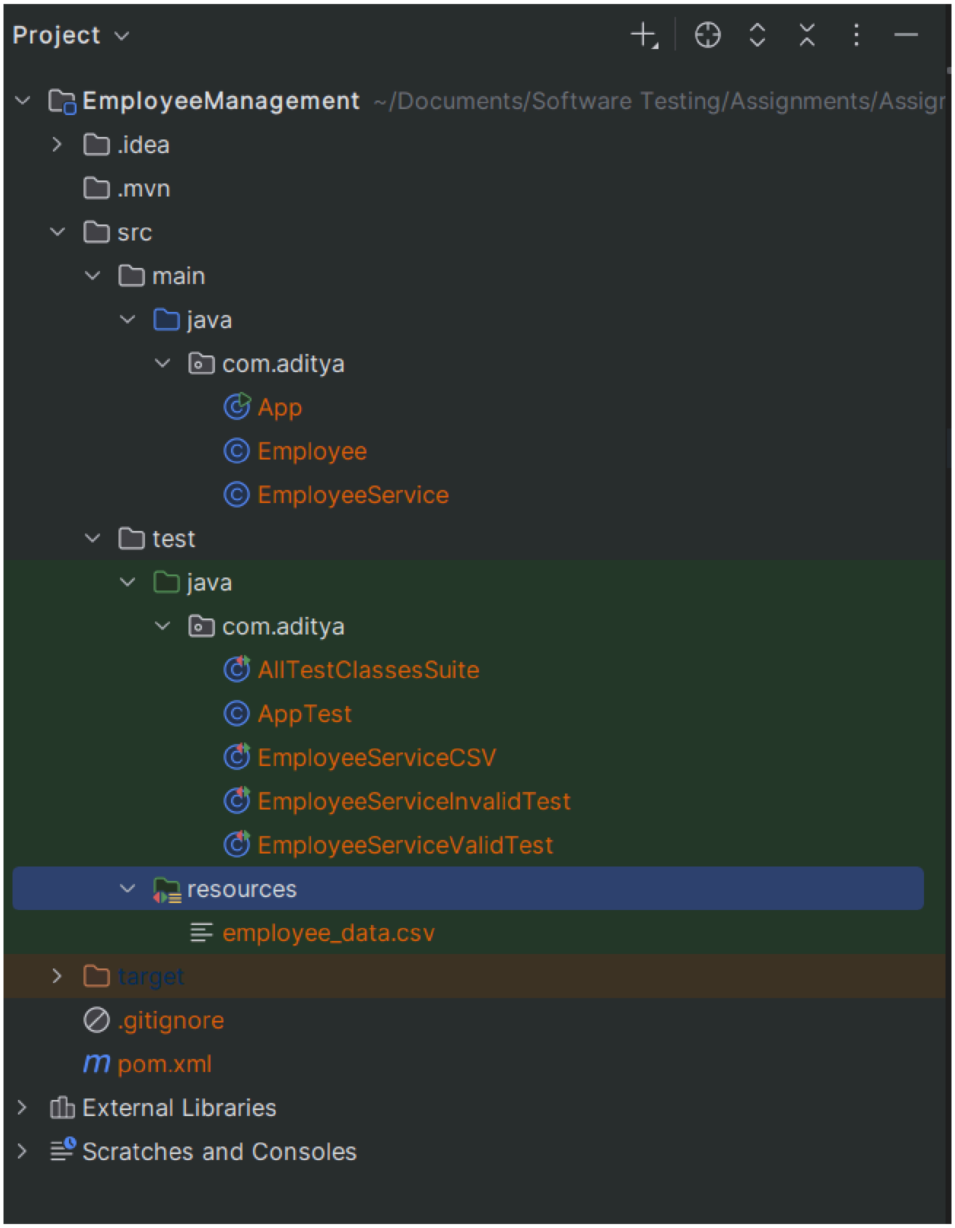
Assignment No. 1

Questions

Name: Aditya Pawar

USN: Aditya_72233061J

Folder Structure:



```
Project
EmployeeManagement ~/Documents/Software Testing/Assignments/Assigr
  .idea
  .mvn
  src
    main
      java
        com.aditya
          App
          Employee
          EmployeeService
    test
      java
        com.aditya
          AllTestClassesSuite
          AppTest
          EmployeeServiceCSV
          EmployeeServiceInvalidTest
          EmployeeServiceValidTest
      resources
        employee_data.csv
  target
  .gitignore
  pom.xml
External Libraries
Scratches and Consoles
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.aditya</groupId>
  <artifactId>EmployeeManagement</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>EmployeeManagement</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <dependencies>

    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-api -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-api</artifactId>
      <version>5.11.4</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-engine -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>5.11.4</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.junit.platform/junit-platform-suite -
    <dependency>
      <groupId>org.junit.platform</groupId>
      <artifactId>junit-platform-suite</artifactId>
      <version>1.9.1</version>
      <scope>test</scope>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.junit.jupiter/junit-jupiter-params -->
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-params</artifactId>
      <version>5.11.4</version>
      <scope>test</scope>
    </dependency>

  </dependencies>

  <build>
    <plugins>
      <plugin>
        <!-- https://mvnrepository.com/artifact/org.apache.maven.plugins/maven-s
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>3.5.2</version>
      </plugin>
    </plugins>
  </build>
</project>
```

Src —>

Employee.java

```
package com.aditya;

public class Employee {

    private int id;
    private String name;
    private double monthlySalary;

    public Employee(int id, String name, double monthlySalary) {
        this.id = id;
        this.name = name;
        this.monthlySalary = monthlySalary;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getMonthlySalary() {
        return monthlySalary;
    }

    public void setMonthlySalary(double monthlySalary) {
        this.monthlySalary = monthlySalary;
    }
}
```

EmployeeService.java

```
package com.aditya;

public class EmployeeService {

    public double calculateYearlySalary(Employee employee){
        return employee.getMonthlySalary() * 12;
    }

    public double calculateSalaryHike(Employee employee){
        return (employee.getMonthlySalary() < 10000) ? 2000 : 1000;
    }
}
```

