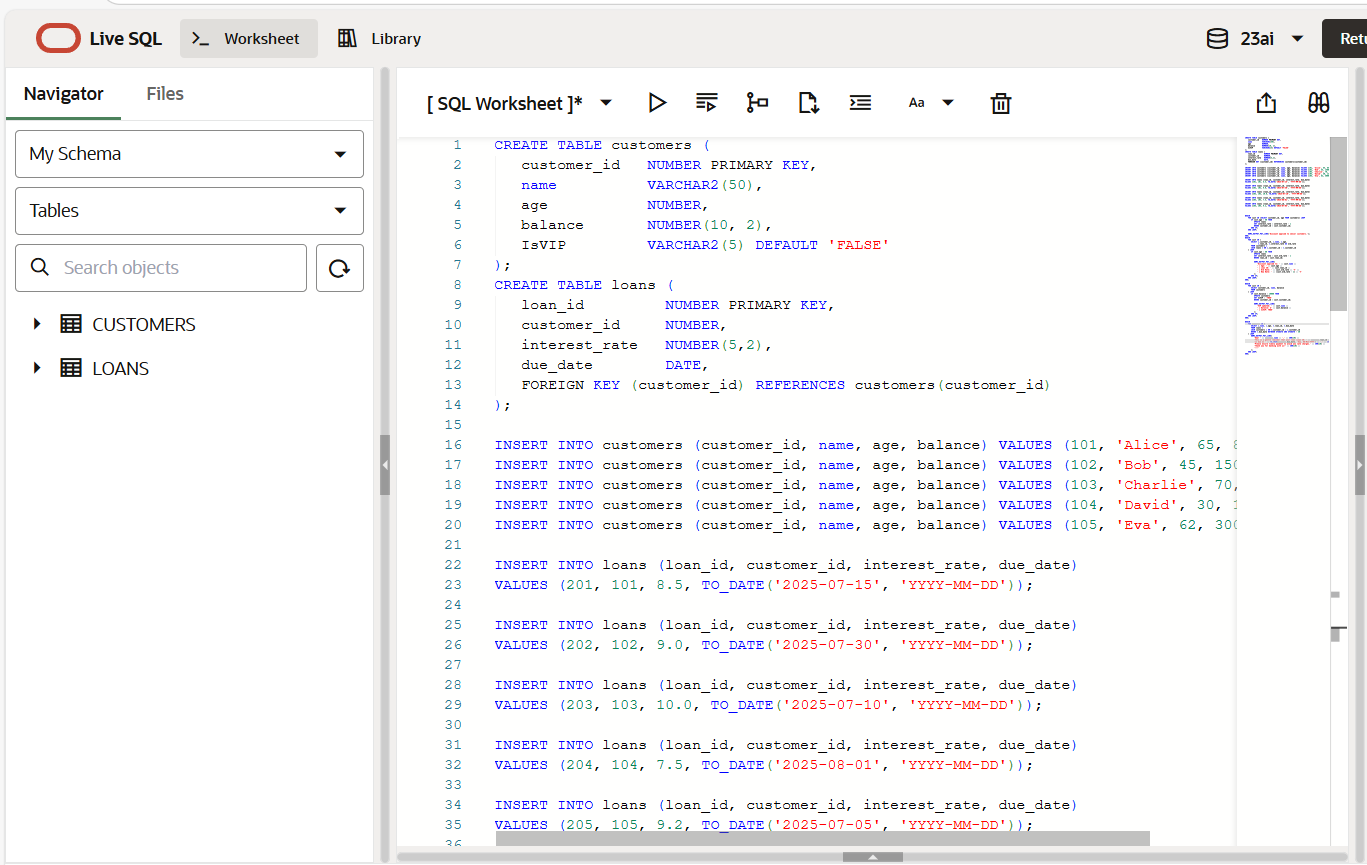
**WEEK – 2 [PL/SQL]**

**Exercise 1: Control Structures**

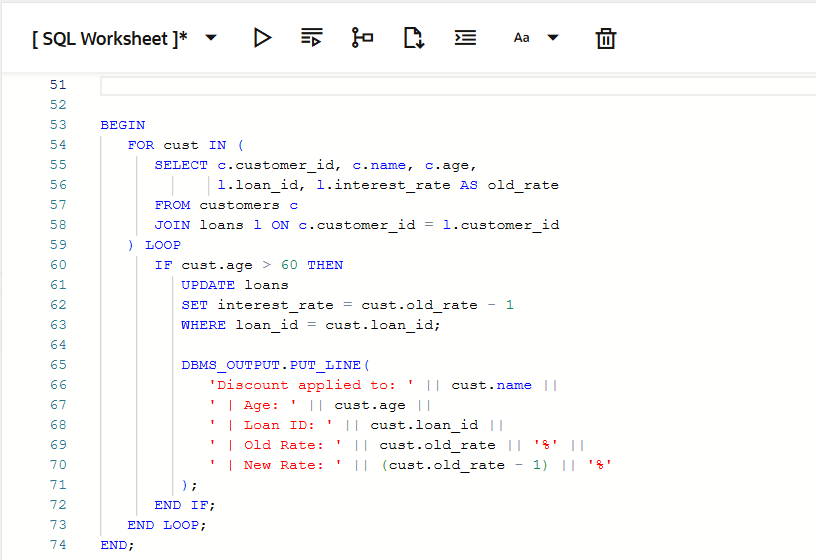
**Table created and inserted the value**

****

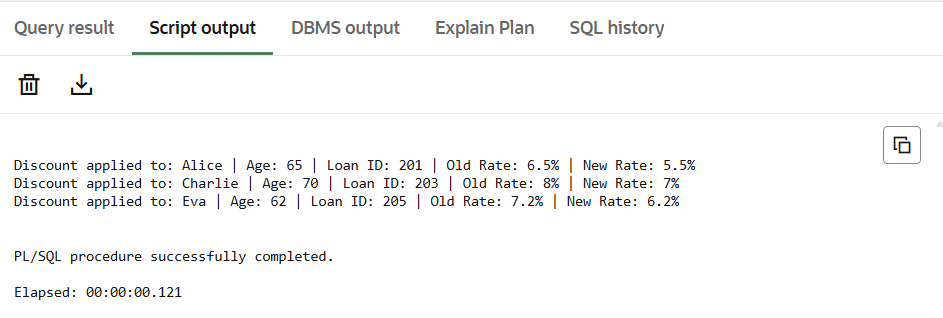
**i) Scenario 1:**

Write a PL/SQL block that loops through all customers, checks their age, and if they are above 60, apply a 1% discount to their current loan interest rates.

**Coding:**



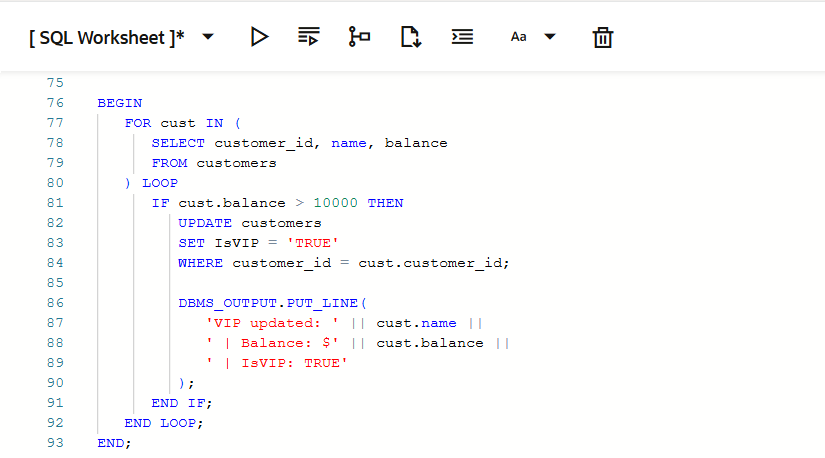
**Output:**

****

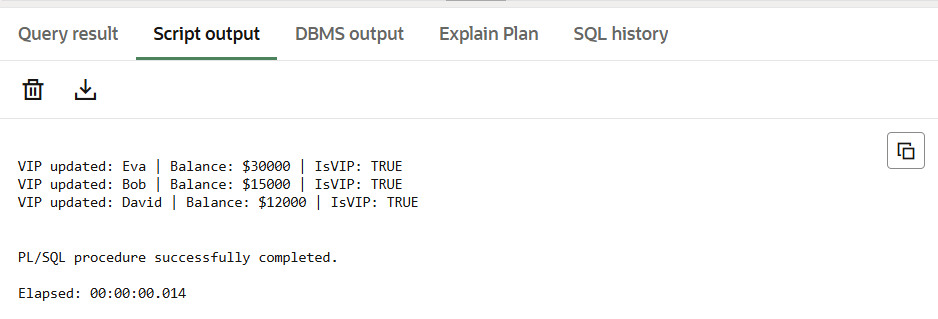
**ii) Scenario 2:**

Write a PL/SQL block that iterates through all customers and sets a flag IsVIP to TRUE for those with a balance over $10,000.

