

Taller Individual 2

 Daniel Castillo Giraldo

 dcastillogi@unal.edu.co

POO_Talleres/Taller 2 at main · danielcgiraldo/POO_Talleres

Talleres de Programación Orientada a Objetos 2022-2S - POO_Talleres/Taller 2 at main · danielcgiraldo/POO_Talleres

 https://github.com/danielcgiraldo/POO_Talleres/tree/main/Taller%202

danielcgiraldo/

POO_Talleres

Talleres de Programación Orientada a Objetos
2022-2S



 1 Contributor  0 Issues  0 Stars  0 Forks



Capítulo 3

Ejercicios Propuestos

[Ejercicio 18](#)

[Ejercicio 19](#)

Capítulo 4

Ejercicios Resueltos

[Ejercicio 7](#)

[Ejercicio 10](#)

Ejercicios Propuestos

[Ejercicio 22](#)

[Ejercicio 23](#)

[Ejercicio 40](#)

[Ejercicio 41](#)

Partes 2

[Codificación](#)

[Diagrama de clases](#)

Capítulo 3

Ejercicios Propuestos

Ejercicio 18

Clase Principal

```
package poo.cap3propuesto18;

public class Cap3Propuesto18 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}
```

Clase Empleado

```
package poo.cap3propuesto18;

public class Empleado {
```

```
String codigo, nombres;
double num_horas, valor_horas, retefuente, bruto;
public double getBruto() {
    bruto = valor_horas * num_horas;
    return bruto;
}
public double getNeto() {
    return bruto * (1 - retefuente/100);
}
}
```

JFrame

Bienvenido.
Por favor llena los siguientes datos:

Código empleado Nombre completo empleado

Número horas trabajadas Valor hora trabajada Porcentaje retefuente (Sin %)

Continuar

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
boolean empleadoExists = false;
private void btnActionPerformed(java.awt.event.ActionEvent evt) {
    if (empleadoExists) {
        // Set visible again

        txtCode.setEditable(true);
        txtName.setEditable(true);
        txtNumHoras.setEditable(true);
        txtValorHora.setEditable(true);
        txtRetefuente.setVisible(true);
        lblRetefuente.setVisible(true);

        // reset text
        txtCode.setText("");
        txtValorHora.setText("");
        txtNumHoras.setText("");
        txtRetefuente.setText("");
        txtName.setText("");
        lblValorHora.setText("Valor horas trabajadas");
        lblNumHoras.setText("Número horas trabajadas");
        lblDescription.setText("Por favor llena los siguientes datos:");

        // reset button
        btn.setText("Continuar");
        empleadoExists = false;
    } else {
```

```

// GET ALL THE INFO

Empleado empleado = new Empleado();
empleado.codigo = txtCode.getText();
empleado.nombres = txtName.getText();
try {
    empleado.num_horas = Double.parseDouble(txtNumHoras.getText());
    empleado.valor_horas = Double.parseDouble(txtValorHora.getText());
    empleado.retefuente = Double.parseDouble(txtReteFuente.getText());
    // SET ALL THE INFO
    lblDescription.setText(empleado.nombres + '.');
    txtCode.setEditable(false);
    txtName.setEditable(false);
    lblNumHoras.setText("Salario Bruto");
    txtNumHoras.setText(String.valueOf(empleado.getBruto()));
    txtNumHoras.setEditable(false);
    lblValorHora.setText("Salario Neto");
    txtValorHora.setText(String.valueOf(empleado.getNeto()));
    txtValorHora.setEditable(false);

    // HIDE NOT USED
    txtReteFuente.setVisible(false);
    lblReteFuente.setVisible(false);

    // BUTTON CHANGE
    btn.setText("Limpiar");
    empleadoExists = true;
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(null,
        "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
        "Datos Incorrectos",
        JOptionPane.ERROR_MESSAGE);
}

}
}

```

Ejercicio 19

Clase Principal

```

package poo.cap3propuesto19;

public class Cap3Propuesto19 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}

```

Clase Triangulo

