Software Engineering

Continuous Assessment Laboratory 2

Burak Yasar Parlak

Emre TEK

We created unit tests for our test project using Junit 5 and Mokito. For the first test we are using Math.addExact function of java and testing if everything is correct by executing the statement 2 + 2 = 4

```java
import static org.junit.jupiter.api.Assertions.assertEquals;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Assertions;


22 usages
class CalculatorTests {

    @Test
    @DisplayName("2 + 2 = 4")
    void addNumberTest()
    {
        final int expected = 4;
        final int actual = Math.addExact(2, 2);
        assertEquals(expected, actual);
    }
```

This function should give us an exception if one of the parameters exceeds INT_MAX defined in environment (INT Overflow)

On the next part we are testing this exception

```
@Test
@DisplayName("addExact Overflow Exception")
void addNumberExceptionTest()
{
    ArithmeticException thrown = Assertions.assertThrows(ArithmeticException.class, () -> {
        Math.addExact(Integer.MAX_VALUE, 200);
    });
}
```

If the integer value overflows it correctly issues an exception

We also have another test scenario for division which works by dividing 10/2 which should give 5 as result

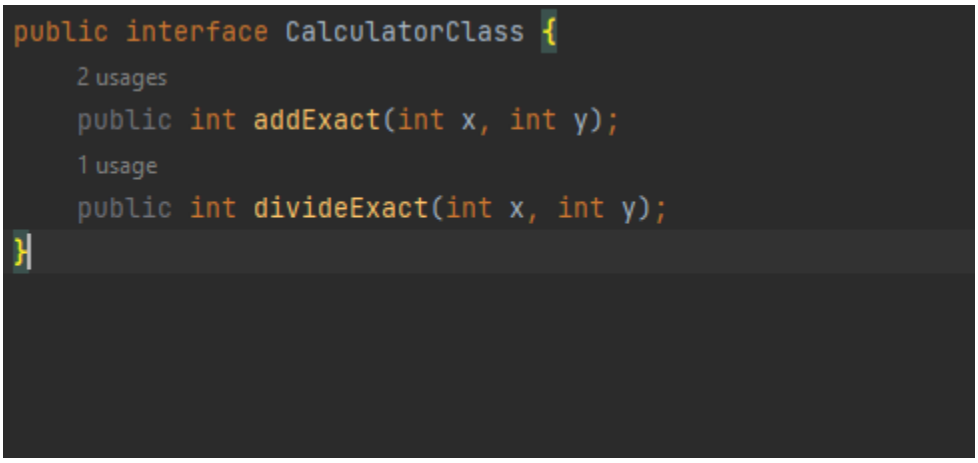
```
@Test
@DisplayName("10 / 2 = 5")
void divideIntegerTest()
{
    final int expected = 5;
    final int actual = Math.divideExact(10, 2);
    assertEquals(expected, actual);
}
```

There is also a exception test for this scenario which checks for divide by zero exception

