Advanced JUnit Testing Exercises:

Exercise 1: Parameterized Tests:

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>aex1</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>24</maven.compiler.source>
        <maven.compiler.target>24</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

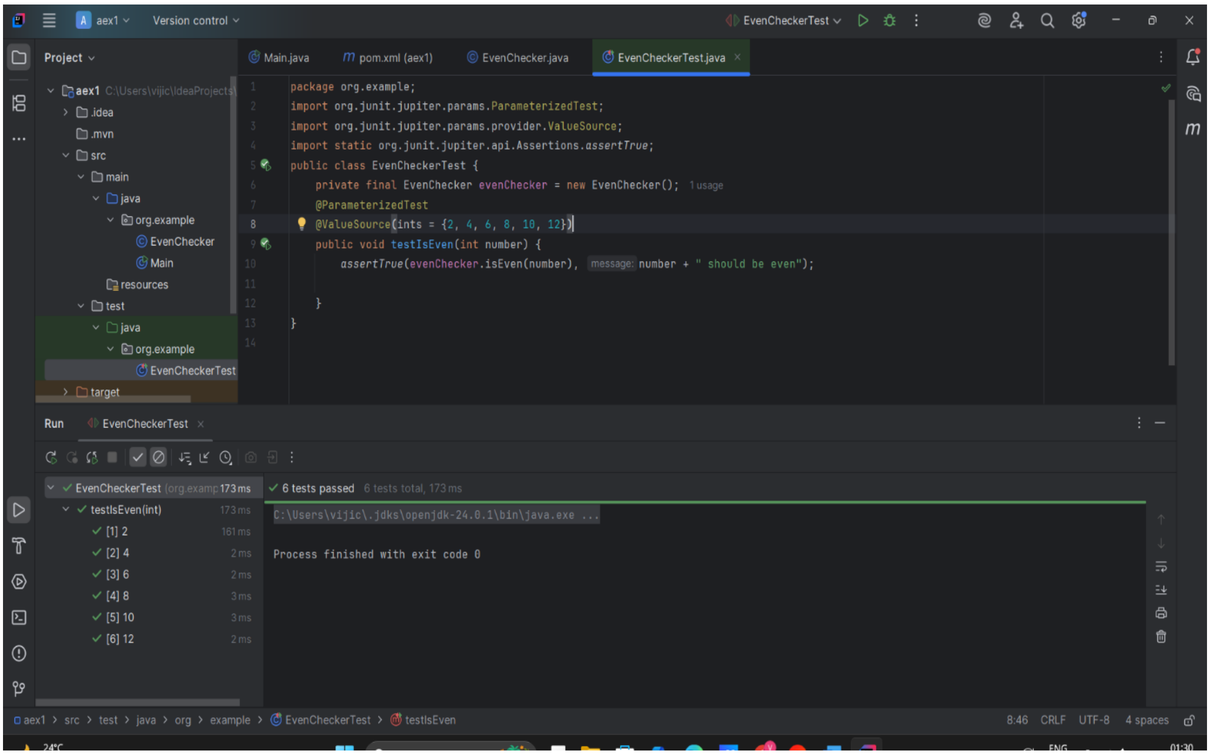
EventChecker.java:

```
package org.example;
public class EvenChecker {
        public boolean isEven(int number) {
           return number % 2 == 0;
        }
    }
```

EventCheckerTest.java:

```
package org.example;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;
import static org.junit.jupiter.api.Assertions.assertTrue;
public class EvenCheckerTest {
    private final EvenChecker evenChecker = new EvenChecker();
    @ParameterizedTest
    @ValueSource(ints = {2, 4, 6, 8, 10, 12})
    public void testIsEven(int number) {
        assertTrue(evenChecker.isEven(number), number + " should be even");
    }
}
```

