

EXAMEN 3ª EV - DAM

Atento al formato de entrega de cada uno de los ejercicios. Cualquiera de ellos que no esté entregado en el formato requerido **no** será puntuado.

En las preguntas de tres opciones, tres respuestas incorrectas restan un acierto. En aquellas con dos opciones, dos respuestas erróneas restan una acertada.

PARTE 2 – PRUEBAS UNITARIAS

EJERCICIO 1 (3 puntos)

A partir de la siguiente de la siguiente clase en Java, crea las pruebas unitarias para cada uno de los métodos, incluido el constructor, tal y como se detalla posteriormente:

```
public class operacionesNumeros {
    private java.util.ArrayList<Integer> numeros; // referencia a la lista
    de números
    /**
     * Constructor de operacionesNumeros.
     * @param datos arreglo con los números para la lista
     */
    public operacionesNumeros(int[] datos) {
        if (datos == null || datos.length == 0) { // Verificamos que la
        lista tenga valores
            throw new IllegalArgumentException();
        }
        this.numeros = new java.util.ArrayList<>();
        for (int elemento : datos) {
            numeros.add(elemento);
        }
    }

    /**
     * @return el número máximo de la lista.
     */
    public int getMaximo() {
        int max = Integer.MIN_VALUE;
        for (int elemento : numeros) {
            if (elemento > max) {
                max = elemento;
            }
        }
        return max;
    }
}
```

```
/**
 * @return el número mínimo de la lista.
 */
public int getMinimo() {
    int min = Integer.MAX_VALUE;
    for (int elemento : numeros) {
        if (elemento < min) {
            min = elemento;
        }
    }
    return min;
}

/**
 * @return la suma total de los números de la lista.
 */
public int getSuma() {
    int suma = 0;
    for (int elemento : numeros) {
        suma += elemento;
    }
    return suma;
}

/**
 * Retorna el índice de un número buscado.
 *
 * @param numero Número buscado.
 * @return Retorna la posición de un número dentro de la lista.
 * @throws java.util.NoSuchElementException Si el número no existe en
la lista.
 */
public int getIndiceDe(int numero) throws
java.util.NoSuchElementException {

    int pos = 0;
    for (int elemento : numeros) {
        if (elemento == numero) {
            return pos;
        }
        pos++;
    }
    throw new
java.util.NoSuchElementException(String.valueOf(numero)); // El número no
existe, lanzamos la excepción
}
}
```

Establece los siguientes casos de prueba:

1. **Constructor** (1,5 puntos)
 - a. con un null
 - b. con una matriz vacía
 - c. con la matriz 'prueba'

Integer[] prueba = new Integer [] {10, 1, 20, 2, 25, 65, 125, 12, 31, 85, 78, 31, 89, 67, 95, 98};

2. **getMaximo** (0,25 puntos)
 - a. Comprueba que arroja el número máximo de la matriz.
3. **getMinimo** (0,25 puntos)
 - a. Comprueba que arroja el número mínimo de la matriz.
4. **getSuma** (0,25 puntos)
 - a. Comprueba que la suma total es la correcta.
5. **getIndiceDe** (0,75 puntos)
 - a. Prueba que con el número 2 se devuelve el número correcto
 - b. Prueba con el número 169, que no se encuentra en la lista.

Recuerda que debe pasar todos los test para que las pruebas unitarias estén diseñadas de forma correcta, puesto que los métodos son correctos.

En el caso de tener que devolver una excepción, será correcto si se lanza esa excepción.

EN LA TAREA DEL EXAMEN TIENES QUE ENTREGAR EN ENLACE DEL REPOSITORIO DONDE HAYAS GUARDADO EL PROYECTO.