```

package poo.cap3propuesto19;

public class Triangulo {
    double lado;
    public double getAltura() {
        return (Math.sqrt(3) / 2) * lado;
    }
    public double getPerimetro() {
        return lado*3;
    }
    public double getArea() {
        return getAltura() * lado / 2;
    }
}

```

Bienvenido. Triángulo Equilátero.

Lado Triángulo

Perímetro

Altura

Área

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
private void txtLadoKeyReleased(java.awt.event.KeyEvent evt) {
    try {
        String txt = txtLado.getText();
        if (!txt.equals("")) {
            Triangulo mitriangulo = new Triangulo();
            mitriangulo.lado = Double.parseDouble(txt);
            txtPerimetro.setText(String.valueOf(mitriangulo.getPerimetro()));
            txtAltura.setText(String.valueOf(mitriangulo.getAltura()));
            txtArea.setText(String.valueOf(mitriangulo.getAltura()));
        } else {
            txtPerimetro.setText("");
            txtAltura.setText("");
            txtArea.setText("");
        }
    }

    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. El lado debe ser un número.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
        txtPerimetro.setText("");
        txtAltura.setText("");
        txtArea.setText("");
        txtLado.setText("");
    }
}
```

Capítulo 4

Ejercicios Resueltos

Ejercicio 7

Clase Principal

```
package poo.cap4resueltos7;

public class cap4resueltos7 {

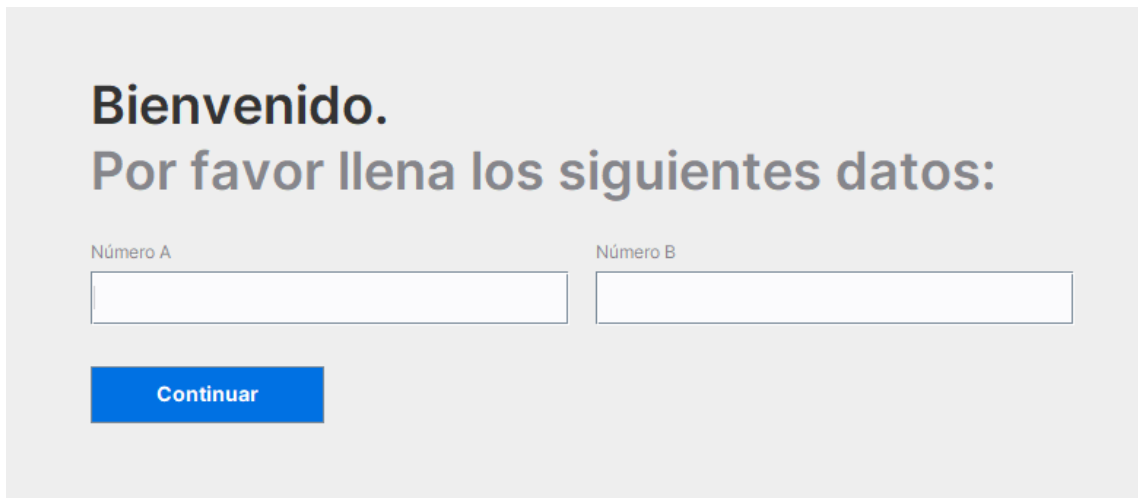
    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}
```

Clase Numeros

```
package poo.cap4resueltos7;

public class Numeros {
    double A, B;
    public String check() {
        if(A > B) return "A es MAYOR que B.";
        else if (A < B) return "A es MENOR que B.";
        else return "A es IGUAL a B.";
    }
}
```

JFrame



The screenshot shows a Java Swing window with a light gray background. At the top, the text "Bienvenido." is displayed in a large, bold, black font. Below it, the text "Por favor llena los siguientes datos:" is displayed in a slightly smaller, bold, black font. Underneath this text, there are two text input fields. The first field is labeled "Número A" and the second field is labeled "Número B". Both fields are empty. Below the input fields, there is a blue button with the text "Continuar" in white.

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
boolean procesoHecho = false;
private void btnActionPerformed(java.awt.event.ActionEvent evt) {
    if (procesoHecho) {
        // Set visible again

        lblDescription.setText("Por favor llena los siguientes datos:");
        txtA.setEditable(true);
        txtB.setEditable(true);
        txtA.setText("");
        txtB.setText("");

        // reset button
    }
}
```

```

        btn.setText("Continuar");
        procesoHecho = false;
    } else {
        // GET ALL THE INFO
        try {
            Numeros nuevonumero = new Numeros();
            nuevonumero.A = Double.parseDouble(txtA.getText());
            nuevonumero.B = Double.parseDouble(txtB.getText());

            lblDescription.setText(nuevonumero.check());
            txtA.setEditable(false);
            txtB.setEditable(false);
            // BUTTON CHANGE
            btn.setText("Limpiar");
            procesoHecho = true;
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null,
                "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
                "Datos Incorrectos",
                JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

