

# Conditional Tests



# Conditional Tests

We may not need to run all of the tests ....

# Conditional Tests - Use Cases

- Don't run a test because the method to **test is broken** ... and we are waiting on dev team to fix it
- A test should only run for a **specific version of Java** (Java 18) or range of versions (13 - 18)
- A test should **only run on a given operating system**: MS Windows, Mac, Linux
- A test should **only run if specific environment variables or system properties** are set

# Why not just comment the code???

- We could do that ... but then the tests will not be displayed in reports
- Easy to forget about broken tests
- Manual process to enable / disable tests for a given operating system etc ...
- Let's report the tests ... so that management and QA are aware of the issue

# Annotations

Name	Description
@Disabled	Disable a test method
@EnabledOnOs	Enable test when running on a given operating system
...	...

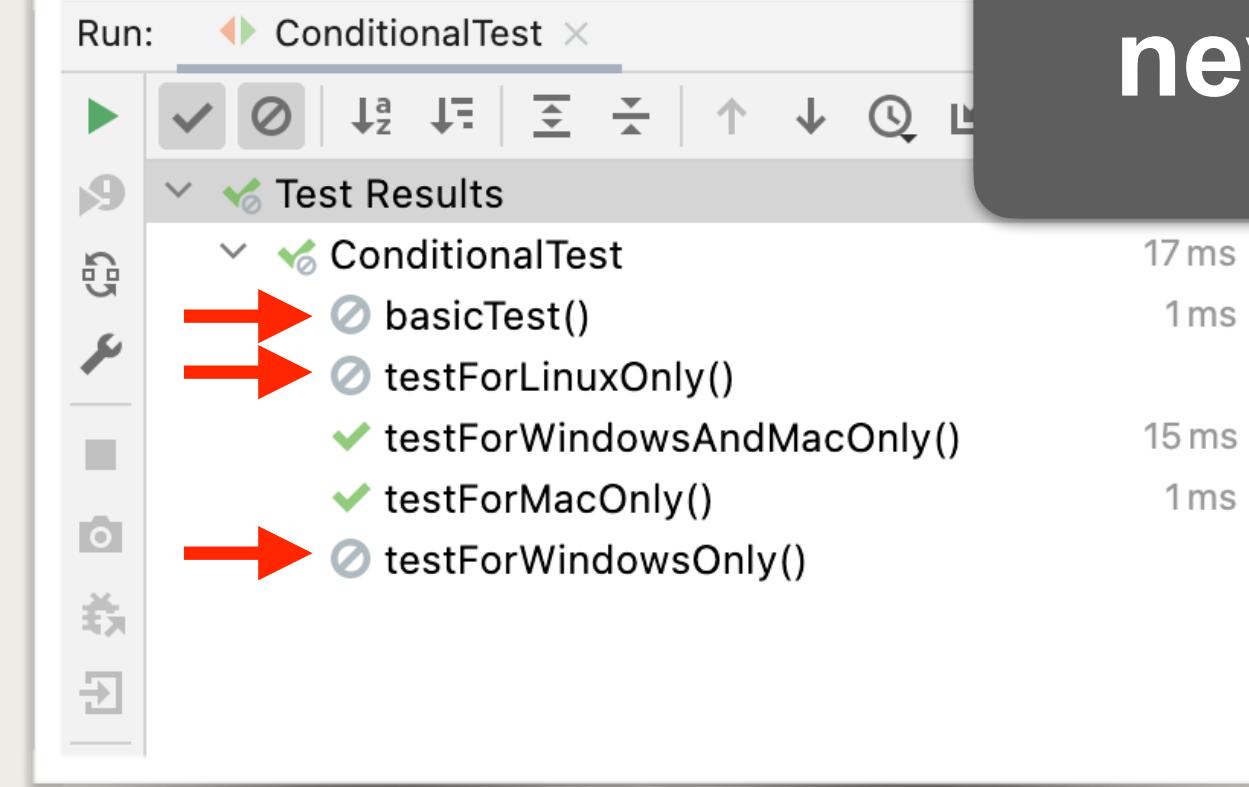
**Annotations can be applied at  
the class level or method level**

