

UNIVERSIDAD ANDINA DEL CUSCO

FACULTAD DE INGENIERIA Y ARQUITECTURA

CARRERA PROFESIONAL DE INGENIERIA DE SISTEMAS



TEMA:

“Calculadora en JAVA utilizando el POO”

ASIGNATURA: ALGORITMICA Y LABORATORIO DE PROGRAMACION II

DOCENTE: ESPETIA HUAMANGA HUGO

ALUMNO: ZAPATA KIDA CESAR ALVARO

HUAMAN SUTTA ANTONY ESTIF

CUSCO –PERU

2019

Índice:

Presentación:.....	3
Antecedentes:.....	4
Programación orientad a objetos(POO):.....	4
Interfaces graficas o JFrame:	4
Desarrollo de código:.....	5
Clase Principal:	5
Clase Operaciones:	5
Clase formulario:	7
Interface de la calculadora:	10
Lista de cotejos:	10
Bibliografía:	11

Presentación:

En el siguiente trabajo realizamos una calculadora en java usando programación orientada a objetivo, además de utilizar herramientas como son el JFrame de java el cual es una interface gráfica.

Nuestro objetivo es realizar una calculadora que pueda realizar diferentes tipos de operaciones para esto emplearemos las clases y los métodos para poder ejecutar nuestra calculadora.

Las clases que construiremos son una en la cual contendrá las operaciones y una interface.

Antecedentes:

Programación orientad a objetos(POO):

Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos.

Está basado en varias técnicas:

- Herencia.
- Abstracción.
- Polimorfismo.
- Encapsulamiento.

Interfaces graficas o JFrame:

En java para hacer interfaces de usuario se puede utilizar los JFrame, que vienen siendo las ventanas en las que el usuario puede navegar por la aplicación. En esta entrada y las siguientes aprenderemos sobre este elemento y sobre muchos otros.

Desarrollo de código:

Clase Principal:

```
package proyectocalcu;
import Formulario.newpackage.frmProyectoCalcu;
public class ProyectoCalcu {
    public static void main(String[] args) {
        frmProyectoCalcu frm = new frmProyectoCalcu ();
        frm.setVisible(true);
    }
}
```

Clase Operaciones:

```
package proyectocalcu;
public class Operaciones {
    protected double nro1;
    protected double nro2;

    public Operaciones(double nro1, double nro2)
    {
        this.nro1 = nro1;
        this.nro2 = nro2;
    }
    public double getNro1()
    {
        return this.nro1;
    }
    public double getNro2() {
        return this.nro2;
    }
    public void setNro1(double nro1)
    {
        this.nro1 = nro1;
    }
    public void setNro2(double nro2)
    {
        this.nro2 = nro2;
    }

    public double Sumar()
    {
        return this.nro1 + this.nro2;
    }
    public double Restar()
    {
        return this.nro1 - this.nro2;
    }
    public double Multiplicar()
    {
        return this.nro1 * this.nro2;
    }
}
```

```
public double Multiplicar()  
{  
    return this.nro1 * this.nro2;  
}  
public double Dividir()  
{  
    return this.nro1 / this.nro2;  
}  
public double Raiz()  
{  
    return Math.sqrt(this.nro1);  
}  
public double Porcentaje()  
{  
    return (this.nro1*this.nro2)/100;  
}  
public double Inversa()  
{  
    return 1/this.nro1;  
}  
}
```

Clase formulario:

```
package Formulario.newpackage;

import javax.swing.JOptionPane;

public class frmProyectoCalcu extends javax.swing.JFrame {
    private boolean punto=true;
    String nro1,nro2,signo,contenido;
    Double resultado;
    public frmProyectoCalcu() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void txtmostrarActionPerformed(java.awt.event.ActionEvent evt) {
        |
    }

    private void btnNum0ActionPerformed(java.awt.event.ActionEvent evt) {
        txtmostrar.setText(txtmostrar.getText()+"0");
    }

    private void btnNum1ActionPerformed(java.awt.event.ActionEvent evt) {
        txtmostrar.setText(txtmostrar.getText()+"1");
    }

    private void btnNum2ActionPerformed(java.awt.event.ActionEvent evt) {
        txtmostrar.setText(txtmostrar.getText()+"2");
    }

    private void btnNum3ActionPerformed(java.awt.event.ActionEvent evt) {
        txtmostrar.setText(txtmostrar.getText()+"3");
    }
}
```

```

private void btnNum4ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"4");
}

private void btnNum5ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"5");
}

private void btnNum6ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"6");
}

private void btnNum7ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"7");
}

private void btnNum8ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"8");
}

private void btnNum9ActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText(txtmostrar.getText()+"9");
}

private void btnCActionPerformed(java.awt.event.ActionEvent evt) {
    txtmostrar.setText("");
}