**Coding:**

****

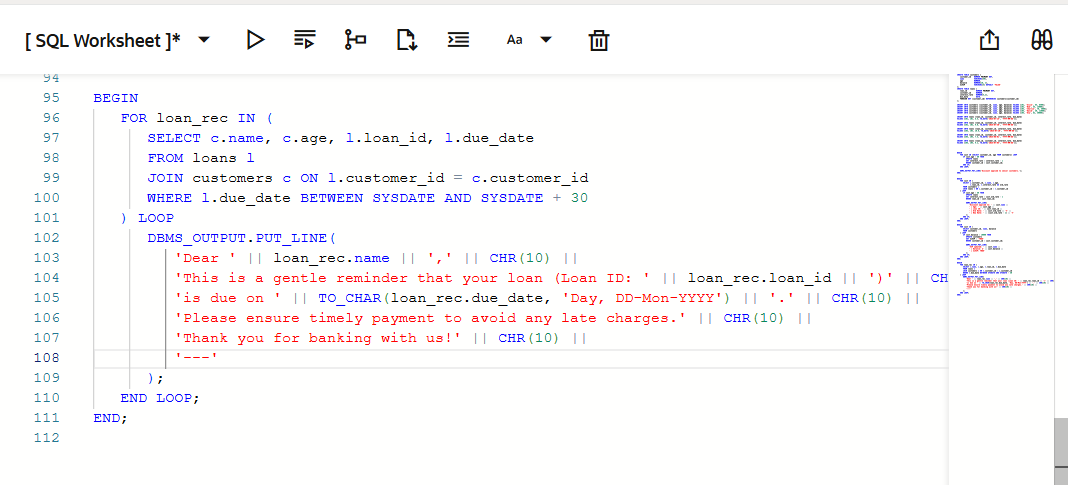
**Output :**

****

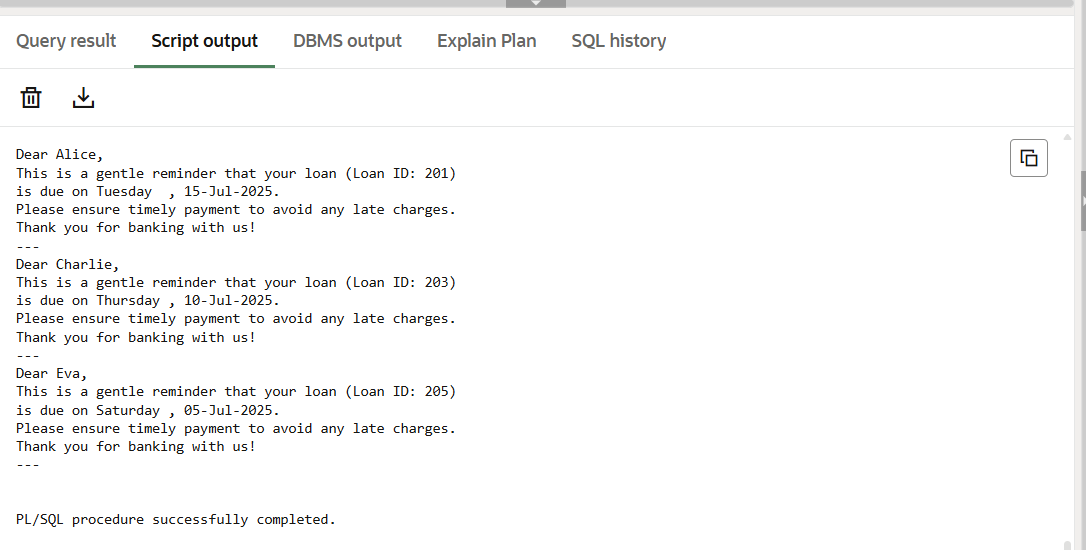
**ii) Scenario 3**:

Write a PL/SQL block that fetches all loans due in the next 30 days and prints a reminder message for each customer

