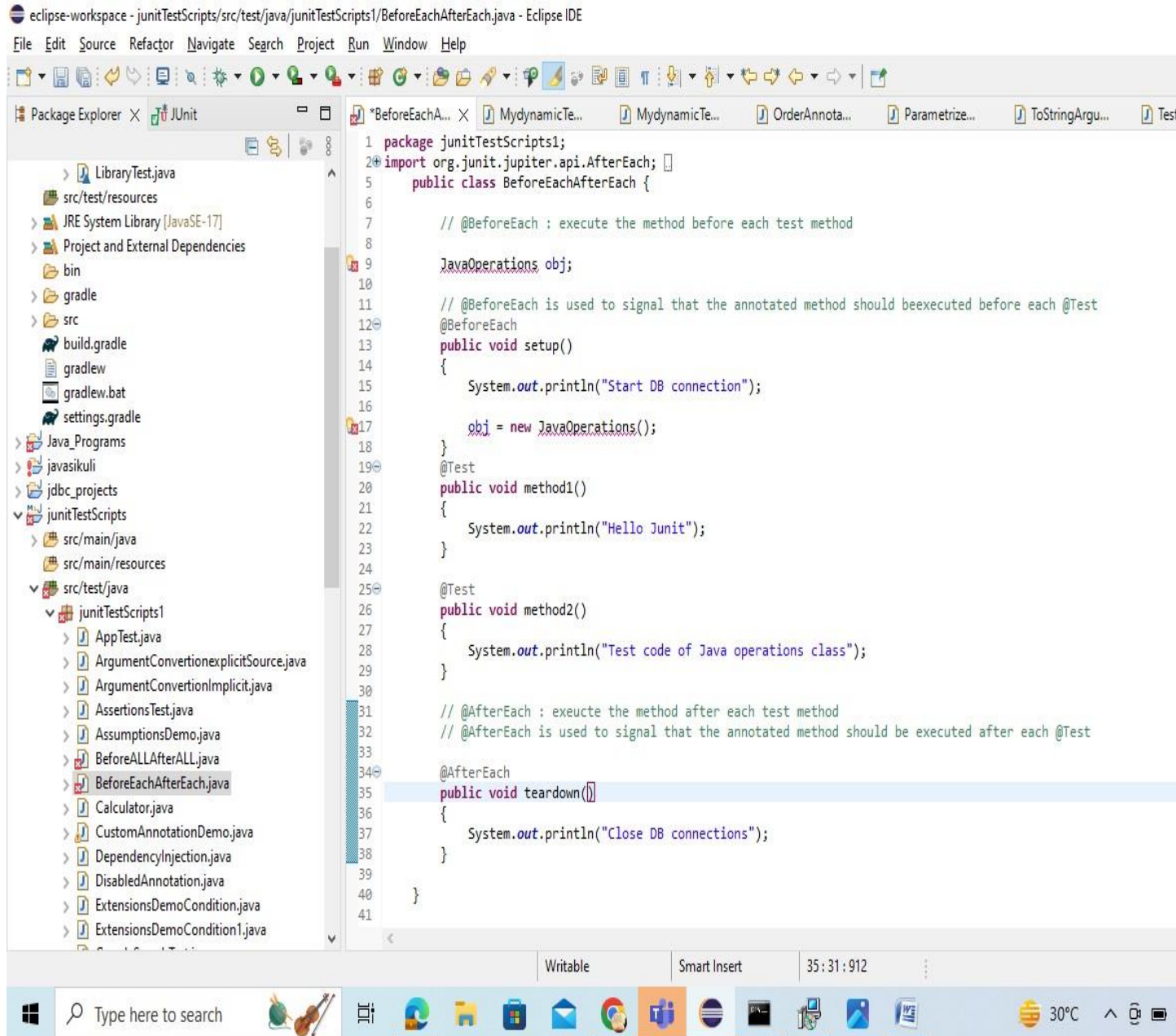


# PHASE-2 ASSISTED PRACTICE PROJECT

EMP ID: 2587325  
NAME: HIRANMAI

## 1. Lifecycle Methods.



```
eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/BeforeEachAfterEach.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X JUnit
  > LibraryTest.java
  > src/test/resources
  > JRE System Library [JavaSE-17]
  > Project and External Dependencies
  > bin
  > gradle
  > src
  > build.gradle
  > gradlew
  > gradlew.bat
  > settings.gradle
  > Java_Programs
  > javasikuli
  > jdbc_projects
  > junitTestScripts
    > src/main/java
    > src/main/resources
    > src/test/java
      > junitTestScripts1
        > AppTest.java
        > ArgumentConversionexplicitSource.java
        > ArgumentConversionImplicit.java
        > AssertionsTest.java
        > AssumptionsDemo.java
        > BeforeAllAfterAll.java
        > BeforeEachAfterEach.java
        > Calculator.java
        > CustomAnnotationDemo.java
        > DependencyInjection.java
        > DisabledAnnotation.java
        > ExtensionsDemoCondition.java
        > ExtensionsDemoCondition1.java

1 package junitTestScripts1;
2 import org.junit.jupiter.api.AfterEach;
3
4 public class BeforeEachAfterEach {
5
6     // @BeforeEach : execute the method before each test method
7
8     JavaOperations obj;
9
10    // @BeforeEach is used to signal that the annotated method should be executed before each @Test
11    @BeforeEach
12    public void setup()
13    {
14        System.out.println("Start DB connection");
15
16        obj = new JavaOperations();
17    }
18
19    @Test
20    public void method1()
21    {
22        System.out.println("Hello Junit");
23    }
24
25    @Test
26    public void method2()
27    {
28        System.out.println("Test code of Java operations class");
29    }
30
31    // @AfterEach : execute the method after each test method
32    // @AfterEach is used to signal that the annotated method should be executed after each @Test
33
34    @AfterEach
35    public void teardown()
36    {
37        System.out.println("Close DB connections");
38    }
39
40 }
41
```

## 2. Assertions

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/AssertionsTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x JUnit

