Karthika Vijay - 6377005

Advanced JUnit Testing Exercises

Exercise 1: Parameterized Tests

Scenario:

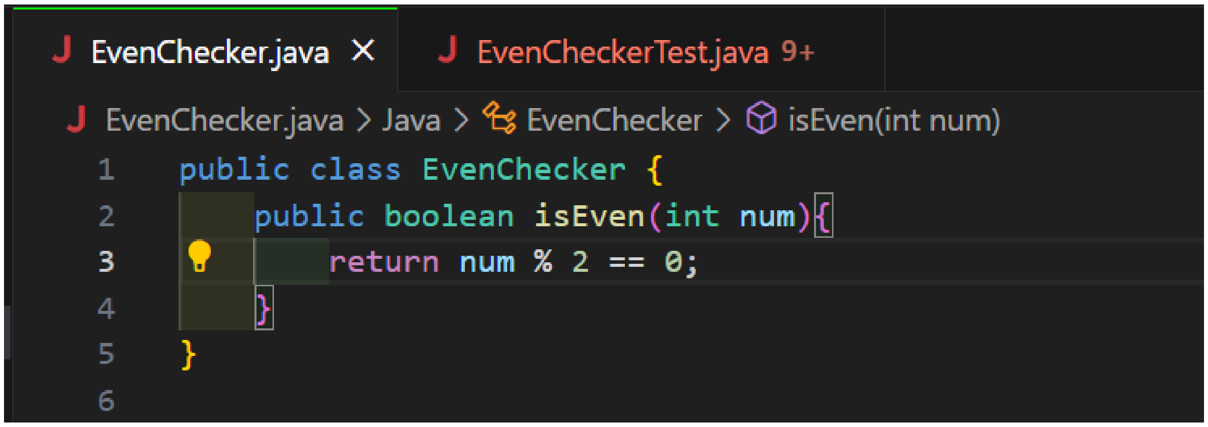
You want to test a method that checks if a number is even. Instead of writing multiple test cases, you will use parameterized tests to run the same test with different inputs.

Steps:

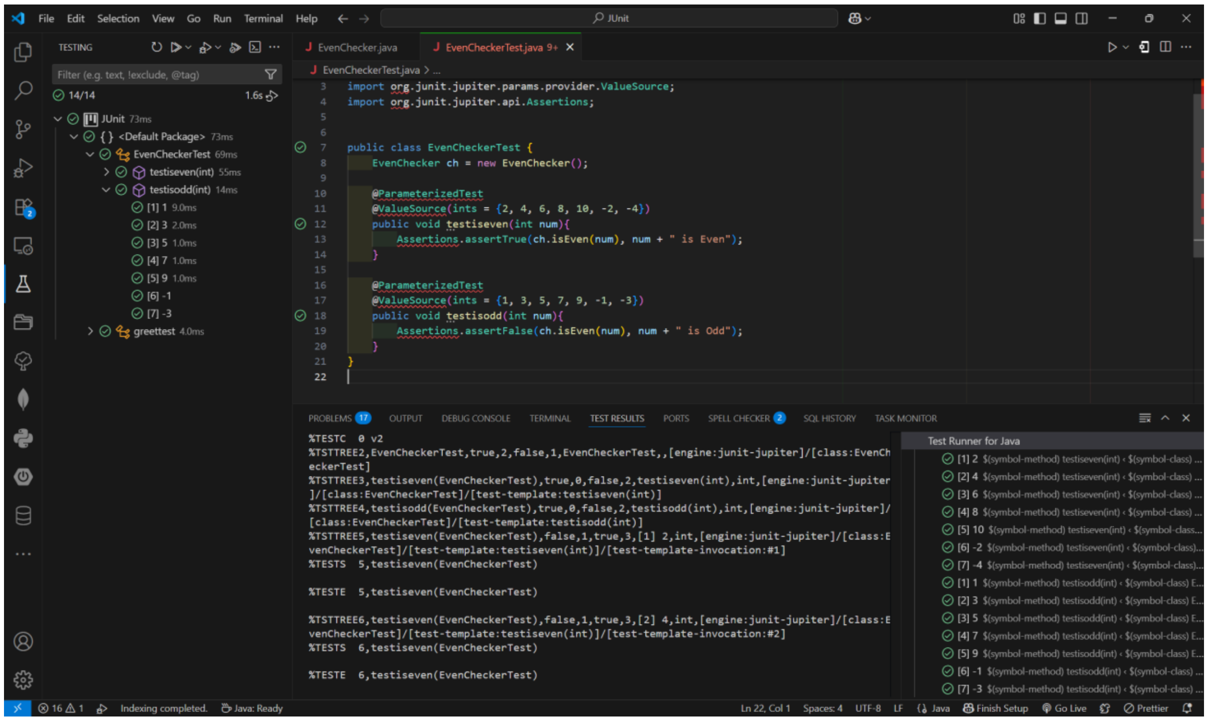
1. Create a new Java class `EvenChecker` with a method `isEven(int number)`.

