

USAC

Manual técnico

GRAPH-USAC

Cristian Daniel Gomez Escobar

202107190

Practica 2, IPC

Clase principal

Librerías

```
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;
import javax.swing.table.DefaultTableModel;
import org.jfree.chart.ChartRenderingInfo;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.entity.StandardEntityCollection;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
import org.json.simple.JSONValue;
```

Ventana principal

Aquí creo la ventana principal que tendrá mi programa, es donde se podrá visualizar todo

```
57     iniciars = true;
58     second = 0;
59     tpasos = 0;
60     pantalla = new JFrame("PRACTICA 2");
61     pantalla.setSize(990, 650);
62     pantalla.setBackground(Color.white);
63     pantalla.setLayout(null);
64     pantalla.setLocationRelativeTo(null); //centre la ventana
65     pantalla.setDefaultCloseOperation(EXIT_ON_CLOSE);
66     pantalla.addWindowListener(new WindowAdapter() {
67         public void WindowClosing(WindowEvent WindowEnvent) {
68             System.exit(0);
69         }
70     });
```

Etiquetas

Estas mostraran texto en pantalla como lo son la dirección de archivo o el cronometro

```
file = new JLabel("", JLabel.LEFT);
file.setFont(new Font("Ubuntu", Font.BOLD, 20));
file.setBounds(10, 20, 760, 30);
file.setForeground(Color.black);
file.setBackground(Color.LIGHT_GRAY);
file.setVisible(true);

JLabel titulo1 = new JLabel("TITULO: ", JLabel.LEFT);
titulo1.setFont(new Font("Ubuntu", Font.BOLD, 20));

titulo1.setForeground(Color.BLACK);
titulo1.setBounds(10, 60, 760, 30);
titulo1.setVisible(true);

title = new JLabel("", JLabel.LEFT);
title.setFont(new Font("Ubuntu", Font.BOLD, 20));
title.setForeground(Color.black);
title.setBounds(100, 60, 250, 30);
title.setBackground(Color.LIGHT_GRAY);
title.setVisible(true);
```

Botón Examinar

Este botón nos permitirá buscar el archivo json y poder visualizar el botón ordenar

```
JButton cargar = new JButton("Examinar");// creo mi boton
cargar.setBounds(800, 20, 140, 30);//le doy posicion a mi boton
// cargar.setBackground(Color.white);
cargar.setFont(new Font("Ubuntu", Font.BOLD, 15));
cargar.setBorder(BorderFactory.createLineBorder(Color.BLACK));
cargar.setVisible(true);
cargar.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
```

Acción

```
String aux = "";
String texto = "";
JFileChooser j = new JFileChooser();
j.showSaveDialog(null);
File abre = j.getSelectedFile();
if (abre != null) {
    file.setText(j.getSelectedFile().getAbsolutePath().toUpperCase());
    FileReader archivos = new FileReader(abre);
    BufferedReader lee = new BufferedReader(archivos);
    while ((aux = lee.readLine()) != null) {
        texto += aux + "\n";
    }
    lee.close();
    Object jsonObyeto = JSONValue.parse(texto);
    JSONObject obyeto = (JSONObject) jsonObyeto;

    Object jsontitleobyeto = obyeto.get("title");
    title.setText(jsontitleobyeto.toString().toUpperCase());
    Object jsonarrayobyeto = obyeto.get("dataset");
    JSONArray arrayobyeto = (JSONArray) jsonarrayobyeto;

    int tam = 0;
    for (Object obyeto_inarray : arrayobyeto) {
        tam++;
    }
    dataset = new int[tam];
    int index = 0;
    for (Object obyeto_inarray : arrayobyeto) {
        dataset[index] = Integer.parseInt(obyeto_inarray.toString());
        index++;
    }
}

pantalla.remove(grafica);
grafica = new clase2(dataset, title.getText());
pantalla.add(grafica);
iniciar.setVisible(true);
```

Botón ordenar

```
iniciar = new JButton("Ordenar");
iniciar.setBounds(800, 60, 140, 30);
// iniciar.setBackground(Color.white);
iniciar.setFont(new Font("Ubuntu", Font.BOLD, 15));
iniciar.setBorder(BorderFactory.createLineBorder(Color.BLACK));
iniciar.setVisible(false);
iniciar.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {
            new Thread(grafica).start();
            iniciar.setVisible(false);
        } catch (Exception ex) {
            System.out.println(ex.toString());
        }
    }
});

pantalla.add(file);
pantalla.add(grafica);
pantalla.add(title);
pantalla.add(titulo1);
pantalla.add(cargar);
pantalla.add(iniciar);
pantalla.setVisible(true);
}
```

