

Programación Orientada a Objetos (POO)
Actividad 3

Presentado por:

Danny Johan Santa Ruiz - dsantar@unal.edu.co

Cc 1000900382

Grupo 3

Profesor:

Arboleda Mazo Walter Hugo

awalter@unal.edu.co

2024-1S.



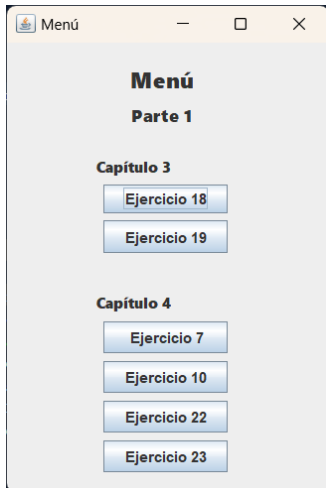
Universidad Nacional de Colombia
Facultad de Minas
Sede Medellín

Nota: Se omitirá el código que genera la interfaz debido a su gran longitud, únicamente se mostrará el código del método que ocurre al presionar el botón de calcular.

Parte 1: Usando una interfaz gráfica, creación de clases, métodos y atributos, realice los siguientes ejercicios del libro de lógica de programación de Efraín Oviedo.

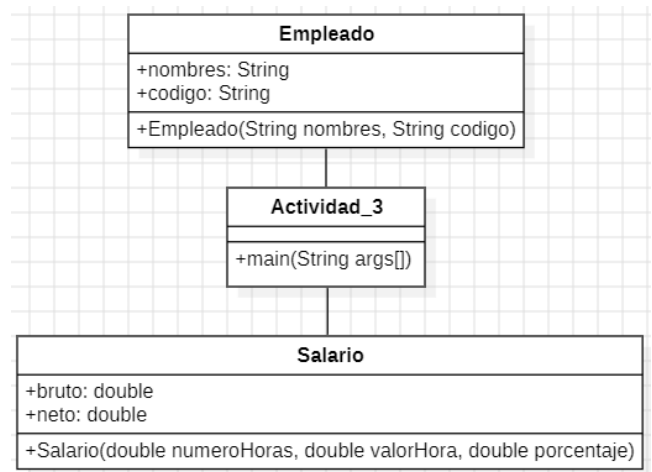
URL: https://github.com/DannyKerim/POO/tree/main/Actividad_3

Menú generado para todos los ejercicios de la parte 1:



```
package logica;
import igu.Menu;
public class Actividad_3 {
    public static void main(String[] args) {
        //Crear pantalla
        Menu pantalla = new Menu();
        //Visualizar pantalla
        pantalla.setVisible(true);
        pantalla.setLocationRelativeTo(null);
    }
}
```

Ejercicio 18.



Clases:

```
package logica;

public class Salario {
    public double bruto;
    public double neto;
    //Constructor
    public Salario(double numeroHoras, double valorHora, double porcentaje){
        bruto = numeroHoras*valorHora;
        neto = bruto - (bruto*porcentaje/100);
    }
}
```

```
package logica;

public class Empleado {
    public String nombres;
    public String codigo;
    //Constructor
    public Empleado(String nombres,String codigo){
        this.nombres = nombres;
        this.codigo = codigo;
    }
}
```

Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    //Guardar los datos de usuario
    String nombres = nombresEmpleado.getText();
    String codigo = codEmpleado.getText();
    float numeroHoras = Float.parseFloat(numeroHorasEmpleado.getText());
    float valorHora = Float.parseFloat(valorHoraEmpleado.getText());
    float porcentaje = Float.parseFloat(porcentajeEmpleado.getText());

    //Crear instancias
    Salario Salario = new Salario(numeroHoras, valorHora, porcentaje);
    Empleado Empleado = new Empleado(nombres,codigo);

    //Mostrar en pantalla
    viCodEmpleado.setText(Empleado.codigo);
    viNombresEmpleado.setText(Empleado.nombres);
    viBrutoEmpleado.setText(String.valueOf(Salario.bruto));
    viNetoEmpleado.setText(String.valueOf(Salario.neto));
}
```

Ejercicio 19.

Ejercicio 19

Dado el valor del lado en un triángulo equilátero, haga un algoritmo que obtenga:

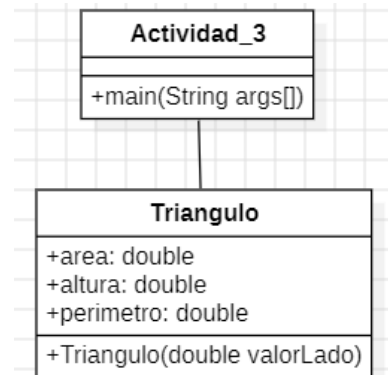
- El perímetro.
- El valor de la altura.
- El área del triángulo.

Valor del lado del triángulo equilátero.

5

Perímetro.	Valor de la altura.	Área del triángulo.
15,00	4,33	10,83

Calcular Limpiar Menú



Clases:

```
package logica;

public class Triangulo {
    public double altura;
    public double area;
    public double perimetro;
    //Constructor
    public Triangulo(double valorLado){
        perimetro = 3*valorLado;
        area = Math.pow(valorLado, 2) * Math.sqrt(3) / 4;
        altura = valorLado * Math.sqrt(3) / 2;
    }
}
```

Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    //Guardar los datos de usuario
    double valorLado = Float.parseFloat(txtValorLado.getText());

    //Crear instancias
    Triangulo Triangulo = new Triangulo(valorLado);

    //Mostrar en pantalla
    viAltura.setText(String.format("%.2f",Triangulo.altura));
    viArea.setText(String.format("%.2f",Triangulo.area));
    viPerimetro.setText(String.format("%.2f",Triangulo.perimetro));
}
```

Ejercicio 7.

Ejercicio 7

Hacer un algoritmo que, dados dos valores numéricos A y B:
escriba un mensaje diciendo si A es mayor, menor o igual a B.

Número A. Número B.

3 2

A es mayor que B

Calcular Limpiar Menú

Actividad_3

+main(String args[])

Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    //Guardar los datos de usuario  
    float A = Float.parseFloat(numA.getText());  
    float B = Float.parseFloat(numB.getText());  
  
    //Mostrar en pantalla  
    if (A > B) {  
        viResultado.setText("A es mayor que B");  
    } else if (A < B) {  
        viResultado.setText("A es menor que B");  
    } else {  
        viResultado.setText("A que es igual que B");  
    }  
}
```

Ejercicio 10.

Ejercicio 10

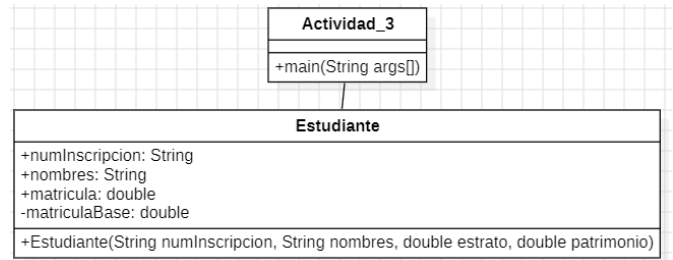
Cierta universidad para liquidar el pago de matrícula de un estudiante le exige los siguientes datos:

- Número de inscripción.
- Nombres.
- Patrimonio.
- Estrato social.

La universidad cobra un valor constante para cada estudiante de \$50.000. Si el patrimonio es mayor que \$2'000.000 y el estrato superior a 3, se le incrementa un porcentaje del 3% sobre el patrimonio. Hacer un algoritmo que muestre:

- Número de inscripción.
- Nombres.
- Pago de matrícula.

Número de inscripción.	Número de inscripción.
<input type="text" value="1"/>	<input type="text" value="1"/>
Nombres.	Nombres.
<input type="text" value="Danny"/>	<input type="text" value="Danny"/>
Patrimonio.	Pago de matrícula.
<input type="text" value="2500000"/>	<input type="text" value="\$125000.00"/>
Estrato social.	
<input type="text" value="4"/>	



Clases:

```
package logica;
public class Estudiante {
    public String numInscripcion;
    public String nombres;
    public double matricula;
    double matriculaBase = 50000;
    //Constructor
    public Estudiante(String numInscripcion, String nombres, double estrato, double patrimonio){
        this.numInscripcion = numInscripcion;
        this.nombres = nombres;
        if (estrato > 3 && patrimonio > 2000000){
            matricula = matriculaBase + 0.03*patrimonio;
        }else{
            matricula = matriculaBase;
        }
    }
}
```

Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    //Guardar los datos de usuario
    String numInscripcion = txtNumInscripcion.getText();
    String nombres = txtNombres.getText();
    float estrato = Float.parseFloat(txtEstrato.getText());
    float patrimonio = Float.parseFloat(txtPatrimonio.getText());

    //Generar instancias
    Estudiante Estudiante = new Estudiante(numInscripcion, nombres, estrato, patrimonio);

    //Mostrar en pantalla
    viNumInscripcion.setText(Estudiante.numInscripcion);
    viNombres.setText(Estudiante.nombres);
    viMatricula.setText(String.format("$%.2f", Estudiante.matricula));
}
```

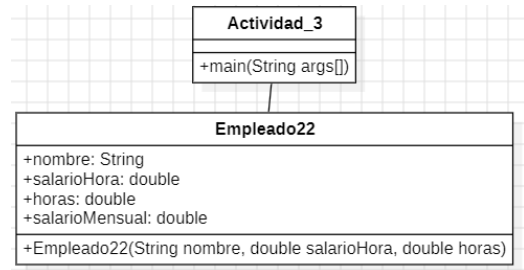
Ejercicio 22.