**Coding:**

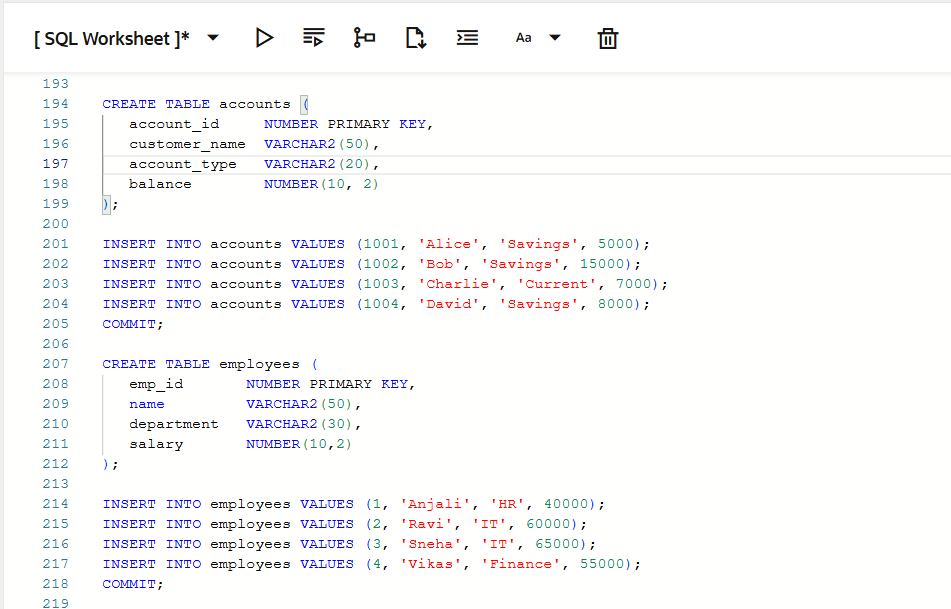
****

**Output:**

****

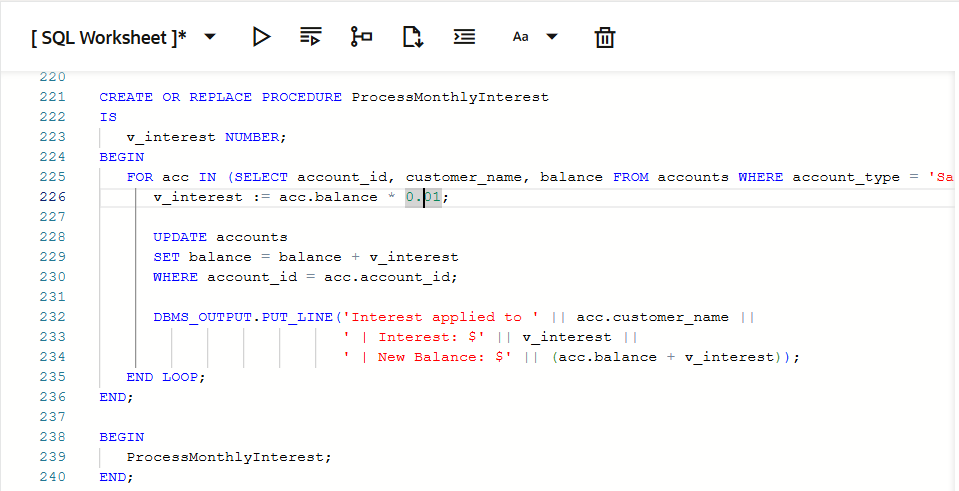
**Exercise 3: Stored Procedures**

**Table Created and Row inserted**

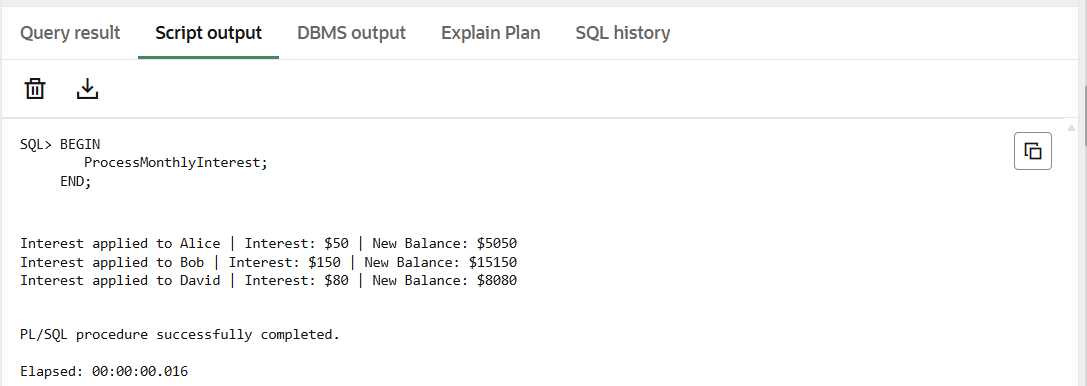
****

**i)Scenario 1:**

Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

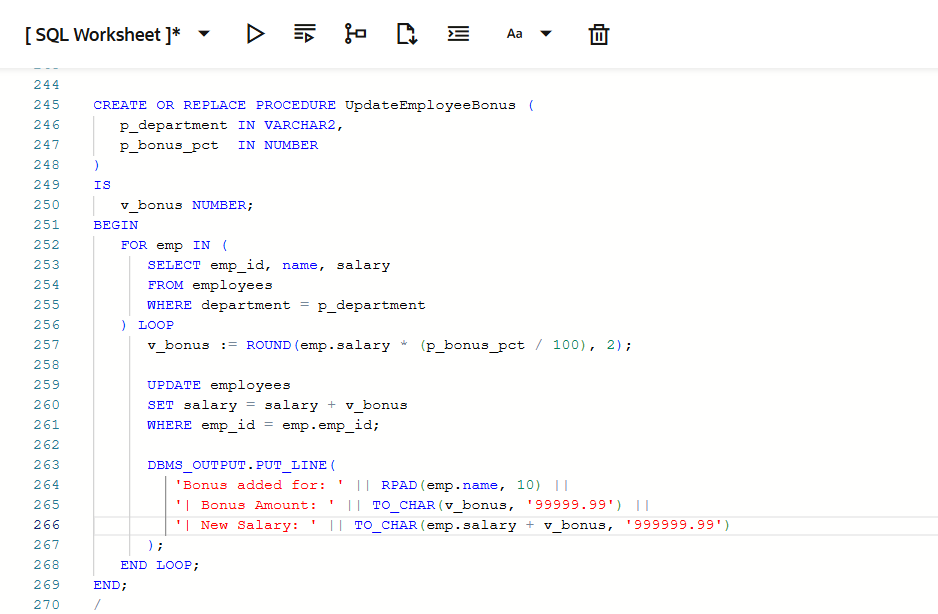


**Output**

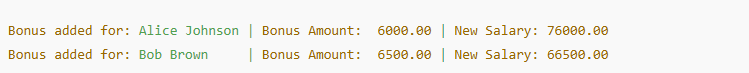
****

**ii)Scenario 2:**

Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter

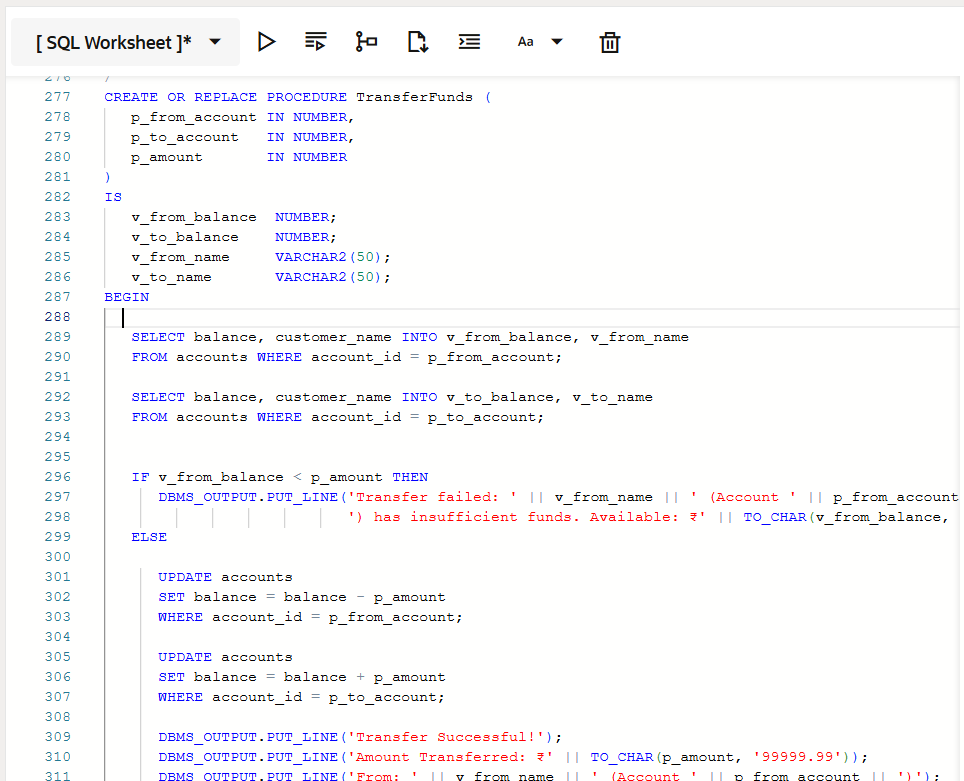


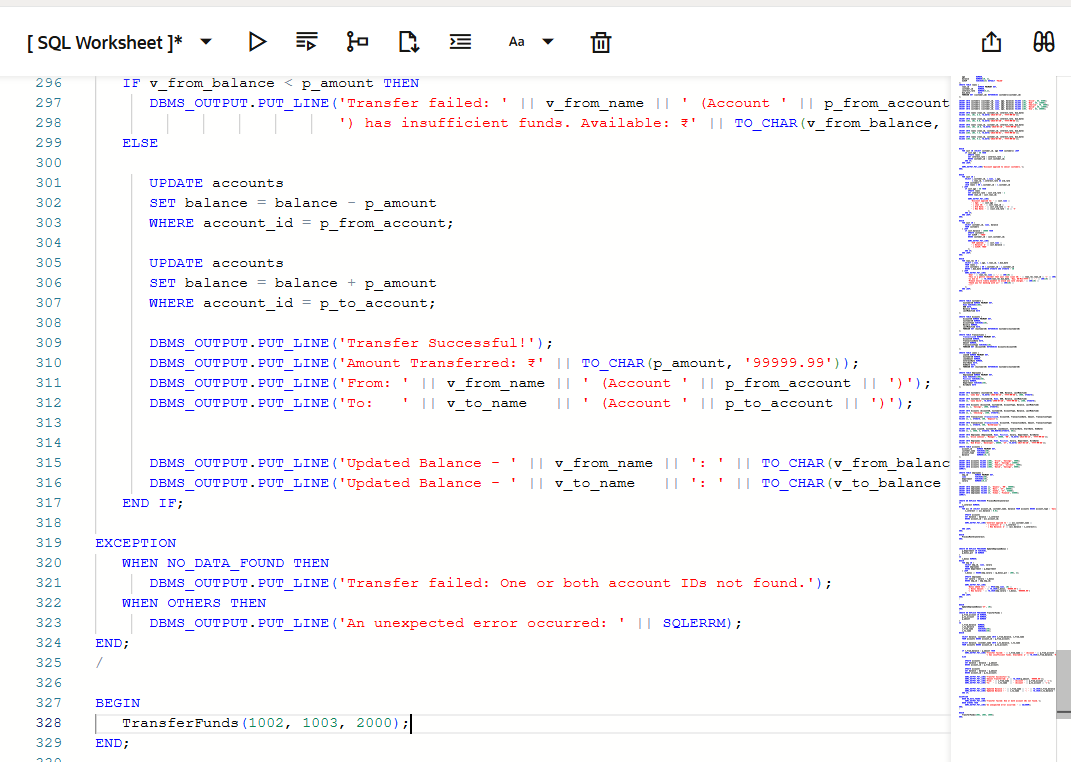
**Output**

****

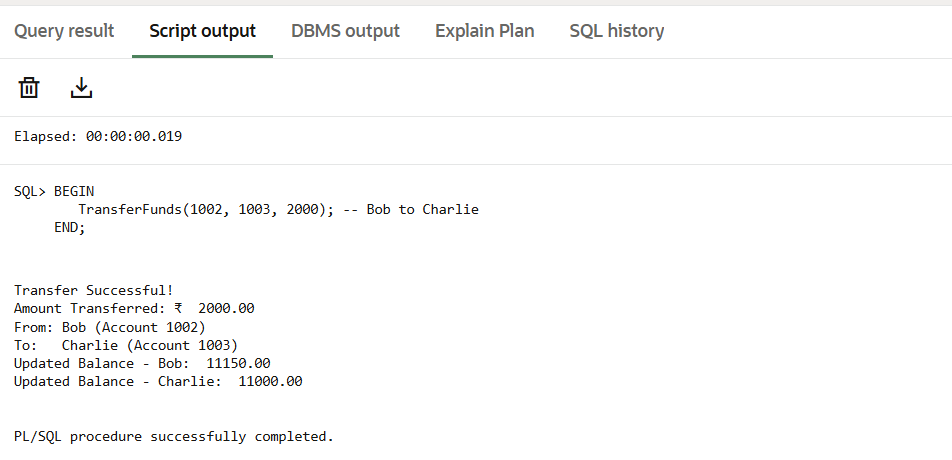
**iii)Scenario 3:**

Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.