# @Disabled and @EnabledOnOs

## ConditionalTest.java

```
class ConditionalTest {  
  
    @Test  
    @Disabled("Don't run until JIRA #123 is resolved")  
    void basicTest() {  
        // execute method and perform assertions  
    }  
  
    @Test  
    @EnabledOnOs(OS.WINDOWS)  
    void testForWindowsOnly() {  
        // execute method and perform assertions  
    }  
  
    @Test  
    @EnabledOnOs(OS.MAC)  
    void testForMacOnly() {  
        // execute method and perform assertions  
    }  
  
    @Test  
    @EnabledOnOs({OS.WINDOWS, OS.MAC})  
    void testForWindowsAndMacOnly() {  
        // execute method and perform assertions  
    }  
  
    @Test  
    @EnabledOnOs(OS.LINUX)  
    void testForLinuxOnly() {  
        // execute method and perform assertions  
    }  
}
```

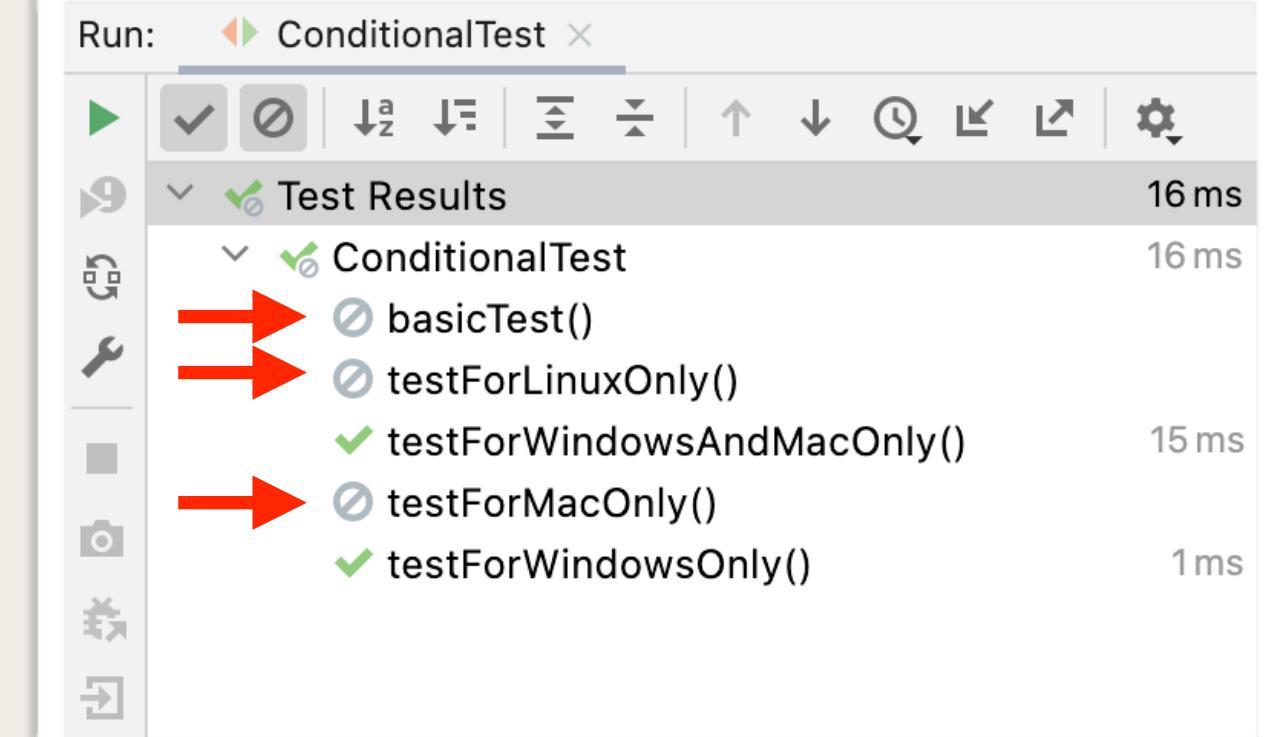
Mac



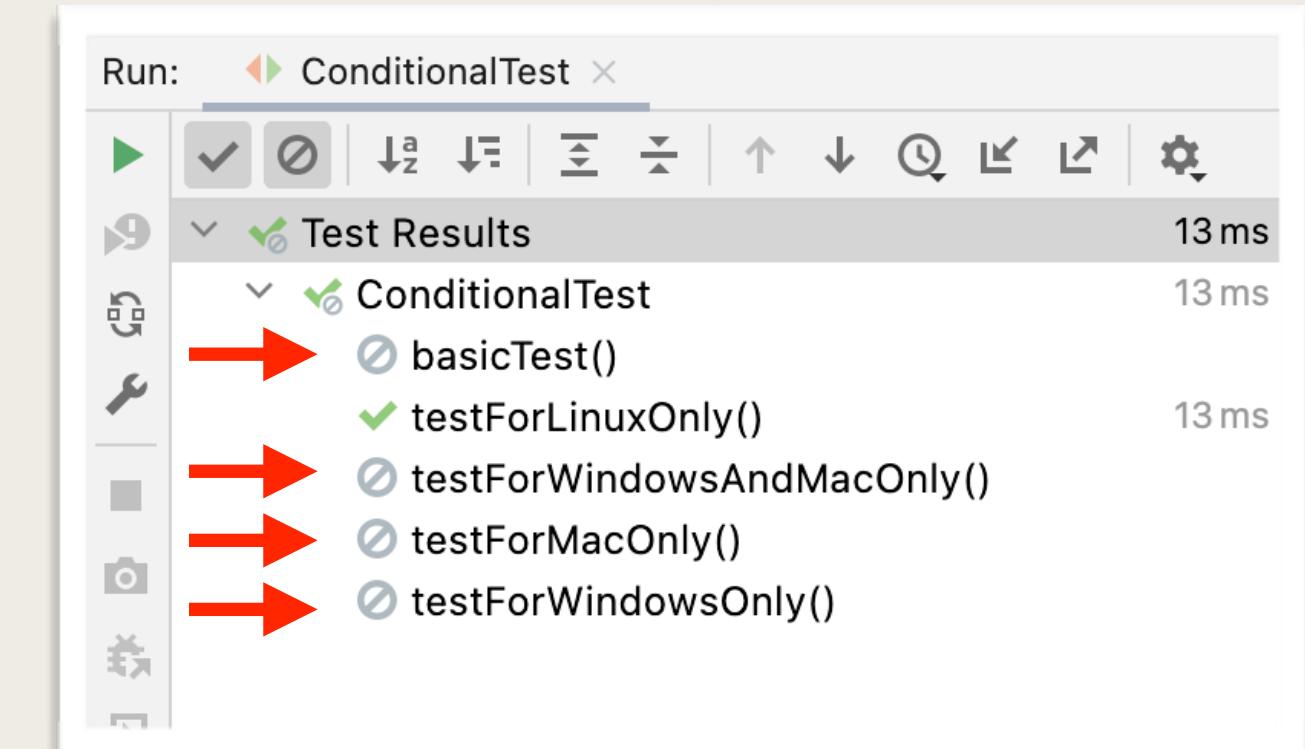
Disabled tests never execute

Disabled tests never execute

MS Windows



Linux



Disabled tests never execute

# Annotations

Name	Description
@EnabledOnJre	Enable test for a given Java version
@EnabledForJreRange	Enable test for a given Java version range
...	...