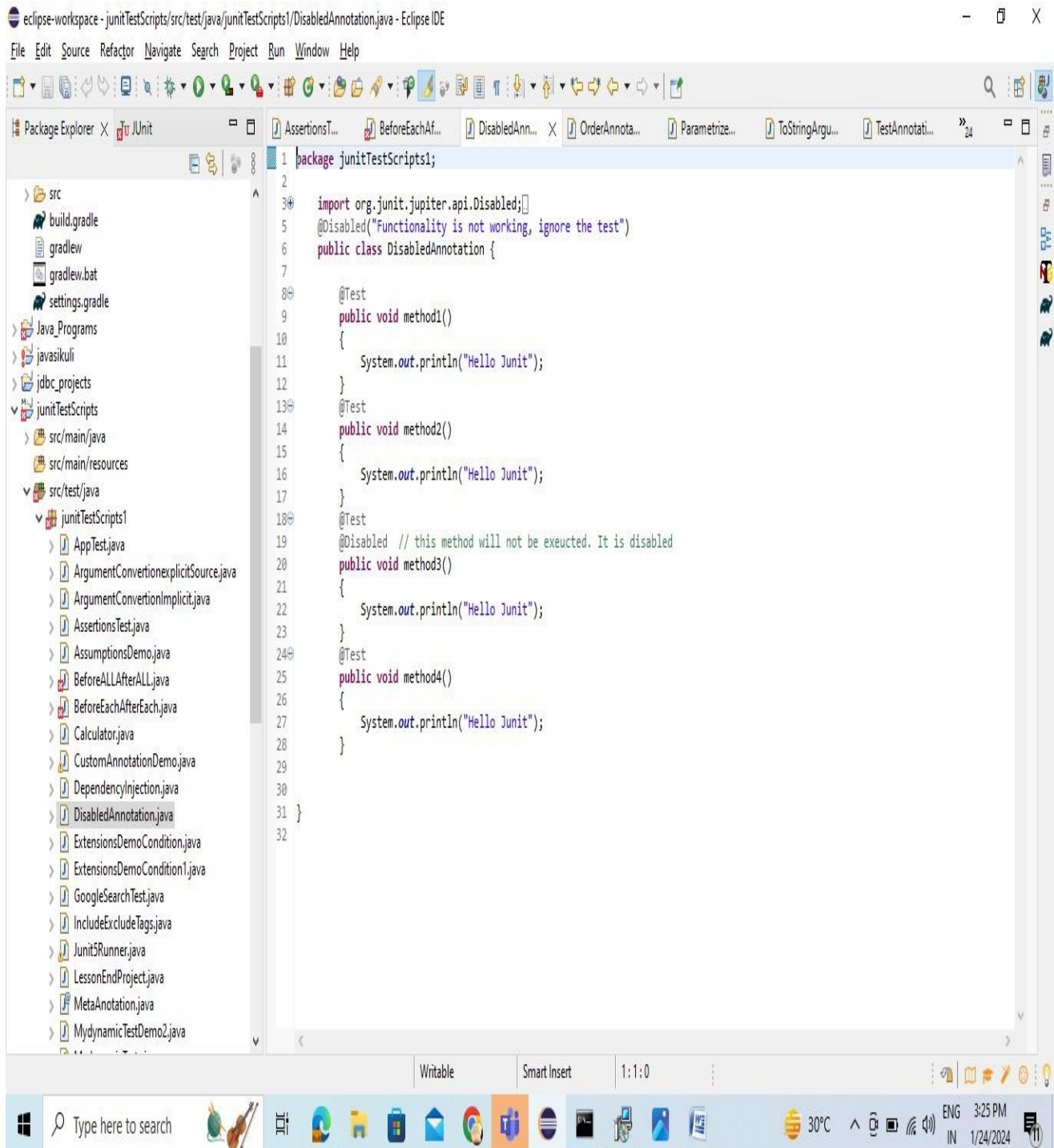
LibraryTest.java  
src/test/resources  
JRE System Library [JavaSE-17]  
Project and External Dependencies  
bin  
gradle  
src  
build.gradle  
gradlew  
gradlew.bat  
settings.gradle  
Java\_Programs  
jasikuli  
jdbc\_projects  
junitTestScripts  
src/main/java  
src/main/resources  
src/test/java  
junitTestScripts1  
AppTest.java  
ArgumentConversionexplicitSource.java  
ArgumentConversionImplicit.java  
AssertionsTest.java  
AssumptionsDemo.java  
BeforeALLAfterALL.java  
BeforeEachAfterEach.java  
Calculator.java  
CustomAnnotationDemo.java  
DependencyInjection.java  
DisabledAnnotation.java  
ExtensionsDemoCondition.java  
ExtensionsDemoCondition1.java

```
1 package junitTestScripts1;
2 import org.junit.jupiter.api.Assertions;
3 public class AssertionsTest {
4
5     Calculator obj;
6
7     @BeforeEach
8     public void setup()
9     {
10         obj = new Calculator();
11     }
12
13     @Test
14     public void Testadd()
15     {
16         int expectnumber = 40;
17         int actualnumber = obj.add(2, 2);
18
19         Assertions.assertEquals(expectnumber, actualnumber); // test will fail with assertion
20
21         // give custom message with assertion method
22         // Assertions.assertEquals(expectnumber, actualnumber, "Sum is Incorrect..recheck the code");
23     }
24
25     @Test
26     public void Testadd2()
27     {
28         int expectnumber = 40;
29         int actualnumber = obj.add(2, 2);
30         Assertions.assertNotEquals(expectnumber, actualnumber); // test will pass
31     }
32
33     @Test
34     public void TestAssertArray()
35     {
36
37         Assertions.assertArrayEquals(new int[] {1,2,3}, new int[] {1,2,3});
38
39         Assertions.assertArrayEquals(new int[] {1,2,3}, new int[] {1,2,3,4,5});
40
41     }
```

Writable Smart Insert 1:1:0

Type here to search 30°C ENG IN 3:19 PM 1/24/2024

### 3. Disabling Tests



## 4. Assumptions

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/AssumptionsDemo.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer JUnit

AssumptionsDemo.java

```
1 package junitTestScripts1;
2 import org.junit.jupiter.api.Assumptions;
3
4 public class AssumptionsDemo {
5
6     @RepeatedTest(3)
7     @Test
8     public void AssumptionsDemo()
9     {
10         // Condition Assumed
11         boolean isDBServerUp= true;
12
13         Assumptions.assumeTrue(isDBServerUp);
14         // the test case will aborted as the assumption is not true
15
16         System.out.println("execute the tests");
17     }
18
19     @Test
20     public void AssumptionsDemo2()
21     {
22         // Condition Assumed
23         boolean isDBServerUp= false;
24
25         Assumptions.assumeFalse(isDBServerUp);
26         // the test case will aborted as the assumption is not true
27
28         System.out.println("Dont execute test");
29     }
30 }
31
32
33
34
```

Writable Smart Insert 1:1:0

Type here to search 30°C 3:28 PM 1/24/2024



eclipse-workspace - Eclipse IDE

File Edit Navigate Search Project Run Window Help



Package Explorer JUnit X



Finished after 0.401 seconds

Runs: 5/5 Errors: 0 Failures: 0



> AssumptionsDemo [Runner: JUnit 5] (0.064 s)

Failure Trace



Console X AssertionsT... Assumptions...

```
<terminated> AssumptionsDemo [JUnit] C:\Users\GVH\p2\pool\plugin
Jan 24, 2024 3:28:07 PM org.junit.jupiter.engine.disco
WARNING: Possible configuration error: method [public
Dont execute test
execute the tests
execute the tests
execute the tests
execute the tests
```



Type here to search



## 5. Test Interfaces and Default Methods

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/MydynamicTests.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X JUnit Run MydynamicTests Console ArgumentConv... AssertionsT... Assumptions... BeforeEachAf... DisabledAnn... MydynamicTe... X 24

```
1 package junitTestScripts1;
2
3 import java.util.Arrays;
4 import java.util.Collection;
5 import org.junit.jupiter.api.Assertions;
6 import org.junit.jupiter.api.DynamicTest;
7 import org.junit.jupiter.api.TestFactory;
8 public class MydynamicTests {
9
10     // A DynamicTest is a test case generated at runtime.
11
12     @TestFactory
13     public Collection<DynamicTest> generatedynamictests()
14     {
15
16         return Arrays.asList(
17
18             //lets write dynamic test code that will generate tests at runtime.
19             DynamicTest.dynamicTest("Test1", () -> Assertions.assertTrue(true)),
20             DynamicTest.dynamicTest("Test2", () -> Assertions.assertFalse(false)),
21             DynamicTest.dynamicTest("Test3", () -> Assertions.assertEquals(4,2*2)),
22             DynamicTest.dynamicTest("exception Handling", () -> {throw new Exception("Exception for Dynamic test");})
23         );
24     }
25 }
26
27
```

src/test/resources  
Maven Dependencies  
JRE System Library [JavaSE-11]

Writable Smart Insert 15:5:384

Type here to search

ENG 3:35 PM  
IN 1/24/2024

## 6. Repeating Tests

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/DependencyInjection.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x JUnit