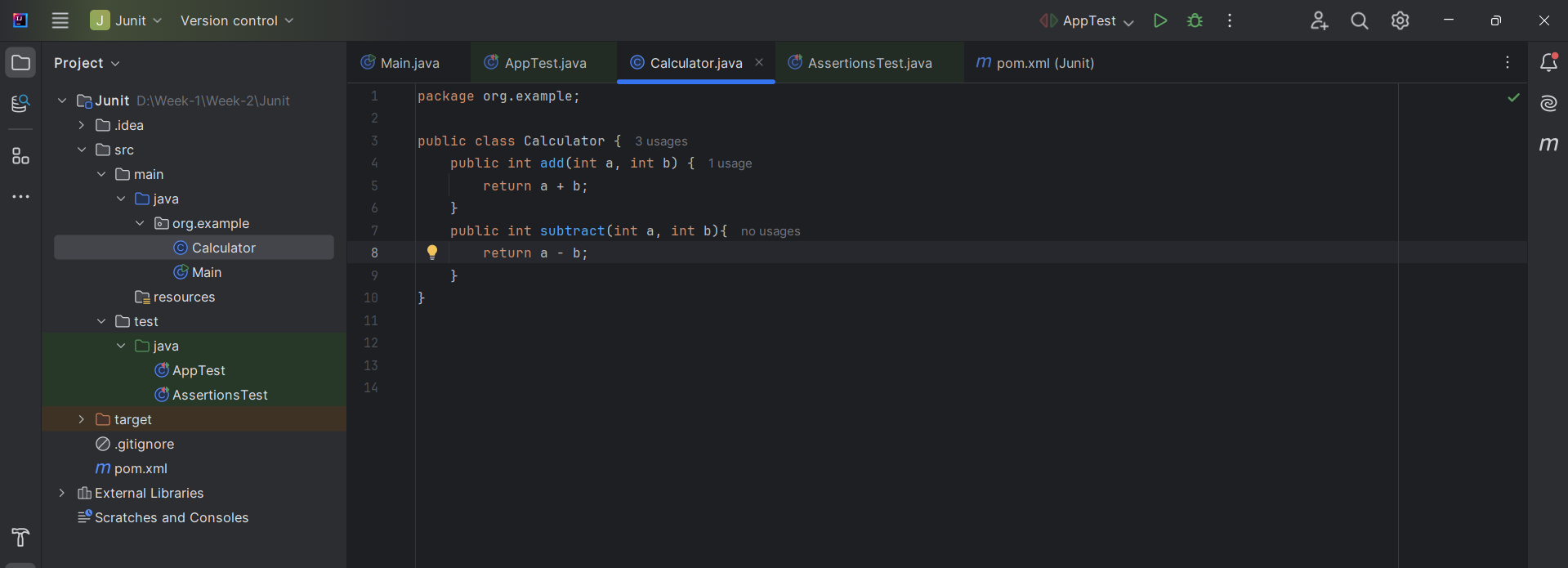


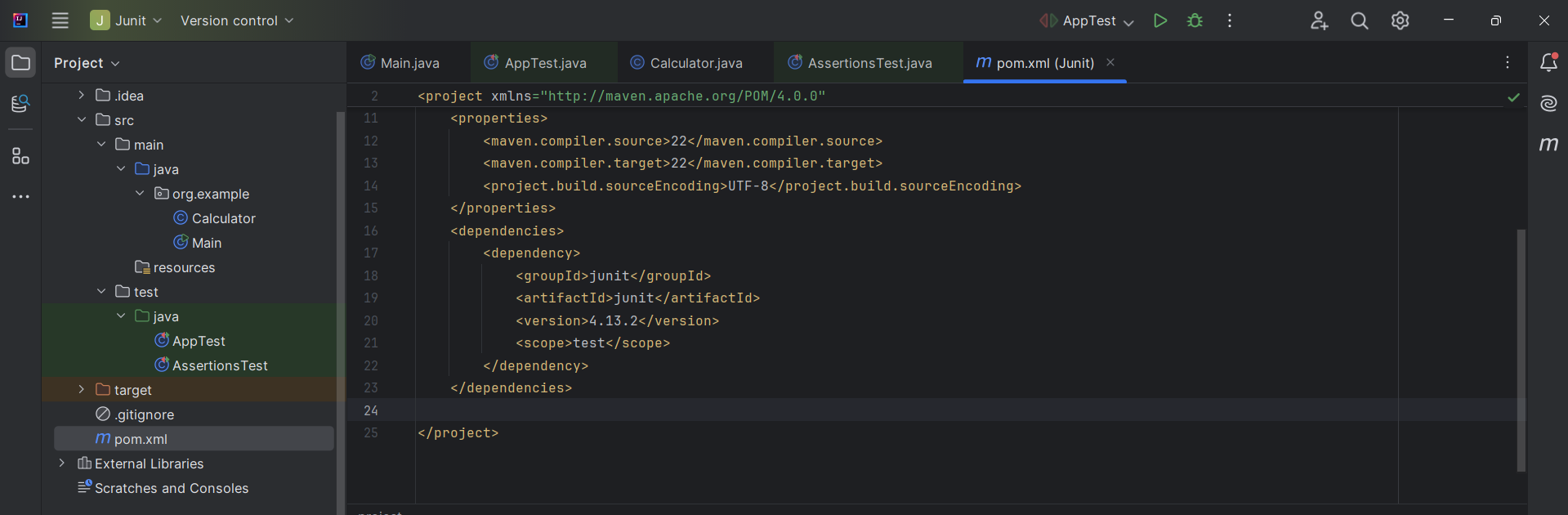
**Output**

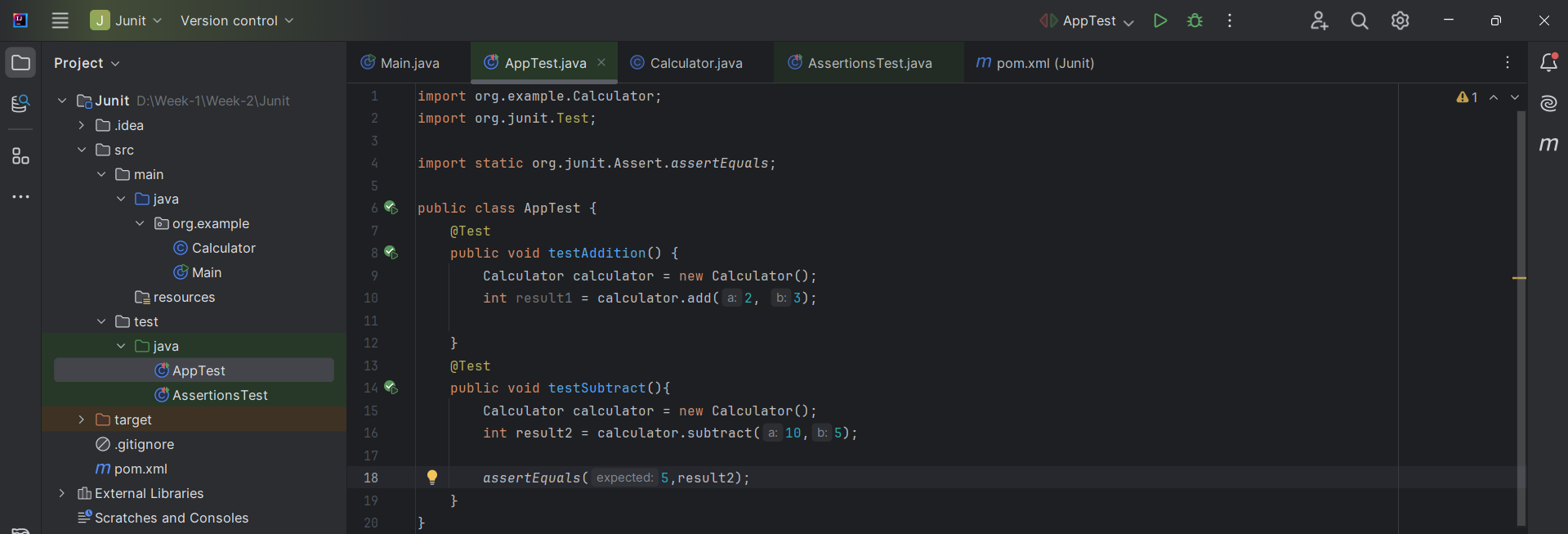
****

**Week – 2 Junit**

**Exercise 1: Setting Up Junit**

****

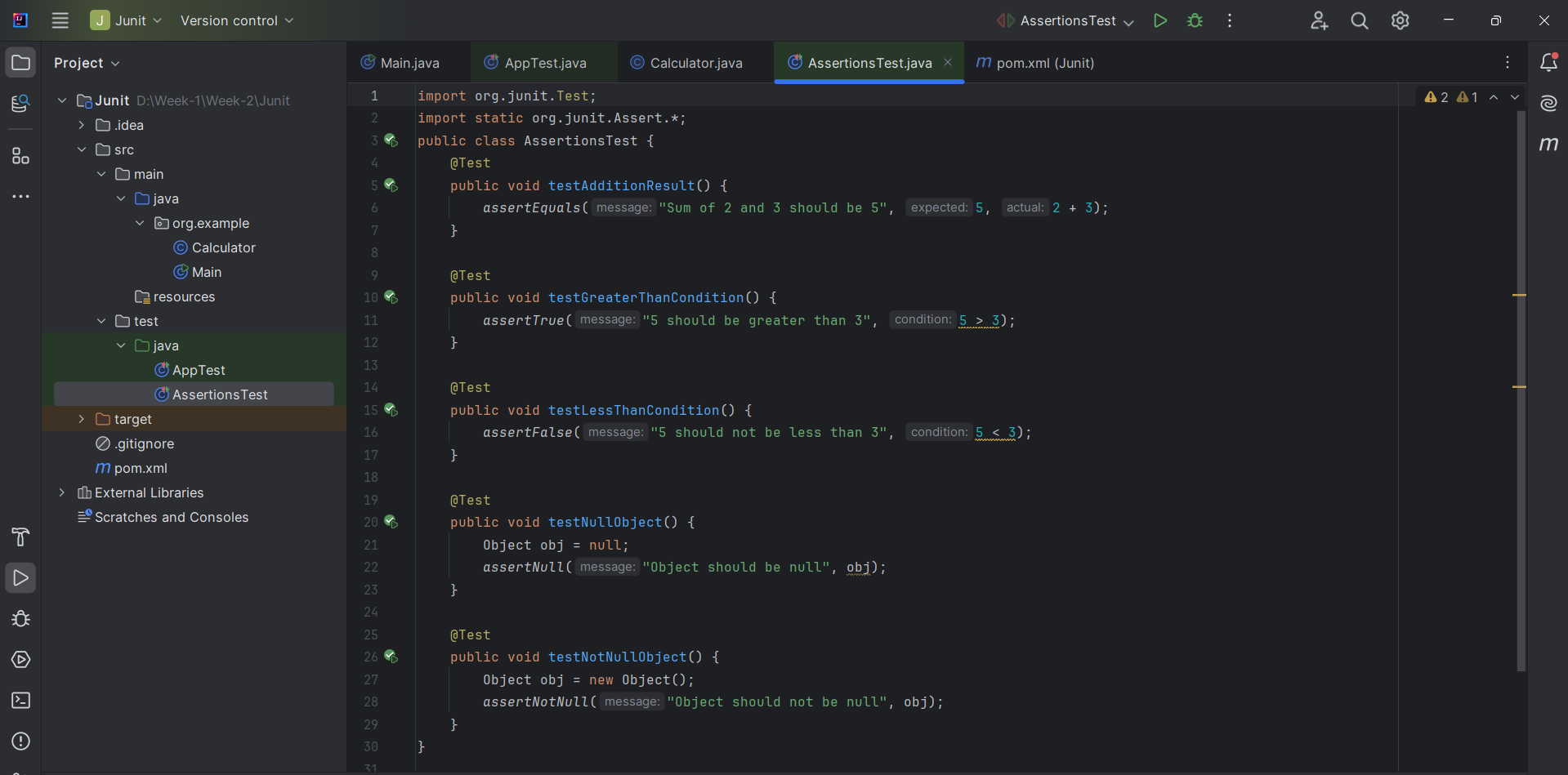
****

****

**Output:**

****

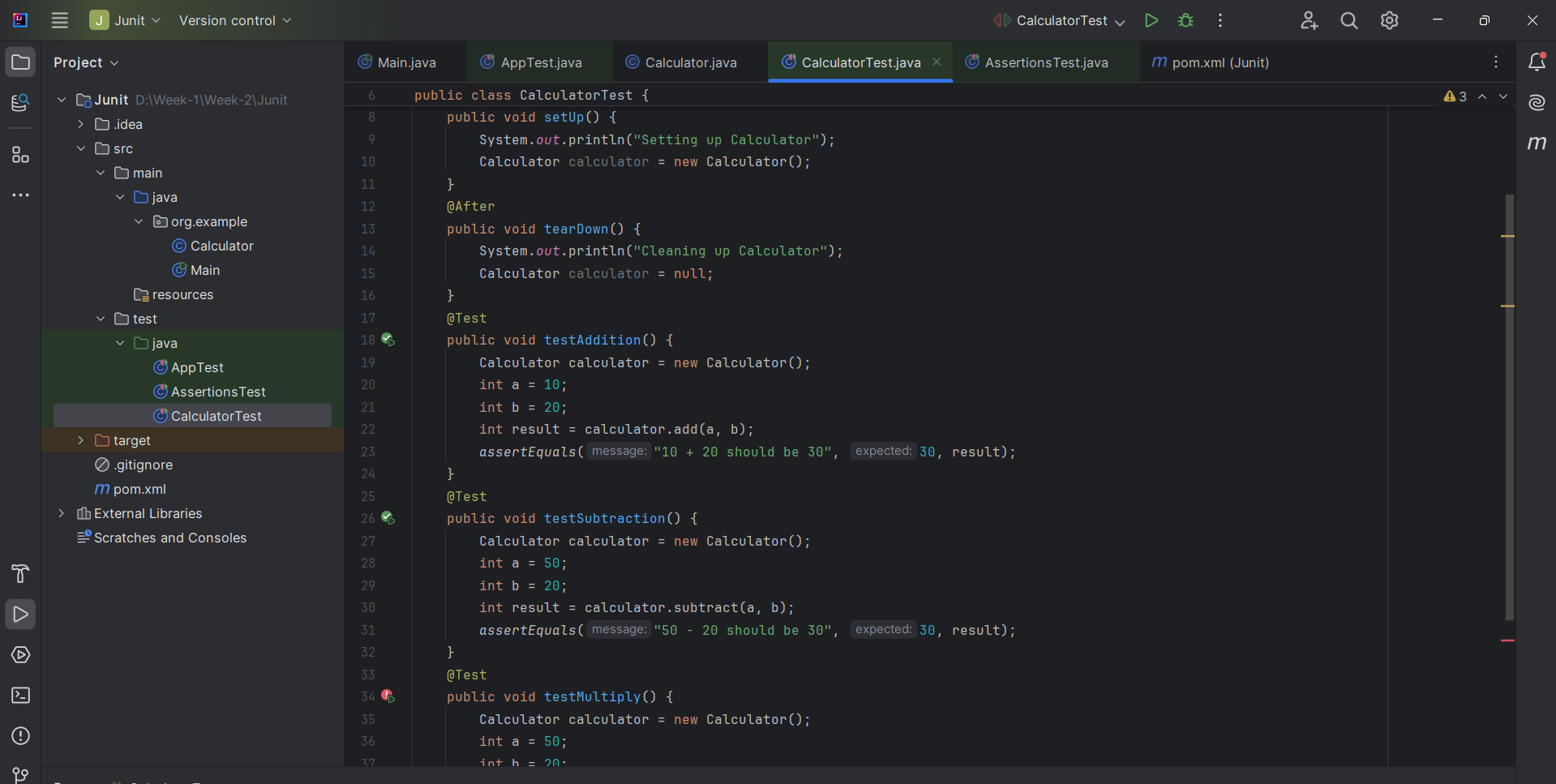
**Exercise 3: Assertions in Junit**

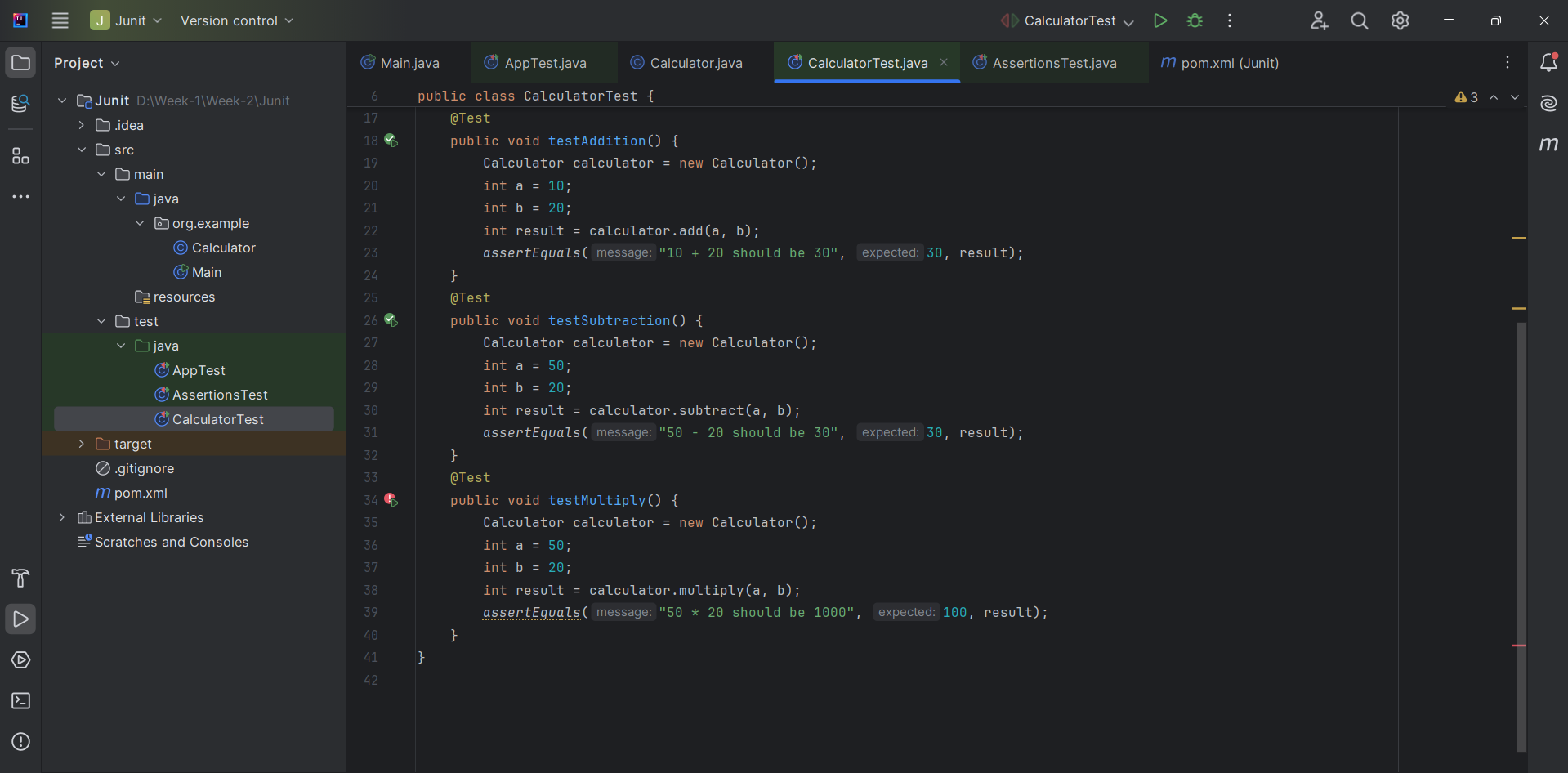
****

**Output:**

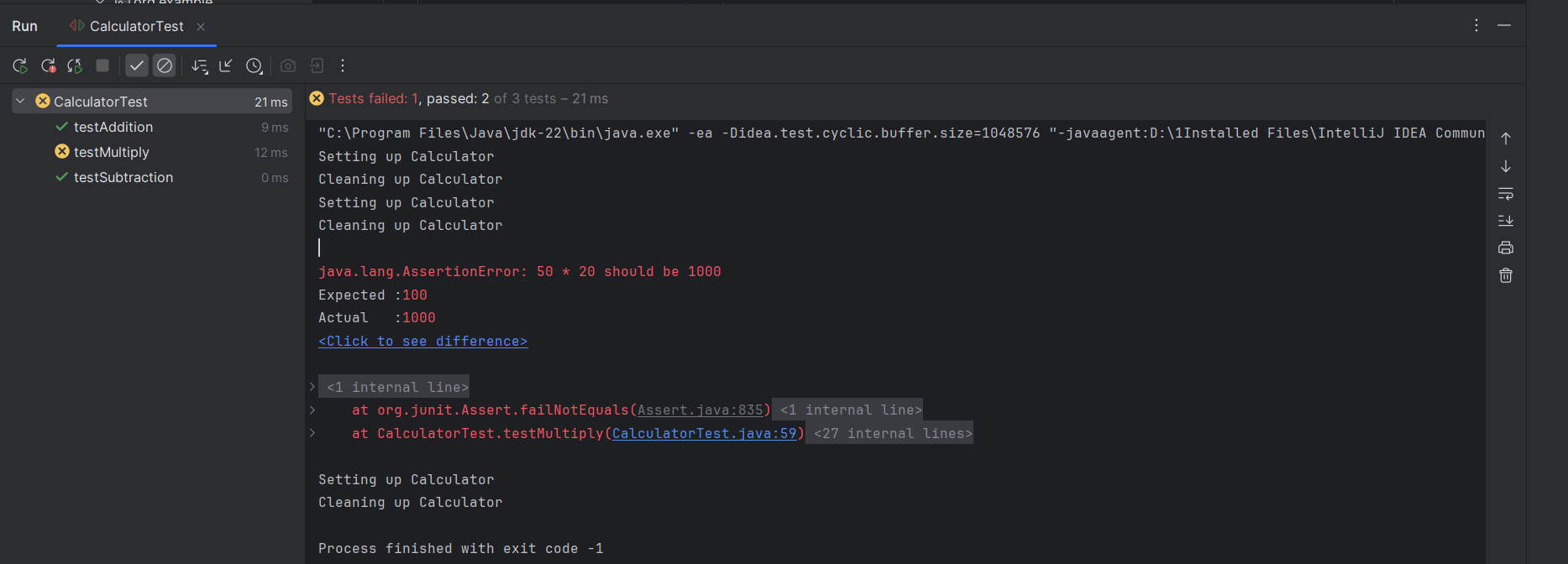
****

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

****

****

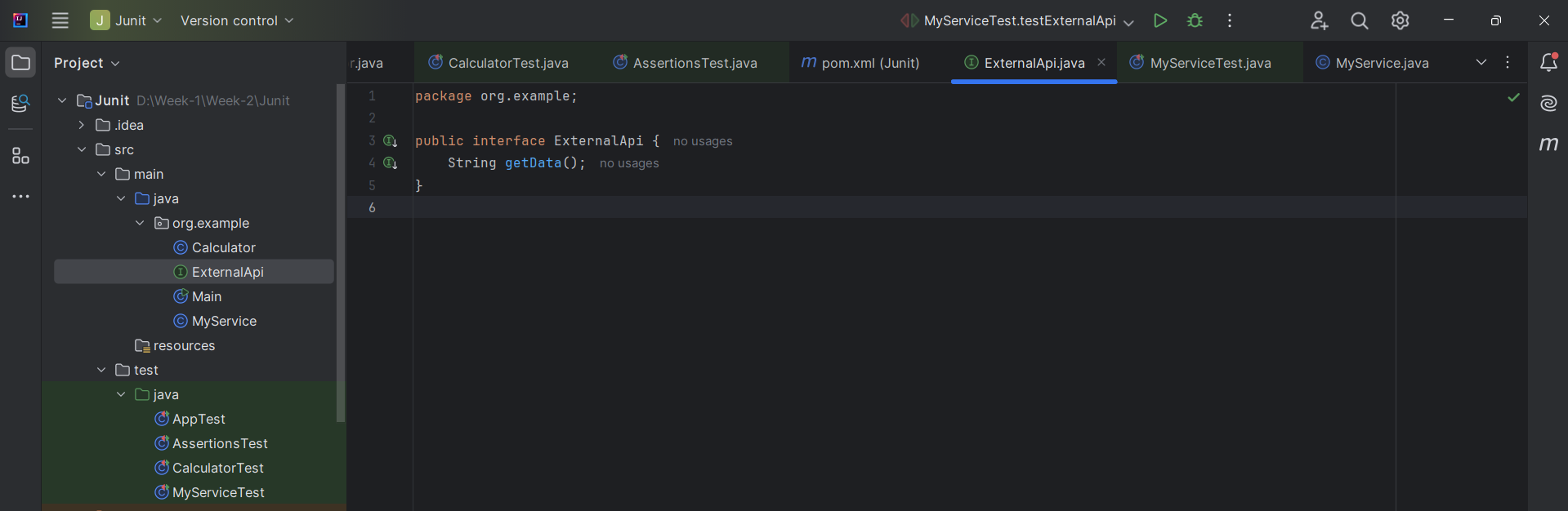