Clase 2

Librerías

```
3 import com.itextpdf.text.BaseColor;
4 import java.awt.BorderLayout;
5 import java.awt.Color;
6 import java.awt.Dimension;
7 import java.awt.Font;
8 import java.io.File;
9 import javax.swing.BorderFactory;
10 import javax.swing.JLabel;
11 import javax.swing.JPanel;
12 import org.jfree.chart.ChartFactory;
13 import org.jfree.chart.ChartPanel;
14 import org.jfree.chart.ChartRenderingInfo;
15 import org.jfree.chart.ChartUtilities;
16 import org.jfree.chart.JFreeChart;
17 import org.jfree.chart.entity.StandardEntityCollection;
18 import org.jfree.chart.plot.CategoryPlot;
19 import org.jfree.chart.plot.PlotOrientation;
20 import org.jfree.data.category.DefaultCategoryDataset;
21 import com.itextpdf.text.Chunk;
22 import com.itextpdf.text.Document;
23 import com.itextpdf.text.Image;
24 import com.itextpdf.text.Paragraph;
25 import com.itextpdf.text.FontFactory;
26 import com.itextpdf.text.pdf.PdfTable;
27 import com.itextpdf.text.pdf.PdfWriter;
28 import java.io.BufferedWriter;
29 import java.io.FileOutputStream;
30 import java.io.FileWriter;
31 import java.io.IOException;
32 import java.io.PrintWriter;
33 import java.util.Calendar;
34 import javax.swing.JScrollPane;
35 import javax.swing.JTable;
36 import javax.swing.table.DefaultTableModel;
37 import static proyecto2.principal.mod2;
```

Etiqueta pasos

En esta mostrara la cantidad de pasos que recorre el programa por cada vuelta al momento de ordenar los datos

```
titleLabel = new JLabel("pasos = 0");
titleLabel.setFont(new Font("Ubuntu", Font.BOLD, 20));
titleLabel.setForeground(Color.black);
titleLabel.setPreferredSize(new Dimension(760, 30));
titleLabel.setVisible(true);
add(titleLabel, BorderLayout.PAGE_START); //le indica visualizarse al inicio
```

Grafica

En esta parte crearemos la grafica que mostrara el proceso de ordenamiento, la grafica esta sobre un panel que esta sobre la ventana principal

```
barchart = ChartFactory.createBarChart(
    title, "", "Valor", new DefaultCategoryDataset() //definimos desde cero el data set
    ,
    PlotOrientation.VERTICAL, true, true, true); //
ChartPanel chartPanel = new ChartPanel(barchart);
chartPanel.setPreferredSize(new Dimension(760, 400));
add(chartPanel, BorderLayout.CENTER);
setVisible(true);
}

public void graficar(int[] datos) {
    CategoryPlot plot = (CategoryPlot) barchart.getPlot();
    final DefaultCategoryDataset dataset = new DefaultCategoryDataset(); //esto es lo que vamos a llenar
    for (int k = 0; k < datos.length; k++) {
        dataset.addValue(datos[k], String.valueOf(k), "Datos");
    }
    plot.setDataset(dataset);
}
```

Ordenamiento burbuja

He escogido este ordenamiento ya que he probado con tres ordenamientos diferentes y este es el que menos ha tardado en hacer el ordenamiento.

```
for (int i = 0; i < (n - 1); i++) {
    for (int j = 0; j < (n - 1); j++) {
        if (datos[j] > datos[j + 1]) {
            temp = datos[j];
            datos[j] = datos[j + 1];
            datos[j + 1] = temp;
            Thread.sleep(70);

            graficar(datos);

            pasos++;
            title1.setText("pasos:" + pasos);
        }
    }
}
```

