Evaluación 1 - 2da Unidad

Nombre: Juan David Jiménez Romero

Codigo main:

package evaluacion1\_u2\_juanj;

import javax.swing.SwingUtilities;

public class Evaluacion1\_U2\_JuanJ {

public static void main(String[] args) {//inicio del metodo main que ejecuta el programa

SwingUtilities.invokeLater(new Runnable(){//nos dice que con la libreria SwingUtilities crea un nuevo ejecutable

@Override

public void run(){//metodo que con ayuda de la libreria ejecuta lo siguiente

new MenuPrincipal();//esto dice que crea una ventana con el contenido de la clase Ordenarxd()

}//fin del metodo run()

});//fin del SwuingUtilities

}//fin del metodo main

}

Codigo Burbuja:

package evaluacion1\_u2\_juanj;

public class Burbuja {

static void sort(int[] arr) {

// TODO code application logic here

int n = arr.length;

for(int i=0;i<n-1;i++){

for(int j=0;j<n-i-1;j++){

if(arr[j]>arr[j+1]){

int temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

}  
Codigo Ordenar Burbuja;

package evaluacion1\_u2\_juanj;

import java.awt.Color;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import static java.awt.image.ImageObserver.HEIGHT;

import javax.swing.\*;

public class OrdenarBurbuja extends JFrame {

private JTextField InputField;

private JTextField InputField1;

private JButton addButton;

private JButton simularButton;

private JButton buscaButton;

private JButton salirButton;

private JButton RadixButton;

private JTextArea resultArea;

private JTextArea resultArea1;

private JLabel text;

private JLabel text1;

private JLabel text2;

private int[] numbers;

private String[] name;

int resultado;

public OrdenarBurbuja(){

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setTitle("Ordenamiento Burbuja Evaluacion JuanJ");

setSize(500,400);

setLayout(null);

InputField = new JTextField();

InputField1 = new JTextField();

addButton = new JButton("Añadir");

simularButton = new JButton("Ordenar");

salirButton = new JButton("Salir");

buscaButton = new JButton("Buscar");

RadixButton = new JButton("Orde Radix");

resultArea = new JTextArea();

resultArea1 = new JTextArea();

text = new JLabel("=====Ingresar estudiante======");

text1 = new JLabel("Ingresar Nota");

text2 = new JLabel("Ingresar Estudiante");

text.setBounds(150, 5, 300, 30);

text1.setBounds(60, 20, 300, 30);

text2.setBounds(260, 20, 300, 30);

InputField.setBounds(20,50,150,30);

InputField1.setBounds(240,50,150,30);

addButton.setBounds(110, 300, 90, 30);

simularButton.setBounds(230, 300, 90, 30);

salirButton.setBounds(10, 300, 90, 30);

RadixButton.setBounds(350, 300, 90, 30);

buscaButton.setBounds(380, 20, 90,30);

resultArea.setBounds(20, 90, 180, 200);

resultArea.setEditable(false);

resultArea1.setBounds(220, 90, 220, 200);

resultArea1.setEditable(false);

numbers= new int[0];

name = new String[0];

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

addName();

addNumero();

}

});

simularButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

simulateBubbleSort();

}

});

RadixButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

simulateRadixSort();

}

});

salirButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

new MenuPrincipal();

dispose();

}

});

buscaButton.addActionListener(new ActionListener() {//metodo que sirve para darle una funcion al boton buscaButton

@Override

public void actionPerformed(ActionEvent e) {//metodo que sirve para realizar la funcion dentro de las llaves

int xd = Integer.parseInt(JOptionPane.showInputDialog("Ingrese el número a buscar"));//aqui el valor entero xd va a tomar el valor ingresado en la ventana

resultado = busquedaBinaria(numbers,xd);//resultado va a tomar el valor del metodo busquedaBinaria

resultBusqueda();//metodo que muestran los resultados de la busqueda

}//fin del metodo actionPerformed

});//fin del metodo buscaButton.addActionListener

add(text);

add(text1);

add(text2);

add(resultArea);

add(resultArea1);

add(InputField);

add(InputField1);

add(addButton);

add(simularButton);

add(RadixButton);

add(buscaButton);

add(salirButton);

}

private void addNumero(){

try{

int number= Integer.parseInt(InputField.getText());

InputField.setText("");

if(numbers==null){

numbers = new int[]{number};

}else{

int[] newArray = new int[numbers.length+1];