**Output:**

****

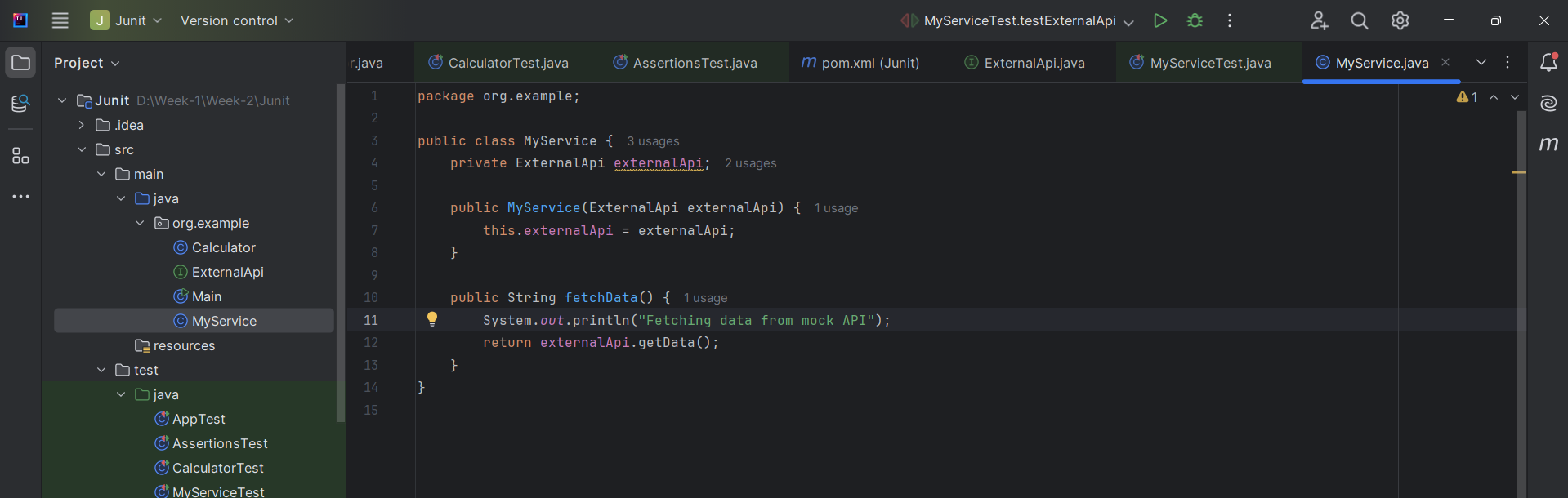
**WEEK 2 - MOKITO**

**Exercise 1: Mocking and Stubbing**

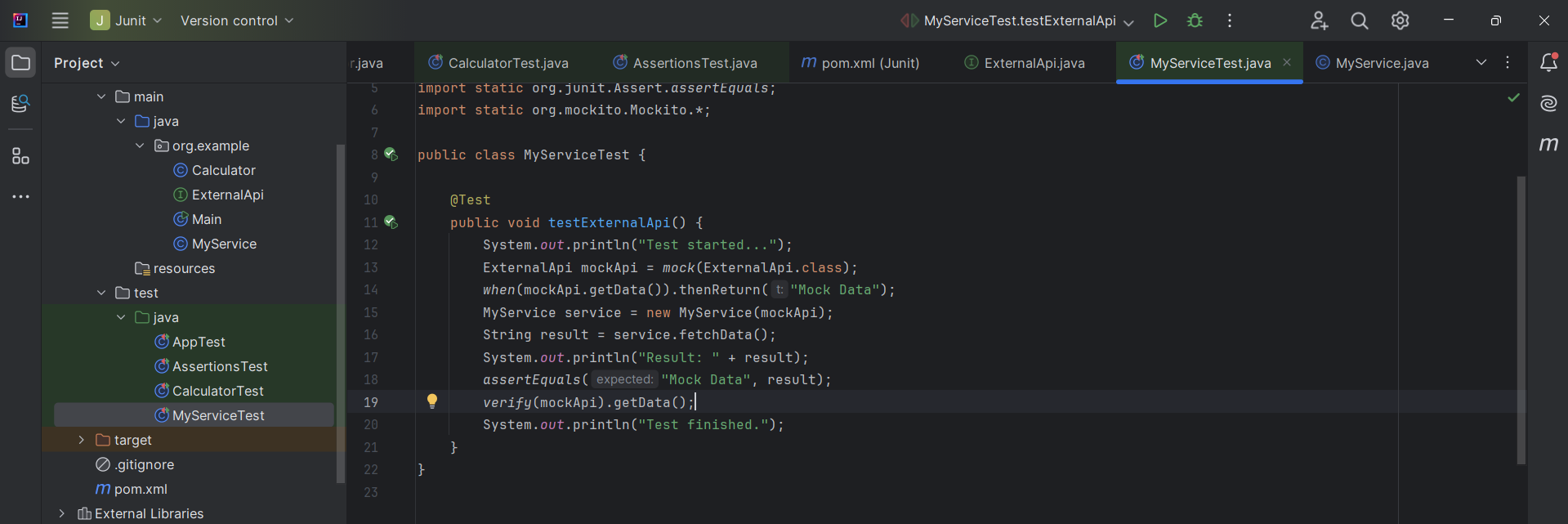
**1.ExternalApi Interface**

****

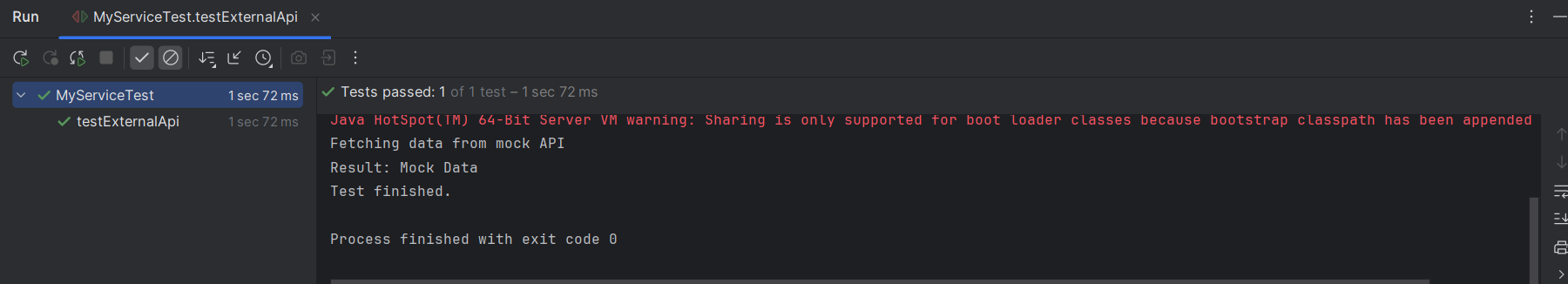
**2.MyService Class**

****

**3.MyServiceTest**

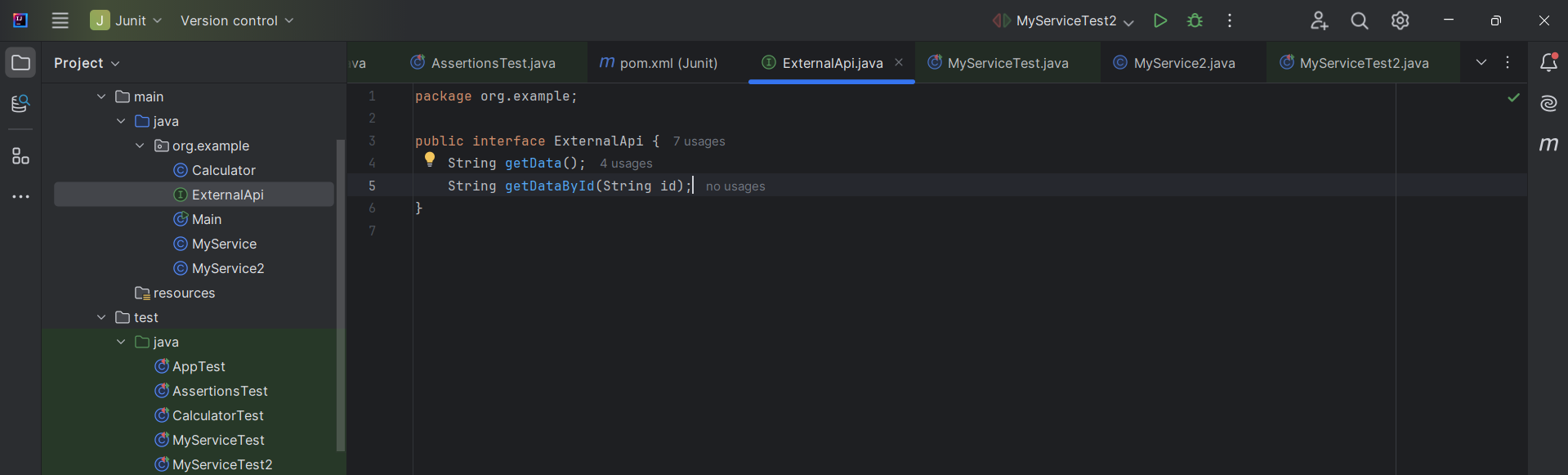