Output:



```
package org.example;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;
import static org.junit.jupiter.api.Assertions.assertTrue;
public class EvenCheckerTest {
    private final EvenChecker evenChecker = new EvenChecker();  1 usage
    @ParameterizedTest
    @ValueSource(ints = {2, 4, 6, 8, 10, 12})
    public void testIsEven(int number) {
        assertTrue(evenChecker.isEven(number), message: number + " should be even");

    }
}
```

```
✓ EvenCheckerTest (org.examp 173 ms
  ✓ testIsEven(int)          173 ms
    ✓ [1] 2                  161 ms
    ✓ [2] 4                    2 ms
    ✓ [3] 6                    2 ms
    ✓ [4] 8                    3 ms
    ✓ [5] 10                   3 ms
    ✓ [6] 12                   2 ms
```

```
✓ 6 tests passed  6 tests total, 173 ms
C:\Users\vijic\.jdks\openjdk-24.0.1\bin\java.exe ...

Process finished with exit code 0
```

Exercise 2: Test Suites and Categories

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>aex2</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>24</maven.compiler.source>
        <maven.compiler.target>24</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version>
            <scope>test</scope>
        </dependency>

        <dependency>
            <groupId>org.junit.platform</groupId>
            <artifactId>junit-platform-suite</artifactId>
            <version>1.10.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

Calculator.java:

```
package org.example;
public class Calculator {
   public int add(int a, int b) {
      return a + b;
   }
 public int multiply(int a, int b) {
      return a * b;
   }
}
```

AdditionTest.java:

```
package org.example;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;
public class AdditionTest {
   Calculator calc = new Calculator();
   @Test
   public void testAdd() {
      assertEquals(5, calc.add(2, 3));
   }
}
```

MultiplicationTest.java:

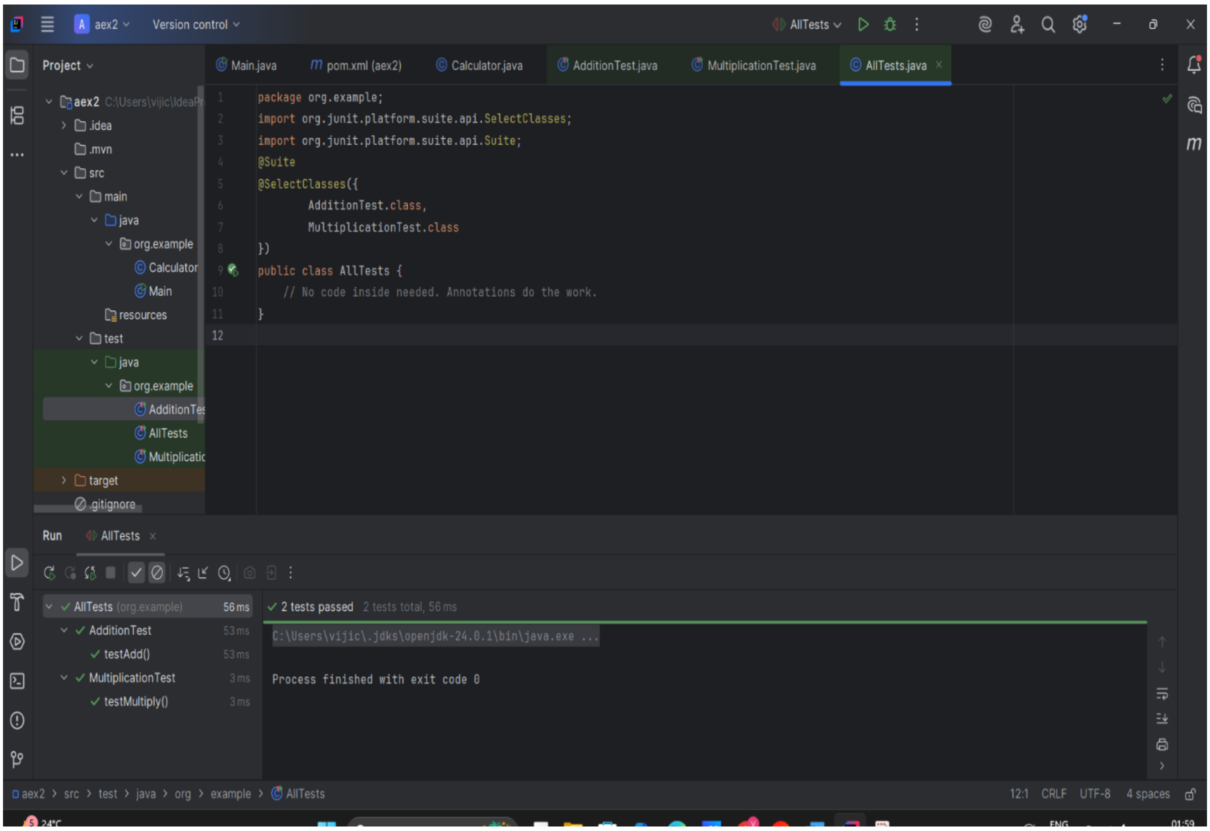
```
package org.example;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;

public class MultiplicationTest {
   Calculator calc = new Calculator();
   @Test
   public void testMultiply() {
      assertEquals(6, calc.multiply(2, 3));
   }
}
```

