EJERCICIO 1

Clase

import java.util.Arrays;

/\*\*

\* Esta clase contiene métodos para trabajar con arrays.

\*

\* @author Gabriel Vargas Reyes

\* @version 1.0

\*/

public class ArrayOperations {

/\*\*

\* Imprime los valores de un array dentro de []

\*

\* @param array Array a imprimir

\* @return array

\*/

public static int[] print(int[] array) {

System.out.print("[");

for (int i = 0; i < array.length; i++) {

System.out.print(array[i] + " ");

}

System.out.println("]");

System.out.println();

return array;

}

/\*\*

\* Devuelve el array en orden inverso

\* @param array Array a invertir

\* @return reversed

\*/

public static int[] reverse(int[] array) {

int[] reversed = new int[array.length];

for (int i = 0; i < array.length; i++) {

reversed[i] = array[array.length - 1 - i];

}

return reversed;

}

/\*\*

\* Devuelve el valor máximo de un array.

\*

\* @param array Array a analizar

\* @return Valor máximo.

\*/

public static int max(int[] array) {

int max = array[0];

for (int i = 0; i < array.length; i++) {

if (array[i] > max) {

max = array[i];

}

}

return max;

}

/\*\*

\* Devuelve el valor mínimo de un array.

\*

\* @param array Array a analizar

\* @return Valor mínimo.

\*/

public static int min(int[] array) {

int min = array[0];

for (int i = 0; i < array.length; i++) {

if (array[i] < min) {

min = array[i];

}

}

return min;

}

/\*\*

\* Devuelve la suma de todos los valores de un array.

\*

\* @param array Array a analizar

\* @return Suma de todos los valores.

\*/

public static int sum(int[] array) {

int sum = 0;

for (int i = 0; i < array.length; i++) {

sum += array[i];

}

return sum;

}

/\*\*

\* Devuelve si dos arrays son iguales.

\*

\* @param array Array a analizar

\* @return true si son iguales, false en caso contrario.

\*/

public static boolean equals(int[] array1, int[] array2) {

if (array1.length != array2.length) {

return false;

}

for (int i = 0; i < array1.length; i++) {

if (array1[i] != array2[i]) {

return false;

}

}

return true;

}

/\*\*

\* Devuelve si dos arrays son iguales.

\*

\* @param array Array a analizar

\* @return true si son iguales, false en caso contrario.

\*/

public static boolean equalsWithOrder(int[] array1, int[] array2) {

if (array1.length != array2.length) {

return false;

}

for (int i = 0; i < array1.length; i++) {

if (array1[i] != array2[i]) {

return false;

}

}

return true;

}

/\*\*

\* Devuelve si un array está contenido en otro.

\*

\* @param array Array a analizar

\* @return true si está contenido, false en caso contrario.

\*/

public static boolean IsOn(int[] src, int[] dst) {

boolean isOn = false;

Arrays.sort(src);

Arrays.sort(dst);

for (int i = 0; i < src.length; i++) {

for (int j = 0; j < dst.length; j++) {

if (src[i] == dst[j]) {

isOn = true;

}

}

}

return isOn;

}

/\*\*

\* Devuelve un array copiado.

\*

\* @param array Array de enteros.

\* @param copyArray Array de enteros.

\* @return Array copiado.

\*/

public static int[] copy(int[] src, int[] dst) {

for (int i = 0; i < src.length; i++) {

dst[i] = src[i];

}

return dst;

}

/\*\*

\* Elimina los valores impares de un array.

\*

\* @param array Array a analizar

\* @return array sin impares

\*/

public static int[] removeOddNumbers(int[] array) {

int[] newArray = new int[array.length];

int j = 0;

for (int i = 0; i < array.length; i++) {

if (array[i] % 2 == 0) {

newArray[j] = array[i];

j++;

} else {

newArray[j] = 0;

j++;

}

}

return newArray;

}

/\*\*

\* Imprime los valores de un array en un rango determinado.

\*

\* @param array Array a imprimir

\* @param start Inicio del rango

\* @param end Fin del rango

\*/

public static void printRange(int[] array, int inicio, int fin) {

System.out.print("[");

for (int i = inicio; i < fin; i++) {

System.out.print(array[i] + " ");

}

System.out.println("]");

System.out.println();

}

/\*\*

\* Devuelve un array ordenado.