- junitTestScripts1
  - AppTest.java
  - ArgumentConventionexplicitSource.java
  - ArgumentConventionImplicit.java
  - AssertionsTest.java
  - AssumptionsDemo.java
  - BeforeALLAfterALL.java
  - BeforeEachAfterEach.java
  - Calculator.java
  - CustomAnnotationDemo.java
  - DependencyInjection.java
  - DisabledAnnotation.java
  - ExtensionsDemoCondition.java
  - ExtensionsDemoCondition1.java
  - GoogleSearchTest.java
  - IncludeExcludeTags.java
  - JUnit5Runner.java
  - LessonEndProject.java
  - MetaAnotation.java
  - MydynamicTestDemo2.java
  - MydynamicTests.java
  - Myutils.java
  - OrderAnnotation.java
  - ParameterizedCSVsource.java
  - ParameterizedValueSource.java
  - ParameterizedEnumSource.java
  - ParameterizedTestMethodSource.java
  - TestAnnotationDemo1.java
  - ToStringArgumentCoverter.java
- src/test/resources
- Maven Dependencies
- JRE System Library [JavaSE-11]

Console

```
1 package junitTestScripts1;
2 import org.junit.jupiter.api.RepetitionInfo;
3 import org.junit.jupiter.api.TestInfo;
4 public class DependencyInjection {
5
6     // write the code to inject test and repeatedTest annotation as an Object
7
8     @org.junit.jupiter.api.RepeatedTest(5)
9     public void repeatedtestdemo(TestInfo testinfo, RepetitionInfo repInfo)
10    {
11
12        System.out.println(testinfo.getDisplayName() + " " + repInfo.getCurrentRepetition() );
13
14    }
15
16
17
18 }
19
```

Writable Smart Insert 1:27:26

Type here to search

30°C ENG IN 3:41 PM 1/24/20



Finished after 0.431 seconds

Runs: 5/5 Errors: 0 Failures: 0



&gt; DependencyInjection [Runner: JUnit 5] (0.089 s)

Failure Trace



&lt;terminated&gt; DependencyInjection [JUnit] C:\Users\GVH\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.9.v20231028-0858\jre\bin\javaw.exe (Jan 24, 2024, 3:43 PM)

repetition 1 of 51  
repetition 2 of 52  
repetition 3 of 53  
repetition 4 of 54  
repetition 5 of 55





## 7. Dynamic Tests

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/MydynamicTestDemo2.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x JUnit Console ArgumentConv... DependencyIn... Assumptions... DisabledAnn... MydynamicTe... x

junitTestScripts1

- AppTest.java
- ArgumentConversionexplicitSource.java
- ArgumentConversionImplicit.java
- AssertionsTest.java
- AssumptionsDemo.java
- BeforeALLAfterALL.java
- BeforeEachAfterEach.java
- Calculator.java
- CustomAnnotationDemo.java
- DependencyInjection.java
- DisabledAnnotation.java
- ExtensionsDemoCondition.java
- ExtensionsDemoCondition1.java
- GoogleSearchTest.java
- IncludeExcludeTags.java
- JUnit5Runner.java
- LessonEndProject.java
- MetaAnnotation.java
- MydynamicTestDemo2.java
- MydynamicTests.java
- Myutils.java
- OrderAnnotation.java
- ParameterizedCSVsource.java
- ParameterizedValueSource.java
- ParameterizedEnumSource.java
- ParameterizedTestMethodSource.java
- TestAnnotationDemo1.java
- ToStringArgumentConverter.java

src/test/resources

Maven Dependencies

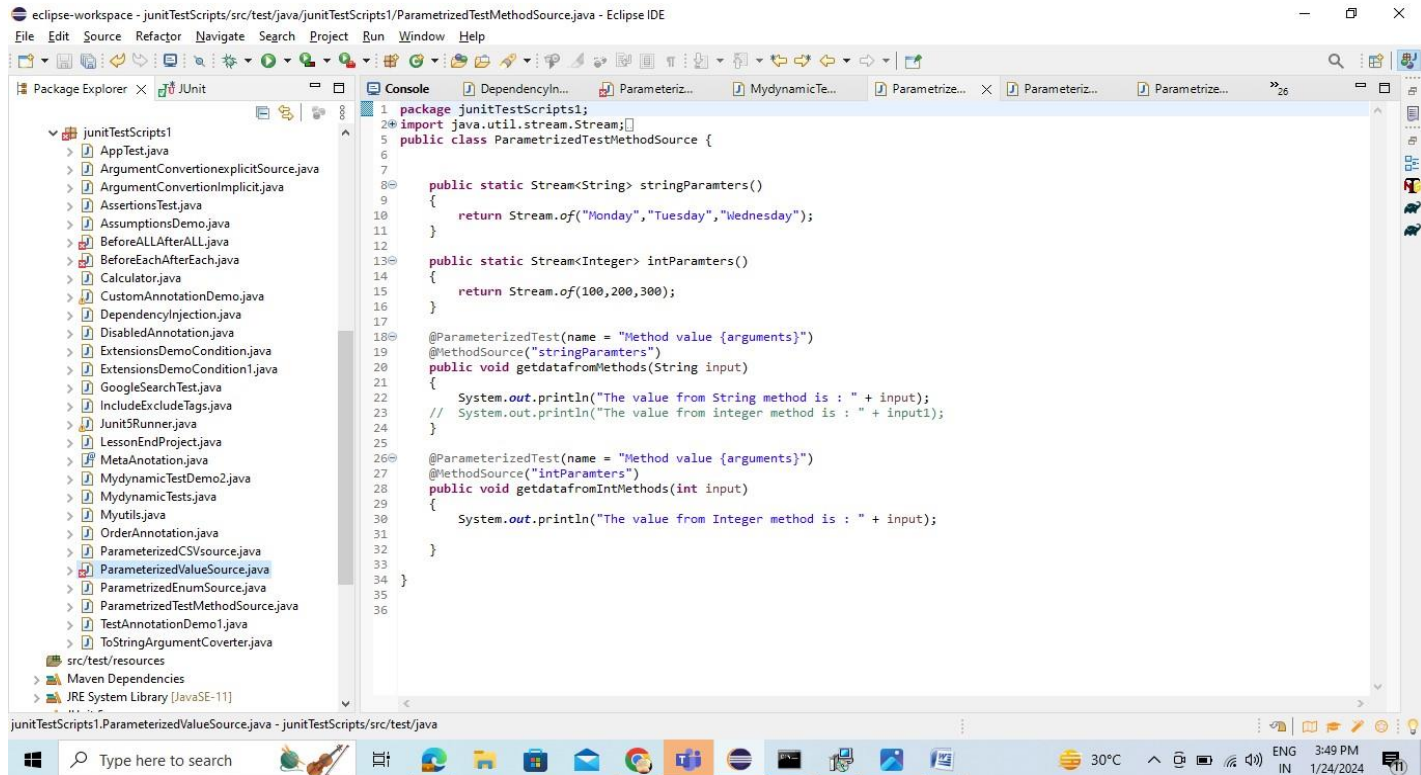
JRE System Library [JavaSE-11]

```
1 package junitTestScripts1;
2 import java.util.stream.Stream;
3 import org.junit.jupiter.api.Assertions;
4 import org.junit.jupiter.api.DynamicTest;
5 import org.junit.jupiter.api.TestFactory;
6 public class MydynamicTestDemo2 {
7
8     @TestFactory
9     public Stream<DynamicTest> generatedynamicTest()
10     {
11
12         return Stream.of("pop", "mom", "madam", "dad", "mam")
13             .map(inputText ->
14                 DynamicTest.dynamicTest(inputText,
15                     () -> Assertions.assertTrue(Myutils.isPalindrome(inputText)))
16             );
17
18     }
19
20 }
21
22 }
23
24 }
```

