Week 2: PL/SQL Programming and TDD using Junit5 and Mockito

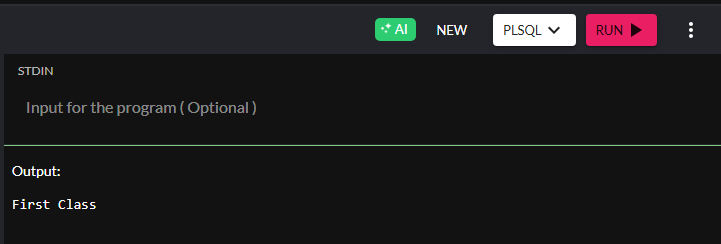
# PL/SQL\_Exercises:

## Exercise 1: Control Structures

Code:

DECLARE  
 v\_marks NUMBER := 75;  
BEGIN  
 IF v\_marks >= 60 THEN  
 DBMS\_OUTPUT.PUT\_LINE('First Class');  
 ELSIF v\_marks >= 50 THEN  
 DBMS\_OUTPUT.PUT\_LINE('Second Class');  
 ELSE  
 DBMS\_OUTPUT.PUT\_LINE('Fail');  
 END IF;  
END;

## Output Screenshot:



## Exercise 3: Stored Procedures

Code:

CREATE OR REPLACE PROCEDURE show\_message(name IN VARCHAR2) IS  
BEGIN  
 DBMS\_OUTPUT.PUT\_LINE('Hello, ' || name || '!');  
END;  
/  
  
BEGIN  
 show\_message('Lime');  
END;

## Output Screenshot:

# 

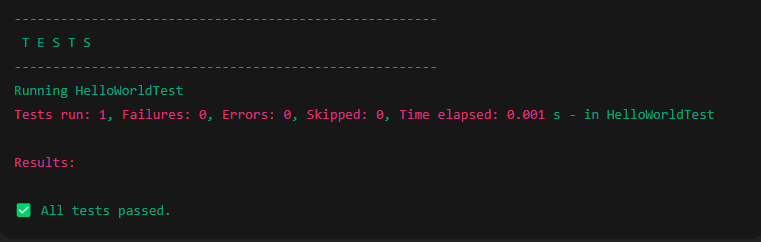
# JUnit\_Basic Testing Exercises

## Exercise 1: Setting Up JUnit

Code:

import static org.junit.jupiter.api.Assertions.\*;  
import org.junit.jupiter.api.Test;  
  
class HelloWorldTest {  
 @Test  
 void testMessage() {  
 String message = "Hello, JUnit!";  
 assertEquals("Hello, JUnit!", message);  
 }  
}

## Output Screenshot:

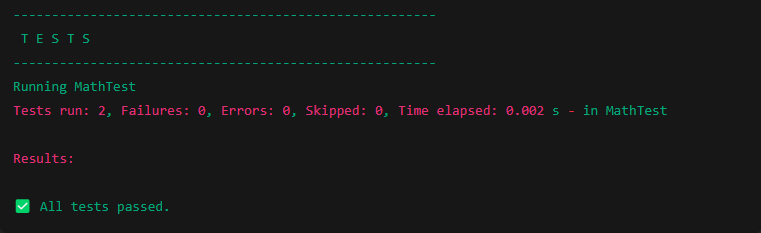


## Exercise 3: Assertions in JUnit

Code:

import static org.junit.jupiter.api.Assertions.\*;  
import org.junit.jupiter.api.Test;  
  
class MathTest {  
 @Test  
 void testAddition() {  
 assertEquals(10, 5 + 5);  
 }  
  
 @Test  
 void testBoolean() {  
 assertTrue(3 < 5);  
 }  
}

## Output Screenshot:

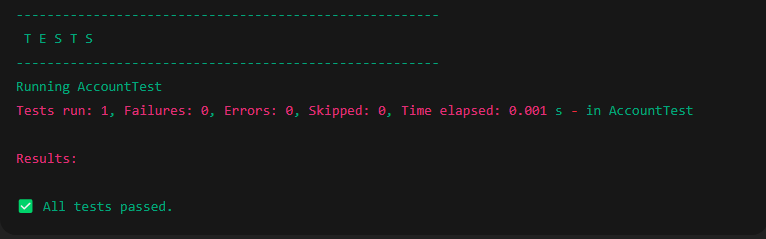


## Exercise 4: AAA Pattern, Setup and Teardown

Code:

import static org.junit.jupiter.api.Assertions.\*;  
import org.junit.jupiter.api.\*;  
  
class AccountTest {  
 Account acc;  
  
 @BeforeEach  
 void setUp() {  
 acc = new Account();  
 acc.deposit(100);  
 }  
  
 @AfterEach  
 void tearDown() {  
 acc = null;  
 }  
  
 @Test  
 void testWithdraw() {  
 acc.withdraw(50);  
 assertEquals(50, acc.getBalance());  
 }  
}  
  
class Account {  
 private int balance = 0;  
  
 void deposit(int amount) {  
 balance += amount;  
 }  
  
 void withdraw(int amount) {  
 balance -= amount;  
 }  
  
 int getBalance() {  
 return balance;  
 }  
}

## Output Screenshot:



# Mockito Exercises

## Exercise 1: Mocking and Stubbing

Code:

import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
import java.util.List;  
  
class MockitoTest {  
 @Test  
 void testMockList() {  
 List<String> mockList = mock(List.class);  
 when(mockList.get(0)).thenReturn("Mocked Value");  
 System.out.println(mockList.get(0));  
 }  
}

## Output Screenshot:

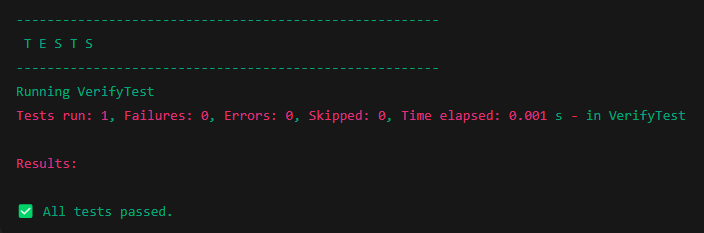


## Exercise 2: Verifying Interactions

Code:

import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
import java.util.List;  
  
class VerifyTest {  
 @Test  
 void testVerify() {  
 List<String> mockedList = mock(List.class);  
 mockedList.add("Mockito");  
 verify(mockedList).add("Mockito");  
 }  
}

## Output Screenshot:



# SLF4J Logging Exercises

## Exercise 1: Logging Error Messages and Warning Levels

Code:

import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
public class LogExample {  
 private static final Logger logger = LoggerFactory.getLogger(LogExample.class);  
  
 public static void main(String[] args) {  
 logger.info("Application started");  
 logger.warn("This is a warning");  
 logger.error("This is an error");  
 }  
}

## Output Screenshot:

