GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

CIP FP BATOI

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

UNIÓ EUROPEA
Fons Social Europeu
L'FSE inverteix en el teu futur

04 de mayo de 2022

Entornos de Desarrollo

Practica 2: JUnit

Por Adrià Jordà Martínez

Practicas enfocadas en la introducción de Test en Java.

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura i Esport

CIP FP BATOI

C/ Societat Unió Musical, 8 - 03802 Alcoi
Tel. 966 52 76 60 - Fax 966 52 76 61
03012165.secret@gva.es
www.cipfpbatoi.es

GOBIERNO DE ESPAÑA
MINISTERIO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL

UNIÓ EUROPEA
Fons Social Europeu
L'FSE inverteix en el teu futur

Índice

A_____1

B_____1

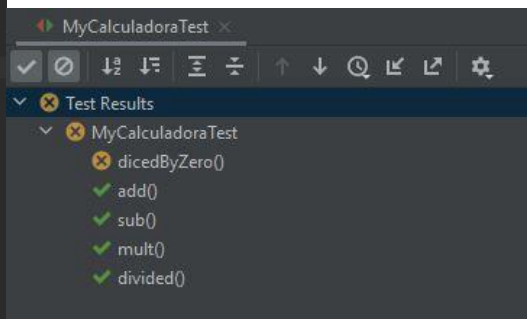
C_____2

D_____2

E_____4

A) Afegeix mètodes de test per a verificar que les operacions: sub(5, 5), mult(5, 5) i div(15, 3)

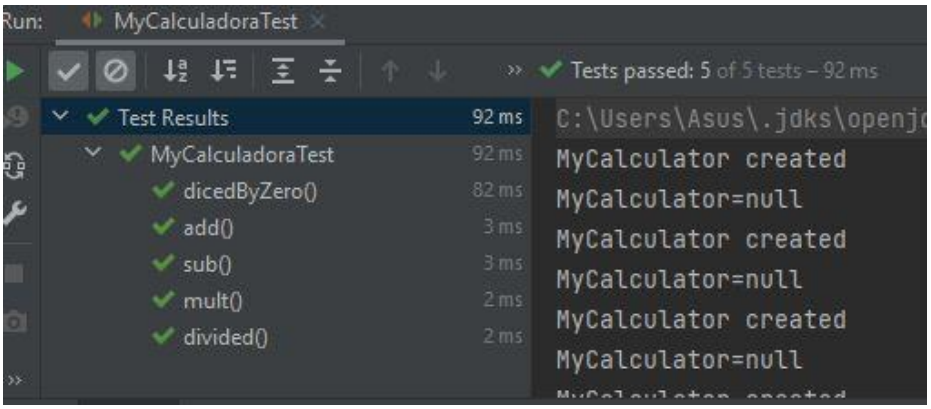
```java
@Test
void sub(){
    int total = 5;
    int sub = 5;
    int expected = 0;
    MyCalculadora myCalculadora = new MyCalculadora();
    assertEquals(expected, myCalculadora.sub(total,sub));
}

@Test
void mult(){
    int total = 5;
    int sub = 5;
    int expected = 25;
    MyCalculadora myCalculadora = new MyCalculadora();
    assertEquals(expected, myCalculadora.multi(total,sub));
}

@Test
void divided(){
    int numerador = 15;
    int denominador = 3;
    int expected = 5;
    MyCalculadora myCalculadora = new MyCalculadora();
    assertEquals(expected, myCalculadora.div(numerador,denominador));
}
```



B)

```java
MyCalculadora myCalculadora;

@BeforeEach
void setUpMyCalculator(){
    myCalculadora = new MyCalculadora();
    System.out.println("MyCalculator created");
}

@Test
void add() {
    int expected = 3;
    assertEquals(expected, myCalculadora.add( a: 1, b: 2));
}

@Test
void sub(){
    int total = 5;
    int sub = 5;
    int expected = 0;
    assertEquals(expected, myCalculadora.sub(total,sub));
}
```

```java
@Test
void divided(){
    int numerador = 15;
    int denominador = 3;
    int expected = 5;;
    assertEquals(expected, myCalculadora.div(numerador,denominador));
}

@Test
void dicedByZero(){
    int numerador = 1;
    int denominador = 0;

    ArithmeticException e = assertThrowsExactly(ArithmeticException.class, ()->{
        myCalculadora.div(numerador, denominador);
    });

    assertEquals( expected: "/ by zero", e.getMessage());
}

@AfterEach
void tearDownMyCalculator(){
    myCalculadora = null;
    System.out.println("MyCalculator=null");
}
```