Ejercicio 10

Clase Principal

```

package poo.cap4resuelto10;

public class Cap4Resuelto10 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}

```

Clase Estudiante

```

package poo.cap4resuelto10;

public class Estudiante {
    String codigo, nombres;
    double patrimonio, estrato;
    public double pago() {
        if(patrimonio > 2000000 & estrato > 3) {
            return 50.000 + patrimonio * 0.03;
        } else return 50.000;
    }
}

```

JFrame

Bienvenido.

Por favor llena los siguientes datos:

Número de inscripción

Nombre Completo

Patrimonio

Estrato Social

Continuar

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
boolean estudianteExists = false;
private void btnActionPerformed(java.awt.event.ActionEvent evt) {
    if (estudianteExists) {
        // Set visible again

        lblEstrato.setVisible(true);
        txtEstrato.setVisible(true);

        txtName.setEditable(true);
        txtCode.setEditable(true);
        txtEstrato.setEditable(true);
        txtPatrimonio.setEditable(true);

        // reset text
        txtName.setText("");
        txtCode.setText("");
        txtEstrato.setText("");
        txtPatrimonio.setText("");
        lblPatrimonio.setText("Patrimonio");
        lblDescription.setText("Por favor llena los siguientes datos:");

        // reset button
        btn.setText("Continuar");
        estudianteExists = false;
    } else {
        // GET ALL THE INFO

        try {
            Estudiante nuevoestudiante = new Estudiante();
            nuevoestudiante.nombres = txtName.getText();
            nuevoestudiante.codigo = txtCode.getText();
            lblDescription.setText(nuevoestudiante.nombres);
            nuevoestudiante.estrato = Double.parseDouble(txtEstrato.getText());
            nuevoestudiante.patrimonio = Double.parseDouble(txtPatrimonio.getText());
            lblPatrimonio.setText("Valor Matricula");
            txtPatrimonio.setText(String.valueOf(nuevoestudiante.pago()));

            txtName.setEditable(false);
            txtCode.setEditable(false);
            txtEstrato.setEditable(false);
            txtPatrimonio.setEditable(false);
        } catch (Exception e) {
            // Handle exception
        }
    }
}
```

```

        // HIDE NOT USED
        lblEstrato.setVisible(false);
        txtEstrato.setVisible(false);

        // BUTTON CHANGE
        btn.setText("Limpiar");
        estudianteExists = true;
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
    }
}

}

```

Ejercicios Propuestos

Ejercicio 22

Clase Principal

```

package poo.cap4propuesto22;

public class Cap4Propuesto22 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}

```

Clase Empleado

```

package poo.cap4propuesto22;

public class Empleado {
    String nombre;
    double num_horas, salario;
    public double salario_mensual() {
        return salario * num_horas;
    }
}

```

JFrame

Bienvenido.

Por favor llena los siguientes datos:

Nombre empleado

Salario básico

Número de horas

Continuar

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
boolean empleadoExists = false;

private void btnActionPerformed(java.awt.event.ActionEvent evt) {
    if (empleadoExists) {
        // Set visible again

        txtName.setEditable(true);
        txtNumHoras.setEditable(true);
        txtSalario.setEditable(true);
        txtName.setText("");
        txtNumHoras.setText("");
        txtSalario.setText("");
        spaceName.setText("");
        titleName.setText("");
        spaceSalario.setText("");
        titleSalario.setText("");
        lblDescription.setText("Por favor llena los siguientes datos:");

        // reset button
        btn.setText("Continuar");
        empleadoExists = false;
    } else {
        // GET ALL THE INFO

        Empleado empleado = new Empleado();
        empleado.nombre = txtName.getText();
        try {
            empleado.salario = Double.parseDouble(txtSalario.getText());
            empleado.num_horas = Double.parseDouble(txtNumHoras.getText());
            // SET ALL THE INFO

            txtName.setEditable(false);
            txtNumHoras.setEditable(false);
            txtSalario.setEditable(false);
            spaceName.setText(empleado.nombre);
            titleName.setText("Nombre:");

            if (empleado.salario_mensual() > 450000) {
                titleSalario.setText("Salario Mensual:");
                spaceSalario.setText(String.valueOf(empleado.salario_mensual()));
            }
        }
    }
}
```

```

        lblDescription.setText("Información Empleado.");

        // BUTTON CHANGE
        btn.setText("Limpiar");
        empleadoExists = true;
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
    }
}

}

}