System.arraycopy(numbers, 0, newArray, 0, numbers.length);//linea que pega los datos del array en la tabla

newArray[numbers.length]=number;

numbers=newArray;

}

updateResultArea();

//

}catch(NumberFormatException e){

JOptionPane.showMessageDialog(this, "Error al ingresar un Numero","Error",JOptionPane.ERROR\_MESSAGE);

}

}

private void addName(){

String nombre = InputField1.getText();

InputField1.setText("");

if(name==null){

name = new String[]{nombre};

}else{

String[] newArray = new String[name.length+1];

System.arraycopy(name, 0, newArray, 0, name.length);//linea que pega los datos del array en la tabla

newArray[name.length]= nombre;

name=newArray;

}

updateResultArea();

//

}

private void simulateBubbleSort(){

if(numbers !=null&& numbers.length > 1){

Burbuja.sort(numbers);

updateResultArea();

}else{

JOptionPane.showMessageDialog(this, "Agregue al menos dos numeros antes de simular el ordenamiento.", "Advertencia", JOptionPane.WARNING\_MESSAGE);

}

}

private void simulateRadixSort(){

if(numbers !=null&& numbers.length > 1){

Radix.ordenacionRadix(numbers);

updateResultArea();

}else{

JOptionPane.showMessageDialog(this, "Agregue al menos dos numeros antes de simular el ordenamiento.", "Advertencia", JOptionPane.WARNING\_MESSAGE);

}

}

private void updateResultArea(){

StringBuilder sb = new StringBuilder();

StringBuilder names = new StringBuilder();

names.append("Nombre: \n");

for(String nam:name){

names.append(nam).append(" ");

}

sb.append("Notas: \n");

for(int num:numbers){

sb.append(num).append(" ");

}

resultArea1.setText(names.toString());

resultArea.setText(sb.toString());

}

public int busquedaBinaria(int elementos[], int x){//Este método toma como parámetros un arreglo de elementos

//ordenados y un valor a buscar, y devuelve la posición del valor en el arreglo o -1 si no se encuentra.

//declaracion y inicializacion de las variables donde elementos[] es el array utilizado y x es el valor separado

int l = 0, r = elementos.length - 1; //l inicializa en 0 y r en la cantidad de elementos del array menos uno

//l significa Left (izquierda) osea el valor de la posicion inicial del array

//r significa Right (derecha) osea el valor de la posicion del tamaño del array menos 1

while (l <= r) {// buque while (mientras) que nos dice que va a seguir haciendose mientras l sea menor o igual a r

//se declara un valor entero "m" que sera igual a la operacion l + (r - l) / 2

int m = l + (r - l) / 2;

//un if que nos dice que devolvera verdadero si la posicion m del array sea igual a x

if (elementos[m] == x)

//en caso de ser positivo el valor regresado sera m

return m;

//if que nos dice que en caso de que el valor de la posicion de m del array sea menor que x

if (elementos[m] < x)

//este hara que l sea igual al valor de m + 1

l = m + 1;

// else (demas) que nos dice que si el ultimo if que se realizo tambien debe realizar

else

//Que r tome el valor de m - 1

r = m - 1;

}

//al cerrar el bucle while este retornara el valor de -1 y si ninguna de las otras funciones se cumplia se retorna

return -1;

}//fin del metodo busqueda binaria

private void resultBusqueda(){//metodo que muestra el resultBusqueda

if (resultado == -1)//if que verifica que resultado no sea igual a -1

System.out.println("No se ha encontrado el elementos buscado");//muestra en consola que no se encontro el elemento buscado

else//sino

System.out.println("Elemento encontrado en la posicion: "+ resultado );//muestra que es en contro el elemento y la posicion en el que esta

}//fin del metodo resultBusqueda()

}  
  
Codigo MenuPrincipal:

package evaluacion1\_u2\_juanj;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.SwingUtilities;

public class MenuPrincipal extends JFrame {

private JTextField InputField;

private JButton ingreButton;

private JButton salirButton;

private JTextArea resultArea;

private JLabel text;

private JLabel text1;

private JLabel text3;

OrdenarBurbuja burbujaingre = new OrdenarBurbuja();

private int[] numbers;

public MenuPrincipal(){

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setTitle("Evaluacion 1 U2 JuanJ");

setSize(600,400);

setLayout(null);

InputField = new JTextField();

ingreButton = new JButton("Ingresar");

salirButton = new JButton("Salir");

text = new JLabel("==========MENU PRINCIPAL==========");

text1 = new JLabel("Pulse el boton Ingresar para ingresar los datos del estudiante");

text3 = new JLabel("Pulse el boton de salir para cerrar el programa");

resultArea = new JTextArea();

