**Exercise 4:**

Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in JUnit

Scenario:

You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup

and teardown methods.

Steps:

1. Write tests using the AAA pattern.

2. Use @Before and @After annotations for setup and teardown methods.

**CODE:**

**Java Class: BankAccount.java**

package com.example;

public class BankAccount {

private int balance;

public BankAccount(int initialBalance) {

this.balance = initialBalance;

}

public void deposit(int amount) {

balance += amount;

}

public void withdraw(int amount) {

if (amount <= balance) {

balance -= amount;

}

}

public int getBalance() {

return balance;

}

}

**Test Class: BankAccountTest.java**

package com.example;

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

public class BankAccountTest {

private BankAccount account;

@Before

public void setUp() {

// Arrange: Initialize test fixture

account = new BankAccount(100);

}

@After

public void tearDown() {

// Cleanup (if needed)

account = null;

}

@Test

public void testDeposit() {

// Act

account.deposit(50);

// Assert

assertEquals(150, account.getBalance());

}

@Test

public void testWithdraw() {

// Act

account.withdraw(30);

// Assert

assertEquals(70, account.getBalance());

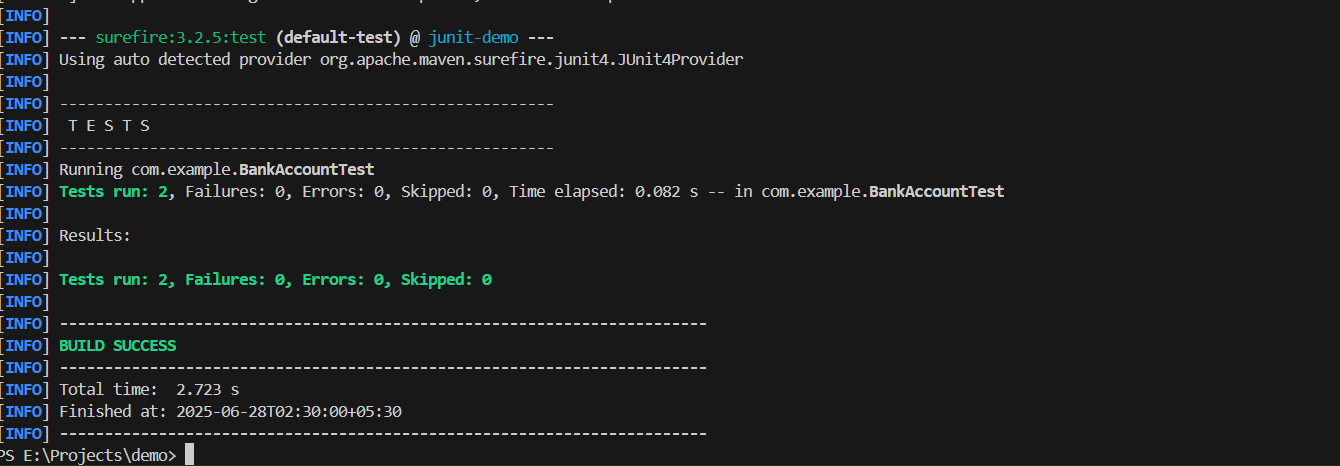
}

}

**Run This Test:**

mvn -Dtest=BankAccountTest test

**Output:**

**

**Conclusion:**

* Demonstrated the Arrange-Act-Assert (AAA) pattern.
* Used @Before and @After to manage test setup and teardown.
* Ensured test isolation and clarity by separating preparation, execution, and verification.