Test —>

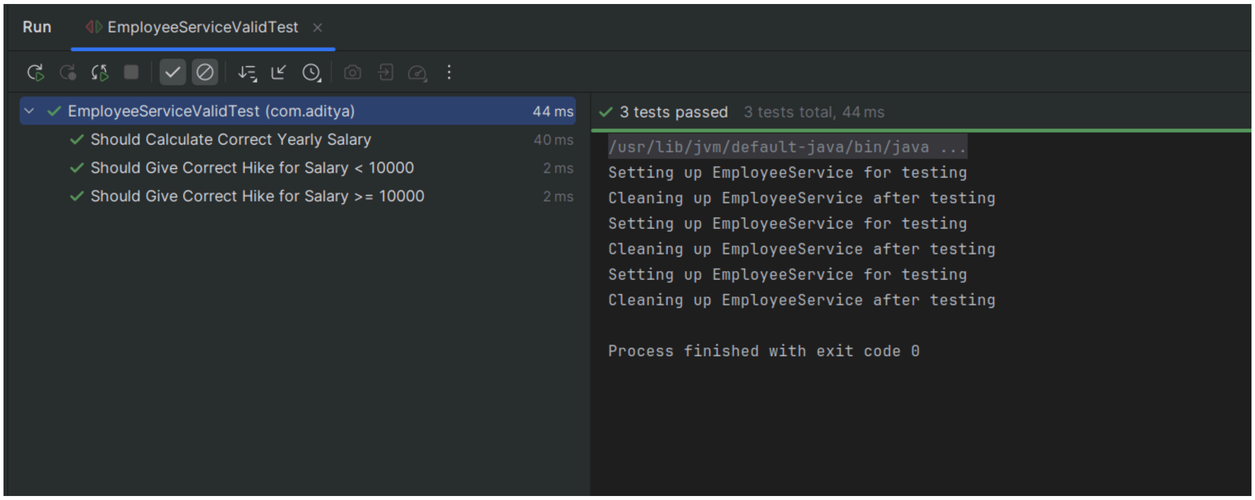
EmployeeServiceValidTest.java

```
package com.aditya;

import org.junit.jupiter.api.*;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class EmployeeServiceValidTest {

    private EmployeeService employeeService;

    @BeforeEach
    void setup()
    {
        employeeService = new EmployeeService();
        System.out.println("Setting up EmployeeService for testing");
    }
```

```
@AfterEach
void cleanup()
{
    employeeService = null;
    System.out.println("Cleaning up EmployeeService after testing");
}


@Test
@Order(1)
@DisplayName("Should Calculate Correct Yearly Salary")
void calculateYearlySalary_valid_test(){
    Employee employee = new Employee(1, "Aditya", 50000);
    double expectedYearlySalary = 600000;
    Assertions.assertEquals(expectedYearlySalary,
    employeeService.calculateYearlySalary(employee));
}

@Test
@DisplayName("Should Give Correct Hike for Salary < 10000")
@Order(2)