Creación del documento pdf

Se añaden 5 párrafos en el pdf, en estos se mostrara el nombre del estudiante, el carnet, los pasos y el tiempo. Posteriormente se mostrara

```
Paragraph parrafo = new Paragraph();
parrafo.setAlignment(Paragraph.ALIGN_LEFT);
parrafo.setFont(FontFactory.getFont("Tahoma", 16, Font.BOLD, BaseColor.BLACK));
parrafo.add("\n Nombre: Cristian Daniel Gomez Escobar ");

Paragraph parraf = new Paragraph();
parraf.setAlignment(Paragraph.ALIGN_LEFT);
parraf.setFont(FontFactory.getFont("Tahoma", 15, Font.BOLD, BaseColor.BLACK));
parraf.add("\n Carnet: 202107190 ");

Paragraph parr = new Paragraph();
parr.setAlignment(Paragraph.ALIGN_LEFT);
parr.setFont(FontFactory.getFont("Tahoma", 15, Font.BOLD, BaseColor.BLACK));
parr.add("\n Pasos:" + pasos);

Paragraph parra = new Paragraph();
parra.setAlignment(Paragraph.ALIGN_LEFT);
parra.setFont(FontFactory.getFont("Tahoma", 15, Font.BOLD, BaseColor.BLACK));
parra.add("\n Tiempo:" + tiempo + "s");

String ruta = System.getProperty("user.home");
PdfWriter.getInstance(document, new FileOutputStream(ruta + "/Desktop/" + a + d + me + h + m + s + ".pdf"));
Image header = Image.getInstance(ruta + "/Desktop/proyectos neathbeans/proyecto1/char.png");
header.scaleToFit(500, 400);
header.setAlignment(Chunk.ALIGN_CENTER);

Paragraph par = new Paragraph();
par.setAlignment(Paragraph.ALIGN_CENTER);
par.add("\n ");
par.setFont(FontFactory.getFont("Tahoma", 18, Font.BOLD, BaseColor.GREEN));

ppas = new PdfPTable(1);
ppas.addCell("\nDatos de entrada");
for (int i = 0; i < tabla2.getRowCount(); i++) {
    String id = tabla2.getValueAt(i, 0).toString();
    ppas.addCell(id);
}
```

Genero las tablas que se utilizaran

```
ppa = new PdfPTable(1);
ppa.addCell("\nDatos ordenados");
for (int i = 0; i < tabla.getRowCount(); i++) {
    String id = tabla.getValueAt(i, 0).toString();
    ppa.addCell(id);
}
```

Creación del documento html

Se creara un documento html posteriormente de hacer el ordenamiento, este contendrá el nombre del estudiante, carnet, pasos, tiempo, grafica del ordenamiento y las tablas de los datos de entrada y ordenados

```
try {

    String ruta = System.getProperty("user.home");
    File f = new File(ruta + "/Desktop/" + a + d + me + h + m + s + ".html");
    BufferedWriter bw = new BufferedWriter(new FileWriter(f));
    String html = "<div><h1>Reporte html</h1><p>Nombre: Cristsian Daniel Gomez Escobar</p>"
        + "<p>Carnet: 202107190 </p>"
        + "<p>Pasos: " + pasos + "</p>"
        + "<p>Tiempo: " + tiempoclae2 + ":" + texecute + "</p>"
        + "<img src=file:///C:/Users/USER/Desktop/proyectos%20neatbeans/proyecto1/char.png >"
        + "<table border=1>"
        + "<tr>"
        + "<td>Datos de entrada</td>"

        + "</tr>"
        + "<tr>"
        + "<td>Datos de en</td>"
        + "</tr>"
        + "</table>"
        + "</div>";

    bw.write(html);
    bw.close();

} catch (IOException e) {
    e.printStackTrace();
}
```

Clase tiempo

Esta clase contendrá las acciones que realizara el cronometro.

Etiqueta inicial

```
public Tiempo() {  
    setPreferredSize(new Dimension(760, 30));  
    setText("0:00");  
    setFont(new Font("Ubuntu", Font.BOLD, 30));  
    setForeground(Color.BLACK);  
    setVisible(true);  
    setAlignmentX(JLabel.HEIGHT);  
}
```

Con un hilo se podrá controlar el tiempo de pausa y en que momento podrá ser interrumpido por el programa, esto será luego de que el ordenamiento termine su trabajo.

```
@Override  
public void run() {  
    setForeground(Color.BLACK);  
    try {  
        int minutes = 0;  
        while (true) {  
            clase2 obj = new clase2();  
            tiempo++;  
            if (tiempo >= 60) {  
                tiempo = 0;  
                minutes++;  
                obj.tiempoclase2++;  
            }  
            int seconds = tiempo;  
            setText(String.format("%d:%02d", minutes, seconds));  
  
            Thread.sleep(1000);  
        }  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
}
```