PRACTICA SOBRE METODOS DE ORDENACION Y BUSQUEDA CON LA ESTRUCTURA ADAPTER

CLASE MAIN:

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

package OrganizarBuscarAdapter;

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)){
```

```
                        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("*********** ¿  QUE PROCESO REALIZARÁS ?
************ ");
        System.out.println("  1)  Busqueda ");
        System.out.println("  2)  Ordenacion ");
        System.out.println();
        int respu=leer.nextInt();
        int resp;

        switch(respu){
            case 1:
                System.out.print("La letra a buscar es:  ");
                String lb=leer.next();
                System.out.println("*********** Selecciona un Tipo de Busqueda
************ ");
                System.out.println("  1)  Binaria ");
                System.out.println("  2)  Secuencial ");
                resp=leer.nextInt();

                if(resp==1)
                    adap.convierte(vect, lb, "binaria");
                else
                    adap.convierte(vect, lb, "secuencial");

            break;

            case 2:
                System.out.println("*********** Selecciona un Tipo de Busqueda
************ ");
                System.out.println("  1)  Burbuja ");
                System.out.println("  2)  Insercion ");
                System.out.println("  3)  Seleccion ");
                System.out.println();
                resp=leer.nextInt();
```

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

            break;
        }
    }
}
```

CLASE ADAPTADOR:

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

package OrganizarBuscarAdapter;

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

    public void convierte (char[] vector, String lb, String tipo){
        int[] vector2 = new int[vector.length];
        for (int i = 0; i < vector.length ; i++){
            String test="";
            test = test + vector[i];
            vector2[i] = vector[i];
        }
        int letbus = lb.charAt(0);
        int lbi=0;
        if(tipo.equals("binaria") || tipo.equals("secuencial")){
            BusBinaria bi = new BusBinaria();
            if(tipo.equals("binaria")){
                lbi=bi.buscar(vector2 , letbus);
            }
            else{
                BusSecuencial sec = new BusSecuencial();
                lbi=sec.buscar(vector2 , letbus);
            }
            String mandar;
            if(lbi>-1){
                mandar="Felicidades... La letra fue encontrada";
            }else{
                mandar="Lo siento, esta letra no se encontro";
            }
            bi.Imprimir(mandar);

        }else{
            MetSeleccion sel=new MetSeleccion();
            if(tipo.equals("burbuja")){
                MetBurbuja b=new MetBurbuja();
                vector2=b.ordenar(vector2);
            }
            else if(tipo.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 SUPERCLASE:

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

package OrganizarBuscarAdapter;

/**
 *
 * @author HP PAVILION
 */
public class SuperClase {
    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 ORDENACION:

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

package OrganizarBuscarAdapter;

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

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

}
```

CLASE METBURBUJA:

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

package OrganizarBuscarAdapter;

/**
 *
 * @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 OrganizarBuscarAdapter;

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

    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 OrganizarBuscarAdapter;

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

    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;
    }

}
```

CLASE BUSCAR:

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

package OrganizarBuscarAdapter;

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

```
public abstract class Buscar extends SuperClase{

    abstract int buscar( int [] vector, int Num);

}
```

CLASE BUSBINARIA:

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

package OrganizarBuscarAdapter;

/**
 *
 * @author HP PAVILION
 */
public class BusBinaria extends Buscar {

     public int buscar (int[] vector, int Num) {
        int inicio=0, pos;
        int fin=vector.length-1;
        int[] f = new int[1];

        vector=this.buscar(vector);


        while (inicio<=fin) {
            pos=(inicio+fin)/2;
            if (vector[pos]==Num){
                f[0] = pos;
                return pos;
            }
            else if (vector[pos]<Num) {
                inicio = pos+1;
            }
            else
                fin = pos-1;
        }
        int[] a = new int[1];
        a[0] = -1;

        return -1;