Writable Smart Insert 1:1:0

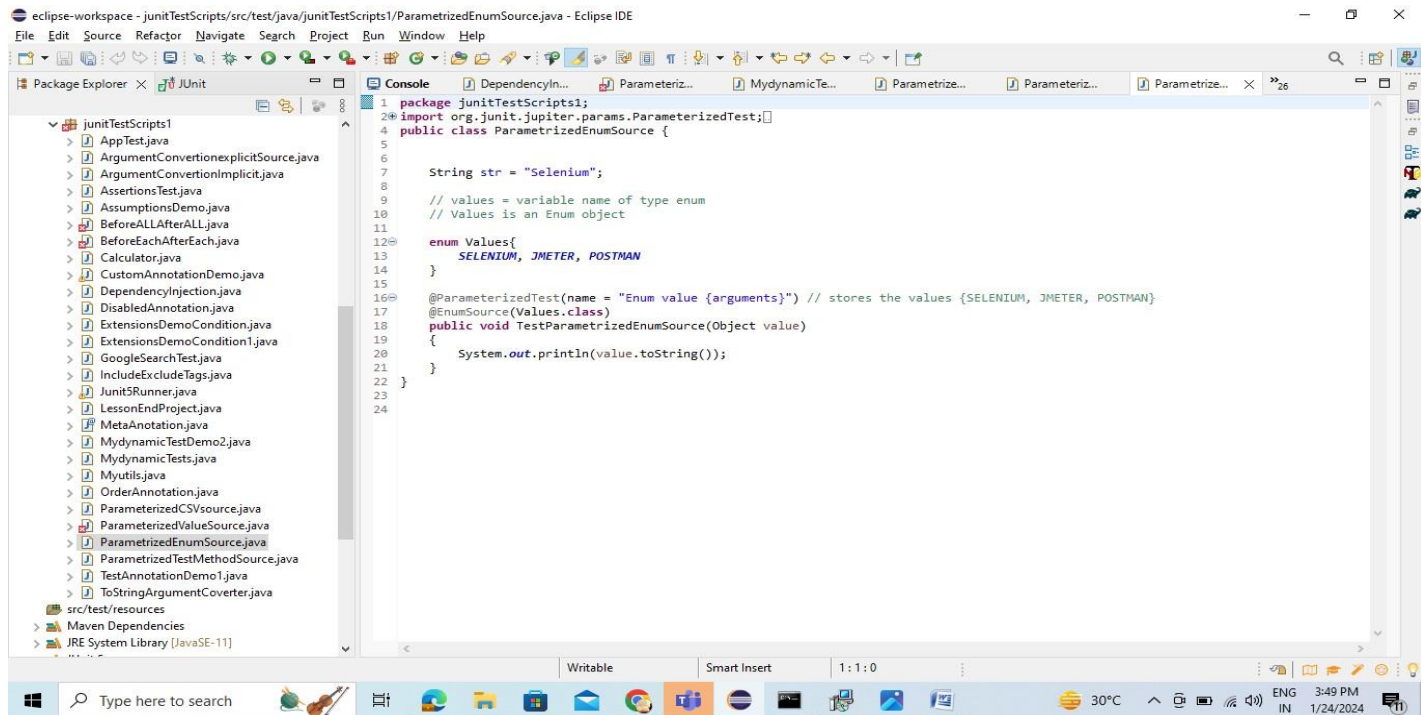
Type here to search 30°C

## 8. Parameterized Tests



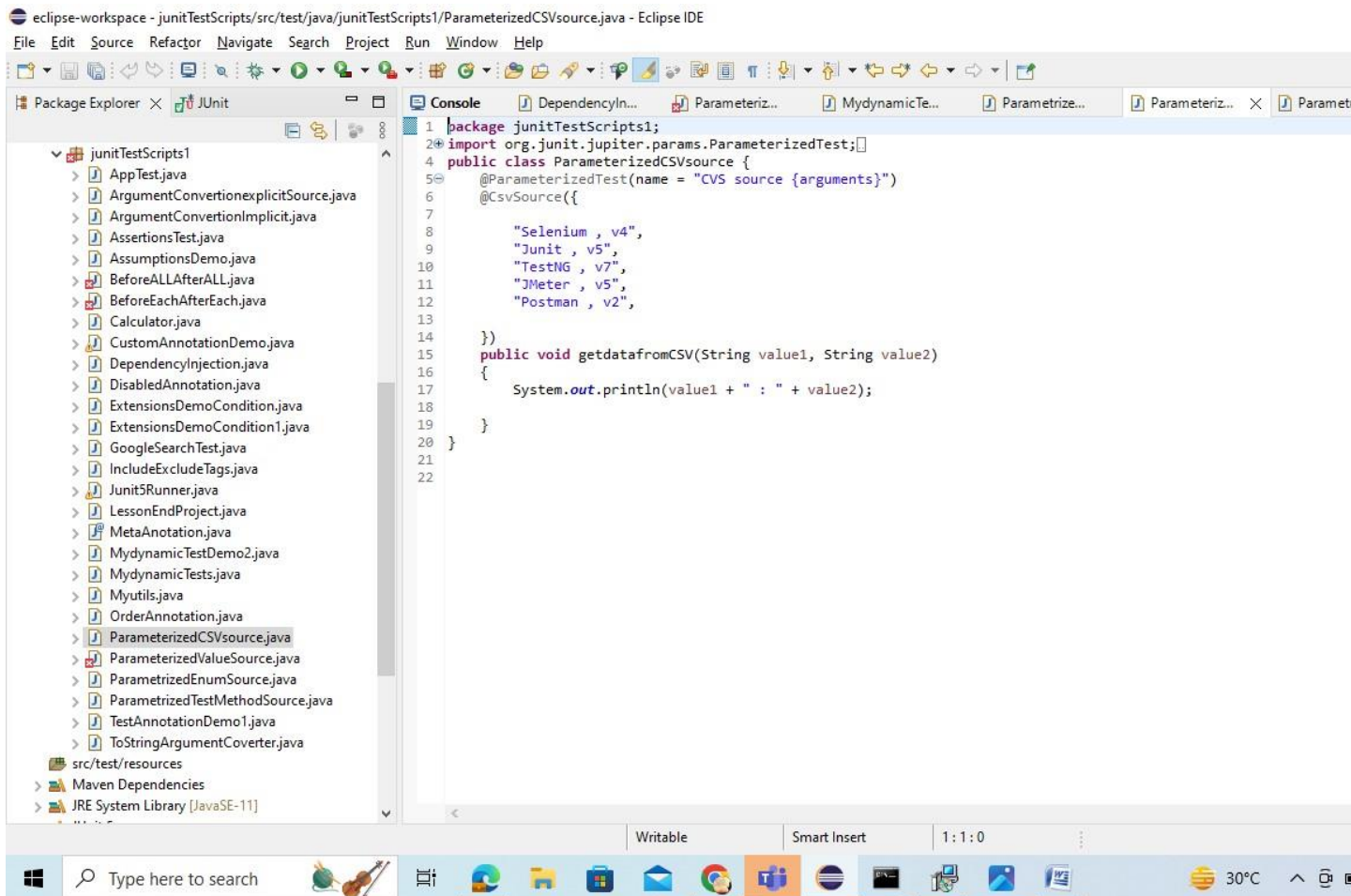
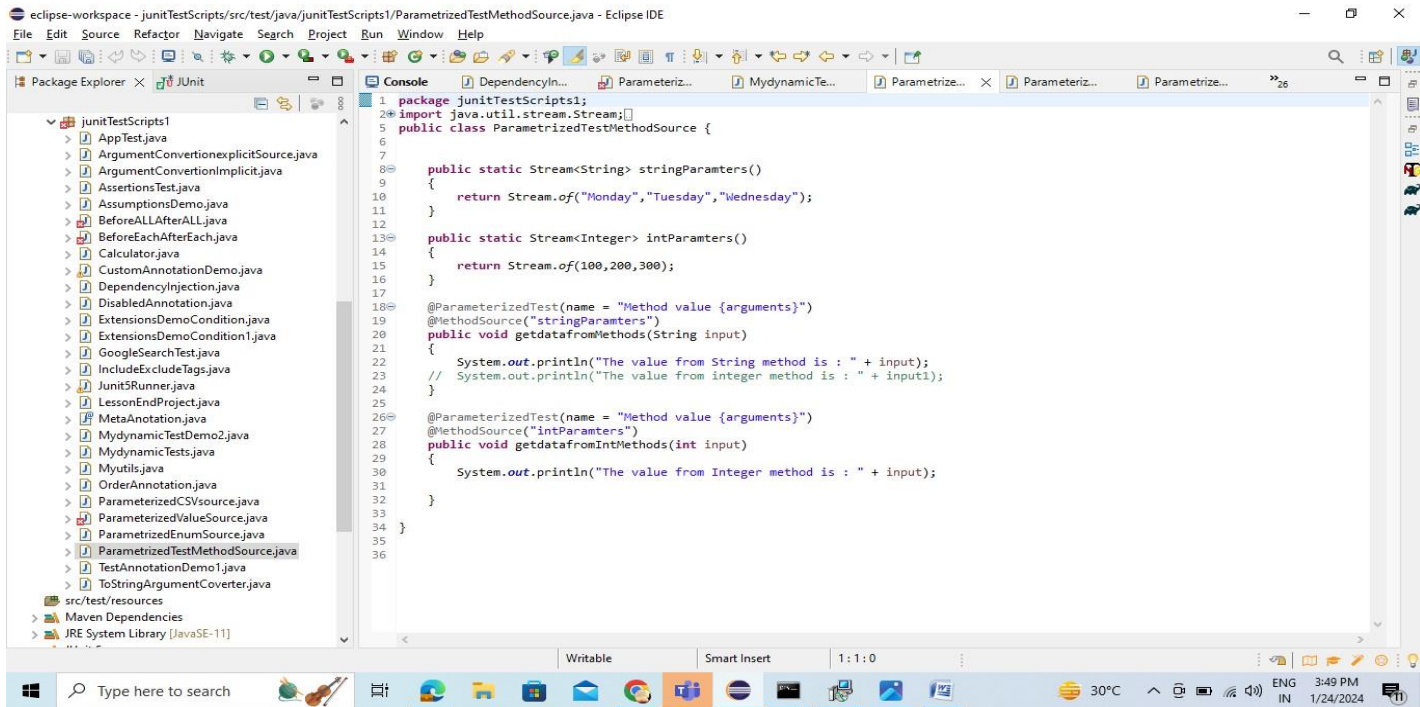
eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/ParametrizedTestMethodSource.java - Eclipse IDE

```
1 package junitTestScripts1;
2 import java.util.stream.Stream;
3
4 public class ParametrizedTestMethodSource {
5
6
7
8     public static Stream<String> stringParameters()
9     {
10         return Stream.of("Monday", "Tuesday", "Wednesday");
11     }
12
13     public static Stream<Integer> intParameters()
14     {
15         return Stream.of(100, 200, 300);
16     }
17
18     @ParameterizedTest(name = "Method value {arguments}")
19     @MethodSource("stringParameters")
20     public void getDatafromMethods(String input)
21     {
22         System.out.println("The value from String method is : " + input);
23         // System.out.println("The value from integer method is : " + input1);
24     }
25
26     @ParameterizedTest(name = "Method value {arguments}")
27     @MethodSource("intParameters")
28     public void getDatafromIntMethods(int input)
29     {
30         System.out.println("The value from Integer method is : " + input);
31     }
32
33 }
34
35
36
```