void testCalculateHike_LowSalary() {
    Employee emp = new Employee(2, "Alex", 8000);
    double expectedSalary = 2000;
    Assertions.assertEquals(expectedSalary,
    employeeService.calculateSalaryHike(emp));
}

@Test
@DisplayName("Should Give Correct Hike for Salary >= 10000")
@Order(3)
void testCalculateHike_HighSalary() {
    Employee emp = new Employee(3, "Priya", 15000);
    double expectedSalary = 1000;
    Assertions.assertEquals(expectedSalary,
    employeeService.calculateSalaryHike(emp));
    }

}
```

Output:

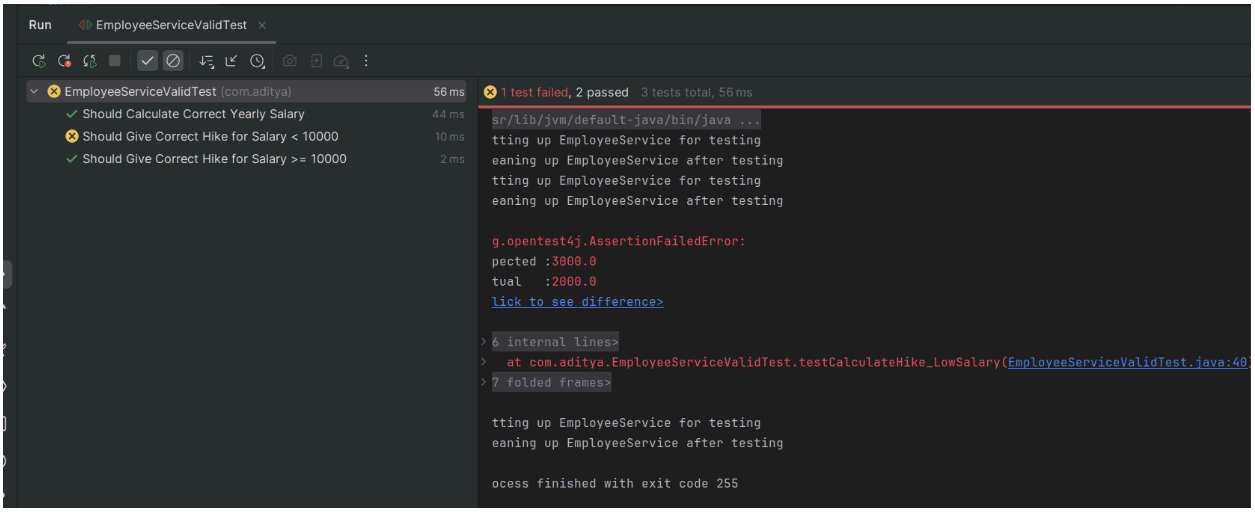
- Valid input



```
✓ EmployeeServiceValidTest (com.aditya)          44 ms
    ✓ Should Calculate Correct Yearly Salary      40 ms
    ✓ Should Give Correct Hike for Salary < 10000  2 ms
    ✓ Should Give Correct Hike for Salary >= 10000 2 ms
```

```
✓ 3 tests passed  3 tests total, 44 ms
/usr/lib/jvm/default-java/bin/java ...
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing

Process finished with exit code 0
```

