

```

/*-----
File name      : MatrixTest.java
Author(s)     : Kevin Farine, Timothée Van Hove
Created        : 3 nov. 2022
Description    : Test program for the Matrix class
Remark(s)     : Use "mvn clean test" command to launch the test
JDK           : OpenJDK Runtime Environment Temurin-17.0.5+8 (build 17.0.5+8)
-----*/

package ch.heigvd.poo.labo5;

import ch.heigvd.poo.labo5.Matrix.Matrix;
import ch.heigvd.poo.labo5.operations.*;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertThrows;

public class MatrixTest {

    @Test
    public void testNominalCase() {
        Matrix m1 = new Matrix(new int[][]{
            {1, 3, 1, 1},
            {3, 2, 4, 2},
            {1, 0, 1, 0}}, 5);

        Matrix m2 = new Matrix(new int[][]{
            {1, 4, 2, 3, 2},
            {0, 1, 0, 4, 2},
            {0, 0, 2, 0, 2}}, 5);

        Matrix resultAdd = new Matrix(new int[][]{
            {2, 2, 3, 4, 2},
            {3, 3, 4, 1, 2},
            {1, 0, 3, 0, 2}}, 5);

        Matrix resultSub = new Matrix(new int[][]{
            {0, 4, 4, 3, 3},
            {3, 1, 4, 3, 3},
            {1, 0, 4, 0, 3}}, 5);

        Matrix resultMult = new Matrix(new int[][]{
            {1, 2, 2, 3, 0},
            {0, 2, 0, 3, 0},
            {0, 0, 2, 0, 0}}, 5);

        assertEquals(resultAdd.toString(),
            m1.executeOperation(m2, new Addition()).toString());

        assertEquals(resultSub.toString(),
            m1.executeOperation(m2, new Subtraction()).toString());

        assertEquals(resultMult.toString(),
            m1.executeOperation(m2, new Multiplication()).toString());
    }

    @Test
    public void testChainOperations() {
        Matrix m1 = new Matrix(new int[][]{
            {1, 1, 1},
            {2, 2, 2},
            {3, 3, 3}}, 10);

        Matrix m2 = new Matrix(new int[][]{
            {1, 1, 1, 1, 1},
            {1, 1, 1, 1, 1},
            {1, 1, 1, 1, 1}}, 10);
    }
}

```

```
Matrix result =
    new Matrix(new int[][]{
        {3, 3, 3, 2, 2},
        {4, 4, 4, 2, 2},
        {5, 5, 5, 2, 2}}, 10);

m1 = m1.executeOperation(m2, new Addition())
    .executeOperation(m2, new Addition());

assertEquals(result.toString(), m1.toString());
}

@Test
public void testAdditionWithDifferentModulus() {
    assertThrows(RuntimeException.class,
        () -> new Matrix(3, 4, 5).executeOperation(
            new Matrix(3, 4, 6), new Addition()));
}

@Test
public void testSubtractionWithDifferentModulus() {
    assertThrows(RuntimeException.class,
        () -> new Matrix(1, 4, 1).executeOperation(
            new Matrix(10, 11, 6), new Subtraction()));
}

@Test
public void testMultiplicationWithDifferentModulus() {
    assertThrows(RuntimeException.class,
        () -> new Matrix(100, 200, 1000).executeOperation(
            new Matrix(66, 1, 1001), new Multiplication()));
}

@Test
public void testConstructionWithValuesHigherThanModulus() {
    assertThrows(RuntimeException.class, () ->
        new Matrix(new int[][]{{6}, {0}, {3}}, 5));
}

@Test
public void testConstructionWithIrregularMatrix() {
    assertThrows(RuntimeException.class, () ->
        new Matrix(new int[][]{{6, 3, 1}, {0}, {3, 1}}, 5));
    assertThrows(RuntimeException.class, () ->
        new Matrix(new int[][]{{6}, {0}, {3, 1}}, 5));
}

@Test
public void testConstructionWithNullArray() {
    assertThrows(RuntimeException.class, () -> new Matrix(null, 5));
}

@Test
public void testConstructionWithArrayContainingNullRow() {
    assertThrows(RuntimeException.class, () -> new Matrix(
        new int[][]{{0, 0, 2, 0, 0}, {0, 0, 2, 0, 0}, null}, 5));
    assertThrows(RuntimeException.class, () -> new Matrix(
        new int[][]{null, {0, 0, 2, 0, 0}, {0, 0, 2, 0, 0}}, 5));
}

@Test
public void testConstructionWith0Row() {
    assertThrows(RuntimeException.class, () -> new Matrix(0, 2, 2));
}

@Test
public void testConstructionWith0Column() {
```

```
        assertThrows(RuntimeException.class, () -> new Matrix(2, -1, 2));
    }

    @Test
    public void testConstructionWithNegativeAnd0Modulus() {
        assertThrows(RuntimeException.class, () -> new Matrix(3, 3, -1));
    }

    @Test
    public void testConstrictionWith0Modulus(){
        assertThrows(RuntimeException.class, () ->
            new Matrix(new int[][]{{6}, {0}, {3}}, 0));
    }
}
```