



/TheVampi



Métodos para manejo de Arreglos de una dimensión

Programación
orientada a objetos

Martinez Rivera Luis Fernando

“Nunca pares de aprender”





Descripción del problema

Desarrolle una clase donde se incluyan los métodos que se han visto para el manejo de arreglos de una dimensión.

TODO ESTO PARA ARREGLOS DE TIPO ENTERO

- 1) Cargar un arreglo.
- 2) Imprimir un arreglo.
- 3) Ordenar un arreglo. (De menor a mayor y de mayor a menor)
- 4) Buscar un elemento de un arreglo.

En un método main (puede implementarlo en otra clase o en la misma) implemente un menú que presente las opciones de los puntos anteriores.

Proyecto completo libre para descarga en el repositorio:

https://github.com/TheVampi/POO-TecNM/tree/main/Practica_2_1





Código fuente

```
/*  
  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this  
 license  
  
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template  
  
 */  
  
package practica_2_1;  
  
  
import javax.swing.JOptionPane;  
  
  
public class Practica_2_1 {  
  
    /**  
     * @param args the command line arguments  
     */  
  
    public static void main(String[] args) {  
        Arreglos ob = new Arreglos();  
        //Creando el arreglo principal  
        int n = Integer.parseInt(JOptionPane.showInputDialog("Digite el numero de elementos para el  
 arreglo"));  
        int[] arr = new int[n];  
        //Menu para seleccionar operacion  
        int num;  
  
        do {  
            num = Integer.parseInt(JOptionPane.showInputDialog("Seleccione la operacion deseada:  
 \n1- Cargar arreglo\n2- Ordenar de menor a mayor\n3- Ordenar de mayor a menor\n4- Buscar  
 numeros en el arreglo\n5- Mostrar el arreglo\n6- Salir"));  
            switch (num) {  
                case 1:
```





```
        arr = ob.cargarArreglo(n);

        break;

    case 2:

        ob.ordenamientoMenorAMayor(arr);

        break;

    case 3:

        ob.ordenamientoMayorAMenor(arr);

        break;

    case 4:

        ob.buscarElemento(arr);

        break;

    case 5:

        JOptionPane.showMessageDialog(null, ob.mostrarArreglo(arr));

        break;

    case 6:

        System.exit(0);

        break;

    default:

        JOptionPane.showMessageDialog(null, "Opción incorrecta digite de nuevo");

    }

} while (true);

}

}

class Arreglos {

    int[] cargarArreglo(int filas) {
```





```
int[] arregloDevuelto = new int[filas];

//Llenando con datos
for (int posicion = 0; posicion < arregloDevuelto.length; posicion++) {
    arregloDevuelto[posicion] = Integer.parseInt(JOptionPane.showInputDialog("Ingrese un
valor para la posición: " + (posicion + 1)));
}

return (arregloDevuelto);
}

void ordenamientoMenorAMayor(int[] arreglo) {

    int variableAuxiliar;

    //Ordenandolos elementos
    for (int pasadas = 1; pasadas < arreglo.length; pasadas++) {
        for (int comparaciones = 1; comparaciones <= arreglo.length - 1; comparaciones++) {

            if (arreglo[comparaciones - 1] > arreglo[comparaciones]) {
                variableAuxiliar = arreglo[comparaciones - 1];
                arreglo[comparaciones - 1] = arreglo[comparaciones];
                arreglo[comparaciones] = variableAuxiliar;
            }

            //LO SIGUIENTE ES PARA VERIFICAR EN CONSOLA COMO FUNCIONA EL ALGORITMO EN
LAS COMPARACIONES

            //La siguiente linea es solo para verificar en consola
            System.out.println("Comparacion Actual:" + comparaciones);
```





```
//Mostrando los resultados ordenados del arreglo

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

    System.out.println(arreglo[i]);

}

}