- Invalid Input



```
☒ EmployeeServiceValidTest (com.aditya)          56 ms
    ✓ Should Calculate Correct Yearly Salary      44 ms
    ☒ Should Give Correct Hike for Salary < 10000 10 ms
    ✓ Should Give Correct Hike for Salary >= 10000 2 ms
```

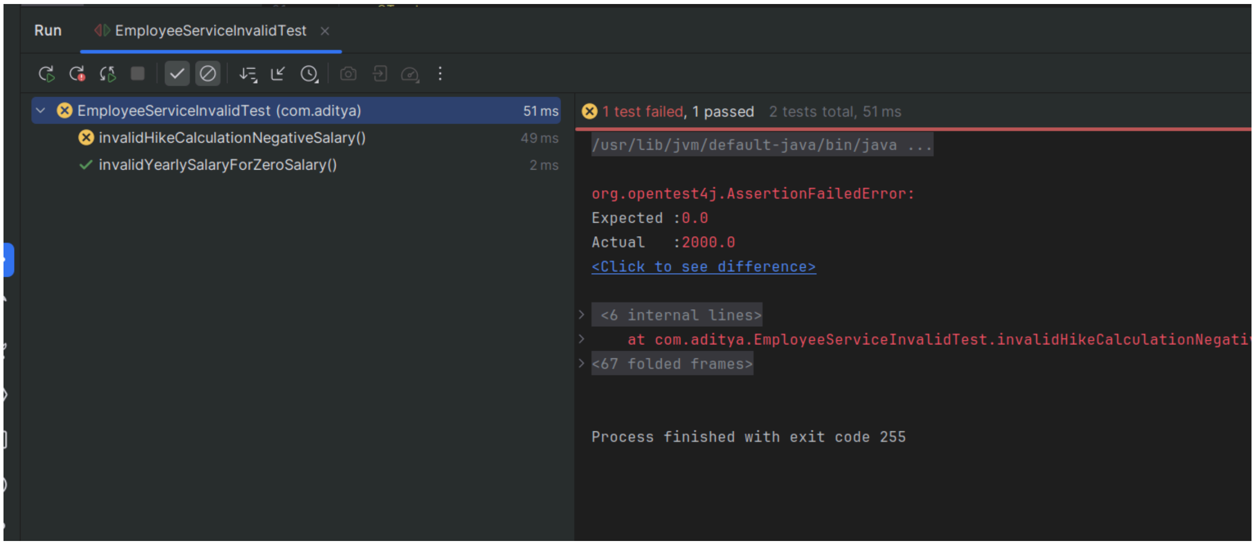
```
☒ 1 test failed, 2 passed  3 tests total, 56 ms
sr/lib/jvm/default-java/bin/java ...
tting up EmployeeService for testing
eaning up EmployeeService after testing
tting up EmployeeService for testing
eaning up EmployeeService after testing

g.opentest4j.AssertionFailedError:
pected :3000.0
tual   :2000.0
lick to see difference>

> 6 internal lines>
>   at com.aditya.EmployeeServiceValidTest.testCalculateHike_LowSalary(EmployeeServiceValidTest.java:40)
> 7 folded frames>

tting up EmployeeService for testing
eaning up EmployeeService after testing

ocess finished with exit code 255
```

