Supersetid:6383982

**1.**

**package** com.eg;

**public** **class** TestUnit

{

**public** **int** Sq(**int** n) {

**return** n\*n;

}

}

**package** com.eg;

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** SqUnit {

@Test

**public** **void** test() {

TestUnit obj1=**new** TestUnit();

**int** output=obj1.Sq(4);

*assertEquals*(16,output);

}

}

A computer screen with a green box

AI-generated content may be incorrect.

2.

**package** com.eg;

**import** **static** org.junit.Assert.\*;

**import** org.junit.Test;

**public** **class** AssertionsTest {

@Test

**public** **void** testAssertions() {

// Assert equals

*assertEquals*(5, 2 + 3);

// Assert true

*assertTrue*(5 > 3);

// Assert false

*assertFalse*(5 < 3);

// Assert null

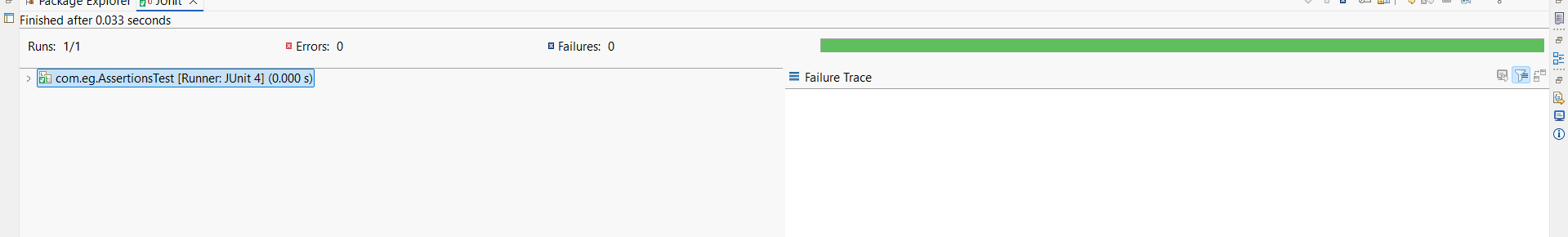
*assertNull*(**null**);

// Assert not null

*assertNotNull*(**new** Object());

}

}



3.  
**package** com.eg;

**import** **static** org.junit.Assert.\*;

**import** org.junit.Before;

**import** org.junit.After;

**import** org.junit.Test;

**public** **class** CalculatorTest {

**private** Calculator calculator;

// Setup method: runs before each test

@Before

**public** **void** setUp() {

calculator = **new** Calculator();

System.***out***.println("Setup: Calculator initialized.");

}

// Teardown method: runs after each test

@After

**public** **void** tearDown() {

calculator = **null**;

System.***out***.println("Teardown: Calculator set to null.");

}

@Test

**public** **void** testAddition() {

// Arrange

**int** a = 5;

**int** b = 3;

// Act

**int** result = calculator.add(a, b);

// Assert

*assertEquals*(8, result);

}

@Test

**public** **void** testSubtraction() {

// Arrange

**int** a = 10;

**int** b = 4;

// Act

**int** result = calculator.subtract(a, b);

// Assert

*assertEquals*(6, result);

}

}

**package** com.eg;

**public** **class** Calculator {

**public** **int** add(**int** a, **int** b) {

**return** a + b;

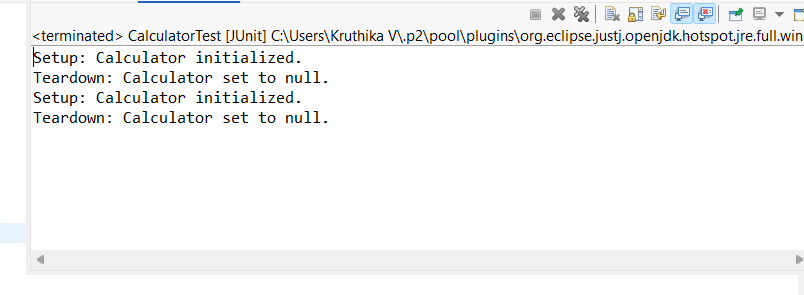
}

**public** **int** subtract(**int** a, **int** b) {

**return** a - b;

}

}



public interface ExternalApi {

String getData();

}  
public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}  
  
import static org.junit.jupiter.api.Assertions.assertEquals;

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testExternalApi() {

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

when(mockApi.getData()).thenReturn("Mock Data");

MyService service = new MyService(mockApi);

String result = service.fetchData();

assertEquals("Mock Data", result);

}

}  
  
  
  
  
  
  
public interface ExternalApi {

String getData();

}

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

import static org.mockito.Mockito.\*;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

public class MyServiceTest {

@Test

public void testVerifyInteraction() {

// Step 1: Create mock

ExternalApi mockApi = Mockito.mock(ExternalApi.class);

// Step 2: Call method that should use mock

MyService service = new MyService(mockApi);

service.fetchData();

// Step 3: Verify method was called

verify(mockApi).getData();

}

}

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Saving book: " + bookName);

}

}  
  
  
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter for dependency injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("BookService: Adding book...");

bookRepository.saveBook(bookName);

}

}  
  
package com.library.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = (BookService) context.getBean("bookService");

bookService.addBook("Wings of Fire");

}

}

package com.library.main;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

private static final Logger logger = LoggerFactory.getLogger(MainApp.class);

public static void main(String[] args) {

logger.info("Application started");

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = (BookService) context.getBean("bookService");

bookService.addBook("Wings of Fire");

logger.warn("This is a warning message");

logger.error("This is an error message");

}

}