

EJERCICIO 1

Se han realizado pruebas de la clase sumar y de la clase media. No hay pruebas notOK, porque son metodos que solo pueden acertar, por lo tanto con tal de que los test OK funcione, los test estaran completos. Por otro lado el metodo de inserir valores funciona ya que es atraves de este por donde si insertan los valores y al dar los metodos media y suma correctos quiere decir que este tambien funciona bien. Como solo pide el maximo de numeros, es el unico filtro que esta hecho.

```
public class Calculo {

    private static final int MAX_NUM = 4;
    private double suma = 0.0;
    private double media = 0.0;
    private double[] numeros = new double[MAX_NUM];

    public double[] inserirValores(double[] valor) {
        for(int i = 0; i < this.numeros.length; i++)
            if (valor.length <= this.numeros.length)
                this.numeros[i] = valor[i];
            else
                i = MAX_NUM;
        return numeros;
    }

    public double sumar() {
        double suma = 0.0;
        for (int k = 0; k < numeros.length; k++)
            suma += numeros[k];
        return suma;
    }

    public double media() {
        media = this.sumar() / this.numeros.length;
        return media;
    }

    public double getSuma() {
        return suma;
    }

    public double getMedia() {
        return media;
    }
}

public class CalculoApp {

    public static void main(String[] args) {
```

```

        double[] valores = {1,1,1,1};

        //prueba de media
        Calculo calculo = new Calculo();
        calculo.inserirValores(valores);
        double esperado1 = (1 + 1 + 1 + 1) / 4;
        if(esperado1 == calculo.media())
            System.out.println("Prueba de media correcta");
        else
            System.out.println("Prueba de media erronea");

        //prueba de suma
        double esperado2 = (double) 1 + 1 + 1 + 1;
        calculo.inserirValores(valores);
        if(esperado2 == calculo.sumar())
            System.out.println("Prueba de suma correcta");
        else
            System.out.println("Prueba de suma erronea");

    }

}

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

import org.junit.jupiter.api.Test;

class CalculoTest {

    @Test
    void mediaOK() {
        double [] valores = {1, 1, 1, 1};
        double esperado = (double) (1 + 1 + 1 + 1) / 4;
        Calculo calculo = new Calculo();
        calculo.inserirValores(valores);
        assertEquals(esperado, calculo.media());
    }
    @Test
    void sumarOK() {
        double [] valores = {1, 1, 1, 1};
        double esperado = (double) 1 + 1 + 1 + 1;
        Calculo calculo = new Calculo();
        calculo.inserirValores(valores);
        assertEquals(esperado, calculo.sumar());
    }
}

```

EJERCICIO 2

```

public class Clienta {

    private String dni;

```

```

private double saldo;

public Clienta(String dni, double saldo) {
    this.dni = dni;
    this.saldo = saldo;
}

public String getDni() {
    return dni;
}

public double getSaldo() {
    return saldo;
}

public void setSaldo(double saldo) {
    this.saldo = saldo;
}

@Override
public String toString() {
    return "Clienta{" + "dni='" + dni + '\'' + ", saldo= " + saldo +
}';
}
}

```

```

public class Transferencia {

    private boolean ok = false;
    private static final String OK = "EXITO";
    private static final String NOTOK = "NO REALIZADA";
    private final Clienta orixe;
    private final Clienta destino;

    public Transferencia(Clienta orixe, Clienta destino) {
        this.orixe = orixe;
        this.destino = destino;
    }

    public String mensaxeok() {
        if(this.ok)
            return OK;
        else
            return NOTOK;
    }

    public boolean transferencia(double cantidade) {
        if(this.orixe.getSaldo() >= cantidade) {
            this.orixe.setSaldo(this.orixe.getSaldo() - cantidade);
            this.destino.setSaldo(this.destino.getSaldo() +
cantidade);
            ok = true;

```

```

        return ok;
    } else
        ok = false;
        return ok;
}

```

```

public class TransferenciaApp {

    public static void main(String[] args) {

        Clienta orixe = new Clienta("1A", 100);
        Clienta destino = new Clienta("2B", 200);

        Transferencia transferencia = new Transferencia(orixe, destino);
        double cantidad = 200;
        boolean ok = transferencia.transferencia(cantidad);
        String prueba = transferencia.mensaxeok();
        System.out.println(prueba);

        cantidad = 50;
        ok = transferencia.transferencia(cantidad);
        prueba = transferencia.mensaxeok();
        System.out.println(prueba);

    }
}

```

```

class TransferenciaTest {

    @Test
    void transferenciaOKTest() {
        Clienta orixe = new Clienta("1A", 100);
        Clienta destino = new Clienta("2B", 200);
        Transferencia transferencia = new Transferencia(orixe, destino);
        double cantidad = 200;
        boolean esperado = false;
        assertEquals(esperado, transferencia.transferencia(cantidad));
    }

    @Test
    void transferenciaNotOKTest() {
        Clienta orixe = new Clienta("1A", 100);
        Clienta destino = new Clienta("2B", 200);
        Transferencia transferencia = new Transferencia(orixe, destino);
        double cantidad = 80;
        boolean esperado = true;
        assertEquals(esperado, transferencia.transferencia(cantidad));
    }
}

```

}