2. Write a parameterized test class `EvenCheckerTest` that tests the `isEven` method with different inputs.

3. Use JUnit's `@ParameterizedTest` and `@ValueSource` annotations.



```java
public class EvenChecker {
    public boolean isEven(int num){
        return num % 2 == 0;
    }
}
```



```java
import org.junit.jupiter.params.provider.ValueSource;
import org.junit.jupiter.api.Assertions;


public class EvenCheckerTest {
    EvenChecker ch = new EvenChecker();

    @ParameterizedTest
    @ValueSource(ints = {2, 4, 6, 8, 10, -2, -4})
    public void testiseven(int num){
        Assertions.assertTrue(ch.isEven(num), num + " is Even");
    }

    @ParameterizedTest
    @ValueSource(ints = {1, 3, 5, 7, 9, -1, -3})
    public void testisodd(int num){
        Assertions.assertFalse(ch.isEven(num), num + " is Odd");
    }
}
```

```
%TESTC  0 v2
%TSTTREE2,EvenCheckerTest,true,2,false,1,EvenCheckerTest,,[engine:junit-jupiter]/[class:EvenCh
eckerTest]
%TSTTREE3,testiseven(EvenCheckerTest),true,0,false,2,testiseven(int),int,[engine:junit-jupiter
]/[class:EvenCheckerTest]/[test-template:testiseven(int)]
%TSTTREE4,testisodd(EvenCheckerTest),true,0,false,2,testisodd(int),int,[engine:junit-jupiter]/
[class:EvenCheckerTest]/[test-template:testisodd(int)]
%TSTTREE5,testiseven(EvenCheckerTest),false,1,true,3,[1] 2,int,[engine:junit-jupiter]/[class:E
venCheckerTest]/[test-template:testiseven(int)]/[test-template-invocation:#1]
%TESTS  5,testiseven(EvenCheckerTest)

%TESTE  5,testiseven(EvenCheckerTest)

%TSTTREE6,testiseven(EvenCheckerTest),false,1,true,3,[2] 4,int,[engine:junit-jupiter]/[class:E
venCheckerTest]/[test-template:testiseven(int)]/[test-template-invocation:#2]
%TESTS  6,testiseven(EvenCheckerTest)

%TESTE  6,testiseven(EvenCheckerTest)
```

```
Test Runner for Java
✓ [1] 2  $(symbol-method) testiseven(int) ‹ $(symbol-class) ...
✓ [2] 4  $(symbol-method) testiseven(int) ‹ $(symbol-class) ...
✓ [3] 6  $(symbol-method) testiseven(int) ‹ $(symbol-class) ...
✓ [4] 8  $(symbol-method) testiseven(int) ‹ $(symbol-class) ...
✓ [5] 10  $(symbol-method) testiseven(int) ‹ $(symbol-class...
✓ [6] -2  $(symbol-method) testiseven(int) ‹ $(symbol-class)...
✓ [7] -4  $(symbol-method) testiseven(int) ‹ $(symbol-class)...
✓ [1] 1  $(symbol-method) testisodd(int) ‹ $(symbol-class) E...
✓ [2] 3  $(symbol-method) testisodd(int) ‹ $(symbol-class) E...
✓ [3] 5  $(symbol-method) testisodd(int) ‹ $(symbol-class) E...
✓ [4] 7  $(symbol-method) testisodd(int) ‹ $(symbol-class) E...
✓ [5] 9  $(symbol-method) testisodd(int) ‹ $(symbol-class) E...
✓ [6] -1  $(symbol-method) testisodd(int) ‹ $(symbol-class) ...
✓ [7] -3  $(symbol-method) testisodd(int) ‹ $(symbol-class) ...
```

```
14/14
JUnit 73ms
  <Default Package> 73ms
    EvenCheckerTest 69ms
      testiseven(int) 55ms
      testisodd(int) 14ms
        [1] 1 9.0ms
        [2] 3 2.0ms
        [3] 5 1.0ms
        [4] 7 1.0ms
        [5] 9 1.0ms
        [6] -1
        [7] -3
    greettest 4.0ms
```