\*

\* @param array Array a ordenar

\* @return Array ordenado.

\*/

public static int[] sort(int[] array) {

int[] sorted = new int[array.length];

for (int i = 0; i < array.length; i++) {

sorted[i] = array[i];

}

Arrays.sort(sorted);

return sorted;

}

/\*\*

\* Elimina los elementos de un solo digito en un array.

\* @param array

\* @return newArray

\*/

public static int[] deleteOneDigitNumbers(int[] array) {

int[] newArray = new int[array.length];

int j = 0;

for (int i = 0; i < array.length; i++) {

if (array[i] > 9) {

newArray[j] = array[i];

j++;

} else {

newArray[j] = 0;

j++;

}

}

return newArray;

}

/\*\*

\* Cuenta cuantas veces se repite un numero en un array.

\* @param array

\* @param number

\* @return count

\*/

public static int count(int[] array, int number) {

int count = 0;

for (int i = 0; i < array.length; i++) {

if (array[i] == number) {

count++;

}

}

return count;

}

}

Main

public class Main extends ArrayOperations {

public static void main(String[] args) {

int[] array = { -2, 45, -29, 77, 56, 31 };

int[] copyArray = new int[array.length];

System.out.println("Valor máximo: " + max(array));

System.out.print("Copia: ");

print(copy(array, copyArray));

System.out.print("Ordenado: ");

int[] arrayOrdenado = sort(array);

print(arrayOrdenado);

System.out.print("Igual con orden: ");

System.out.println(equalsWithOrder(array, arrayOrdenado));

System.out.print("Igual sin orden: ");

System.out.println(equals(array, copyArray));

System.out.println("Sin números impares: ");

print(removeOddNumbers(array));

System.out.println("Suma total: ");

System.out.print(sum(array));

System.out.println("Mínimo: ");

System.out.print(min(array));

System.out.println("Imprimir rango: ");

printRange(array, 1, 4);

System.out.println("Array inverso: ");

print(reverse(array));

System.out.println("Array ordenado inverso: ");

print(sort(reverse(array)));

}

}

EJERCICIO 2

Clase

public class MesCollector {

private int mes, diaMes, anyo;

private int[] temperatura;

MesCollector() {} // Constructor por defecto

MesCollector(int mes, int anyo) {

this.mes = mes;

this.anyo = anyo;

this.temperatura = new int[32];

switch (mes) {

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

this.diaMes = 31;

break;

case 4:

case 6:

case 9:

case 11:

this.diaMes = 30;

break;

case 2:

this.diaMes = 28;

break;

default:

this.diaMes = 0;

break;

}

} // Constructor con parámetros

public int getTemperatura(int dia) {

if (dia > 0 && dia <= this.diaMes) {

return temperatura[dia - 1];

} else {

return -1;

}

}

public void setTemperatura(int dia, int temperatura) {

if (dia > 0 && dia <= this.diaMes) {

this.temperatura[dia - 1] = temperatura;

}

}

public int getDiaMes() {

return this.diaMes;

}

@Override

public String toString() {

/\*

Lo he hecho asi porque queria probar StringBuilder

pero se podia haber hecho de la otra forma

\*/

StringBuilder sb = new StringBuilder();

sb.append("Mes: " + this.mes + "\n");

sb.append("Año: " + this.anyo + "\n");

sb.append("Días: " + this.diaMes + "\n");

sb.append("Temperaturas: ");

sb.append("\n");

for (int i = 0; i < this.diaMes; i++) {

sb.append("Dia " + (i + 1) + ": " + this.temperatura[i] + "ºC\n");

}

return sb.toString();

}

}

Main

import java.util.Scanner;

public class MesTester {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

MesCollector enero = new MesCollector(1, 2021);

for (int i = 0; i < enero.getDiaMes(); i++) {

System.out.print(

"Introduce la temperatura del día " + (i + 1) + " (ºC): "

);

int temp = sc.nextInt();

enero.setTemperatura(i + 1, temp);

}

System.out.println(enero.toString());

sc.close();

}

}