InputField.setBounds(20,20,150,30);

text.setBounds(150, 20, 300, 30);

text1.setBounds(100, 70, 500, 30);

text3.setBounds(100, 110, 500, 30);

ingreButton.setBounds(75, 200, 100, 50);

salirButton.setBounds(225, 200, 100, 50);

numbers= new int[0];

ingreButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

SwingUtilities.invokeLater(new Runnable(){//nos dice que con la libreria SwingUtilities crea un nuevo ejecutable

@Override

public void run(){//metodo que con ayuda de la libreria ejecuta lo siguiente

new OrdenarBurbuja();//esto dice que crea una ventana con el contenido de la clase OrdenarBurbuja()

burbujaingre.setVisible(true);

dispose();

}//fin del metodo run()

});//fin del SwuingUtilities

}

});

salirButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

dispose();

burbujaingre.dispose();

}

});

add(text);

add(text1);

add(text3);

add(ingreButton);

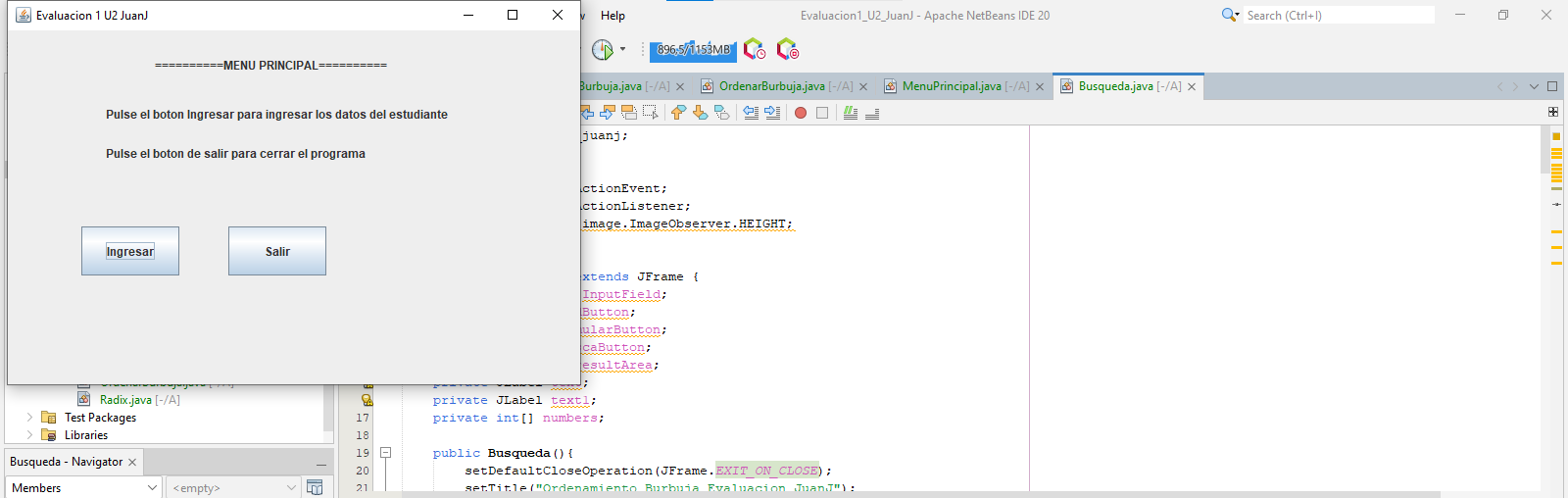
add(salirButton);

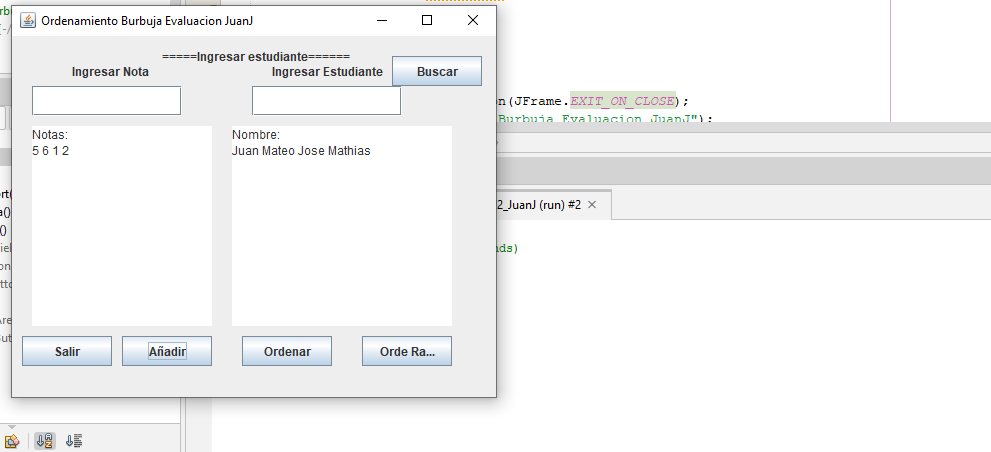
setVisible(true);