//AQUI TERMINA CICLO DE COMPARACIONES


//LO SIGUIENTE ES PARA VERIFICAR EN CONSOLA COMO FUNCIONA EL ALGORITMO EN LAS
PASADAS

//La siguiente linea es solo para verificar en consola

System.out.println("Pasada actual:" + pasadas);


//Mostrando los resultados ordenados del arreglo EN CONSOLA

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

    System.out.println(arreglo[i]);

}

}

//AQUI TERMINA CICLO DE PASADAS


//Mostrando los resultados FINALES ordenados del arreglo ESTO SE MUESTRA EN CONSOLA

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

    System.out.println(arreglo[i]);

}

}

//Mostrando los resultados FINALES ordenados del arreglo en un JOptionPane

String aux = "Arreglo ordenado: \n";

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

    aux = aux + arreglo[i] + "\n";

}

JOptionPane.showMessageDialog(null, aux);

}
```





```
//METODO BURBUJA para ordenar arreglos de UNA DIMENSION (DE TIPO INT)

void ordenamientoMayorAMenor(int[] arreglo) {

    int variableAuxiliar;

    //Ordenandolos elementos
    for (int pasadas = 1; pasadas < arreglo.length; pasadas++) {
        for (int comparaciones = 1; comparaciones <= arreglo.length - 1; comparaciones++) {

            if (arreglo[comparaciones - 1] < arreglo[comparaciones]) {
                variableAuxiliar = arreglo[comparaciones - 1];
                arreglo[comparaciones - 1] = arreglo[comparaciones];
                arreglo[comparaciones] = variableAuxiliar;
            }

            //LO SIGUIENTE ES PARA VERIFICAR EN CONSOLA COMO FUNCIONA EL ALGORITMO EN
            LAS COMPARACIONES

            //La siguiente linea es solo para verificar en consola
            System.out.println("Comparacion Actual:" + comparaciones);

            //Mostrando los resultados ordenados del arreglo
            for (int i = 0; i < arreglo.length; i++) {
                System.out.println(arreglo[i]);
            }

            //AQUI TERMINA CICLO DE COMPARACIONES

            //LO SIGUIENTE ES PARA VERIFICAR EN CONSOLA COMO FUNCIONA EL ALGORITMO EN LAS
            PASADAS

            //La siguiente linea es solo para verificar en consola
```





```
System.out.println("Pasada actual:" + pasadas);

//Mostrando los resultados ordenados del arreglo EN CONSOLA
for (int i = 0; i < arreglo.length; i++) {
    System.out.println(arreglo[i]);
}

}

//AQUI TERMINA CICLO DE PASADAS

//Mostrando los resultados FINALES ordenados del arreglo ESTO SE MUESTRA EN CONSOLA
for (int i = 0; i < arreglo.length; i++) {
    System.out.println(arreglo[i]);
}

//Mostrando los resultados FINALES ordenados del arreglo en un JOptionPane
String aux = "Arreglo ordenado: \n";
for (int i = 0; i < arreglo.length; i++) {
    aux = aux + arreglo[i] + "\n";
}
JOptionPane.showMessageDialog(null, aux);
}

//Buscar elementos en un arreglo
void buscarElemento(int[] unArreglo) {
    int num = Integer.parseInt(JOptionPane.showInputDialog("Teclea el numero a buscar"));
    boolean bandera = false;
    for (int i = 0; i < unArreglo.length; i++) {
        if (unArreglo[i] == num) {
            JOptionPane.showMessageDialog(null, "El numero: " + num + " se encuentra en la
posicion: " + i);
        }
    }
}
```





```
        bandera = true;

        break;
    }
}

if (bandera == false) {

    JOptionPane.showMessageDialog(null, "El numero: " + num + " NO se encuentra en ningun
posicion");

}

}

//Mostrar un arreglo

String mostrarArreglo(int[] unArreglo) {

    String s = "Contenido del arreglo: ";

    for (int fila = 0; fila < unArreglo.length; fila++) {

        s = s + unArreglo[fila] + "\n";

    }

    return (s);

}

}
```



Capturas de ejecución