# @EnabledOnJre

# @EnabledForJreRange

## ConditionalTest.java

```
class ConditionalTest {

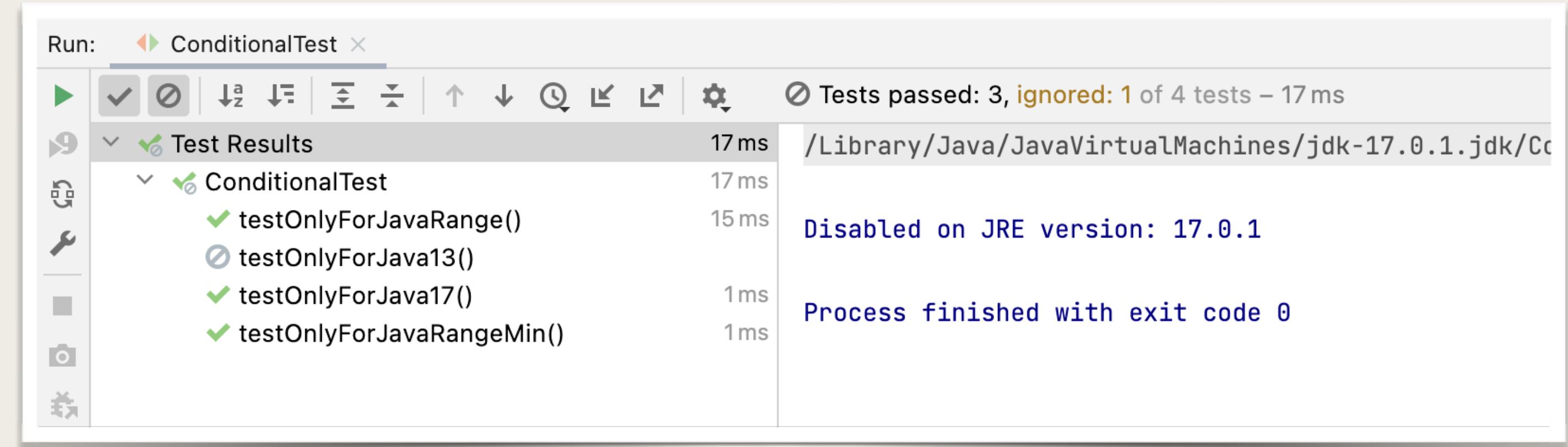
    @Test
    @EnabledOnJre(JRE.JAVA_17)
    void testOnlyForJava17() {
        // execute method and perform assertions
    }

    @Test
    @EnabledOnJre(JRE.JAVA_13)
    void testOnlyForJava13() {
        // execute method and perform assertions
    }

    @Test
    @EnabledForJreRange(min=JRE.JAVA_13, max=JRE.JAVA_18)
    void testOnlyForJavaRange() {
        // execute method and perform assertions
    }

    @Test
    @EnabledForJreRange(min=JRE.JAVA_11)
    void testOnlyForJavaRangeMin() {
        // execute method and perform assertions
    }
}
```

## Running on Java 17



# Annotations

Name	Description
@EnabledIfSystemProperty	Enable test based on system property
@EnabledIfEnvironmentVariable	Enable test based on environment variable
...	...

# @EnabledIfSystemProperty @EnabledIfEnvironmentVariable

-D<propName>=<propValue>

ConditionalTest.java

```
class ConditionalTest {  
  
    @Test  
    @EnabledIfSystemProperty(named="LUV2CODE_SYS_PROP", matches="CI_CD_DEPLOY")  
    void testOnlyForSystemProperty() {  
        // execute method and perform assertions  
    }  
  
    @Test  
    @EnabledIfEnvironmentVariable(named="LUV2CODE_ENV", matches="DEV")  
    void testOnlyForDevEnvironment() {  
        // execute method and perform assertions  
    }  
}
```

Any name

Regular expression

