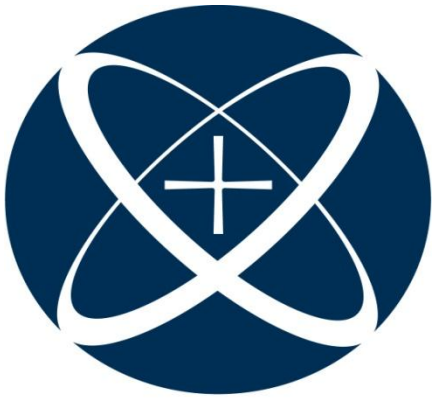
Instituto Tecnológico y de Estudios Superiores de Occidente – ITESO



ITESO

Universidad Jesuita de Guadalajara

Materia: Programación Orientada a Objetos

Maestro: Hugo Iván Piza Dávila

TAREA 1

Sesión: 6

Fecha: 09 de Septiembre el 2025

Temas: Condicionales
Control de Flujo
Try Catch
Arrays

Autor: Alor Santiago Oscar Alberto

PROBLEMA 1

Descripción:

Implementa un método splitArray que reciba un arreglo de números enteros de 16 bits y que devuelva una matriz de dos filas, tal que la primera contenga a todos los números positivos impares del arreglo, y la segunda contenga a los positivos pares, en el orden en que aparecen en el arreglo. No uses estructuras de datos adicionales: sólo el arreglo recibido y la matriz a devolver.

Código fuente:

Se crea una función que devuelve un short[][] y recibe un arreglo de una dimensión de igual manera de tipo short. Después se inspecciona el arreglo para contar cuantos numero pares positivos e impares positivos existen por separados y se guardan en una variable aplicando un for each, en cada iteración se aumenta +1 el valor. Esto sirve para determinar el tamaño de la columna de la matriz:

```
1  public class TareaDos {
2                          static short[][] splitArray(short[] arrayOne){
3                          int even = 0;
4                          int odd = 0;
5                          for(int value : arrayOne){
6                                  if(value % 2 == 0 && value > 0) {
7                                          even += 1;
8                                  }else if (value % 2 != 0 && value > 0) {
9                                          odd += 1;
10                                 }
11                         }
```

Se declara un array de dos dimensiones, las filas se saben que son dos pero las columnas son diferentes, ahí es donde usamos el resultado de "odd" y "even" calculados anteriormente para el tamaño de cada columna en cada dimensión:

```
1                          short twoArrays [][] = new short[2][];
2                          twoArrays[0] = new short[odd];
3                          twoArrays[1] = new short[even];
```

A continuación, se recorre por segunda vez la el arreglo recibido pero esta vez aplicando condiciones mas especificas, se pudo usar un for each quizás pero de esta manera se muestra la implementación de ambas. Aquí nótese un "d1" y "d2" los cuales ayudaran a moverse en las columnas, empiezan en 0 cada una y solo aumentan +1 cuando las condiciones se cumplen, en este caso se revisa el residuo sea 0 (par) o 1 (impar) y ambas evalúan que sea positivo al ser mayor que cero, finalmente se retorna el array:

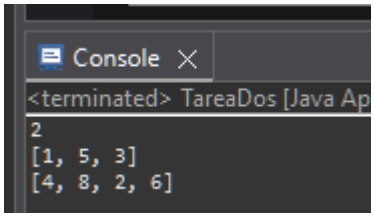
```
1                          int d1 = 0;
2                          int d2 = 0;
```

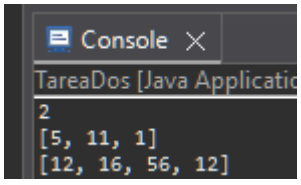
```
3                       // If number is odd, then save it in first row and add a
4 column
5                       // If number is even, then save on second row and add
6 value for column
7                       for(int i = 0; i < arrayOne.length; i++) {
8                               if(arrayOne[i] % 2 != 0 && arrayOne[i] > 0) {
9                                       twoArrays[0][d1] = arrayOne[i];
10                                      d1 += 1;
11                              }
12                              else if (arrayOne[i] > 0){
13                                      twoArrays[1][d2] = arrayOne[i];
14                                      d2 += 1;
15                              }
16
17                      }
                                return twoArrays;
                        }
```

Ahora dentro de main, se declara el arreglo en el código con una serie de números random escogidos por el programador, después se invoca ala función pasando el arreglo y se imprimen los valores de cada fila y sus respectivas columnas (arreglos) usando solo el "Arreglo maestro" como referencia:

```
1       public static void main(String[] args) {
2               /* Part 1 of the homework: splitArrays*/
3               short[] arrayOne = {4, -3, 1, -2, 0, 5, 8, 2, -7, 3, 6};
4               short [][] twoArrays = splitArray(arrayOne);
5               System.out.println(twoArrays.length);
6               System.out.println(Arrays.toString(twoArrays[0]));
7               System.out.println(Arrays.toString(twoArrays[1]));
```

Ejecución:

- short[] arrayOne = {4, -3, 1, -2, 0, 5, 8, 2, -7, 3, 6};
  - 

- short[] arrayOne = {12, -34, 16, -28, 5, -5, 56, 11, 12, 1};
  - 

PROBLEMA 2

EXP:734706

Descripción:

Pregunta al usuario ¿Cuántas calificaciones vamos a capturar?
• Si no escribe un número entero correcto → "Cantidad no válida. Gracias por participar"
• Solicitas cada una de las calificaciones y las almacenas en un arreglo de números reales. Por cada calificación mal capturada, ya sea un número no válido, o un valor fuera del rango [0..10] → "Calificación no válida. Se asignará 0". (try/catch te puede servir)
• Al final despliegas la mediana. Para ello, ordena el arreglo y obtienes el elemento de la mitad.