C)

```
@Test
void dicedByZero2(){
    int numerador = 1;
    int denominador = 0;
    int expected = 0;
    assertEquals(expected, myCalculadora.div(numerador,denominad
}
```

```
MyCalculadoraTest.dicedByZero2
Tests failed: 1 of 1 test – 66 ms
Test Resu 66 ms
MyCa 66 ms
dic 66 ms
C:\Users\Asus\.jdks\openjdk-17.0.1-2\bin\java.exe ...
MyCalculator created
MyCalculator=null

java.lang.ArithmeticException: / by zero

    at daw.proves.MyCalculadora.div(MyCalculadora.java:
    at daw.proves.MyCalculadoraTest.dicedByZero2(MyCalc
    at java.base/java.util.ArrayList.forEach(ArrayList.
    at java.base/java.util.ArrayList.forEach(ArrayList.
```

```
@Test
void dicedByZero2(){
    int numerador = 1;
    int denominador = 0;
    ArithmeticException e = assertThrowsExactly(ArithmeticException.class, ()->{
        myCalculadora.div(numerador, denominador);
    });

    assertEquals( expected: "/ by zero", e.getMessage());
}

@AfterEach
```

```
Tests passed: 1 of 1 test – 86 ms
Test Results 86 ms
MyCalculadoraTest 86 ms
dicedByZero2() 86 ms
C:\Users\Asus\.jdks\openjdk-17.0.1-2\bin\java.exe ...
MyCalculator created
MyCalculator=null

Process finished with exit code 0
```

```
public int div(int a, int b){
    if(b == 0){
        throw new ArithmeticException("Dividido por zero");
    }
    return a / b;
}
```

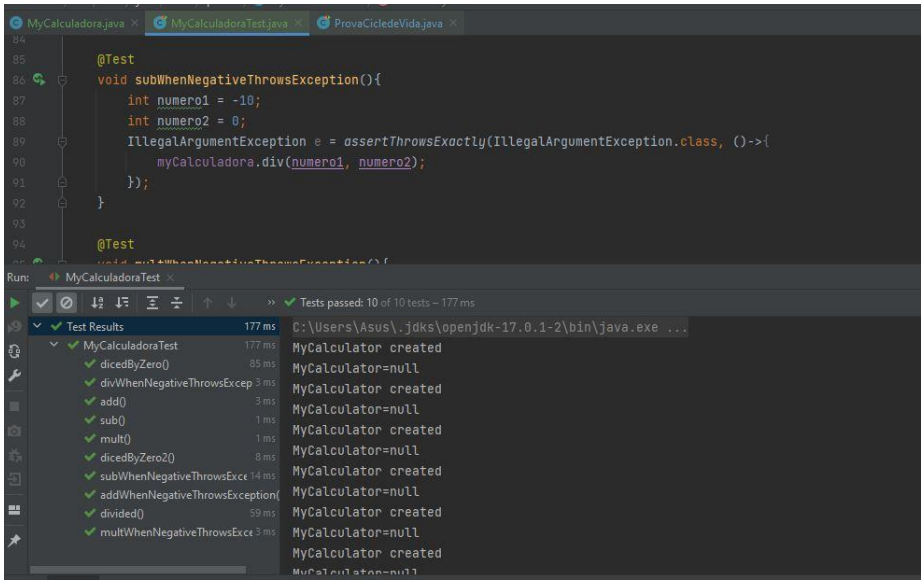
D)

```
@Test
void addWhenNegativeThrowsException(){
    int numero1 = -10;
    int numero2 = 0;
    IllegalArgumentException e = assertThrowsExactly(IllegalArgumentException.class, ()->{
        myCalculadora.div(numero1, numero2);
    });
}

@Test
void subWhenNegativeThrowsException(){
    int numero1 = -10;
    int numero2 = 0;
    IllegalArgumentException e = assertThrowsExactly(IllegalArgumentException.class, ()->{
        myCalculadora.div(numero1, numero2);
    });
}
```

```
@Test
void multWhenNegativeThrowsException(){
    int numero1 = -10;
    int numero2 = 0;
    IllegalArgumentException e = assertThrowsExactly(IllegalArgumentException.class, ()->{
        myCalculadora.div(numero1, numero2);
    });
}

@Test
void divWhenNegativeThrowsException(){
    int numero1 = -10;
    int numero2 = 4;
    IllegalArgumentException e = assertThrowsExactly(IllegalArgumentException.class, ()->{
        myCalculadora.div(numero1, numero2);
    });
}
```