Exercise 2: Test Suites and Categories

Scenario:

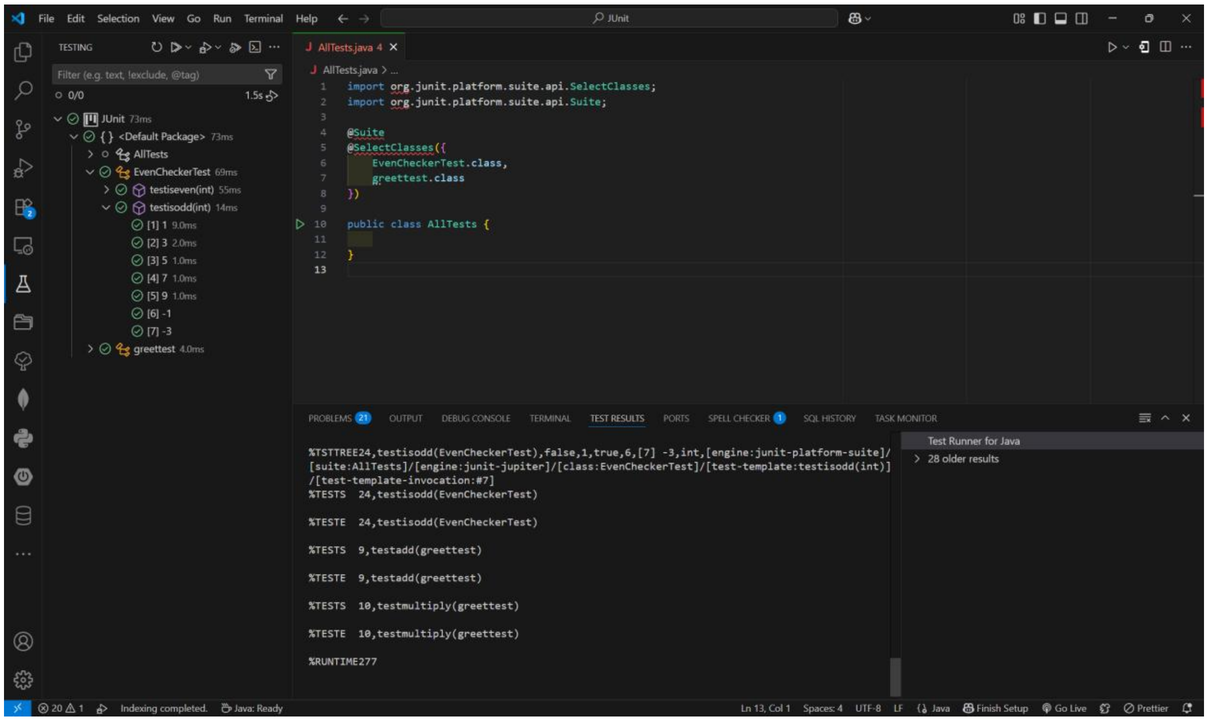
You want to group related tests into a test suite and categorize them.

Steps:

1. Create a new test suite class `AllTests`.

2. Add multiple test classes to the suite.

3. Use JUnit's `@Suite` and `@SelectClasses` annotations.



```java
import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;

@Suite
@SelectClasses({
    EvenCheckerTest.class,
    greettest.class
})

public class AllTests {

}
```

```
%TSTTREE24,testisodd(EvenCheckerTest),false,1,true,6,[7] -3,int,[engine:junit-platform-suite]/
[suite:AllTests]/[engine:junit-jupiter]/[class:EvenCheckerTest]/[test-template:testisodd(int)]
/[test-template-invocation:#7]
%TESTS  24,testisodd(EvenCheckerTest)

%TESTE  24,testisodd(EvenCheckerTest)

%TESTS  9,testadd(greettest)

%TESTE  9,testadd(greettest)

%TESTS  10,testmultiply(greettest)

%TESTE  10,testmultiply(greettest)

%RUNTIME277
```

