

Tarea 3 – Arrays – Tipos de referencia

Enunciado

1. Escribe un método para **revertir** un array. El método no debe devolver nada, debe modificar el array que se le pasa como parámetro y revertirlo.
Ayuda: para revertir un array, debes intercambiar los elementos, es decir, el primer elemento intercambiarlo por el último, el segundo elemento intercambiarlo por el penúltimo, y así sucesivamente. Por ejemplo: [1, 2, 3, 4] revertido sería [4, 3, 2, 1].
Dentro del método reverse, primero imprime el array original, reviértelo y luego vuelvo a imprimir el array revertido. Para el intercambio de variables deberás usar una variable temporal. Utiliza el método que hemos visto en teoría para imprimir arrays.
`public static void reverse(int[] array)`
2. Vamos a crear una clase útil, como las que encontramos dentro del paquete java.util con operaciones relacionadas con arrays. Llama a esta clase **ArrayOperations.java**. Dentro crearemos métodos estáticos, por tanto, **asegúrate que sea una clase estática y nadie pueda llamar al constructor**. Al ser estáticos los métodos, se utilizarán de la forma **ArrayOperations.reverse(array)**. La clase debe incluir los siguientes métodos (algunos realizados en prácticas anteriores):
 - a. Método **print** que imprima el array con formato [1, 2, 3]. No utilices el método Arrays.toString, se trata de que lo programes tú.
 - b. El método **reverse** del ejercicio 1.
 - c. Un método que devuelva el máximo elemento del array, llamado **max**. `public static int max(int[] array)`
 - d. Un método que devuelva el mínimo elemento del array, llamado **min**. `public static int min(int[] array)`
 - e. Método que realice la suma del array. `public static int sum(int[] array)`
 - f. El método **equals** que devuelva un booleano si dos arrays son iguales (ejerc. 9 tarea 2).
 - g. El método equals del ejercicio 10 tarea 2, llamado **equalsWithoutOrder**.
 - h. Método **isArrayOn** (ejerc.11 tarea 2).
 - i. Método **sort** (ejerc.12 tarea 2). Este método se modifica ligeramente, ya que no hay que crear un nuevo array y devolverlo. Se ordena el array pasado por parámetro. `public static void sort(int[] array)`
 - j. Método **copy**, que copie el array fuente en el destino. En este método cada array apunta a una referencia distinta. `public static void copy(int[] src, int[] dst)`
 - k. Método **removeOddNumbers** que elimine del array los números impares. Es decir, cuando encuentre un número impar establezca un 0 en la celda. `public static void removeOddNumbers(int[] array)` [1,2,3] → [0,2,0]
3. Crea un método dentro de la clase ArrayOperations que se llame **printRange** y reciba el array, más dos enteros, inicio y fin. Debe imprimir los elementos del array que estén dentro del rango inicio y fin ambos inclusive. Imprime los elementos con el siguiente formato: [2, 3]
Ejemplo [1, 2, 3, 4, 5] printRange(array, 1, 3) [1, 2, 3]
Si el usuario inserta límites que no se encuentran en el array lanza una excepción de tipo ArrayIndexOutOfBoundsException con el mensaje "El índice X excede los límites del array".
`public static void printRange(int[] array, int start, int end)`
4. Crea una clase **Main.java** con un método main para testear la clase ArrayOperations.
Imagina que quiero realizar las siguientes operaciones con el array: [-2, 45, -29, 77, 56, 31]

```

max(array) → devuelve 77
copy(array, copyArray)
print(copyArray) → [-2, 45, -29, 77, 56, 31]
arrayOrdenado = sort(array)
print(arrayOrdenado) → [-29, -2, 31, 45, 56, 77]
equalsWithoutOrder(array, arrayOrdenado) → true
equals(array, copyArray) → true
removeOddNumbers(array)
sum(array) → 54
min(array) → -29
printRange(array, 1, 4) → [-2, 45, -29, 77]
reverse(arrayOrdenado)
printRange(arrayOrdenado, 2, 5) → [-2, 31, 45, 56]

```

- a. ¿Funciona todo como se espera? ¿Por qué? Explica razonadamente.
 - b. ¿Cómo harías para solucionarlo?
5. Crea un método que sea **deleteOneDigitNumbers**, y elimine del array los números que solo tengan un dígito. Ayuda. Crea un nuevo array con la longitud adecuada, antes deberás contar cuantos datos vas a eliminar para crear el nuevo array. En el nuevo array copia solo los números que te interesan, devuelve el nuevo array.
Ejemplo: [-6, 33, 98, 10, 9] → [33, 98, 10]
6. Piensa y crea un método de utilidad para realizar con arrays e inclúyelo en la clase. Explica qué realiza tú método.
7. Documenta la clase ArrayOperations.java. Donde expliques la clase y los métodos.
Ejemplo:

```

3  /**
4   * Añade una breve descripción de lo que hace la clase
5   * @author Patricia
6   */
7   public class ArrayOperations {
8
9       /**
10      * Método que revierte un array de enteros específico
11      * @param array Array de enteros a revertir
12      */
13      public static void reverse(int[] array) {
14
15      }
16  }
17

```

Ayuda: si escribes /** + Intro te genera el comentario listo para rellenar.

Entrega

- Contesta a las preguntas en PDF (copia y pega el código de los ejercicios que lo requieran, no capturas de pantalla).