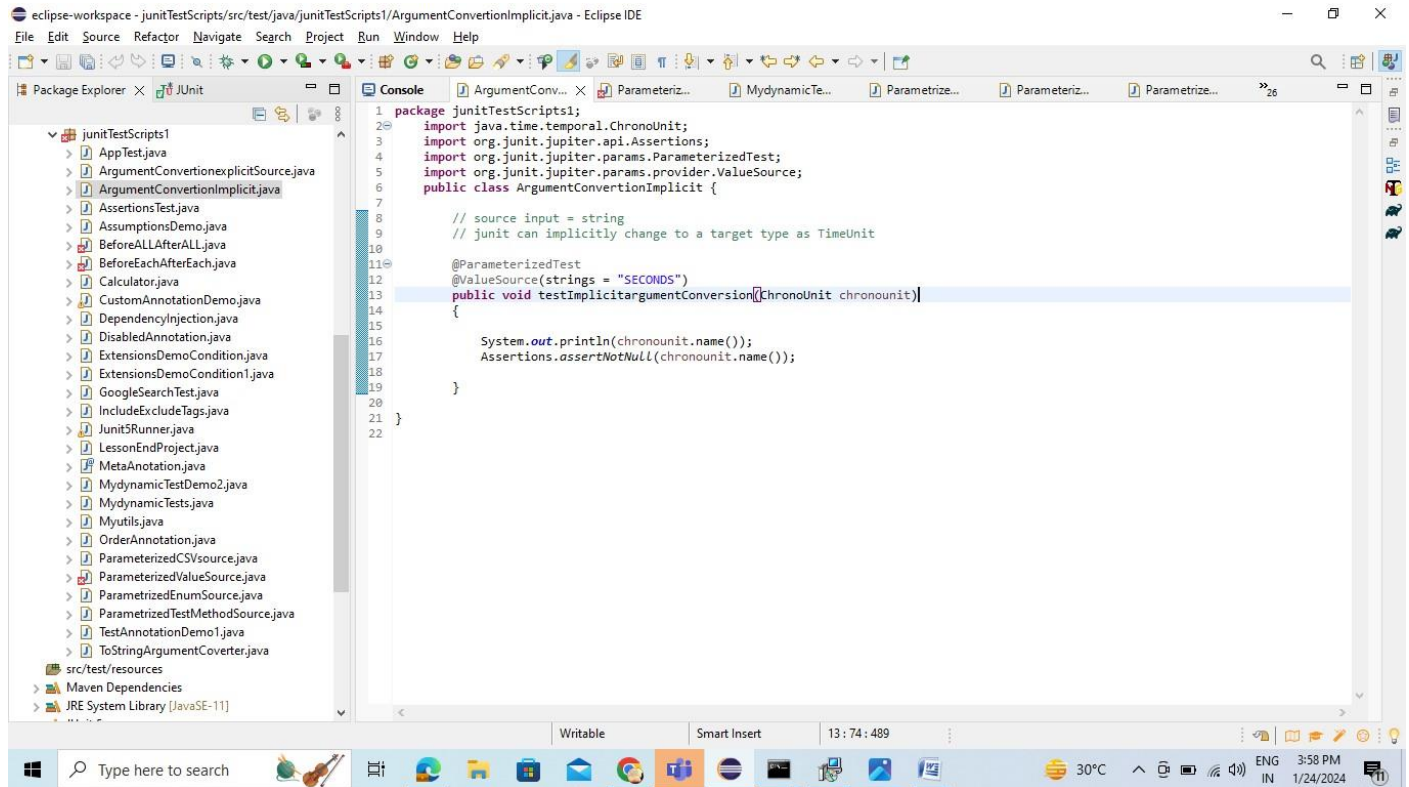
eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/ParametrizedEnumSource.java - Eclipse IDE

```
1 package junitTestScripts1;
2 import org.junit.jupiter.params.ParameterizedTest;
3
4 public class ParametrizedEnumSource {
5
6
7     String str = "Selenium";
8
9     // values = variable name of type enum
10    // Values is an Enum object
11
12    enum Values{
13        SELENIUM, JMETER, POSTMAN
14    }
15
16    @ParameterizedTest(name = "Enum value {arguments}") // stores the values {SELENIUM, JMETER, POSTMAN}
17    @EnumSource(Values.class)
18    public void TestParametrizedEnumSource(Object value)
19    {
20        System.out.println(value.toString());
21    }
22
23 }
24
```

