ParameterizedTest with @EnumSource

This lesson demonstrates use of @EnumSource to pass different arguments to @ParameterizedTest.

We'll cover the following

- @EnumSource

@EnumSource

`@EnumSource` allows us to pass enum constants to `@ParameterizedTest` method.

Let's look at a demo.

Step 1 - Let's assume that we have to write a parameterized test that takes a value of the `Animal` enum as `@EnumSource`. The Pet enum looks as follows.

Animal.java

```java
package com.hubberspot.junit5.

public enum Animal {
	ELEPHANT,
	TIGER,
	DOG,
	CAT,
	MOUSE
}
```

Step 3 - We create a test class by name, `EnumSourceTest.java`.

Step 4 - It contains a test method by name, `testEnumSource`. In order to provide different parameters/values to the same test method, this method is marked as `@ParameterizedTest` instead of `@Test`.

Step 5 - In order to provide different and multiple values through enums. We mark this test method with `@EnumSource` annotation. This annotation takes

`Animal.class` instance.

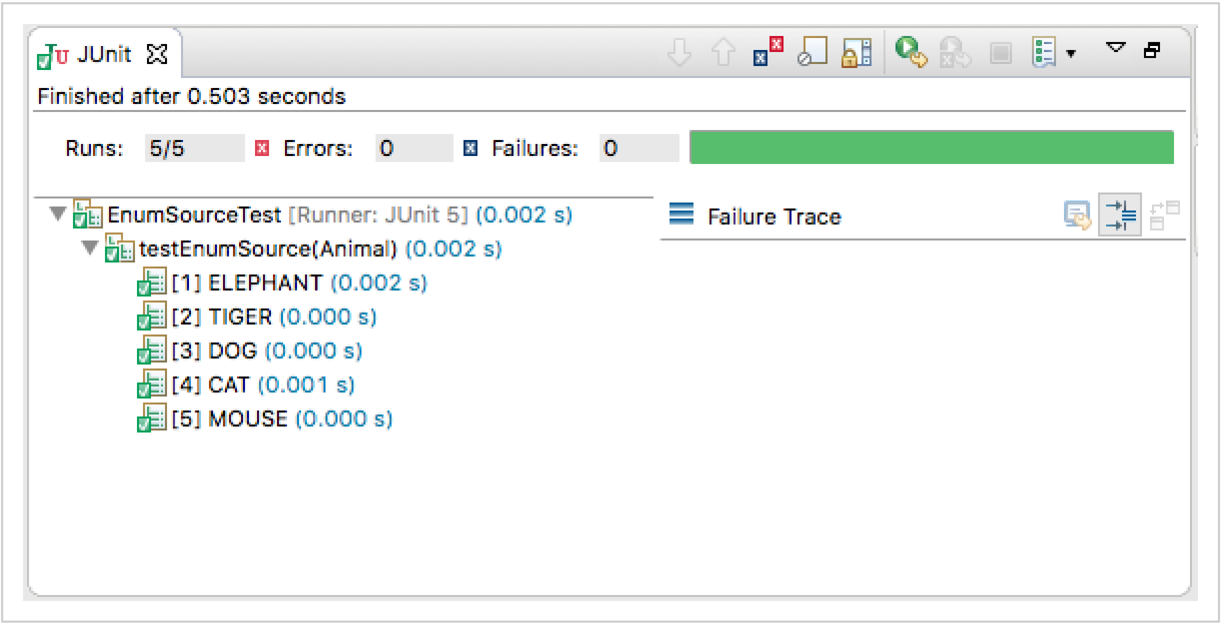
Step 6 - Let's pass Animal enum type to test method. There are 5 animal such as ELEPHANT, TIGER, DOG, CAT, MOUSE, so `@ParameterizedTest` will execute 5 times. In each iteration, it will assert one enum value that it's not null.

Step 7 - Run it as, JUnit Test Case.

Animal.java

EnumSourceTest.java

```
package io.educative.junit5;

import static org.junit.jupiter.api.Assertions.*;

import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.EnumSource;

class EnumSourceTest {

        @ParameterizedTest
        @EnumSource(Animal.class)
        void testEnumSource(Animal animal) {
                assertNotNull(animal);
        }

}
```



```
JUnit
Finished after 0.503 seconds
Runs: 5/5   Errors: 0   Failures: 0

EnumSourceTest [Runner: JUnit 5] (0.002 s)
  testEnumSource(Animal) (0.002 s)
    [1] ELEPHANT (0.002 s)
    [2] TIGER (0.000 s)
    [3] DOG (0.000 s)
    [4] CAT (0.001 s)
    [5] MOUSE (0.000 s)
```

Output of @ParameterizedTest demo

Above image demonstrates the working of `@ParameterizedTest`. As we have provided 5 different enum values, the test case ran 5 times. As all enum values are not null, therefore `assertNotNull` passes for all values passed.

In the next lesson we will be studying parameterized tests with `@MethodSource`.