Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit

import org.junit.\*;

import static org.junit.Assert.\*;

import java.util.ArrayList;

import java.util.List;

public class ShoppingCartTest {

private ShoppingCart cart;

private List<String> testItems;

@Before

public void setUp() {

cart = new ShoppingCart();

testItems = new ArrayList<>();

testItems.add("Apple");

testItems.add("Banana");

testItems.add("Orange");

}

@After

public void tearDown() {

cart = null;

testItems = null;

}

@Test

public void testAddItem() {

// Arrange

String item = "Mango";

int initialSize = cart.getItemCount();

// Act

cart.addItem(item);

// Assert

assertEquals(initialSize + 1, cart.getItemCount());

assertTrue(cart.containsItem(item));

}

@Test

public void testRemoveItem() {

// Arrange

String item = testItems.get(0);

cart.addItem(item);

int initialSize = cart.getItemCount();

// Act

boolean result = cart.removeItem(item);

// Assert

assertTrue(result);

assertEquals(initialSize - 1, cart.getItemCount());

assertFalse(cart.containsItem(item));

}

@Test

public void testTotalPriceCalculation() {

// Arrange

for (String item : testItems) {

cart.addItem(item);

}

double expectedTotal = 15.97;

// Act

double actualTotal = cart.calculateTotal();

// Assert

assertEquals(expectedTotal, actualTotal, 0.001);

}

@Test

public void testEmptyCart() {

// Arrange/Act (nothing needed)

// Assert

assertEquals(0, cart.getItemCount());

assertTrue(cart.isEmpty());

}

private static class ShoppingCart {

private List<String> items = new ArrayList<>();

public void addItem(String item) {

items.add(item);

}

public boolean removeItem(String item) {

return items.remove(item);

}

public boolean containsItem(String item) {

return items.contains(item);

}

public int getItemCount() {

return items.size();

}

public boolean isEmpty() {

return items.isEmpty();

}

public double calculateTotal() {

double total = 0.0;

for (String item : items) {

switch(item) {

case "Apple": total += 5.99; break;

case "Banana": total += 3.99; break;

case "Orange": total += 5.99; break;

}

}

return total;

}

}

}