****

**Output**

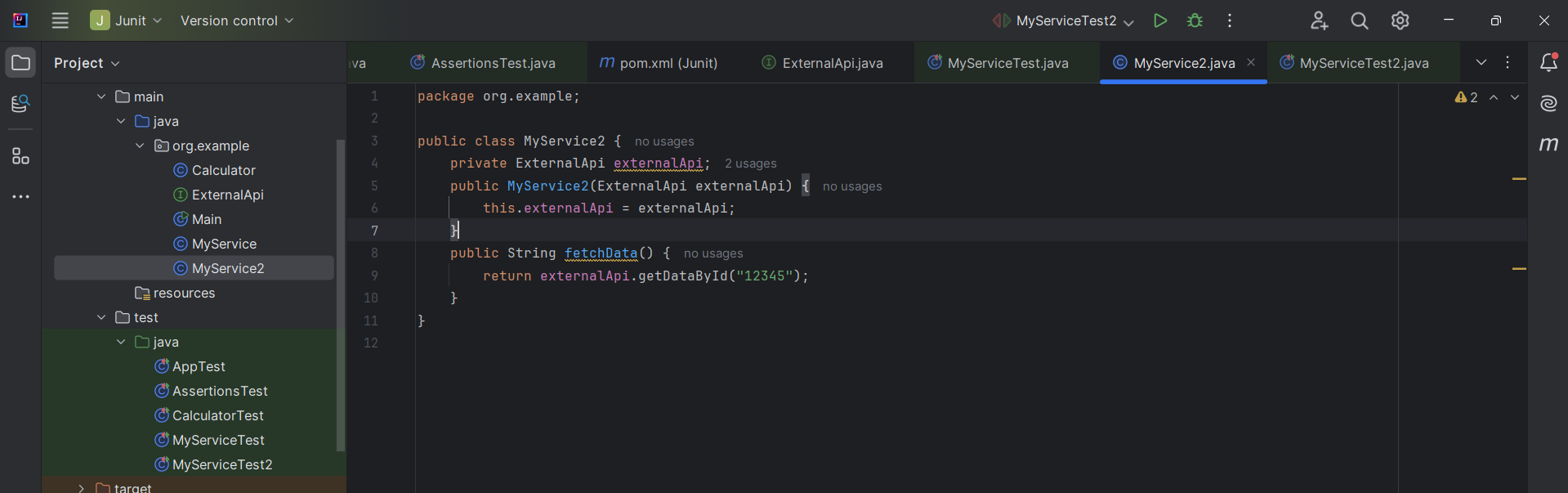
****

**Exercise 2: Verifying Interactions**

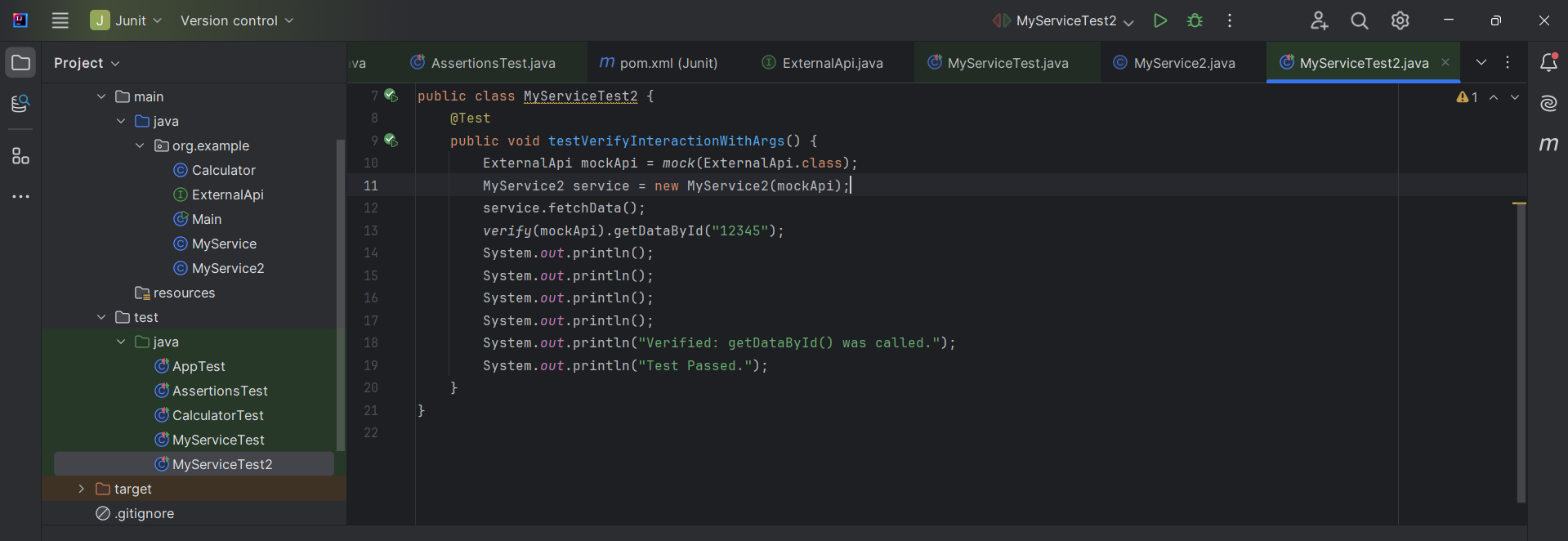
**1.ExternalApi**

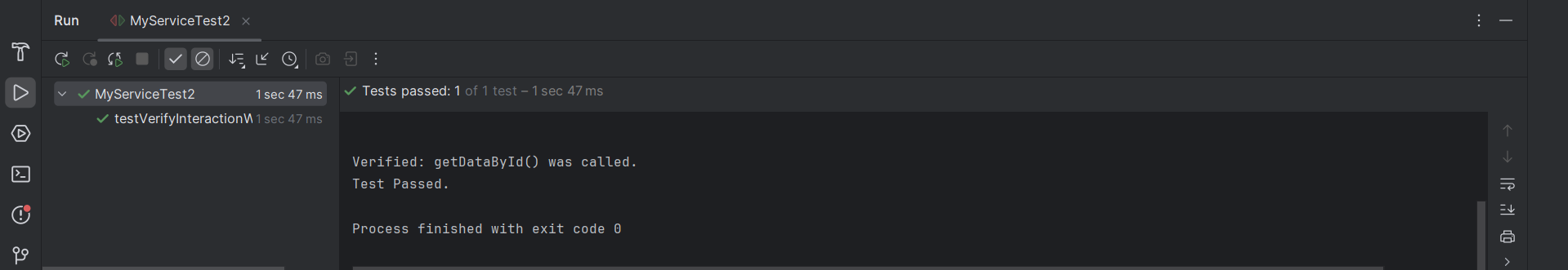
****

**2.MyService Class**

****

**3.MyServiceTest**

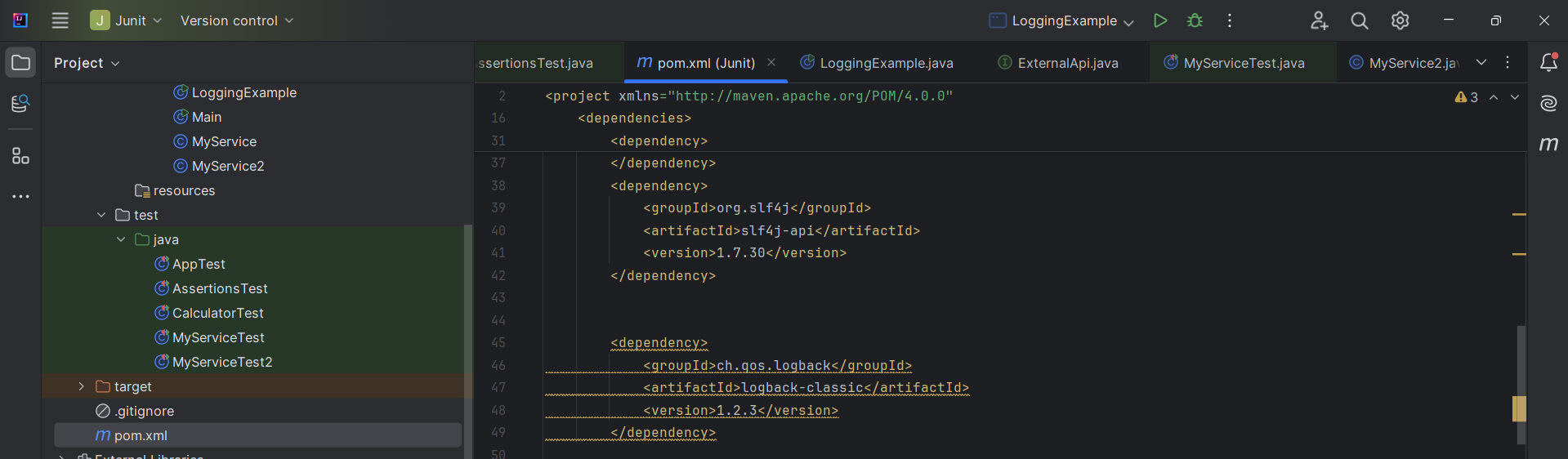