```

Ejercicio 23

Clase Principal

```

package poo.cap4propuesto23;

public class cap4propuesto23 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}

```

Clase Ecuacion

```

package poo.cap4propuesto23;

public class Ecuacion {
    double A,B,C;
    public double getDiscriminante(){
        return Math.pow(B, 2) - 4*A*C;
    }
    public double getOne() {
        return -B / 2*A;
    }
    public double getSecondPart() {
        return getDiscriminante()/(2*A);
    }
}

```

JFrame

Bienvenido.
Sea $Ax^2 + Bx + C$, Ingresa:

A	B	C
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Continuar"/>		

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
boolean solved = false;
private void btnActionPerformed(java.awt.event.ActionEvent evt) {
    if (solved) {
        // Set visible again

        txtA.setEditable(true);
        txtB.setEditable(true);
        txtC.setEditable(true);
        txtA.setVisible(true);
        txtB.setVisible(true);
        txtC.setVisible(true);
        lblA.setVisible(true);
        lblB.setVisible(true);
        lblC.setVisible(true);

        // reset text
        txtA.setText("");
        txtB.setText("");
        txtC.setText("");
        lblDescription.setText("Sea  $Ax^2 + Bx + C$ , Ingresa:");
        lblA.setText("A");
        lblB.setText("B");

        // reset button
        btn.setText("Continuar");
        solved = false;
    } else {
        // GET ALL THE INFO

        Ecuacion cuadratica = new Ecuacion();
        try {
            cuadratica.A = Double.parseDouble(txtA.getText());
            cuadratica.B = Double.parseDouble(txtB.getText());
            cuadratica.C = Double.parseDouble(txtC.getText());
            // SET ALL THE INFO
            lblDescription.setText("Sea " + cuadratica.A + " $x^2$  + " + cuadratica.B + "x + " + cuadratica.C + ". Solución");
            lblA.setText("x1");
            lblB.setText("x2");

            txtA.setEditable(false);
            txtB.setEditable(false);
            txtC.setEditable(false);

            if (cuadratica.getDiscriminante() < 0) {
                txtA.setText("INFINITAS SOLUCIONES");
                txtB.setVisible(false);
            }
        } catch (Exception e) {
            // Handle exception
        }
    }
}
```

```

        lblB.setVisible(false);
    } else if (cuadratica.getDiscriminante() == 0) {
        txtA.setText(String.valueOf(cuadratica.getOne()));
        txtB.setVisible(false);
        lblB.setVisible(false);

    } else {
        txtA.setText(String.valueOf(cuadratica.getOne() + cuadratica.getSecondPart()));
        txtB.setText(String.valueOf(cuadratica.getOne() - cuadratica.getSecondPart()));
    }

    // HIDE NOT USED
    txtC.setVisible(false);
    lblC.setVisible(false);

    // BUTTON CHANGE
    btn.setText("Limpiar");
    solved = true;
} catch (NumberFormatException e) {
    JOptionPane.showMessageDialog(null,
        "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
        "Datos Incorrectos",
        JOptionPane.ERROR_MESSAGE);
}

}

}

```

Ejercicio 40

Clase Principal

```

package poo.Cap4Propuesto40;

public class Cap4Propuesto40 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}

```

Clase Operaciones

```

package poo.Cap4Propuesto40;

import javax.swing.JOptionPane;

public class Operaciones {

    String grupo;
    String[] lista;

    public void genLista() {
        lista = grupo.split(",");
    }

    public String raiz() {
        try {
            String a = "";
            for (String lista1 : lista) {
                a += String.valueOf(Math.sqrt(Double.parseDouble(lista1))) + ",";
            }
            return removeLast(a);
        } catch (NumberFormatException e) {

```