Karthika Vijay - 6377005

Exercise 3: Test Execution Order

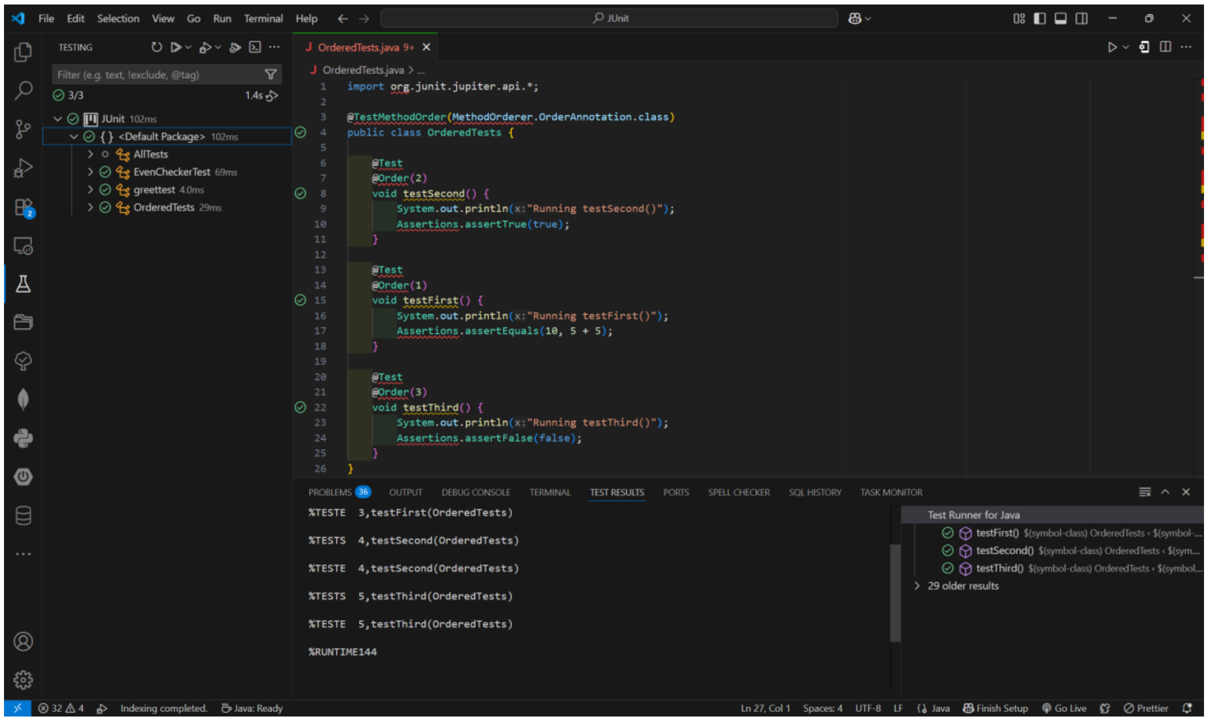
Scenario:

You want to control the order in which tests are executed.

Steps:

1. Create a test class `OrderedTests`.

2. Use JUnit's `@TestMethodOrder` and `@Order` annotations.


File Edit Selection View Go Run Terminal Help
JUnit
TESTING
3/3
JUnit 102ms
<Default Package> 102ms
AllTests
EvenCheckerTest 69ms
greettest 4.0ms
OrderedTests 29ms
OrderedTests.java 9+
PROBLEMS 36 OUTPUT DEBUG CONSOLE TERMINAL TEST RESULTS PORTS SPELL CHECKER SQL HISTORY TASK MONITOR
Indexing completed. Java: Ready
Ln 27, Col 1 Spaces: 4 UTF-8 LF Java Finish Setup Go Live Prettier

```java
import org.junit.jupiter.api.*;

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

    @Test
    @Order(2)
    void testSecond() {
        System.out.println(x:"Running testSecond()");
        Assertions.assertTrue(true);
    }

    @Test
    @Order(1)
    void testFirst() {
        System.out.println(x:"Running testFirst()");
        Assertions.assertEquals(10, 5 + 5);
    }

    @Test
    @Order(3)
    void testThird() {
        System.out.println(x:"Running testThird()");
        Assertions.assertFalse(false);
    }
}
```

```
%TESTE  3,testFirst(OrderedTests)

%TESTS  4,testSecond(OrderedTests)

%TESTE  4,testSecond(OrderedTests)

%TESTS  5,testThird(OrderedTests)

%TESTE  5,testThird(OrderedTests)

%RUNTIME144
```

```
Test Runner for Java
  testFirst() $(symbol-class) OrderedTests ‹ $(symbol-...
  testSecond() $(symbol-class) OrderedTests ‹ $(sym...
  testThird() $(symbol-class) OrderedTests ‹ $(symbol...
> 29 older results
```

