



- Junit5 annotations
- Custom argument provider for non-primitive inputs

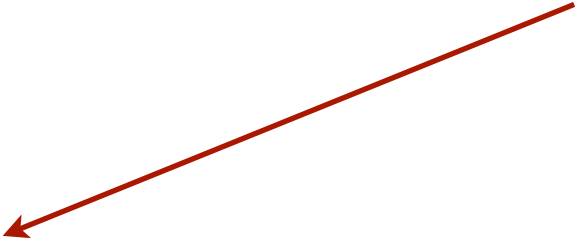
```
class DistanceConverterTest {

    @ParameterizedTest
    @ArgumentsSource(TestInputProvider::class)
    fun `check distance parser`(input: Input) {
        val converter = DistanceConverter()

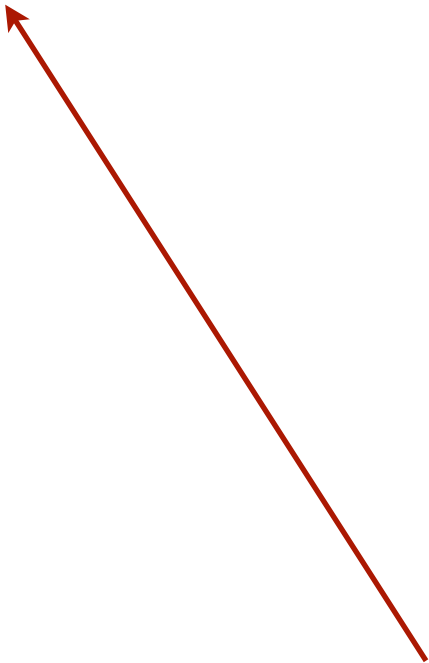
        val actual = converter.parse(input.parameter)
        val expected = input.expected

        assertEquals(actual, expected)
    }

    class TestInputProvider : ArgumentsProvider {
        override fun provideArguments(
            context: ExtensionContext?
        ): Stream<out Arguments> {
            return Stream.of(
                Input(500, 0.5),
                Input(750, 0.8)
            ).map { Arguments.of(it) }
        }
    }
}
```



parameterized test method



# ArgumentsProvider class






# parameterized test method

```
class DistanceConverterTest {  
  
    @ParameterizedTest  
    @ArgumentsSource(TestInputProvider::class)  
    fun `check distance parser`(input: Input) {  
        val converter = DistanceConverter()  
  
        val actual = converter.parse(input.parameter)  
        val expected = input.expected  
  
        assertEquals(actual, expected)  
    }  
  
    class TestInputProvider : ArgumentsProvider {  
        override fun provideArguments(  
            context: ExtensionContext?  
        ): Stream<out Arguments> {  
            return Stream.of(  
                Input(500, 0.5),  
                Input(750, 0.8)  
            ).map { Arguments.of(it) }  
        }  
    }  
}
```

**ArgumentsProvider class**



 Instantiating tests...

