**PLSQL\_Exercises - Answers**

Exercise 1 :

//Scenario 1   
  
BEGIN

FOR cust IN (SELECT customer\_id FROM customers WHERE age > 60) LOOP

UPDATE loans

SET interest\_rate = interest\_rate - 1

WHERE customer\_id = cust.customer\_id;

END LOOP;

COMMIT;

END;

//Scenario 2

BEGIN

FOR cust IN (SELECT customer\_id, balance FROM customers) LOOP

IF cust.balance > 10000 THEN

UPDATE customers

SET is\_vip = 'TRUE'

WHERE customer\_id = cust.customer\_id;

END IF;

END LOOP;

COMMIT;

END;

// Scenario 3

DECLARE

CURSOR due\_loans IS

SELECT l.loan\_id, l.customer\_id, c.name, l.due\_date

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date <= SYSDATE + 30;

BEGIN

FOR loan IN due\_loans LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan.loan\_id ||

' for customer ' || loan.name ||

' is due on ' || TO\_CHAR(loan.due\_date, 'DD-MON-YYYY'));

END LOOP;

END;

Exercise 3

//Scenario 1

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_type = 'SAVINGS';

COMMIT;

END;

//Scenario 2

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept\_id IN NUMBER,

p\_bonus\_pct IN NUMBER

) AS

BEGIN

UPDATE employees

SET salary = salary + (salary \* p\_bonus\_pct)

WHERE department\_id = p\_dept\_id;

COMMIT;

END;

//Scenario 3

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_acc IN NUMBER,

p\_to\_acc IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_acc

FOR UPDATE;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account');

END IF;

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_acc;

UPDATE accounts

SET balance = balance + p\_amount

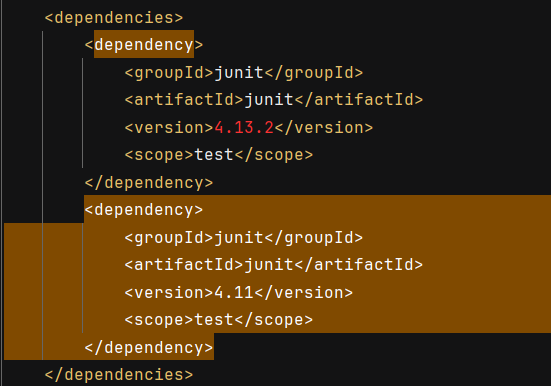
WHERE account\_id = p\_to\_acc;

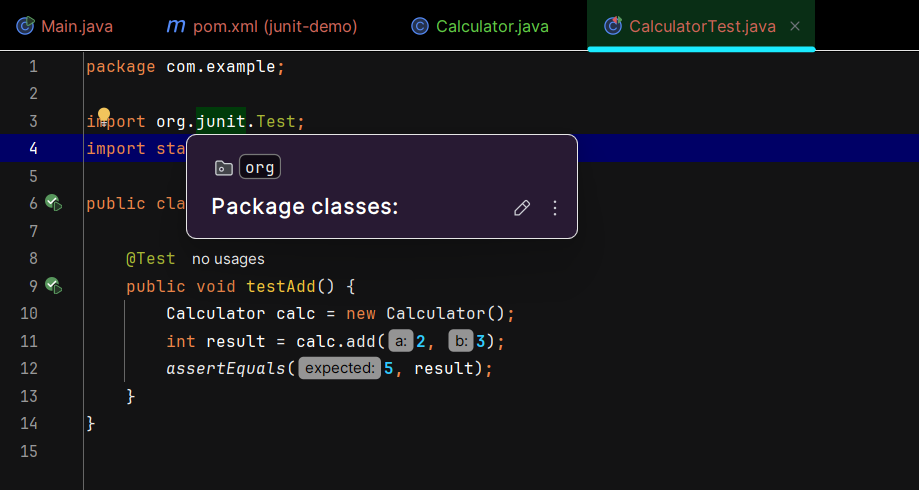
COMMIT;

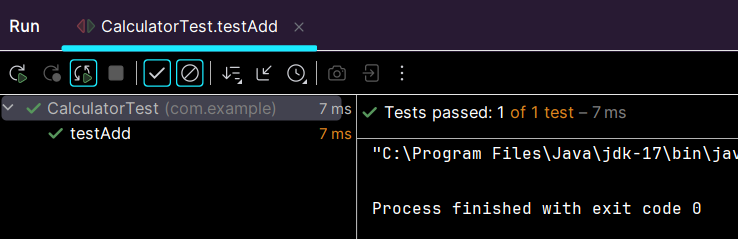
END;