EmployeeServiceInvalidTest.java

```
package com.aditya;

import org.junit.jupiter.api.*;

public class EmployeeServiceInvalidTest {

    private EmployeeService employeeService;
    private Employee employee;

    @BeforeEach
    void setUp() {
        employeeService = new EmployeeService();
    }

    @Test
    void invalidYearlySalaryForZeroSalary() {
        employee = new Employee(1, "Invalid Employee", 0.0);
        Assertions.assertEquals(0.0,
        employeeService.calculateYearlySalary(employee));
    }

    @Test
    void invalidHikeCalculationNegativeSalary() {
        employee = new Employee(2, "Invalid Negative Salary", -5000.0);
        Assertions.assertEquals(0.0,
        employeeService.calculateSalaryHike(employee));
    }
}
```

Output:

- for invalid input



```
EmployeeServiceInvalidTest (com.aditya)          51 ms
    invalidHikeCalculationNegativeSalary()       49 ms
    invalidYearlySalaryForZeroSalary()            2 ms
```

```
1 test failed, 1 passed   2 tests total, 51 ms
/usr/lib/jvm/default-java/bin/java ...

org.opentest4j.AssertionFailedError: 
Expected :0.0
Actual   :2000.0
<Click to see difference>

 <6 internal lines>
    at com.aditya.EmployeeServiceInvalidTest.invalidHikeCalculationNegati
<67 folded frames>


Process finished with exit code 255
```

EmployeeServiceCSVTest.java

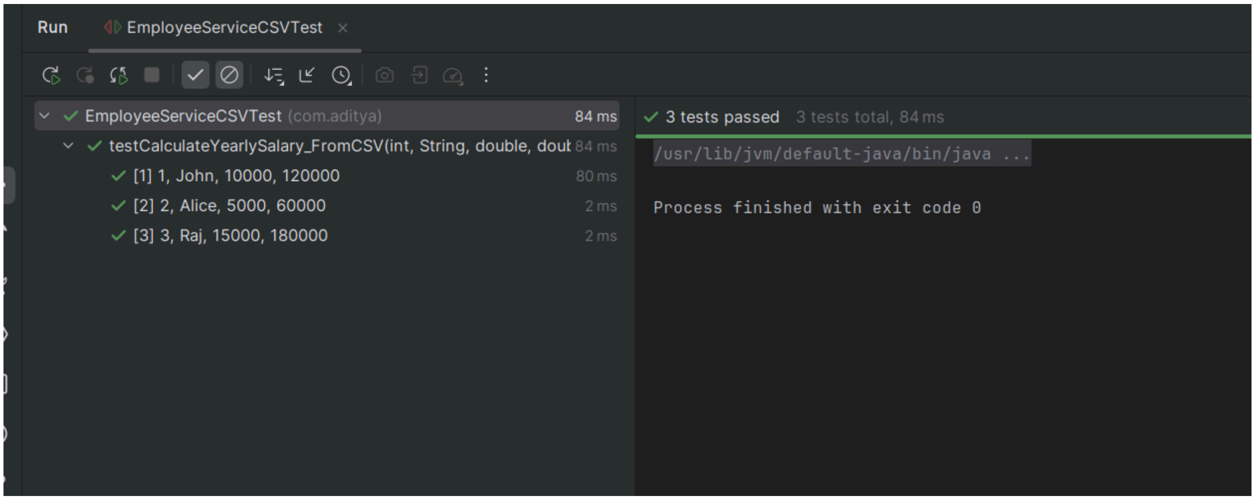
```
package com.aditya;

import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.CsvFileSource;

public class EmployeeServiceCSVTest {

    private final EmployeeService employeeService = new EmployeeService();

    @ParameterizedTest
    @CsvFileSource(resources = "/employee_data.csv", numLinesToSkip = 1)
    void testCalculateYearlySalary_FromCSV(
    int id,
    String name,
    double monthlySalary,
    double expectedYearly)
    {
        Employee emp = new Employee(id, name, monthlySalary);
        Assertions.assertEquals(expectedYearly,
            employeeService.calculateYearlySalary(emp));
    }
}
```

resources —>

employee_data.csv

```
id,name,monthlySalary,expectedYearly
1,John,10000,120000
2,Alice,5000,60000
3,Raj,15000,180000
```