```

        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
        return "";
    }
}

public String cuadrado() {
    try {
        String a = "";
        for (String lista1 : lista) {
            a += String.valueOf(Math.pow(Double.parseDouble(lista1), 2)) + ",";
        }
        return removelast(a);
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
        return "";
    }
}

public String cubo() {
    try {
        String a = "";
        for (String lista1 : lista) {
            a += String.valueOf(Math.pow(Double.parseDouble(lista1), 3)) + ",";
        }
        return removelast(a);
    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null,
            "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
            "Datos Incorrectos",
            JOptionPane.ERROR_MESSAGE);
        return "";
    }
}

private String removelast(String a) {
    StringBuilder sb = new StringBuilder(a);
    sb.deleteCharAt(sb.length() - 1);
    return sb.toString();
}
}

```

JFrame

Bienvenido.

Sigue las instrucciones a continuación:

Digite un conjunto de números positivos separados por comas. Ejemplo: 1,2,3,4.

Continuar

Resultados.

Raíces cuadradas de cada número

Cuadrados de cada número

Cubos de cada número

Limpiar

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```
private void btnContinuarActionPerformed(java.awt.event.ActionEvent evt) {  
    Operaciones op = new Operaciones();  
    op.grupo = txtInput.getText();  
    op.genLista();  
  
    if (!op.cuadrado().equals("")) {  
        txtRaices.setText((op.raiz()));  
        txtCuadrados.setText((op.cuadrado()));  
        txtCubos.setText((op.cubo()));  
    }  
  
    txtRaices.setEditable(false);  
    txtCuadrados.setEditable(false);  
    txtCubos.setEditable(false);  
}  
  
private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {  
    txtRaices.setText("");  
    txtCuadrados.setText("");  
    txtCubos.setText("");  
    txtInput.setText("");  
}
```

Ejercicio 41

Clase Principal

```
package poo.Cap4Propuesto41;

public class Cap4Propuesto41 {

    public static void main(String[] args) {
        UI miventana = new UI();
        miventana.setVisible(true);
    }
}
```

Clase Operaciones

```
package poo.Cap4Propuesto41;

import javax.swing.JOptionPane;

public class Operaciones {

    String grupo;
    double mayor;

    public String genMayor() {
        try {
            String[] p = grupo.split(",");
            for (String lista1 : p) {
                if (Double.parseDouble(lista1) > mayor) {
                    mayor = Double.parseDouble(lista1);
                }
            }
            return String.valueOf(mayor);
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null,
                "Ten cuidado con los datos ingresados. Algun(os) no tienen el tipo de dato correcto.",
                "Datos Incorrectos",
                JOptionPane.ERROR_MESSAGE);
            return "";
        }
    }
}
```

JFrame

Bienvenido.

Sigue las instrucciones a continuación:

Digite un conjunto de números positivos separados por comas. Ejemplo: 1,2,3,4.

Continuar

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerated por netbeans de la interfaz gráfica.

```
boolean mayorExists = false;

private void btnContinuarActionPerformed(java.awt.event.ActionEvent evt) {
    if (mayorExists) {
        lblDescription.setText("Sigue las instrucciones a continuación:");
        txtInput.setText("");
        txtInput.setEditable(true);
        mayorExists = false;
        btnContinuar.setText("Continuar");
    } else {
        Operaciones op = new Operaciones();
        op.grupo = txtInput.getText();

        if (!(op.genMayor().equals("")) {
            lblDescription.setText("El número mayor es: " + op.genMayor());
        }
        txtInput.setEditable(false);
        mayorExists = true;
        btnContinuar.setText("Limpiar");
    }
}
```

Partes 2

Codificación

Clase Principal

```
package poo.parte2;

public class Parte2 {

    public static void main(String[] args) {
        UI ventana = new UI();
        ventana.setVisible(true);
    }
}
```


Paquete Figuras

```
package figuras;

public class Circulo {
    public double radio;
    public double area(){
        double area = Math.PI*(radio*radio);
        return area;
    }

    public double perimetro(){
        double perimetro = Math.PI*2*radio;
        return perimetro;
    }
}
```

```
package figuras;

public class Cuadrado {
    public double lado;
    public double area(){
        double area = lado * lado;
        return area;
    }

    public double perimetro(){
        double perimetro = (4*lado);
        return perimetro;
    }
}
```

```
package figuras;

public class Rectangulo {
    public double base, altura;

    public double area(){
        double area = base * altura;
        return area;
    }
    public double perimetro(){
        double perimetro = (2*base) + (2*altura);
        return perimetro;
    }
}
```

```
package figuras;

public class Triangulo {
    public double base, altura, hipotenusa;
    public double area(){
        double area = (base*altura)/2;
        return area;
    }

    public double perimetro(){
        double perimetro = hipotenusa + base + altura;
        return perimetro;
    }

    public double hipotenusa(){
        hipotenusa = Math.sqrt((altura*altura)+(base*base));
    }
}
```

```

        return hipotenusa;
    }
    public String tipoTriangulo(){
        if(hipotenusa == altura && altura == base) return "Equilátero";
        else if(hipotenusa == altura || altura == base) return "Isóceles";
        else return "Escaleno";
    }
}