**Junit Basic Testing Exercises:**

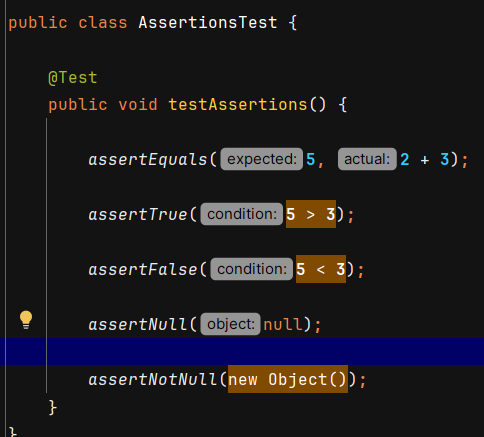
Exercise 1:

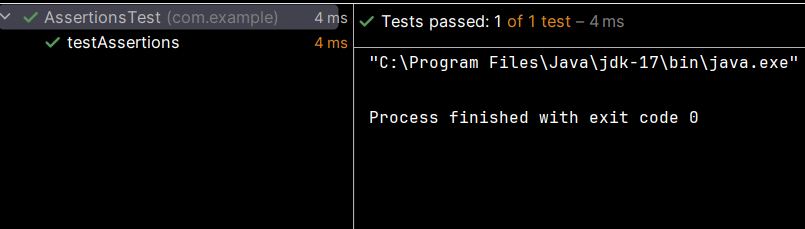




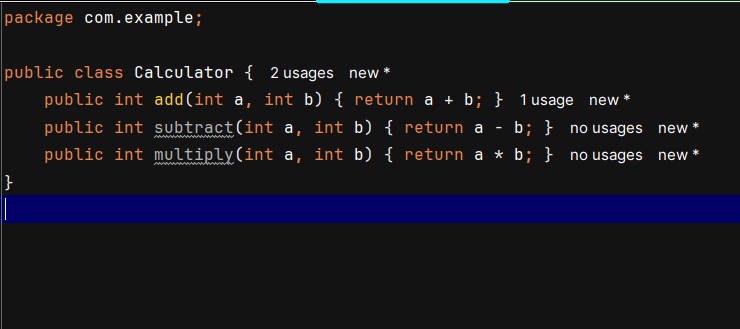


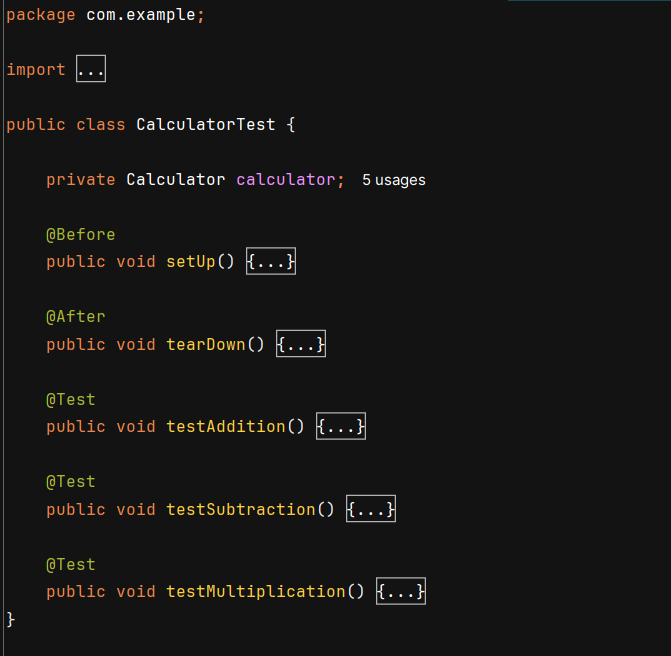
Exercise 3:

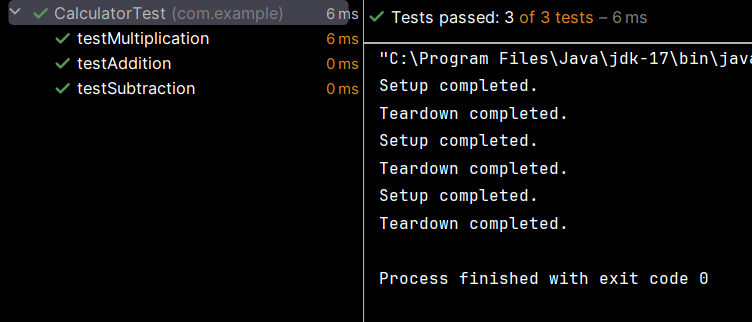




Exercise 4:

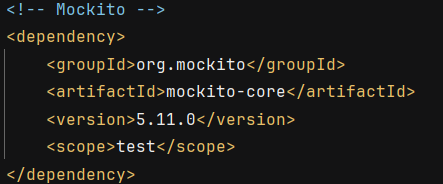


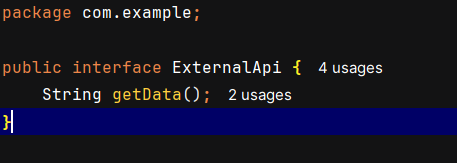


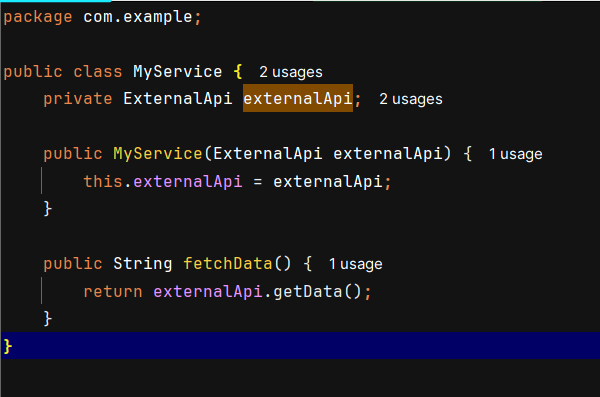


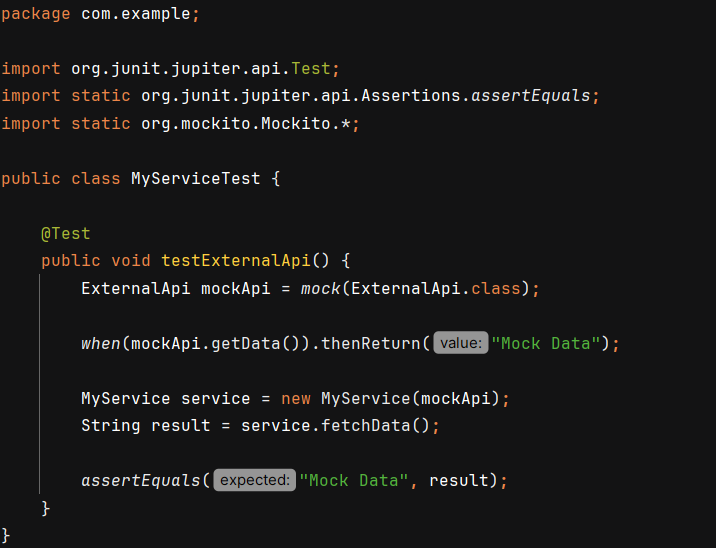
**Mockito Exercises**

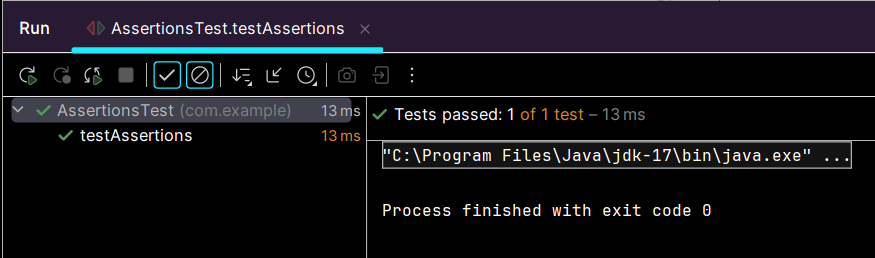
Exercise 1:



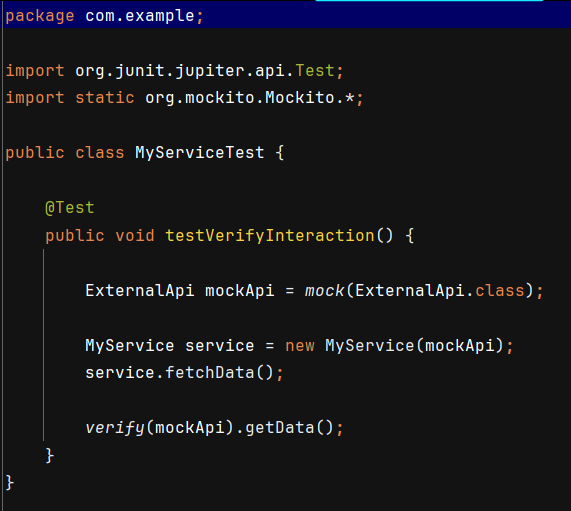


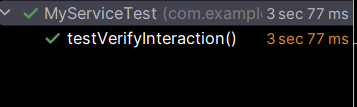






Exercise 2:





**Logging using SLF4J**