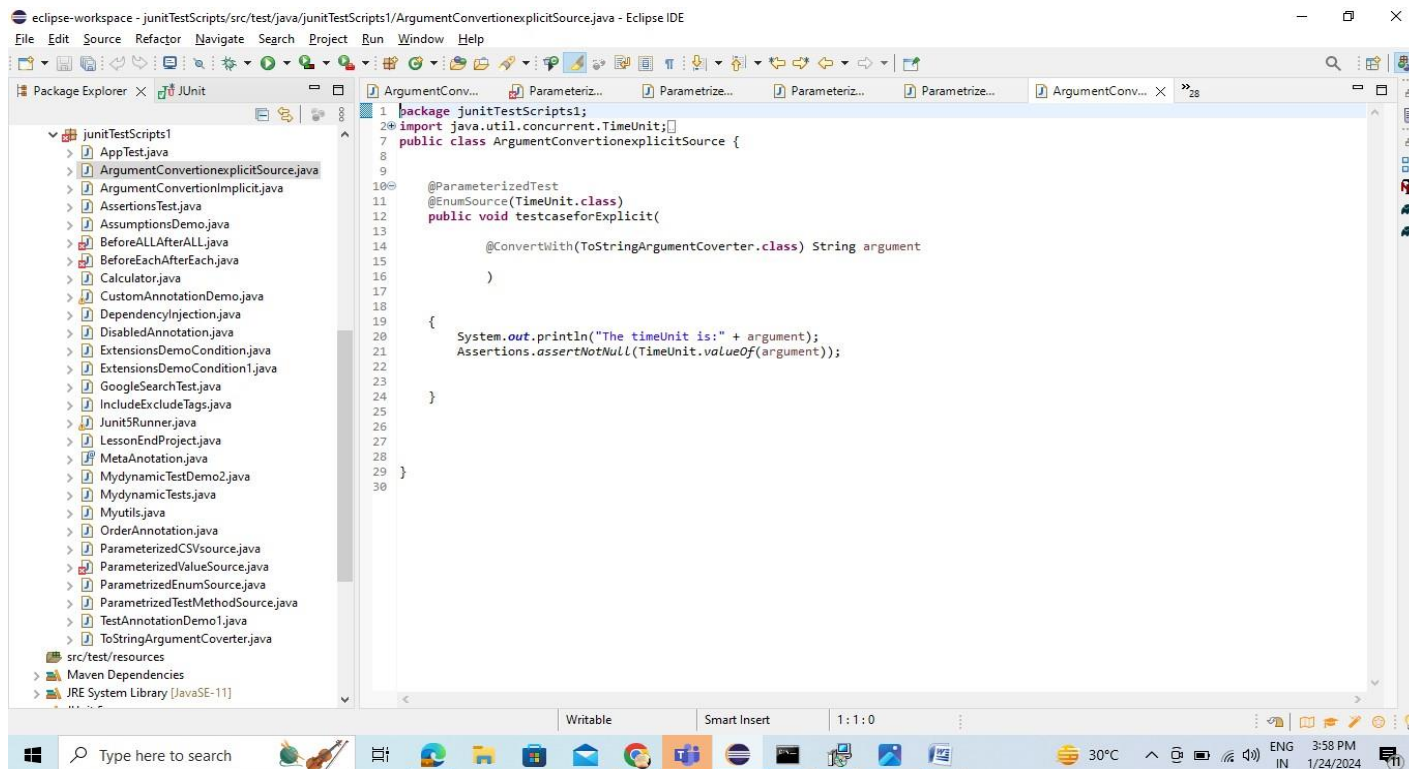




## 10. Argument Conversion



```
1 package junitTestScripts1;
2 import java.time.temporal.ChronoUnit;
3 import org.junit.jupiter.api.Assertions;
4 import org.junit.jupiter.params.ParameterizedTest;
5 import org.junit.jupiter.params.provider.ValueSource;
6 public class ArgumentConversionImplicit {
7
8     // source input = string
9     // junit can implicitly change to a target type as TimeUnit
10
11     @ParameterizedTest
12     @ValueSource(strings = "SECONDS")
13     public void testImplicitArgumentConversion(ChronoUnit chronounit) {
14
15         System.out.println(chronounit.name());
16         Assertions.assertNotNull(chronounit.name());
17     }
18 }
19
20
21
22
```



```
1 package junitTestScripts1;
2 import java.util.concurrent.TimeUnit;
3
4 public class ArgumentConversionExplicitSource {
5
6     @ParameterizedTest
7     @EnumSource(TimeUnit.class)
8     public void testCaseForExplicit(
9
10         @ConvertWith(ToStringArgumentConverter.class) String argument
11     ) {
12
13         System.out.println("The TimeUnit is: " + argument);
14         Assertions.assertNotNull(TimeUnit.valueOf(argument));
15     }
16 }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```



## 11. Extension Points

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/ExtensionsDemoCondition1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x JUnit

- junitTestScripts1
  - AppTest.java
  - ArgumentConversionExplicitSource.java
  - ArgumentConversionImplicit.java
  - AssertionsTest.java
  - AssumptionsDemo.java
  - BeforeAllAfterAll.java
  - BeforeEachAfterEach.java
  - Calculator.java
  - CustomAnnotationDemo.java
  - DependencyInjection.java
  - DisabledAnnotation.java
  - ExtensionsDemoCondition.java
  - ExtensionsDemoCondition1.java
  - GoogleSearchTest.java
  - IncludeExcludeTags.java
  - JUnit5Runner.java
  - LessonEndProject.java
  - MetaAnnotation.java
  - MydynamicTestDemo2.java
  - MydynamicTests.java
  - Myutils.java
  - OrderAnnotation.java
  - ParameterizedCSVSource.java
  - ParameterizedValueSource.java
  - ParameterizedEnumSource.java
  - ParameterizedTestMethodSource.java
  - TestAnnotationDemo1.java
  - ToStringArgumentCoverter.java
- src/test/resources
- Maven Dependencies
- JRE System Library [JavaSE-11]

