

Junit Lab 2

Software Testing



April 6, 2023

Abdullah Shafique

FA19-BSE-094

# BankAccount Class:

public class BankAccount {  
 public String accountNumber;  
 public int balance;  
  
 public BankAccount(String accountNumber, int balance) {  
 this.accountNumber = accountNumber;  
 this.balance = balance;  
 }  
  
 public void depositMoney(int amount) {  
 balance += amount;  
 }  
  
 public void withdrawMoney(int amount) {  
 balance -= amount;  
 }  
  
 public void transferMoney(int amount) {  
 balance -= amount;  
 }  
  
 public int checkBalance() {  
 return balance;  
 }  
}

# BankTests Class:

import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.AfterAll;  
import org.junit.jupiter.api.BeforeAll;  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.*assertEquals*;  
import static org.junit.jupiter.api.Assertions.*assertFalse*;  
  
public class BankTests {  
  
 @BeforeAll  
 public static void beforeAll() {  
 System.*out*.println("Before All of Bank Account tests");  
 }  
  
 @BeforeEach  
 public void before() {  
 System.*out*.println("Before a new test case");  
 }  
  
 @Test  
 public void test1() {  
 BankAccount account = new BankAccount("account-52", 200);  
  
 account.depositMoney(1000);  
// assertEquals(1200, account.balance);  
  
 *assertFalse*(account.balance != 1100);  
 }  
  
 @Test  
 public void test2() {  
 BankAccount first = new BankAccount("account-52", 200);  
 BankAccount second = new BankAccount("account-53", 400);  
  
 first.transferMoney(100);  
 second.depositMoney(100);  
  
 *assertEquals*(100, first.balance);  
  
// assertFalse(second.balance == 400);  
 }  
  
  
 @Test  
 public void test3() {  
 BankAccount account = new BankAccount("account-52", 200);  
  
 account.withdrawMoney(100);  
 *assertEquals*(100, account.balance);  
  
// assertFalse(account.balance == 200);  
 }  
  
 @Test  
 public void test4() {  
 BankAccount account = new BankAccount("account-52", 200);  
  
 int balance = account.checkBalance();  
  
 *assertEquals*(200, account.balance);  
  
// assertFalse(balance != 100);  
 }  
  
 @AfterAll  
 public static void afterAll() {  
 System.*out*.println("After all of Bank Account tests");  
 }  
}

# MISC Class:

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
public class MISC {  
  
 // copy an array by iterating the array  
 public static int[] copyArray(int[] arr) {  
 int[] copy = new int[arr.length];  
 for (int i = 0; i < arr.length; i++) {  
 copy[i] = arr[i];  
 }  
 return copy;  
 }  
  
 // find the common elements between two arrays  
 public static int[] findCommonElements(int[] arr1, int[] arr2) {  
 List<Integer> commonElements = new ArrayList<>();  
 for (int i = 0; i < arr1.length; i++) {  
 for (int j = 0; j < arr2.length; j++) {  
 if (arr1[i] == arr2[j]) {  
 if (!commonElements.contains(arr1[i])) {  
 commonElements.add(arr1[i]);  
 }  
 break;  
 }  
 }  
 }  
 int[] result = new int[commonElements.size()];  
 for (int i = 0; i < commonElements.size(); i++) {  
 result[i] = commonElements.get(i);  
 }  
 return result;  
 }  
  
 // remove duplicate elements from an array  
 public static int[] removeDuplicates(int[] arr) {  
 List<Integer> list = new ArrayList<>();  
 for (int i = 0; i < arr.length; i++) {  
 if (!list.contains(arr[i])) {  
 list.add(arr[i]);  
 }  
 }  
 int[] result = new int[list.size()];  
 for (int i = 0; i < list.size(); i++) {  
 result[i] = list.get(i);  
 }  
 return result;  
 }  
  
 // remove a specific element from an array  
 public static int[] removeElement(int[] arr, int element) {  
 int count = 0;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] == element) {  
 count++;  
 }  
 }  
 int[] result = new int[arr.length - count];  
 int index = 0;  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i] != element) {  
 result[index++] = arr[i];  
 }  
 }  
 return result;  
 }  
  
}

# MISCTests Class:

import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class MISCTest {  
  
 @Test  
 public void testCopyArray() {  
 int[] arr = {1, 2, 3, 4, 5};  
 int[] copy = MISC.*copyArray*(arr);  
 *assertArrayEquals*(arr, copy);  
 }  
  
 @Test  
 public void testFindCommonElements() {  
 int[] arr1 = {1, 2, 3, 4, 5};  
 int[] arr2 = {3, 4, 5, 6, 7};  
 int[] common = MISC.*findCommonElements*(arr1, arr2);  
 int[] expected = {3, 4, 5};  
 *assertArrayEquals*(expected, common);  
 }  
  
 @Test  
 public void testRemoveDuplicates() {  
 int[] arr = {1, 2, 2, 3, 3, 3, 4, 5, 5};  
 int[] unique = MISC.*removeDuplicates*(arr);  
 int[] expected = {1, 2, 3, 4, 5};  
 *assertArrayEquals*(expected, unique);  
 }  
  
 @Test  
 public void testRemoveElement() {  
 int[] arr = {1, 2, 3, 4, 5};  
 int[] removed = MISC.*removeElement*(arr, 3);  
 int[] expected = {1, 2, 4, 5};  
 *assertArrayEquals*(expected, removed);  
 }  
  
}