****

**Output  
**

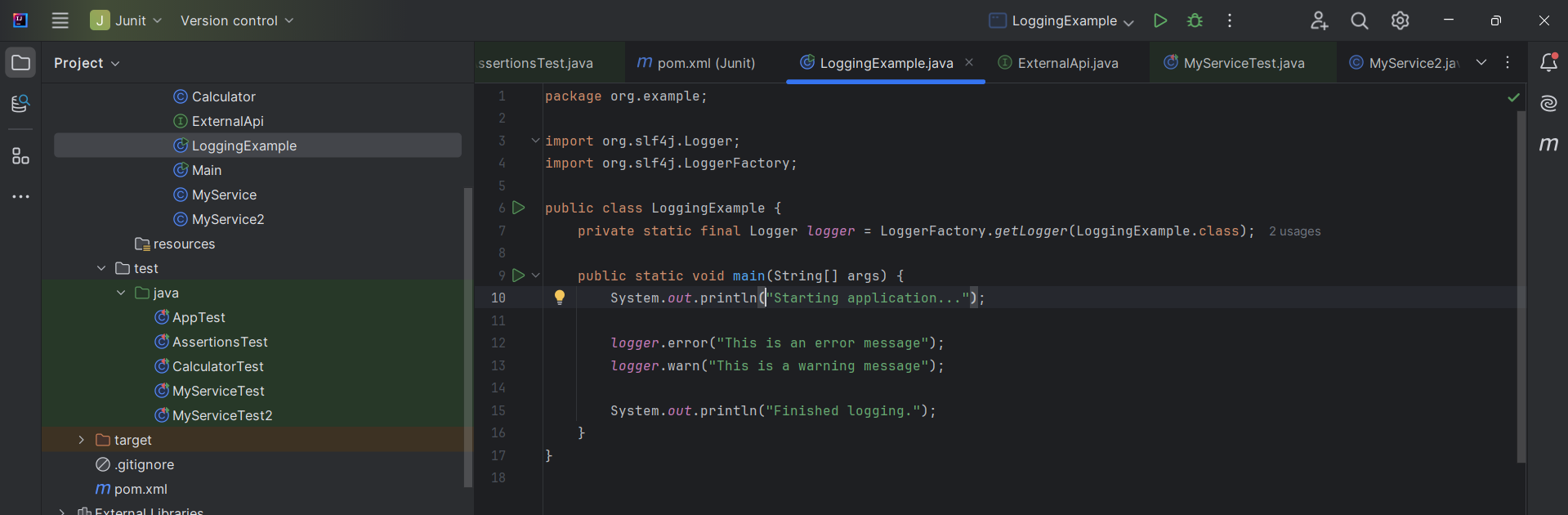
**Week – 2 SL4J**

**Exercise 1: Logging Error Messages and Warning Levels**

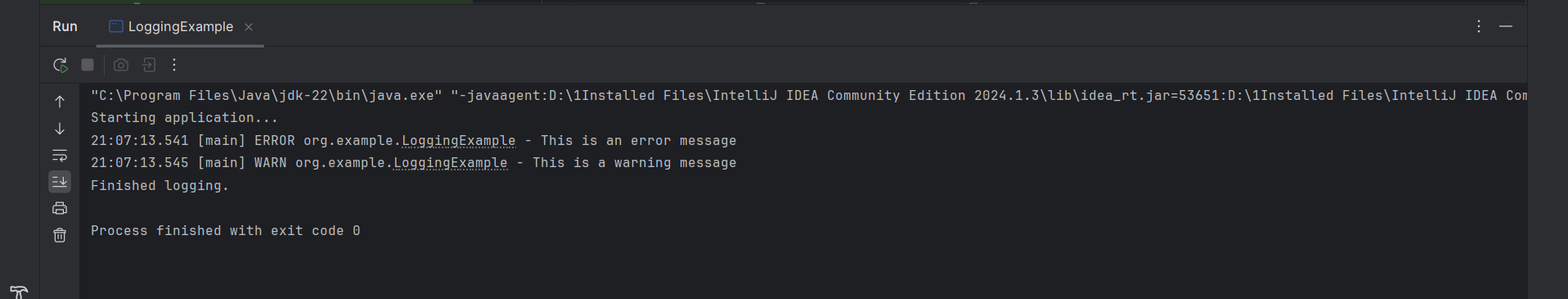
1. **Pom.xlm**

****

1. **LoggingExample**

****

**Output**

****