PRACTICA SOBRE METODOS DE ORDENACION CON LA ESTRUCTURA ADAPTER

CLASE MAIN:

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

import java.util.Scanner;

/**
 *
 * @author HP PAVILION
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args)
    {
        Scanner leer=new Scanner(System.in);

        Adaptador adap=new Adaptador();

        System.out.println("***************** Ingresa los Datos
*****************");

        System.out.println(" Ingresa el Numero de valores del Arreglo ");

        int tam=leer.nextInt();
        char[] vect = new char[tam];
        String cadena = "",aux;
System.out.println("***************** Ingresa los caracteres
****************");
        for(int i=0 ; i<vect.length; i++){
            System.out.print("Letra "+(i+1)+": ");
            aux=leer.next();
                if(aux.length()==1){
                    int auxi = aux.charAt(0);
                    if(auxi>=97)
                    cadena = cadena + aux;
                    else{
                        while(!(auxi>=97)){
```

```java
                        System.out.println("ERROR - Caracter no admitido");
                        aux=leer.next();
                        auxi = aux.charAt(0);
                    }
                }
                }else{
                while(aux.length()!=1){
                    System.out.println("Favor de ingresar solo una letra");
                    aux=leer.next();
                }
                cadena=cadena+aux;
            }
        }
        vect = cadena.toCharArray();

        System.out.println("*********** Selecciona un Tipo de Ordenacion
*********** ");
        System.out.println("  1)  Burbuja ");
        System.out.println("  2)  Insercion ");
        System.out.println("  3)  Seleccion ");
        System.out.println();
        int resp=leer.nextInt();

            if(resp==1)
        adap.conversion(vect, "a", "Burbuja");
        else if(resp==2)
            adap.conversion(vect, "a", "Insercion");
        else
            adap.conversion(vect, "a", "Seleccion");
    }
}
```

CLASE ADAPTADOR:

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
 *
 * @author HP PAVILION
 */
```

```java
public class Adaptador {

    public void conversion(char [] vector, String lb, String tipob){

        int[] vector2 = new int[vector.length];
            for (int i = 0; i < vector.length ; i++){
                String test="";
                test = test + vector[i];
                vector2[i] = vector[i];
            }

        MetSeleccion sel=new MetSeleccion();
        if(tipob.equals("burbuja")){
           MetBurbuja b=new MetBurbuja();
           vector2=b.ordenar(vector2);
        }
        else if(tipob.equals("seleccion")){
            vector2=sel.ordenar(vector2);
        }
        else{
        MetInsercion in=new MetInsercion();
            vector2=in.ordenar(vector2);
        }
        for (int i = 0; i < vector.length ; i++){
            int test=0;
            test = test + vector2[i];
            vector[i] = (char)vector2[i];
        }

            sel.Imprimir(vector);

        }
    }
```

CLASE ORDENACION:
```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
 *
 * @author HP PAVILION
 */
```

```java
public class Ordenacion {

    public void Imprimir(String res){
    System.out.println(res);
    }
    public void Imprimir(char [] vector){
    System.out.println("El Arreglo ordenado es el Sig.: ");
    for(int i=0 ; i<vector.length ; i++){
    System.out.print(vector [i]+ ", ");
    }
  }

}
```

CLASE METODO:

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
 *
 * @author HP PAVILION
 */
public abstract class Metodo extends Ordenacion {

    abstract int[] ordenar(int[] vector);

}
```

CLASE METBURBUJA:

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
```

```
 *
 * @author HP PAVILION
 */
public class MetBurbuja extends Ordenacion {

    int[] ordenar(int[] vector) {
        int x, i, j;
        for(i=0 ; i<vector.length ; i++){
            for(j=0 ; j<vector.length ; j++){
                if(vector[i]<vector[j]){
                    x=vector[i];
                    vector[i]=vector[j];
                    vector[j]=x;
                }
            }
        }
        return vector;
    }
}
```

CLASE METINSERCION:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
 *
 * @author HP PAVILION
 */
public class MetInsercion extends Metodo {

    int[] ordenar(int[] vector) {

        for(int i=1 ; i<vector.length ; i++){
            int aux= vector[i];
            int j=i-1;
            while(j>=0 && j>=0 && vector[j]>aux){
                vector[j+1]=vector[j];
                j--;
            }
            vector[j+1]=aux;
        }
        return vector;
```

```
    }
}
```

CLASE METSELECCION:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package BusquedaAdapter;

/**
 *
 * @author HP PAVILION
 */
public class MetSeleccion extends Metodo{

    int[] ordenar(int[] vector) {
        for(int i=0 ; i<vector.length-1 ; i++){
            int i2=i;
            for(int j=i+1 ; j<vector.length ; j++){
                if(vector[j]<vector[i2]){
                    i2=j;
                }
            }
            if(i!=i2){
                int aux=vector[i];
                vector[i]=vector[i2];
                vector[i2]=aux;
            }
        }
        return vector;
    }

}
```

DIAGRAMA DE CLASES

Main

+ main(String[]) : void

Ordenacion

+ Imprimir(String) : void
+ Imprimir(char[]) : void

Adaptador

+ conversion(char[], String, String) : void

Metodo

~ ordenar(int[]) : int[]

MetBurbuja

~ ordenar(int[]) : int[]

MetInsercion

~ ordenar(int[]) : int[]

MetSeleccion

~ ordenar(int[]) : int[]