClient	retrieveSavings1()	setup1()	-----	False, because the String is incorrect.
TestActiveClient	retrieveSavings2()	setup2()	-----	True, because the object exists and the String is correct.
TestActiveClient	retrieveSavings3()	setup3()	-----	True, all the objects exist in the system.
TestActiveClient	testAddSaving1()	setup1()	-----	False, because the object doesn't exist.
TestActiveClient	testAddSaving1()	setup2()	-----	True and false, because the first element exists, the another one not.
TestActiveClient	testAddSaving1()	setup3()	-----	True, all the objects exist in the system.
TestActiveClient	testCreateSavingsAccount1()	setup1()	-----	True, because the system creates a new account correctly.
TestActiveClient	testCreateSavingsAccount2()	setup2()	-----	True and false, the first element exists, the another one not.

TestActiveClient	testCreateSavingsAccount3()	setup3()	-----	False, because the object doesn’t exist.
TestActiveClient	testCreateCreditCard1()	setup1()	-----	True, because the object exists and the String is correct.
TestActiveClient	testCreateCreditCard2()	setup2()	-----	True and false, the first element exists, the another one not.
TestActiveClient	testCreateCreditCard3()	setup3()	-----	False, because the object doesn’t exist.
TestActiveClient	testGetCreditCardNumbers1()	setup1()	-----	Null, the object doesn’t exist in the system.
TestActiveClient	testGetCreditCardNumbers2()	setup2()	-----	True, the String is equal to the system.
TestActiveClient	testGetCreditCardNumbers3()	setup3()	-----	True, all the objects exist in the system.
TestActiveClient	testGetSavingsAccountsNumbers1()	setup1()	-----	Null, the object doesn’t exist in the system.
TestActiveClient	testGetSavingsAccountsNumbers2()	setup2()	-----	True, the String is equal to the system.
TestActiveClient	testGetSavingsAccountsNumbers3()	setup3()	-----	True, all the objects exist in the system.
TestActiveClient	testClone1()	setup1()	-----	True. First, the String is equal to the system. Second, all the objects exist in the software.
TestActiveClient	testClone1()	setup2()	-----	True. First, the String is equal to the system. Second, all the objects exist in the software.
TestActiveClient	testClone1()	setup3()	-----	False, all the objects don't exist in the software.

6. Diseño de pruebas (Model - Savings Account)

Clase	Método	Escenario	Entrada	Salida
TestSavingsAccount	testDebit1()	setup1()	-----	True, because the amount is correct in the method.
TestSavingsAccount	testDebit1()	setup1()	-----	True, because the amount is correct in the method.
TestSavingsAccount	testDebit1()	setup1()	-----	True, because the amount is correct in the method.
TestSavingsAccount	testCredit1()	setup1()	-----	True, because the amount is correct in the method.
TestSavingsAccount	testCredit1()	setup1()	-----	True, because the amount is correct in the method.
TestSavingsAccount	testCredit1()	setup1()	-----	True, because the amount is correct in the method.

7. Diseño de pruebas (Model - Current Account)

Clase	Método	Escenario	Entrada	Salida
TestCurrentAccount()	testPay1()	setup1()	-----	False, because the amount is incorrect to the system.
TestCurrentAccount()	testPay2()	setup1()	-----	True, because the amount is correct to the system.
TestCurrentAccount()	testPay3()	setup1()	-----	False, because the amount is incorrect to the system.
TestCurrentAccount()	testUse1()	setup1()	-----	True, because the amount is correct to the system.
TestCurrentAccount()	testUse2()	setup1()	-----	True, because the amount is correct to the system.
TestCurrentAccount()	testUse3()	setup1()	-----	True, because the amount is correct to the system.

8. Diseño de pruebas (Modelo - Bank)

Clase	Metodo	Escenario	Entrada	Salida
TestBank	testBank1()	setup1()	-----	Equals, because the name of the bank is correct.
TestBank	testBank2()	setup1()	-----	False, because that name is incorrect to the bank.
TestBank	testBank3()	setup1()	-----	Not null, because the object exists.
TestBank	testAddNewActiveClient1()	setup1()	-----	True, because the object was added to the system.
TestBank	testAddNewActiveClient2()	setup1()	-----	False, because the object wasn't added to the system.
TestBank	testAddNewActiveClient3()	setup1()	-----	Not null, because the object was added to the system.
TestBank	testGetActiveClientsArray1()	setup2()	-----	Not null, because the array exists in the system.
TestBank	testGetActiveClientsArray2()	setup2()	-----	Equals, because the size of the array is three.
TestBank	testGetActiveClientsArray3()	setup2()	-----	False, because the array exists in the system. In this case, it isn't null.
TestBank	testPayCreditCard1()	setup1()	-----	False, because the user doesn't exist in the software.
TestBank	testPayCreditCard2()	setup1()	-----	Equals, because the software can't do the action.