Output:



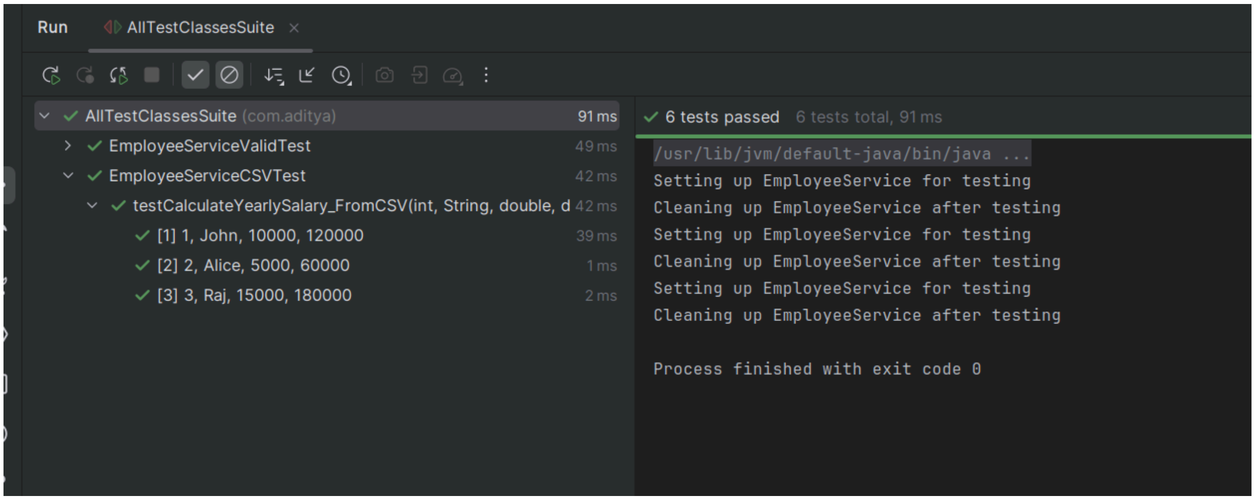
```
✓ EmployeeServiceCSVTest (com.aditya)                          84 ms
  ✓ testCalculateYearlySalary_FromCSV(int, String, double, douk 84 ms
    ✓ [1] 1, John, 10000, 120000                               80 ms
    ✓ [2] 2, Alice, 5000, 60000                                 2 ms
    ✓ [3] 3, Raj, 15000, 180000                                 2 ms
```

```
✓ 3 tests passed   3 tests total, 84 ms
/usr/lib/jvm/default-java/bin/java ...

Process finished with exit code 0
```

AllTestClassesSuite.java

```
package com.aditya;

import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;
import org.junit.platform.suite.api.SuiteDisplayName;

@Suite
@SelectClasses({
        EmployeeServiceValidTest.class,
        EmployeeServiceValidTest.class,
        EmployeeServiceCSVTest.class
})
@SuiteDisplayName("Employee Management - Full Test Suite")
public class AllTestClassesSuite {
}
```

Output:



```
✓ AllTestClassesSuite (com.aditya)                                      91 ms
  ✓ EmployeeServiceValidTest                                             49 ms
  ✓ EmployeeServiceCSVTest                                               42 ms
    ✓ testCalculateYearlySalary_FromCSV(int, String, double, d          42 ms
      ✓ [1] 1, John, 10000, 120000                                       39 ms
      ✓ [2] 2, Alice, 5000, 60000                                         1 ms
      ✓ [3] 3, Raj, 15000, 180000                                         2 ms
```

✓ 6 tests passed   6 tests total, 91 ms

```
/usr/lib/jvm/default-java/bin/java ...
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing
Setting up EmployeeService for testing
Cleaning up EmployeeService after testing

Process finished with exit code 0
```

Thank You !!!