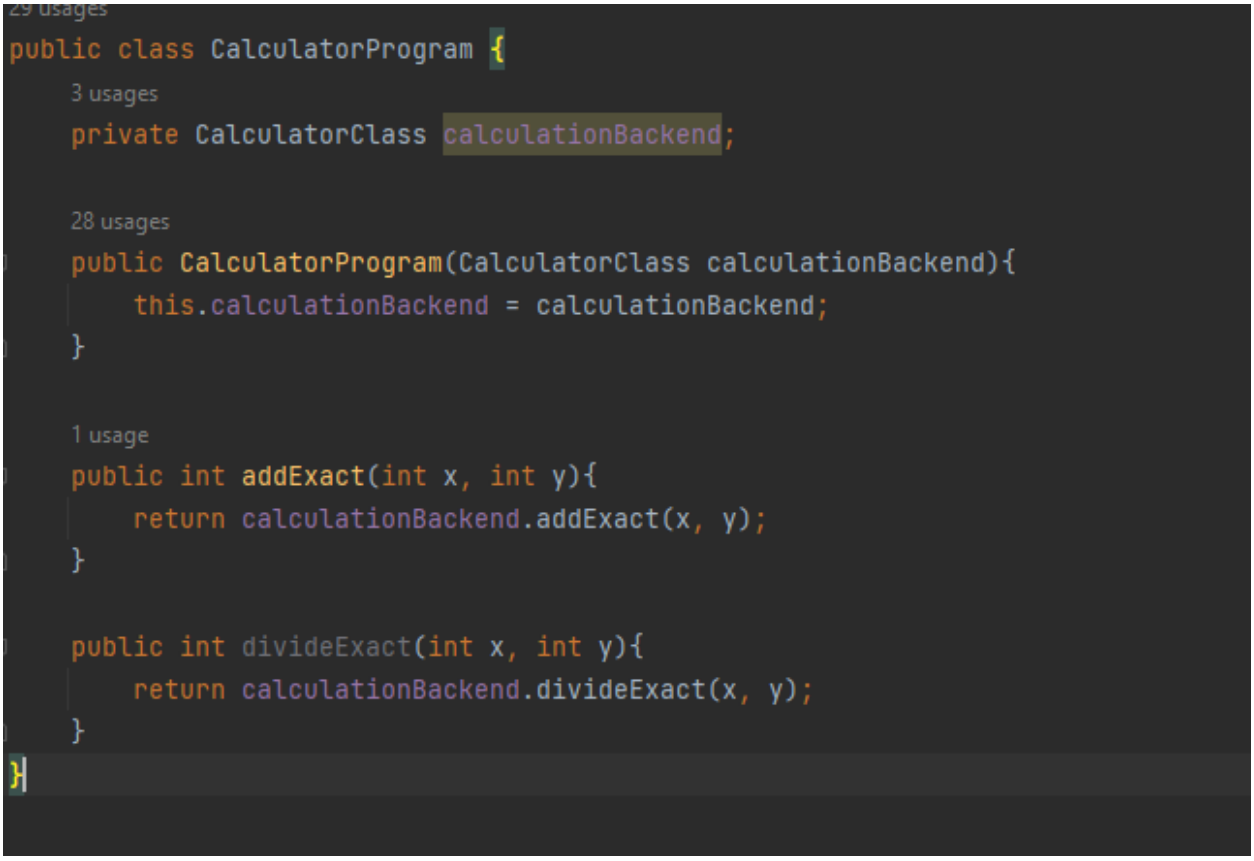
```
@Test
@DisplayName ("divideExact Division by 0 Exception")
void divideByZeroExceptionTest()
{
    ArithmeticException thrown = Assertions.assertThrows(ArithmeticException.class, () -> {
        Math.divideExact(3, 0);
    });
}

}
```

On the next part we used Mockito which is a mock testing platform used in Java project

We have our CalculatorClass here which is an interface without any implemented methods



We also have CalculatorProgram class which uses Calculator interface to add and divide

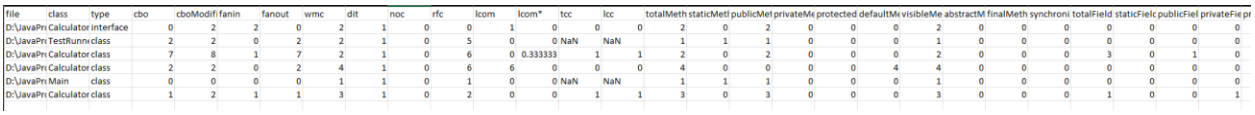


```
29 usages
public class CalculatorProgram {
    3 usages
    private CalculatorClass calculationBackend;

    28 usages
    public CalculatorProgram(CalculatorClass calculationBackend){
        this.calculationBackend = calculationBackend;
    }

    1 usage
    public int addExact(int x, int y){
        return calculationBackend.addExact(x, y);
    }

    public int divideExact(int x, int y){
        return calculationBackend.divideExact(x, y);
    }
}
```

Here is the test function we use to implement mock functionality for this class

```
2 usages
CalculatorProgram calculatorProgram = null;

2 usages
@Mock
CalculatorClass calculationBackend;

@Rule
public MockitoRule rule = MockitoJUnit.rule();

@Before
public void setUp()
{
    calculatorProgram = new CalculatorProgram(calculationBackend);
}

 @Test
 public void addExactTest()
 {
     when(calculationBackend.addExact( x: 2,  y: 2)).thenReturn( t: 4);

     final int expected = 4;
     final int actual = calculatorProgram.addExact( x: 2,  y: 2);
     assertEquals(expected, actual);
 }
}
```

In the last part we have Chidamber Kemerer Java Metrics generated using the tool provided in the document, These measurements could also be found in project root folder.