Ejercicio 22

Elaborar un algoritmo que entre el nombre de un empleado, su salario básico por hora y el número de horas trabajadas en el mes; escriba su nombre y salario mensual si éste es mayor de \$450.000, de lo contrario escriba sólo el nombre.

Nombre.	Salario básico.	Número horas.
<input type="text" value="Danny"/>	<input type="text" value="20000"/>	<input type="text" value="30"/>

Nombre.	Salario mensual.
<input type="text" value="Danny"/>	<input type="text" value="\$600000,00"/>



Clases:

```
package logica;

public class Empleado22 {
    public String nombre;
    public double salarioMensual;
    public double salarioHora;
    public double horas;
    //Constructor
    public Empleado22(String nombre, double salarioHora, double horas){
        this.nombre = nombre;
        this.salarioHora = salarioHora;
        this.horas = horas;
        salarioMensual = salarioHora*horas;
    }
}
```

Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {
    //Guardar los datos de usuario
    String nombre = txtNombre.getText();
    float salarioHora = Float.parseFloat(txtSalarioHora.getText());
    float horas = Float.parseFloat(txtHoras.getText());

    //Generar instancias
    Empleado22 Empleado22 = new Empleado22(nombre, salarioHora, horas);

    //Mostrar en pantalla
    viNombre.setText(Empleado22.nombre);
    if (Empleado22.salarioMensual > 450000){
        viSalarioMensual.setText(String.format("%.2f", Empleado22.salarioMensual));
    }else{
        viSalarioMensual.setText(" ");
    }
}
```

Ejercicio 23.

Ejercicio 23

Dados los valores A, B y C que son los parámetros de una ecuación de segundo grado, elaborar un algoritmo para hallar las posibles soluciones de dicha ecuación.

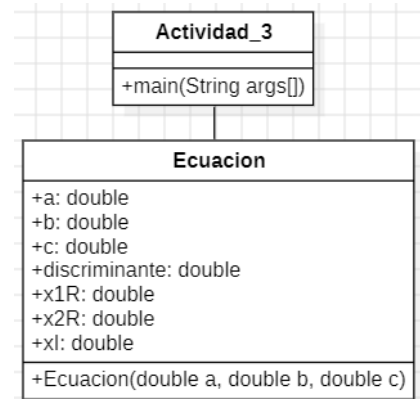
Ecuación de la forma $Ax^2+Bx+C=0$

Número A: 1 Número B: 1 Número C: 1

Resultado

Las soluciones son $x1 = -0,50+0,87i$ y $x2 = -0,50-0,87i$

Calcular Limpiar Menú



Clases:

```
package logica;
public class Ecuacion {
    public double a; public double b; public double c;
    public double discriminante;
    public double x1R; public double x2R; public double xI;
    //Constructor
    public Ecuacion(double a, double b, double c){
        this.a = a;
        this.b = b;
        this.c = c;
        discriminante = b * b - (4 * a * c);
        if (discriminante >= 0) {
            // Dos soluciones reales
            x1R = (-b + Math.sqrt(discriminante)) / (2 * a);
            x2R = (-b - Math.sqrt(discriminante)) / (2 * a);
            xI = 0;
        } else {
            // Soluciones imaginarias
            x1R = (-b) / (2 * a);
            x2R = x1R;
            xI = Math.sqrt(-discriminante) / (2 * a);
        }
    }
}
```


Interfaz:

```
private void btnCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
    //Guardar los datos de usuario  
    double a = Float.parseFloat(txtNumA.getText());  
    double b = Float.parseFloat(txtNumB.getText());  
    double c = Float.parseFloat(txtNumC.getText());  
  
    //Generar instancias  
    Ecuacion Ecuacion = new Ecuacion(a,b,c);  
  