AllTests.java:

```
package org.example;
import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;
@Suite
@SelectClasses({
        AdditionTest.class,
        MultiplicationTest.class
})
public class AllTests {
    // No code inside needed. Annotations do the work.
}
```

Output:



```
package org.example;
import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;
@Suite
@SelectClasses({
        AdditionTest.class,
        MultiplicationTest.class
})
public class AllTests {
    // No code inside needed. Annotations do the work.
}
```

```
✓ AllTests (org.example)   56 ms     ✓ 2 tests passed  2 tests total, 56 ms
  ✓ AdditionTest           53 ms     C:\Users\vijic\.jdks\openjdk-24.0.1\bin\java.exe ...
    ✓ testAdd()            53 ms
  ✓ MultiplicationTest      3 ms     Process finished with exit code 0
    ✓ testMultiply()        3 ms
```

Exercise 3: Test Execution Order:

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>aexx3</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>24</maven.compiler.source>
        <maven.compiler.target>24</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version> <!-- Or latest -->
            <scope>test</scope>
        </dependency>
    </dependencies>

</project>
```

OrderedTests.java:

```
package org.example;
import org.junit.jupiter.api.Order;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.TestMethodOrder;
import org.junit.jupiter.api.MethodOrderer;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
```

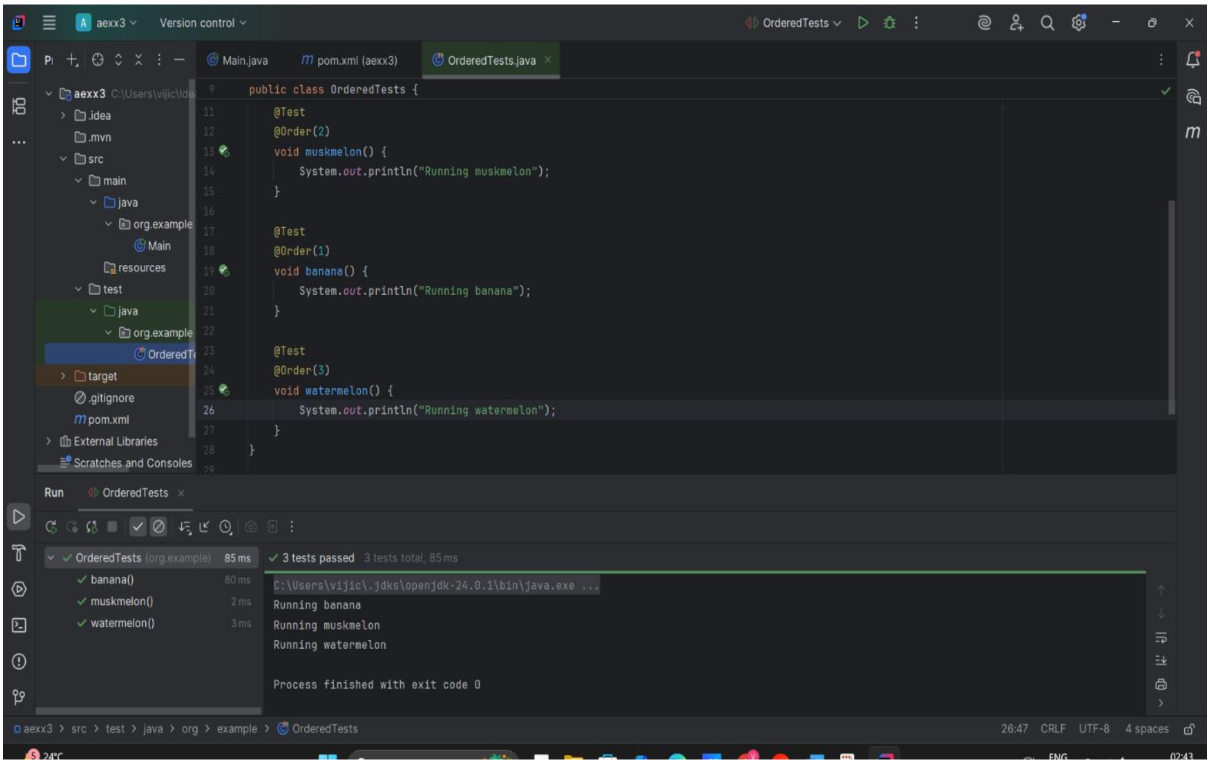
```
public class OrderedTests {

