Clase 9
Arreglos Unidimensionales 2

Taller de preparación para la OCI - PUC

Ejercicio clase 9

Vamos a Eclipse!

Ordenar un arreglo

Cambiar posiciones según los valores internos del arreglo

Ordenar un arreglo

Dado que los arreglos en Java son rígidos, es decir, no permiten cambios en su estructura una vez creados, utilizaremos la creación de un nuevo arreglo para ordenar nuestros datos.

En la lista dada usaremos los siguientes pasos para ir agregando los valores ordenados:

1. Buscar el elemento más pequeño en nuestro arreglo.
2. Conseguir el índice de ese elemento. ej: *Listaza[2]* el índice sería “2”.
3. Guardar el elemento más pequeño en el siguiente espacio disponible del nuevo arreglo.
4. Reemplazar el elemento en el arreglo original por algo más grande a todos los otros elementos.
5. Repetir las veces que sea necesario...

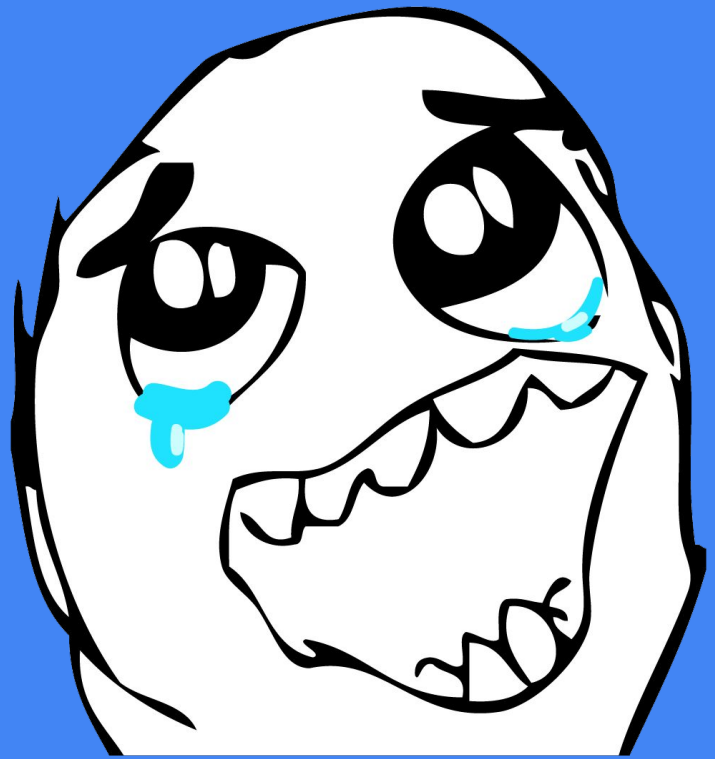
```java
import java.util.Arrays;
class OrdenandoSusNotas{
     public static void main(String[],args){
          //Creamos una lista de notas
          double [] Notas = {7.0, 3.2, 5.0, 2.0, 4.5, 6.2, 5.7, 6.9, 3.94};
          int largo = Notas.length;
          double [] Ordenamiento = new double[largo]; //se crea una nueva lista

          for (int j = 0; j < Lista.length; j++){
               //Primero buscamos el valor mas alto posible para double
               double minimo = Double.MAX_VALUE;
               int indiceMinimo = 0;
               for (int indice = 0; indice < Lista.length; indice++){
                    if (Notas[indice] < minimo){
                         minimo = Notas[indice];
                         indiceMinimo = indice;
                    }
               }
               Ordenamiento[j] = minimo;
               Notas[indiceMinimo] = Double.MAX_VALUE;
          }
     System.out.println(Arrays.toString(Ordenamiento));
     }
}
```

No entendi nada... Ejemplo

Modificando arreglos

Alterar las dimensiones (el largo) del arreglo



Agregar un elemento

Como el largo de los arreglos está fijo para poder “agregar” elementos tendremos que **crear un arreglo nuevo**.

```
import java.util.Arrays;
class Main{
     public static void main(String[],args){
          int [] Lista = {2, 4, 15, 2}; //se crea la lista a la que se le agrega el dato
          int nuevo_dato = 100;
          int [] nuevaLista = new int[Lista.length+1] //se crea una nueva lista
          //ahora iterando agregamos los elementos que ya teniamos mas el elemento nuevo
          for (int i = 0; i < Lista.length; i++){
               nuevaLista[i] = Lista[i];
          }
          nuevaLista[nuevaLista.length - 1] = nuevo_dato;
     }
}
```

Eliminar elementos

Para eliminar un elemento usaremos la misma estrategia que para agregar elementos