TestBank	testPayCreditCard3()	setup1()	-----	True, because the software can't pay with a credit card that doesn't exist.
TestBank	testRetrieveCreditCard1()	setup1()	-----	False, because the user doesn't exist in the software.
TestBank	testRetrieveCreditCard2()	setup1()	-----	Equals, because the software can't do the action.
TestBank	testRetrieveCreditCard3()	setup1()	-----	True, because the software can't pay with a credit card that doesn't exist.
TestBank	testRetrieveSavings1()	setup1()	-----	False, because the user doesn't exist in the software.
TestBank	testRetrieveSavings2()	setup1()	-----	Equals, because the software can't do the action.
TestBank	testRetrieveSavings3()	setup1()	-----	True, because the software can't pay with a credit card that doesn't exist.
TestBank	testAddSavings1()	setup1()	-----	False, because the user doesn't exist in the software.
TestBank	testAddSavings2()	setup1()	-----	False, because a client can't have the same id as another one.
TestBank	testAddSavings3()	setup1()	-----	Not null, because the system adds a new saving, but without any client.
TestBank	testCreateSavingsAccount1()	setup1()	-----	False, because the software can create an account without a client.
TestBank	testCreateSavingsAccount2()	setup1()	-----	Not null, because it creates an element, but doesn't connect with a client.

TestBank	testCreateSavingsAccount3()	setup1()	-----	Equals, because the software can create an account without a client.
TestBank	testCreateCreditCard1()	setup1()	-----	False, because the software can create an account without a client.
TestBank	testCreateCreditCard2()	setup1()	-----	Not null, because it creates an element, but doesn’t connect with a client.
TestBank	testCreateCreditCard3()	setup1()	-----	Equals, because the software can create an account without a client.
TestBank	testUndoLastAction1()	setup1()	-----	False, because the software can take an action that doesn't exist.
TestBank	testUndoLastAction2()	setup1()	-----	Equals, because the software can take an action that doesn't exist.
TestBank	testUndoLastAction3()	setup1()	-----	True, because the software can take an action that doesn't exist.
TestBank	testClearActions1()	setup1()	-----	Not null, because it creates an element, but doesn’t connect with the bank.
TestBank	testClearActions2()	setup1()	-----	True, because all the actions are different to null.
TestBank	testClearActions3()	setup1()	-----	True, because the system clears all the actions correctly.
TestBank	testAssignClientToQueue1()	setup1()	-----	True, because the queue doesn’t have any element.
TestBank	testAssignClientToQueue2()	setup1()	-----	Equals, because the queue doesn’t have any element.

TestBank	testAssignClientToQueue3()	setup1()	-----	True, because the queue doesn’t have any element.
TestBank	testAttendNextClient1()	setup1()	-----	False, because there doesn't exist another client.
TestBank	testAttendNextClient2()	setup1()	-----	Equals, because there doesn't exist another client in the line.
TestBank	testAttendNextClient3()	setup1()	-----	True, because there doesn't exist another client.
TestBank	testRemoveActiveClient1()	setup2()	-----	True, because the system has 3 clients.
TestBank	testRemoveActiveClient2()	setup2()	-----	True, because if the system removes a client, the system will have zero clients.
TestBank	testRemoveActiveClient3()	setup2()	-----	True, because the system has 3 clients.
TestBank	testSearchActiveClientById1()	setup2()	-----	True, because the object exists and this is different to null.
TestBank	testSearchActiveClientById2()	setup2()	-----	Equals, because this is the correct name in the system.
TestBank	testSearchActiveClientById3()	setup2()	-----	Not null, because the element with that id exists.
TestBank	testGetQueue1()	setup1()	-----	Not Null, because the queue exists in the software.
TestBank	testGetQueue2()	setup1()	-----	True, because the queue is different to null in this case.
TestBank	testGetQueue3()	setup1()	-----	Equals, because the size of the queue in the system is zero.

TestBank	testGetPriorityQueue1()	setup1()	-----	Not Null, because the queue exists in the software.
TestBank	testGetPriorityQueue2()	setup1()	-----	True, because the priority queue exists in the software.
TestBank	testGetPriorityQueue3()	setup1()	-----	Equals, because the size of the priority queue in the system is zero.