Exercise 4: Exception Testing

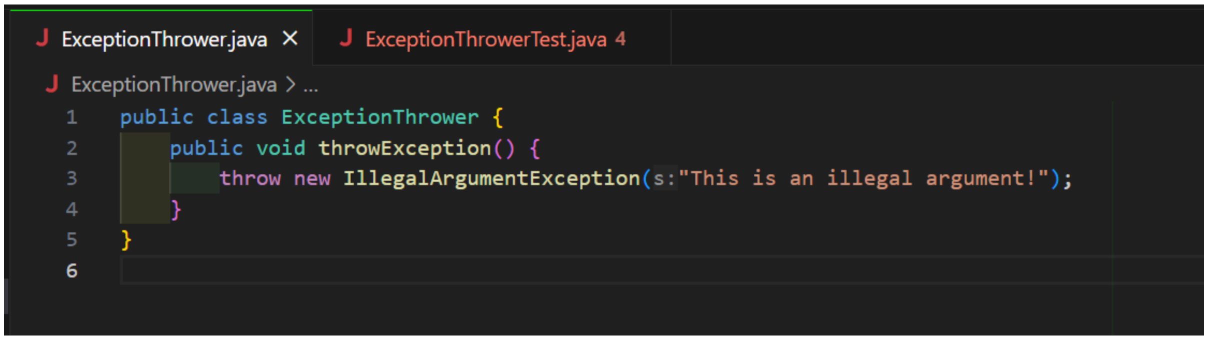
Scenario:

You want to test that a method throws the expected exception.

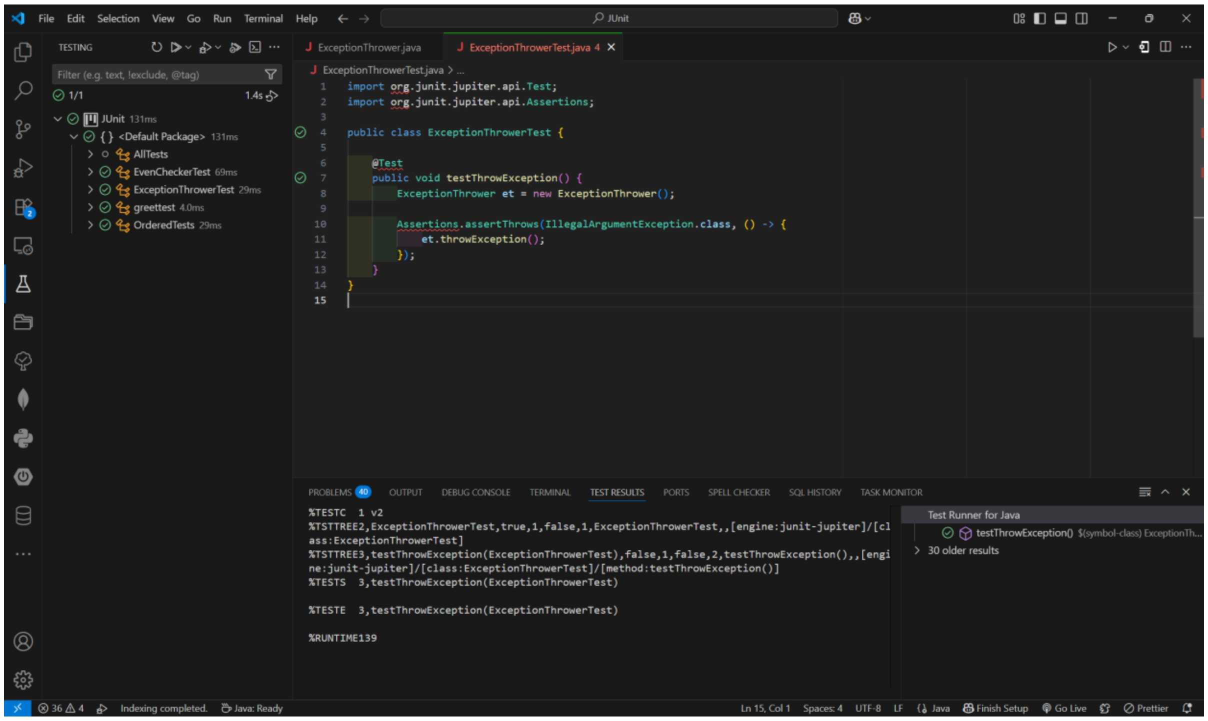
Steps:

1. Create a class `ExceptionThrower` with a method `throwException`.

2. Write a test class `ExceptionThrowerTest` that tests the method for the expected exception.



```java
public class ExceptionThrower {
    public void throwException() {
        throw new IllegalArgumentException(s:"This is an illegal argument!");
    }
}
```



```java
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.Assertions;

public class ExceptionThrowerTest {

    @Test
    public void testThrowException() {
        ExceptionThrower et = new ExceptionThrower();

        Assertions.assertThrows(IllegalArgumentException.class, () -> {
            et.throwException();
        });
    }
}
```

```
%TESTC  1 v2
%TSTTREE2,ExceptionThrowerTest,true,1,false,1,ExceptionThrowerTest,,[engine:junit-jupiter]/[cl
ass:ExceptionThrowerTest]
%TSTTREE3,testThrowException(ExceptionThrowerTest),false,1,false,2,testThrowException(),,[engi
ne:junit-jupiter]/[class:ExceptionThrowerTest]/[method:testThrowException()]
%TESTS  3,testThrowException(ExceptionThrowerTest)

%TESTE  3,testThrowException(ExceptionThrowerTest)

%RUNTIME139
```

Karthika Vijay - 6377005

Exercise 5: Timeout and Performance Testing

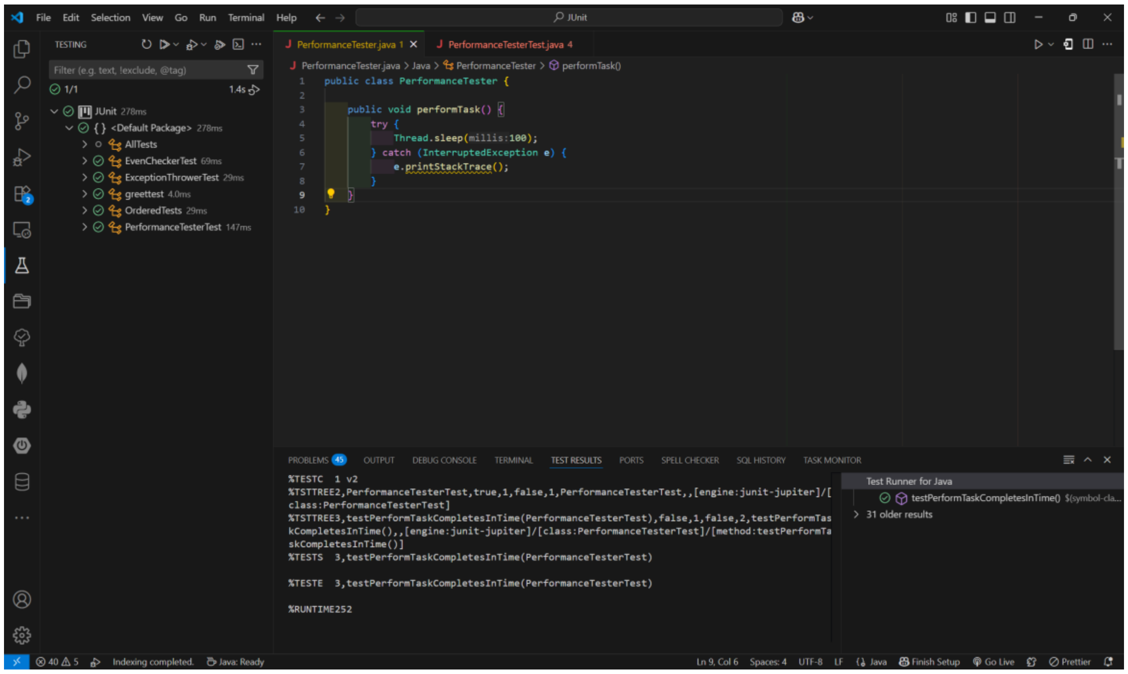
Scenario:

You want to ensure that a method completes within a specified time limit.