```

JFrame

The image shows a Java Swing window titled 'JFrame' containing four panels for calculating geometric shapes:

- Círculo:** Input field for 'Radio', 'Continuar' button, 'Área' input field, 'Perímetro' input field, and 'Limpiar' button.
- Rectángulo:** Input fields for 'Base' and 'Altura', 'Continuar' button, 'Área' input field, 'Perímetro' input field, and 'Limpiar' button.
- Cuadrado:** Input field for 'Lado', 'Continuar' button, 'Área' input field, 'Perímetro' input field, and 'Limpiar' button.
- Triángulo Rectángulo:** Input fields for 'Base', 'Altura', 'Continuar' button, 'Área' input field, 'Perímetro' input field, 'Hipotenusa' input field, 'Tipo Triángulo' input field, and 'Limpiar' button.

Nota: Para evitar sobrecargar este documento, pondré el código realizado por mí sin agregar el autogenerado por netbeans de la interfaz gráfica.

```

private void circuloBtnActionPerformed(java.awt.event.ActionEvent evt) {

    Circulo micirculo = new Circulo();
    micirculo.radio = Double.parseDouble(circuloRadio.getText());
    circuloArea.setText(String.valueOf(micirculo.area()));
    circuloPerimetro.setText(String.valueOf(micirculo.perimetro()));

}

private void circuloBtn1ActionPerformed(java.awt.event.ActionEvent evt) {
    circuloArea.setText("");
    circuloPerimetro.setText("");
    circuloRadio.setText("");
}

private void rectanguloBtnActionPerformed(java.awt.event.ActionEvent evt) {
    Rectangulo mirectangulo = new Rectangulo();
    mirectangulo.altura = Double.parseDouble(rectanguloAltura.getText());
    mirectangulo.base = Double.parseDouble(rectanguloBase.getText());
}

```

```

        rectanguloArea.setText(String.valueOf(mirectangulo.area()));
        rectanguloPerimetro.setText(String.valueOf(mirectangulo.perimetro()));
    }

    private void rectanguloBtnActionPerformed(java.awt.event.ActionEvent evt) {
        rectanguloPerimetro.setText("");
        rectanguloArea.setText("");
        rectanguloAltura.setText("");
        rectanguloBase.setText("");
    }

    private void cuadradoBtnActionPerformed(java.awt.event.ActionEvent evt) {
        Cuadrado micuadrado = new Cuadrado();
        micuadrado.lado = Double.parseDouble(cuadradoLado.getText());
        cuadradoArea.setText(String.valueOf(micuadrado.area()));
        cuadradoPerimetro.setText(String.valueOf(micuadrado.perimetro()));
    }

    private void cuadradoBtnActionPerformed(java.awt.event.ActionEvent evt) {
        cuadradoArea.setText("");
        cuadradoPerimetro.setText("");
        cuadradoLado.setText("");
    }

    private void trianguloBtnActionPerformed(java.awt.event.ActionEvent evt) {
        Triangulo mitriangulo = new Triangulo();
        mitriangulo.altura = Double.parseDouble(trianguloAltura.getText());
        mitriangulo.base = Double.parseDouble(trianguloBase.getText());
        trianguloHipotenusa.setText(String.valueOf(mitriangulo.hipotenusa()));
        trianguloPerimetro.setText(String.valueOf(mitriangulo.perimetro()));
        trianguloArea.setText(String.valueOf(mitriangulo.area()));
        trianguloTipo.setText(mitriangulo.tipoTriangulo());
    }

    private void trianguloBtnActionPerformed(java.awt.event.ActionEvent evt) {
        trianguloHipotenusa.setText("");
        trianguloPerimetro.setText("");
        trianguloArea.setText("");
        trianguloTipo.setText("");
        trianguloAltura.setText("");
        trianguloBase.setText("");
    }
}

```

Diagrama de clases