```
1 package junitTestScripts1;
2 import org.junit.jupiter.api.Test;
3 public class ExtensionsDemoCondition1 {
4
5     // test the method based on the following condition
6
7     /*
8      * > condition OS
9      * > Condition based on JRE
10      * > Condition based on range of Java
11      *
12      */
13     @Test
14     @EnabledOnOs(OS.MAC)
15     // condition is If the OS is Mac -> run the test else ignore/disable the test
16     public void testConditionOS()
17     {
18         System.out.println("OS is matching and test is exeucted");
19     }
20
21     @Test
22     @EnabledOnJre(JRE.JAVA_14)
23     // condition is If the java version on laptop is 14 -> run the test else ignore/disable the test
24     public void testConditionJRE()
25     {
26         System.out.println("Java version is matching and test is exeucted");
27     }
28
29     @Test
30     @EnabledForJreRange(min = JRE.JAVA_10, max= JRE.JAVA_17)
31     // condition is If the java version on laptop is in between 10 to 17 -> run the test else ignore/disable the test
32     public void testConditionJRErange()
33     {
34         System.out.println("Java version is matching and test is exeucted");
35     }
36
37     @Test
38     @EnabledIfSystemProperty(named="java.vm.vendor", matches="Oracle.*")
39 }
```

Writable Smart Insert 46:1:1514

Type here to search 30°C 4:01 PM 1/24/2024

## 12. Meta-Annotations

eclipse-workspace - junitTestScripts/src/test/java/junitTestScripts1/MetaAnnotation.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer X JUnit

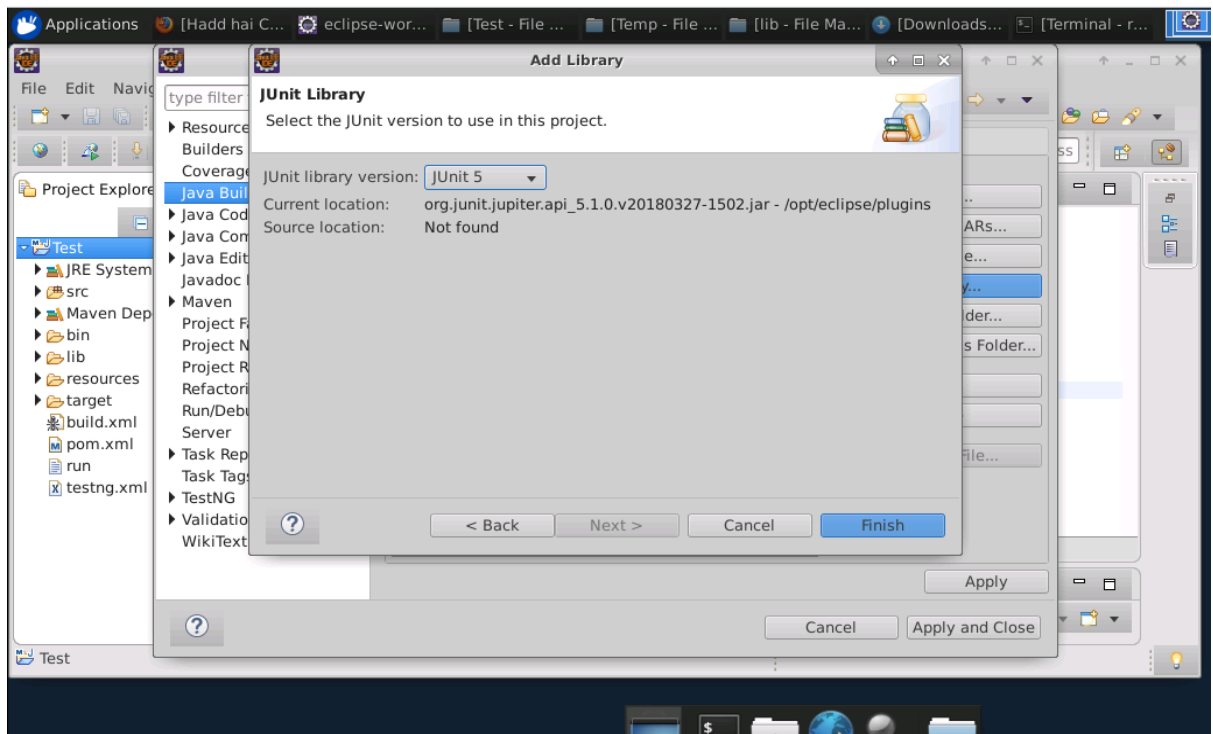
- junitTestScripts1
  - AppTest.java
  - ArgumentConversionexplicitSource.java
  - ArgumentConversionImplicit.java
  - AssertionsTest.java
  - AssumptionsDemo.java
  - BeforeAllAfterAll.java
  - BeforeEachAfterEach.java
  - Calculator.java
  - CustomAnnotationDemo.java
  - DependencyInjection.java
  - DisabledAnnotation.java
  - ExtensionsDemoCondition.java
  - ExtensionsDemoCondition1.java
  - GoogleSearchTest.java
  - IncludeExcludeTags.java
  - JUnit5Runner.java
  - LessonEndProject.java
  - MetaAnnotation.java
  - MydynamicTestDemo2.java
  - MydynamicTests.java
  - Myutils.java
  - OrderAnnotation.java
  - ParameterizedCSVsource.java
  - ParameterizedValueSource.java
  - ParameterizedEnumSource.java
  - ParameterizedTestMethodSource.java
  - TestAnnotationDemo1.java
  - ToStringArgumentConverter.java
- src/test/resources
- Maven Dependencies
- JRE System Library [JavaSE-11]