    @Test
    @Order(2)
    void muskmelon() {
        System.out.println("Running muskmelon");
    }

    @Test
    @Order(1)
    void banana() {
        System.out.println("Running banana");
    }

    @Test
    @Order(3)
    void watermelon() {
        System.out.println("Running watermelon");
    }
}
```

OUTPUT



```
public class OrderedTests {
    @Test
    @Order(2)
    void muskmelon() {
        System.out.println("Running muskmelon");
    }

    @Test
    @Order(1)
    void banana() {
        System.out.println("Running banana");
    }

    @Test
    @Order(3)
    void watermelon() {
        System.out.println("Running watermelon");
    }
}
```

✓ OrderedTests (org.example) 85 ms
✓ banana() 80 ms
✓ muskmelon() 2 ms
✓ watermelon() 3 ms

✓ 3 tests passed 3 tests total, 85 ms

```
C:\Users\vijic\.jdks\openjdk-24.0.1\bin\java.exe ...
Running banana
Running muskmelon
Running watermelon

Process finished with exit code 0
```

Exercise 4: Exception Testing:

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>aaex4</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>24</maven.compiler.source>
        <maven.compiler.target>24</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.10.0</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

ExceptionThrower.java:

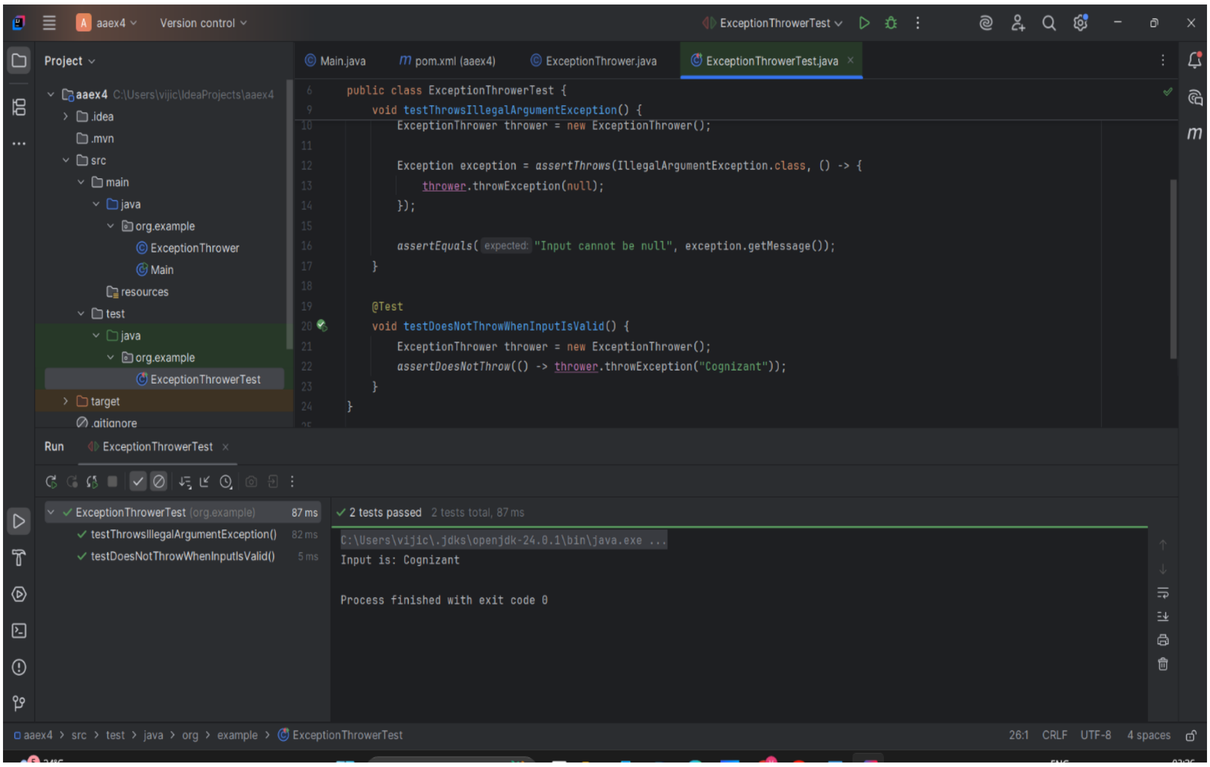
```
package org.example;
public class ExceptionThrower {
    public void throwException(String input) {
        if (input == null) {
            throw new IllegalArgumentException("Input cannot be null");
        }
        System.out.println("Input is: " + input);
    }
}
```

ExceptionThrowerTest.java:

```
package org.example;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;
public class ExceptionThrowerTest {

    @Test
    void testThrowsIllegalArgumentException() {
        ExceptionThrower thrower = new ExceptionThrower();
        Exception exception = assertThrows(IllegalArgumentException.class, () -> {
            thrower.throwException(null);
        });
        assertEquals("Input cannot be null", exception.getMessage());
    }
    @Test
    void testDoesNotThrowWhenInputIsValid() {
        ExceptionThrower thrower = new ExceptionThrower();
        assertDoesNotThrow(() -> thrower.throwException("Cognizant"));
    }
}
```

Output:



```
public class ExceptionThrowerTest {
    void testThrowsIllegalArgumentException() {
        ExceptionThrower thrower = new ExceptionThrower();

        Exception exception = assertThrows(IllegalArgumentException.class, () -> {
            thrower.throwException(null);
        });

        assertEquals( expected: "Input cannot be null", exception.getMessage());
    }

    @Test
    void testDoesNotThrowWhenInputIsValid() {
        ExceptionThrower thrower = new ExceptionThrower();
        assertDoesNotThrow(() -> thrower.throwException("Cognizant"));
    }
}
```

```
✓ ExceptionThrowerTest (org.example)       87 ms
    ✓ testThrowsIllegalArgumentException()  82 ms
    ✓ testDoesNotThrowWhenInputIsValid()     5 ms
```

```
✓ 2 tests passed  2 tests total, 87 ms
C:\Users\vijic\.jdks\openjdk-24.0.1\bin\java.exe ...
Input is: Cognizant

Process finished with exit code 0
```

Exercise 5: Timeout and Performance Testing:

Pom.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>JUnitTimeoutDemo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
        <junit.jupiter.version>5.10.0</junit.jupiter.version>
    </properties>

    <dependencies>
        <!-- JUnit 5 Dependency -->
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>${junit.jupiter.version}</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <!-- Surefire Plugin for running tests -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>3.0.0-M9</version>
                <configuration>
                    <useModulePath>false</useModulePath>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

PerformanceTester.java:

```
package org.example;
public class PerformanceTester {
    public void performTask() {
        try {
            // Simulate task that takes time (e.g., 500ms)
            Thread.sleep(500);
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
}
```

PerformanceTesterTest.java:

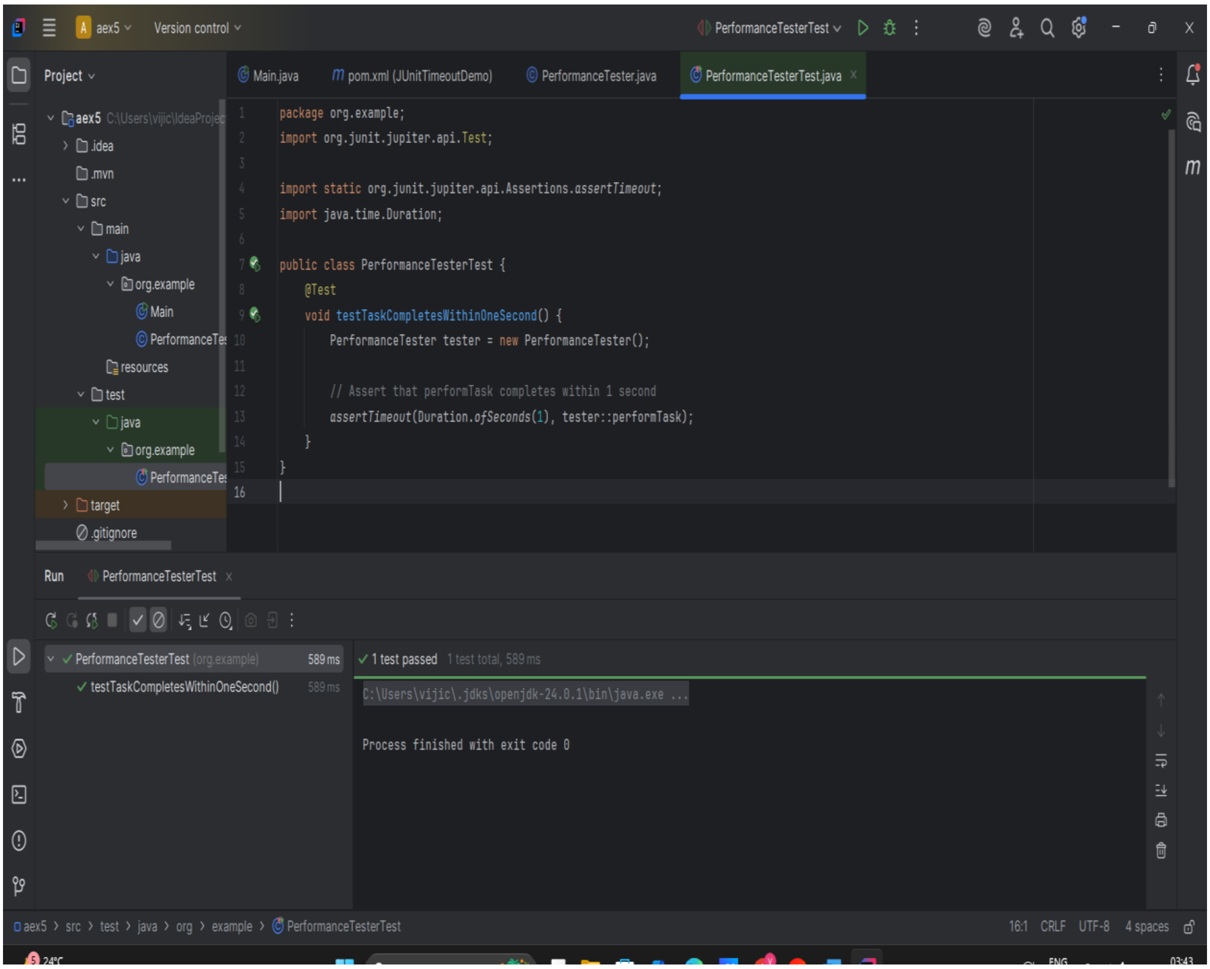
```
package org.example;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertTimeout;
import java.time.Duration;

public class PerformanceTesterTest {
    @Test
    void testTaskCompletesWithinOneSecond() {
        PerformanceTester tester = new PerformanceTester();

        // Assert that performTask completes within 1 second
        assertTimeout(Duration.ofSeconds(1), tester::performTask);
    }
}
```

OUTPUT:



```java
package org.example;
import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.assertTimeout;
import java.time.Duration;

public class PerformanceTesterTest {
    @Test
    void testTaskCompletesWithinOneSecond() {
        PerformanceTester tester = new PerformanceTester();

        // Assert that performTask completes within 1 second
        assertTimeout(Duration.ofSeconds(1), tester::performTask);
    }
}

```

✓ 1 test passed 1 test total, 589 ms

```
C:\Users\vijic\.jdks\openjdk-24.0.1\bin\java.exe ...

Process finished with exit code 0
```