… es decir **crearemos un nuevo arreglo**

```
import java.util.Arrays;
class Main{
    public static void main(String[],args){
        //se crea la lista a la que se le quitara el dato
        int [] Lista = {2, 4, 15, 2};
        int dead_dato = 2; //aquí ponemos el índice del dato que eliminaremos
        int [] nuevaLista = new int[Lista.length-1] //se crea una nueva lista
        int k = 0; //usaremos esta variable para llevar track del índice en
el que vamos
        //ahora iterando agregamos los elementos que ya teniamos mas el
    elemento nuevo
        for (int i = 0; i < Lista.length; i++){
            if (j == dead_dato){
                continue;
            }
            nuevaLista[k] = Lista[i];
        k++; //se aumenta en uno el contador del índice
        }
    }
}
```

eliminando un elemento

Sugerencias

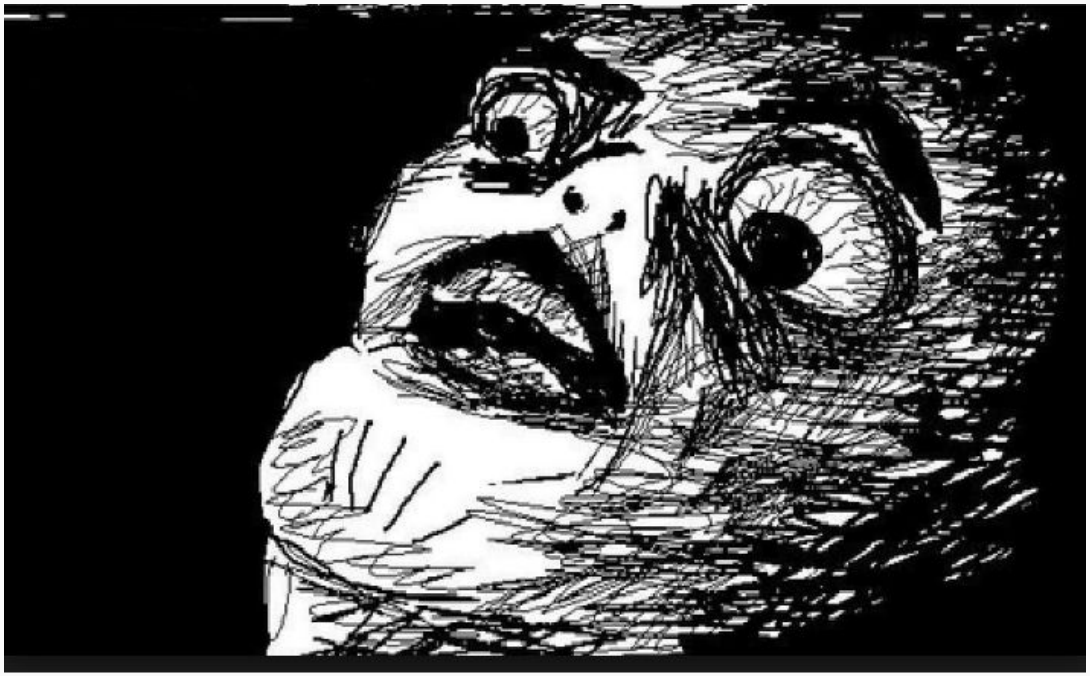
Como estas acciones son bastante útiles y las podríamos llegar a ocupar bastante, es más cómodo (y también se verá mejor en nuestro código) si definimos funciones que realicen estas acciones

```
<type> arreglo_nuevo = borrar_elemento(<arreglo>,<indice>);

<type> arreglo_nuevo = agregar_elemento(<arreglo>,<nuevo_elemento>);
```

Relación entre strings y arreglos

Los strings se pueden ver como un tipo de arreglo!



Funciones tipo arreglo

```
import java.util.Arrays;
class Main{
    public static void main(String[],args){
    String unString = “Tengo caracteres :D”;
    //podemos acceder a elementos
    unString.charAt(<indice>);
    //podemos saber el largo del String
    unString.length();
    }
}
```

Split

Estructura:

```
<el_string_a_separar>.split(<String_Separador>)
```

Para que este funcione se debió haber definido correctamente el String separador y además se tiene que igualar esto a la correcta declaración de un Array

```
import java.util.Arrays;
class Main{
    public static void main(String[],args){
    String unString = “word”;
    /*se separa por el String vacio lo que hace que cada
    caracter del string sea un elemento del Array*/
    String [] unArray = unString.split(“”);
    System.out.println(Arrays.toString(unArray));

    //Ahora se utiliza como separador un String (“X”)
    String [] otroString = “110X101X101X010X101”;
    String [] otroArray = unString.split(“X”);
    System.out.println(Arrays.toString(otroArray));
    }
}
```