```
1 package junitTestScripts1;
2
3 import java.lang.annotation.ElementType;
4 import java.lang.annotation.Retention;
5 import java.lang.annotation.RetentionPolicy;
6 import java.lang.annotation.Target;
7 import org.junit.jupiter.api.Tag;
8 import org.junit.jupiter.api.Test;
9 // We have to create our custom Annotation
10 //1. Where will this annotation be used/targeted
11 @Target({ElementType.TYPE, ElementType.METHOD})
12 //2. When will the custom annotation be triggered
13 @Retention(RetentionPolicy.RUNTIME)
14 //3. Tag the annotation
15 @Tag("dev")
16 @Test // our custom annotation is replica of @Test annotation
17 public @interface MetaAnnotation {
18
19 }
20
21
```

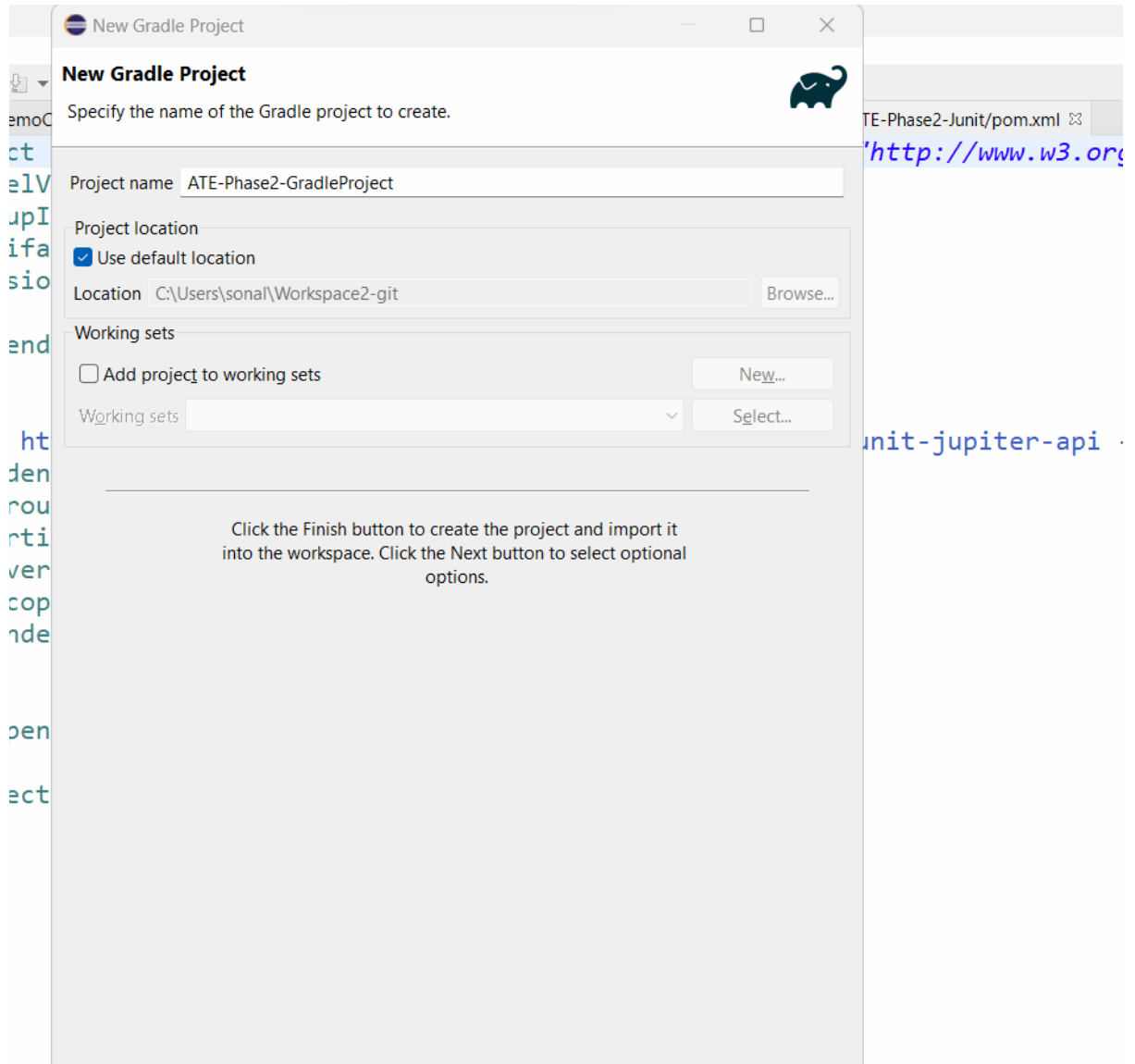
Writable Smart Insert 21:1:656

Type here to search 30°C ENG 4:02 PM IN 1/24/2024

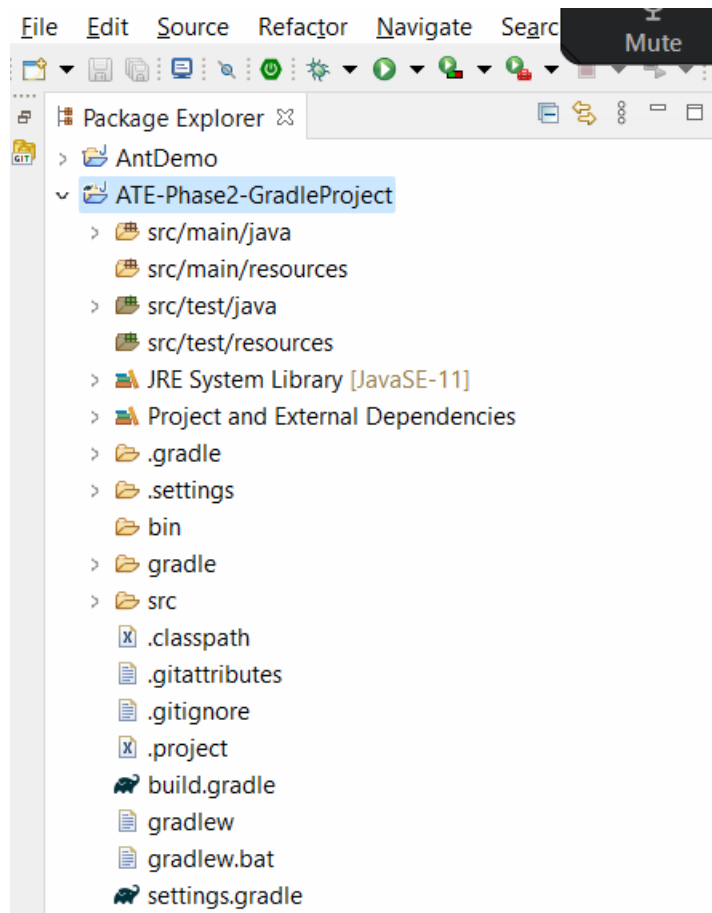
## 13. Running Test from the Console



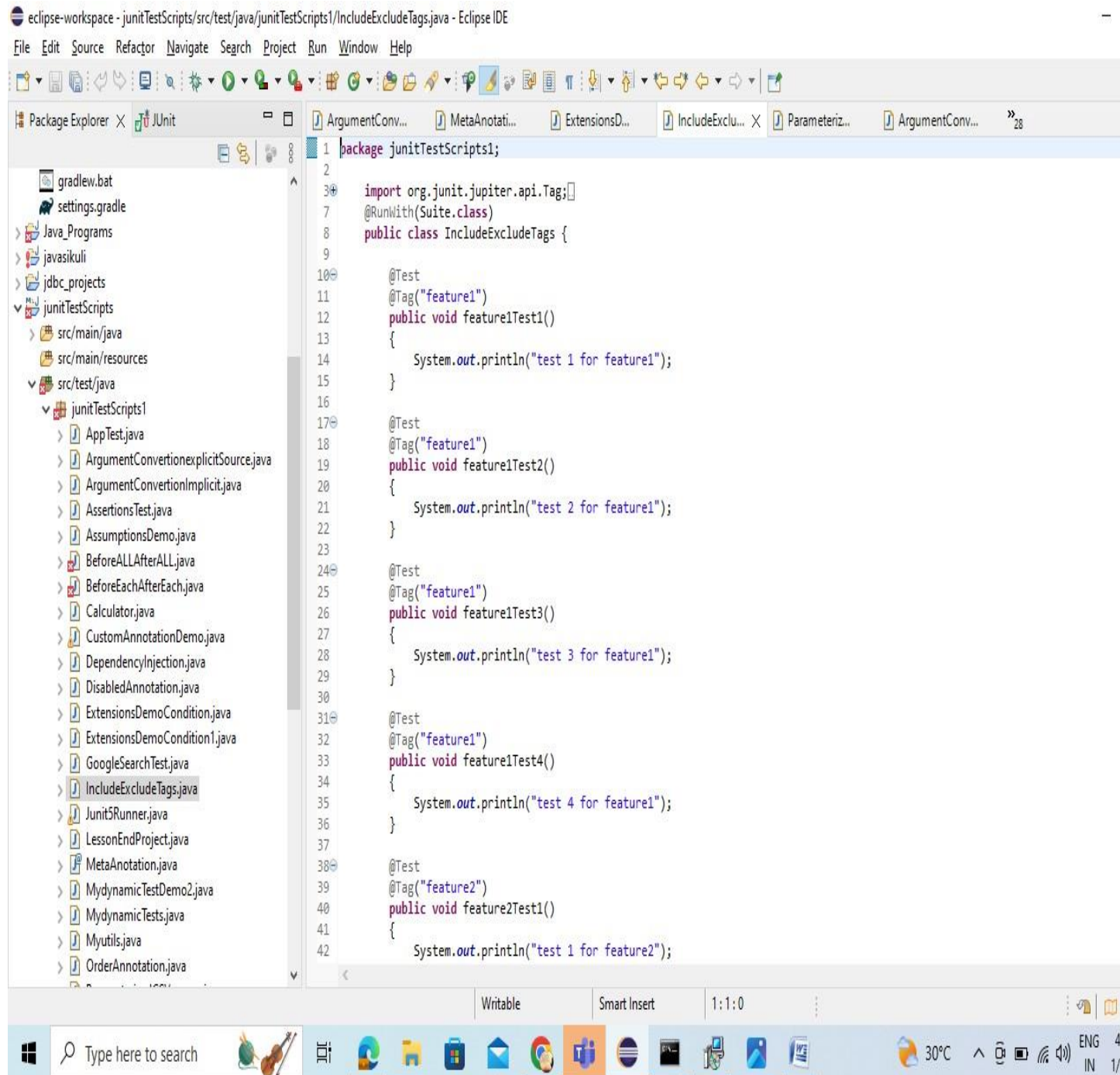
## 14. Running Tests with Gradle







## 16. Include/Exclude Tests with Tags



## 17. Code Conversion