file	class	type	cbo	cboModifi	fanin	fanout	wmc	dit	noc	rfc	lcom	lcom*	tcc	lcc	totalMeth	staticMetl	publicMet	privateMe	protected	defaultMe	visibleMe	abstractM	finalMeth	synchroni	totalField	staticFiel	publicFiel	privateFie	pr
D:\JavaPr	Calculator	interface	0	2	2	0	2	1	0	0	1	0	0	0	2	0	2	0	0	0	2	0	0	0	0	0	0	0	
D:\JavaPr	TestRunn	class	2	2	0	2	2	1	0	5	0	0	NaN	NaN	1	1	1	0	0	0	1	0	0	0	0	0	0	0	
D:\JavaPr	Calculator	class	7	8	1	7	2	1	0	6	0	0.333333	1	1	2	0	2	0	0	0	2	0	0	0	3	0	1	0	
D:\JavaPr	Calculator	class	2	2	0	2	4	1	0	6	6	0	0	0	4	0	0	0	0	4	4	0	0	0	0	0	0	0	
D:\JavaPr	Main	class	0	0	0	0	1	1	0	1	0	0	NaN	NaN	1	1	1	0	0	0	1	0	0	0	0	0	0	0	
D:\JavaPr	Calculator	class	1	2	1	1	3	1	0	2	0	0	1	1	3	0	3	0	0	0	3	0	0	0	1	0	0	1	

file	class	method	variable	usage
D:\JavaPr	Calculator	addExactT	calculatio	1
D:\JavaPr	Calculator	addExactT	calculator	1
D:\JavaPr	Calculator	setUp/0	calculatio	1
D:\JavaPr	Calculator	setUp/0	calculator	1
D:\JavaPr	Calculator	Calculator	calculatio	3
D:\JavaPr	Calculator	divideExa	calculatio	1
D:\JavaPr	Calculator	addExact/	calculatio	1

file	class	method	construct	line	cbo	cboModifi	fanin	fanout	wmc	rfc	loc	returnsQt	variablesC	paramete	methodsIn	methodsIn	methodsIn	loopQty	comparisc	tryCatchQ	parenther	stringLiter	numbersC	assignmer	mathOper	maxNeste	anonymoi	innerClass	la
D:\JavaPr	Calculator	divideExa	false	3	0	1	1	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D:\JavaPr	Calculator	addExact/	false	2	0	2	2	0	1	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D:\JavaPr	TestRunn	main/1[ja	false	6	2	0	0	0	2	5	7	0	1	1	5	0	0	1	0	0	0	0	0	0	1	0	1	0	0
D:\JavaPr	Calculator	addExactT	false	31	3	2	0	2	1	5	6	0	2	0	5	0	0	0	0	0	0	0	0	6	2	0	0	0	0
D:\JavaPr	Calculator	setUp/0	false	25	2	1	0	1	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
D:\JavaPr	Calculator	divideInte	false	30	2	0	0	0	1	2	5	0	2	0	2	0	0	0	0	0	0	0	1	3	2	0	0	0	0
D:\JavaPr	Calculator	addNumb	false	21	2	0	0	0	1	2	6	0	1	0	2	0	0	0	0	0	0	0	1	1	1	0	1	0	0
D:\JavaPr	Calculator	divideByZ	false	39	2	0	0	0	1	2	6	0	1	0	2	0	0	0	0	0	0	0	1	2	1	0	1	0	0
D:\JavaPr	Calculator	addNumb	false	12	2	0	0	0	1	2	5	0	2	0	2	0	0	0	0	0	0	0	1	3	2	0	0	0	0
D:\JavaPr	Main	main/1[ja	false	2	0	0	0	0	1	1	3	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
D:\JavaPr	Calculator	Calculator	true	4	1	1	1	0	1	0	3	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
D:\JavaPr	Calculator	divideExa	false	12	1	1	0	1	1	1	3	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D:\JavaPr	Calculator	addExact/	false	8	1	2	1	1	1	1	3	1	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

file	class	method	variable	usage
D:\JavaPr	Calculator	divideExa	x	0
D:\JavaPr	Calculator	divideExa	y	0
D:\JavaPr	Calculator	addExact/	x	0
D:\JavaPr	Calculator	addExact/	y	0
D:\JavaPr	TestRunne	main/1[ja	args	0
D:\JavaPr	TestRunne	main/1[ja	result	2
D:\JavaPr	TestRunne	main/1[ja	failure	1
D:\JavaPr	Calculator	addExactT	actual	1
D:\JavaPr	Calculator	addExactT	expected	1
D:\JavaPr	Calculator	divideInte	actual	1
D:\JavaPr	Calculator	divideInte	expected	1
D:\JavaPr	Calculator	addNumb	thrown	0
D:\JavaPr	Calculator	divideByZ	thrown	0
D:\JavaPr	Calculator	addNumb	actual	1
D:\JavaPr	Calculator	addNumb	expected	1
D:\JavaPr	Main	main/1[ja	args	0
D:\JavaPr	Calculator	Calculator	calculatio	2
D:\JavaPr	Calculator	divideExa	x	1
D:\JavaPr	Calculator	divideExa	y	1
D:\JavaPr	Calculator	addExact/	x	1
D:\JavaPr	Calculator	addExact/	y	1