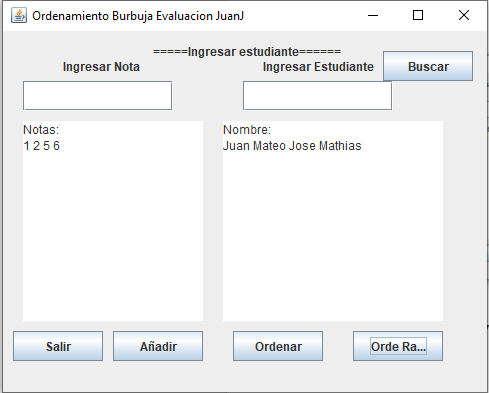
}

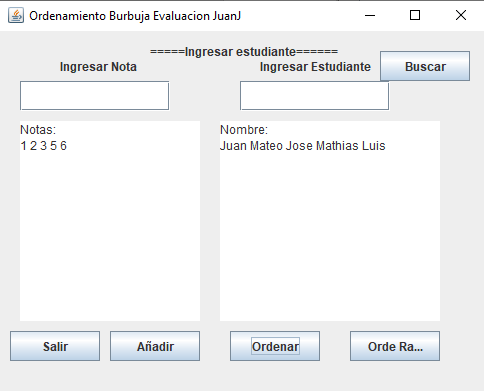
}

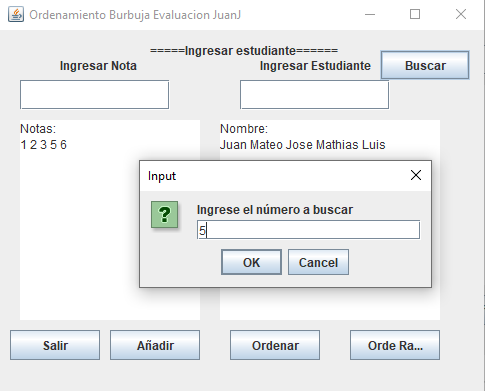
Ejecucion del programa:

  
El programa ejecuta correctamente al inicio muestra un menú principal con 2 opciones para ingresar y otra para salir

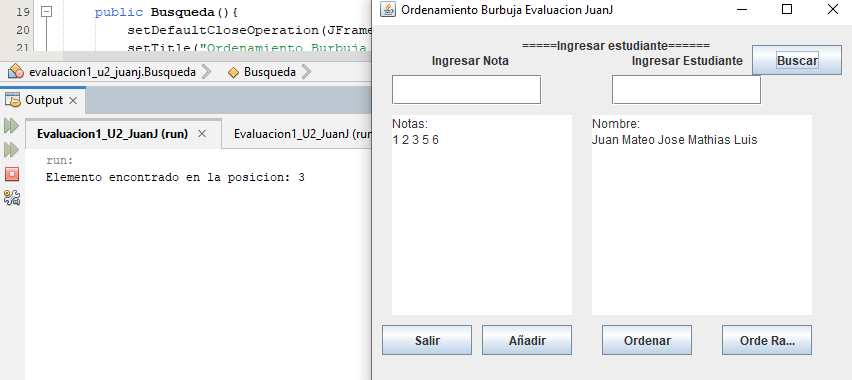
  
Se abre otra ventana al darle al botón de ingresar para subir los datos de los estudiantes y las notas

  
se le uso el botón para ordenar por radix para ver su ejecución en la ordenación de notas

  
se añadió al estudiante Luis con una nota de 3 y se le ordeno esta vez por burbuja



Se dio a la opción de buscar al numero 5

  
y nos muestra q está en la posición 3 del array

Falto poner otro tipo de ordenamiento externo y otro tipo de búsqueda. Y acomodar el programa los botones y el diseño. Además de mostrar en ventana la búsqueda.