Código fuente:

Desde el mismo método main, se declara "numCalf" como entero, el cual solo servirá como el tamaño del arreglo, o en este caso, el numero de calificaciones a escribir, después de ello se usa un try/catch para recibir el numero. Aquí si el formato no es correcto el "catch" toma el error y avisa al usuario que no escribió un numero y termina el programa:

```
                /* Part 2  of the homework: Median*/
                int numCalf = 0;
                // Confirmation that the number is expected, if fails then close
the program, finish it.
                try {
                        numCalf =
Integer.parseInt(JOptionPane.showInputDialog("Dame el numero de calificaciones a
insertar"));
                }catch(NumberFormatException ex) {
                        JOptionPane.showMessageDialog(null, "Cantidad no válida.
 Gracias por participar");
                        };
```

Despues se declara un arreglo de Double, ya que usaremos decimales y el tamaño es el puesto para "numCalf". A continuación, mediante un for por cada elemento/index del arreglo de solicita la calificación. Si el valor es alguno no esperado como una cadena, el programa falla y el try/catch toma el error poniendo cero como valor para la posición de la iteración. Si el valor es valido, entonces un IF verifica que el valor se encuentre en 0 y 11 aplicando un OR, si se sale de dicho rango, el valor es cero para la posición de la iteración.

```

                double calificaciones[] = new double[numCalf];
                for(int i = 0; i < calificaciones.length; i++) {
                        try {
                                calificaciones[i] =
Double.parseDouble(JOptionPane.showInputDialog("Inserta la calificación: " +
(i+1)));
                                if(calificaciones[i] < 0 || calificaciones[i] >
10) {
```

```
                                        calificaciones[i] = 0;

                                    }
                                }catch(NumberFormatException ex) {calificaciones[i] =
0;};
                        }
```

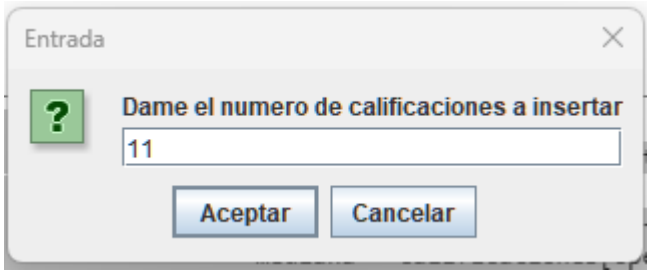
Finalmente se usa un método de "Arrays" para ordenar el arreglo de manera ascendente, por default al usar ".sort", y después con un algoritmo de matemáticas, solo se busca si el arreglo es par, si no lo es, se usan enteros para redondear al entero directamente y se suma mas 1 para estar justo en la mediana, si es par el tamaño del arreglo, se toman los dos valores de en medio y se dividen dando la mediana

```
1                       Arrays.sort(calificaciones);
2                       int operMedi = (calificaciones.length/2);
3                       double mediana;
4                       if(calificaciones.length % 2 == 1) {
5                               mediana = calificaciones[operMedi + 1];
6                       }else {
7                               mediana = (calificaciones[operMedi-1] +
8   calificaciones[operMedi])/2;
9                       }
10                      JOptionPane.showMessageDialog(null, "La mediana es " + mediana);
11          }
    }
```

Ejecución:

mediana 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Solución

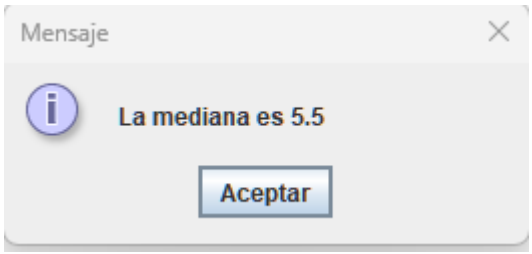
- 6
- 

- 

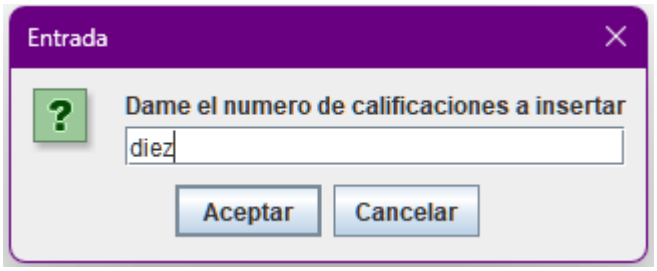
- mediana 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

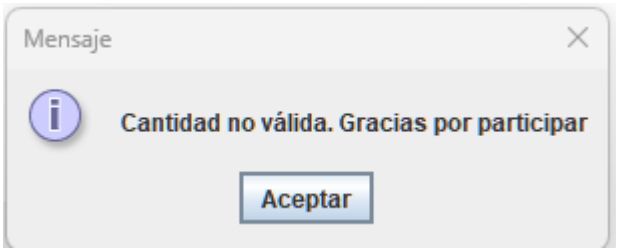
  Solución

  5.5

- 

FALLO:





Codigo Fuente:

https://github.com/OsrKozuki/POO_ESI017O/blob/main/startingPoo/src/tareas/TareaDos.java