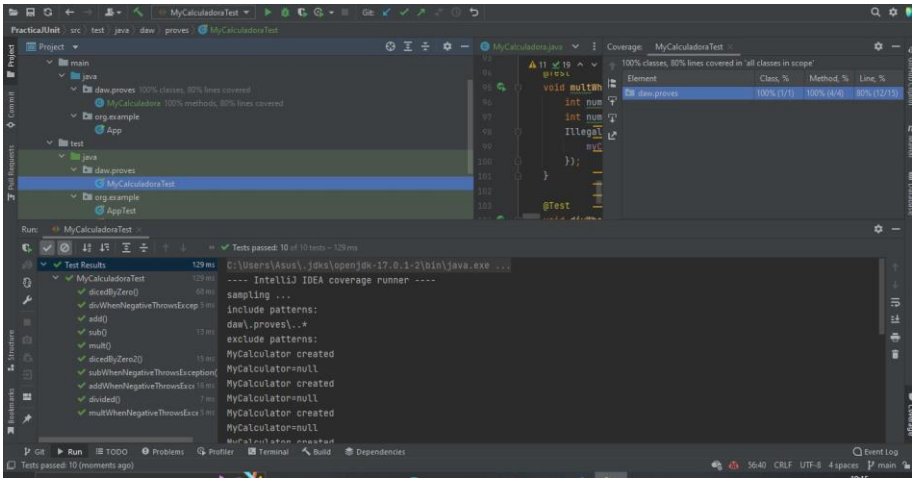
```
MyCalculadoraTest
Tests failed: 4, passed: 6 of 10 tests – 146 ms
Test Results 146 ms
MyCalculadoraTest 146 ms
dicedByZero() 67 ms
divWhenNegativeThrowsExcep 8 ms
add() 16 ms
sub() 7 ms
mult() 5 ms
dicedByZero2() 5 ms
subWhenNegativeThrowsException(
addWhenNegativeThrowsExcr 37 ms
divided()
multWhenNegativeThrowsExcr 6 ms
C:\Users\Asus\.jdks\openjdk-17.0.1-2\bin\java.exe ...
MyCalculator created
MyCalculator=null
MyCalculator created
MyCalculator=null

org.opentest4j.AssertionFailedError: Expected java.lang.IllegalArgumentException to be thrown, but nothing was thrown.
<3 internal lines>
    at daw.proves.MyCalculadoraTest.divWhenNegativeThrowsException(MyCalculadoraTest.java:107) <31 internal lines>
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <9 internal lines>
    at java.base/java.util.ArrayList.forEach(ArrayList.java:1511) <27 internal lines>
```

```java
public int add(int a, int b){
    if (a < 0 || b < 0){
        throw new IllegalArgumentException();
    }
    return a +b;
}

public int sub(int a, int b){
    if (a < 0 || b < 0){
        throw new IllegalArgumentException();
    }
    return a - b;
}

public int multi(int a, int b){
    if (a < 0 || b < 0){
        throw new IllegalArgumentException();
    }
    return a * b;
}

public int div(int a, int b){
    if (a < 0 || b < 0){
        throw new IllegalArgumentException();
    }
    if(b == 0){
        throw new ArithmeticException("Dividido por zero");
```

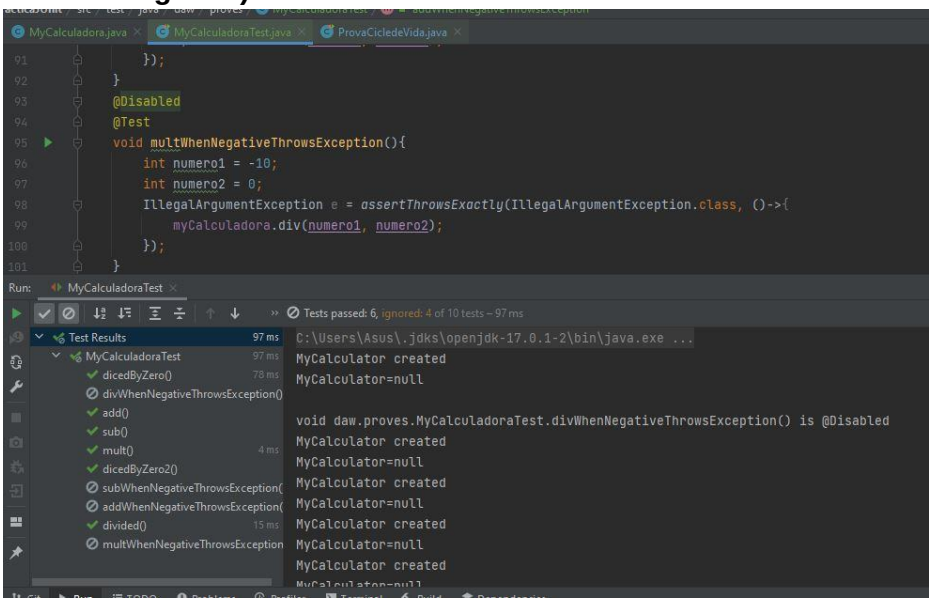


E)

- Executa les proves amb cobertura per a comprovar que les proves que hem programat cobreixen el 100%: classes, mètodes i línies de codi:



Ara, afegeix l'anotació per inhabilitar (seran ignorats) els 4 mètodes de test que has afegit en l'apartat D (els tests dels números negatius) i executa el test de la classe amb cobertura.



Està tot el codi cobert? En quin percentatge? Fes captura del resultat i del codi font.