Steps:

1. Create a class `PerformanceTester` with a method `performTask`.

2. Write a test class `PerformanceTesterTest` that tests the method for timeout.



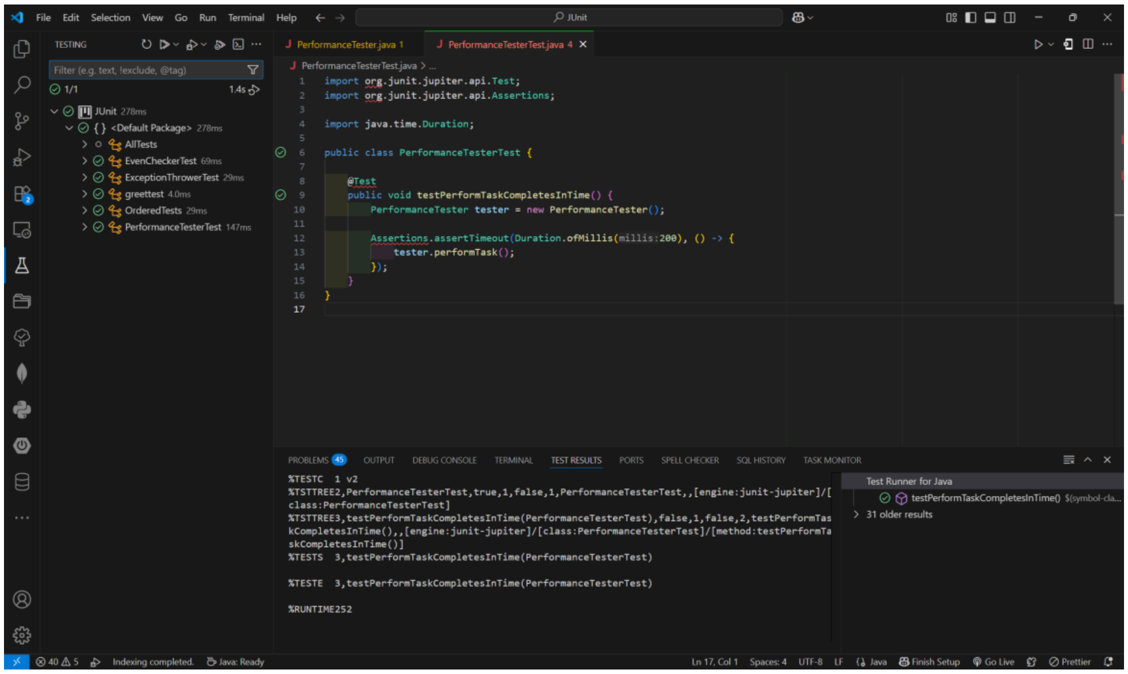
```java
public class PerformanceTester {

    public void performTask() {
        try {
            Thread.sleep(millis:100);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

```
%TESTC  1 v2
%TSTTREE2,PerformanceTesterTest,true,1,false,1,PerformanceTesterTest,,[engine:junit-jupiter]/[
class:PerformanceTesterTest]
%TSTTREE3,testPerformTaskCompletesInTime(PerformanceTesterTest),false,1,false,2,testPerformTas
kCompletesInTime(),,[engine:junit-jupiter]/[class:PerformanceTesterTest]/[method:testPerformTa
skCompletesInTime()]
%TESTS  3,testPerformTaskCompletesInTime(PerformanceTesterTest)

%TESTE  3,testPerformTaskCompletesInTime(PerformanceTesterTest)

%RUNTIME252
```



```java
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.Assertions;

import java.time.Duration;

public class PerformanceTesterTest {

    @Test
    public void testPerformTaskCompletesInTime() {
        PerformanceTester tester = new PerformanceTester();

        Assertions.assertTimeout(Duration.ofMillis(millis:200), () -> {
            tester.performTask();
        });
    }
}
```

```
%TESTC  1 v2
%TSTTREE2,PerformanceTesterTest,true,1,false,1,PerformanceTesterTest,,[engine:junit-jupiter]/[
class:PerformanceTesterTest]
%TSTTREE3,testPerformTaskCompletesInTime(PerformanceTesterTest),false,1,false,2,testPerformTas
kCompletesInTime(),,[engine:junit-jupiter]/[class:PerformanceTesterTest]/[method:testPerformTa
skCompletesInTime()]
%TESTS  3,testPerformTaskCompletesInTime(PerformanceTesterTest)

%TESTE  3,testPerformTaskCompletesInTime(PerformanceTesterTest)

%RUNTIME252
```