```

```

private void btnpuntoActionPerformed(java.awt.event.ActionEvent evt) {
    contenido=txtmostrar.getText();
    if(contenido.length()<=0){
        txtmostrar.setText("0.");
    }else{
        if(txtmostrar.getText().contains(".")){
        }else{
            txtmostrar.setText(txtmostrar.getText()+".");
            punto=false;
        }
    }
}

public static String Operaciones(String valor1,String valor2,String signo){
    Double resultadocalc=0.0;
    String respuesta;
    if(signo.equals("+")){
        resultadocalc=Double.parseDouble(valor1)+Double.parseDouble(valor2);
    }else if(signo.equals("-")){
        resultadocalc=Double.parseDouble(valor1)-Double.parseDouble(valor2);
    }else if(signo.equals("*")){
        resultadocalc=Double.parseDouble(valor1)*Double.parseDouble(valor2);
    }else if(signo.equals("/")){
        resultadocalc=Double.parseDouble(valor1)/Double.parseDouble(valor2);
    }else if(signo.equals("x^y")){
        resultadocalc=Math.pow(Double.parseDouble(valor1),Double.parseDouble(valor2));
    }
    respuesta=resultadocalc.toString();
    return respuesta;
}

private void btnmasmenosActionPerformed(java.awt.event.ActionEvent evt) {
    contenido=txtmostrar.getText();
    if(contenido.length()>0){
        resultado=(-1)*Double.parseDouble(contenido);
        txtmostrar.setText(resultado.toString());
    }
}

```



```
private void btnSumaActionPerformed(java.awt.event.ActionEvent evt) {
    if(!txtmostrar.getText().equals("")){
        nro1=txtmostrar.getText();
        signo="+";
        txtmostrar.setText("");
    }
}
```

```
private void btnRestaActionPerformed(java.awt.event.ActionEvent evt) {
    if(!txtmostrar.getText().equals("")){
        nro1=txtmostrar.getText();
        signo="-";
        txtmostrar.setText("");
    }
}
```

```
private void btnoperActionPerformed(java.awt.event.ActionEvent evt) {
    if(!txtmostrar.getText().equals("")){
        nro1=txtmostrar.getText();
        signo="*";
        txtmostrar.setText(" ");
    }
}
```

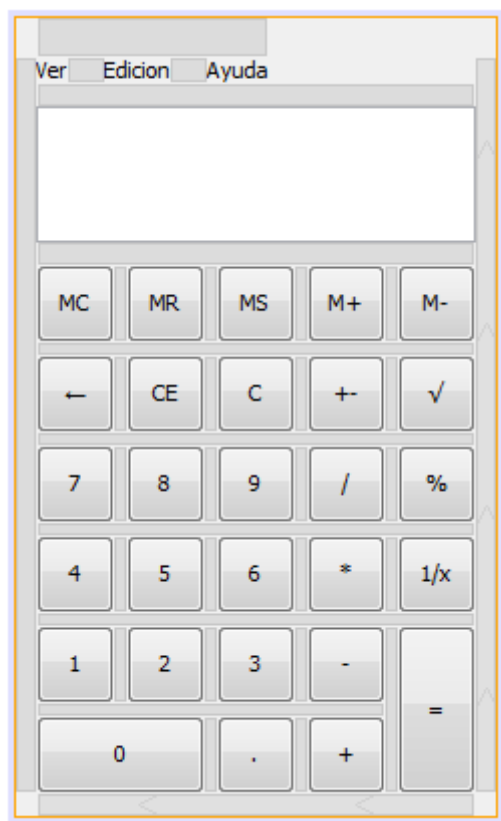
```
private void btnDivicionActionPerformed(java.awt.event.ActionEvent evt) {
    if(!txtmostrar.getText().equals("")){
        nro1=txtmostrar.getText();
        signo="/";
        txtmostrar.setText("");
    }
}
```

```
private void btnigualActionPerformed(java.awt.event.ActionEvent evt) {
    String resultadototal;
    nro2=txtmostrar.getText();
    if(!nro2.equals("")){
        resultadototal=Operaciones(nro1,nro2,signo);
        txtmostrar.setText(resultadototal);
    }
}
```

```
/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new frmProyectoCalcu().setVisible(true);
        }
    });
}
```

Interface de la calculadora:



Lista de cotejos:

Funcionalidad	Si	No
Sumar	✓	
Restar	✓	
Multiplicar	✓	
Dividir	✓	
Raíz	✓	
Inversa	✓	
Porcentaje	✓	
Limpiar	✓	
Memory Clear(MC)		✓
Memory Recall(MR)		✓
Memory Storage(MS)		✓
M+		✓
M-		✓

Conclusiones:

Después de realizar este trabajo de la calculadora obtuvimos nuevos conocimientos en la programación orientada a objetos(POO), y de las interfaces graficas (JFrame) q nos ayudaran en el futuro de nuestro entorno en la programación

Aprendimos generar clases y programar las interfaces con la herramienta JFrame haciendo uso de botones, espacio de texto y etiquetas.

Bibliografía:

- Anthony Castillo. (2012). Programación Orientada a Objetos. 22/08/2019, de EcuRed Sitio web: https://www.ecured.cu/Programaci%C3%B3n_Orientada_a_Objeto
- Cesar Rivera Pacheco. (2015). Java: Swing: JFrame. 22/08/2019, de Franvarvil Sitio web: <http://franvarvil.blogspot.com/2012/12/jframe-programacion-java.html>