    }
```

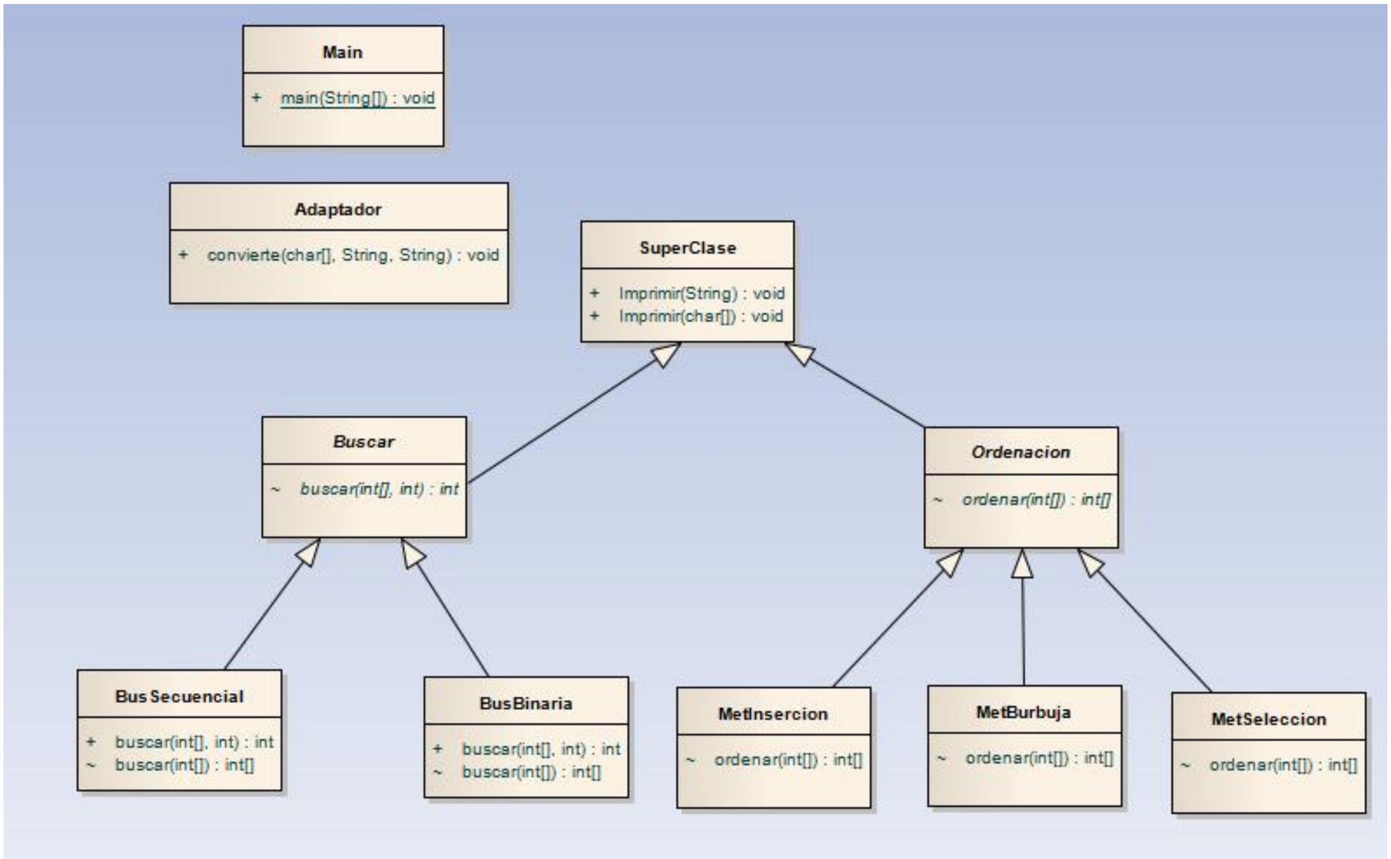
```
     int[] buscar(int[] n){
         int aux;
         for (int i = 1; i < n.length; i++) {
             for (int j = n.length- 1; j >= i; j--) {
                 if(n[j]<n[j-1]){
                     aux=n[j];
                     n[j]=n[j-1];
                     n[j-1]=aux;
                 }
             }
         }
         return n;
     }

}
```

CLASE BUSSECUENCIAL:

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

package OrganizarBuscarAdapter;

/**
 *
 * @author HP PAVILION
 */
public class BusSecuencial extends Buscar {


        public int buscar( int [] vector, int Num) {

            int[] vector2 = new int[1];
            vector = this.buscar(vector);
            int tope = 0;
            for(int i = 0 ; i < vector.length ; i++){
                if(vector[i]==Num){
                    tope = 1;
                    vector2[0] = i;
                    return 2;
                }
            }
            if(tope==0){
```

```
            vector2[0] = -1;

        }
        return -1;

    }
    int[] buscar (int[] vector){
        int x;
        for (int i = 1; i < vector.length; i++) {
            for (int j = vector.length- 1; j >= i; j--){
                if(vector[j] < vector[j-1]){
                    x = vector[j];
                    vector[j] = vector[j-1];
                    vector[j-1] = x;
                }
            }
        }
        return vector;
    }
}
```

DIAGRAMA DE CLASES


Main
+ main(String[]) : void
Adaptador
+ convierte(char[], String, String) : void
SuperClase
+ Imprimir(String) : void
+ Imprimir(char[]) : void
Buscar
~ buscar(int[], int) : int
Ordenacion
~ ordenar(int[]) : int[]
Bus Secuencial
+ buscar(int[], int) : int
~ buscar(int[]) : int[]
BusBinaria
+ buscar(int[], int) : int
~ buscar(int[]) : int[]
MetInsercion
~ ordenar(int[]) : int[]
MetBurbuja
~ ordenar(int[]) : int[]
MetSeleccion
~ ordenar(int[]) : int[]