    //Mostrar en pantalla  
    viResultado.setText("Las soluciones son x1 = " + String.format("%.2f", Ecuacion.x1) + "+" + String.format("%.2f", Ecuacion.x2) +  
    "i" + " y x2 = " + String.format("%.2f", Ecuacion.x2) + "-" + String.format("%.2f", Ecuacion.x1) + "i");  
}
```

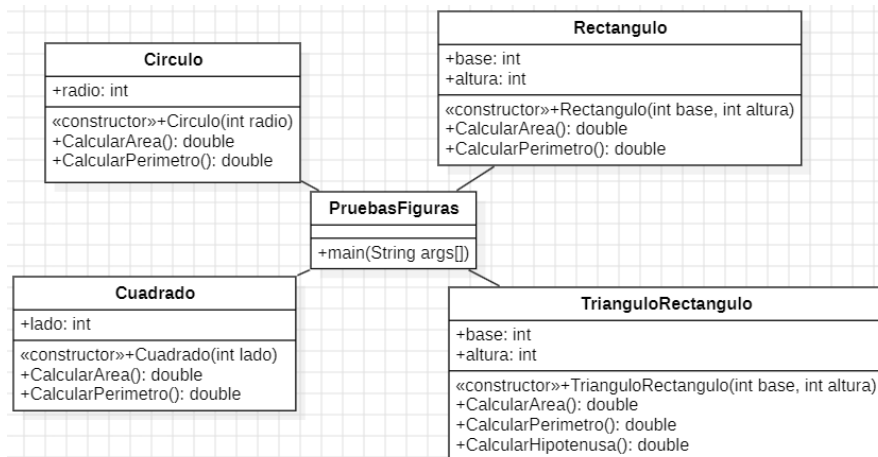
Parte 2.

Realizar el ejercicio “clases sobre figuras geométricas” del libro Ejercicios de programación orientada a objetos con Java y UML de Leonardo Bermón Angarita.

URL: https://github.com/DannyKerim/POO/tree/main/Actividad_3_2

The screenshot shows a Java Swing window titled "Parte 2" with a light gray background. It contains four columns of input fields and calculated values for different geometric shapes. The columns are labeled "Círculo", "Cuadrado", "Rectángulo", and "Triángulo Rectángulo". Each column has input fields for "Radio" (Círculo), "Lado" (Cuadrado), "Base" (Rectángulo), and "Base" (Triángulo Rectángulo). Below these are calculated values for "Área" and "Perímetro". At the bottom, there are "Calcular" and "Limpiar" buttons. A "Tipo" label is also present with a text box showing "Es un triángulo isósceles".

Círculo	Cuadrado	Rectángulo	Triángulo Rectángulo
Radio	Lado	Base	Base
2	2	2	2
		Altura	Altura
		2	2
Área	Área	Área	Área
12,57	4,00	4,00	2,00
Perímetro	Perímetro	Perímetro	Perímetro
12,57	8,00	8,00	6,83
			Tipo
			Es un triángulo isósceles



Main:

```
package logica;
import igu.Pantalla;
public class PruebaFiguras {
    public static void main(String[] args) {
        //Crear pantalla
        Pantalla pantalla = new Pantalla();
        //Visualizar pantalla
        pantalla.setVisible(true);
        pantalla.setLocationRelativeTo(null);
    }
}
```

Clases:

```
package logica;
public class Circulo {
    float radio; // Atributo que define el radio de un círculo
    // Constructor que recibe el radio de un círculo
    public Circulo(float radio) {
        this.radio = radio;
    }
    // Metodo para calcular el area de un círculo
    public double calcularArea() {
        return Math.PI*Math.pow(radio,2);
    }
    // Metodo para calcular el perimetro de un círculo
    public double calcularPerimetro() {
        return 2*Math.PI*radio;
    }
}
```

```
package logica;
public class Cuadrado {
    float lado; // Atributo que define el lado de un cuadrado
    // Constructor que recibe el lado de un cuadrado
    public Cuadrado(float lado) {
        this.lado = lado;
    }
    // Metodos para calcular el area de un cuadrado
    public double calcularArea() {
        return lado*lado;
    }
    // Metodo para calcular el perimetro de un cuadrado
    public double calcularPerimetro() {
        return (4*lado);
    }
}
```

```

package logica;

public class Rectangulo {
    float base; // Atributo que define la base de un rectangulo
    float altura; // Atributo que define la altura de un rectangulo
    // Constructor que recibe la base y la altura de un rectangulo
    public Rectangulo(float base, float altura) {
        this.base = base;
        this.altura = altura;
    }
    // Metodo que calcula el area de un rectangulo
    public double calcularArea() {
        return base * altura;
    }
    // Metodo que calcula el perimetro de un rectangulo
    public double calcularPerimetro() {
        return (2 * base) + (2 * altura);
    }
}

```

```

package logica;

public class TrianguloRectangulo {
    float base; // Atributo que define la base de un triangulo rectangulo
    float altura; // Atributo que define la altura de un triangulo rectangulo
    // Constructor que recibe la base y la altura de un triangulo rectangulo
    public TrianguloRectangulo(float base, float altura) {
        this.base = base;
        this.altura = altura;
    }
    // Metodo que calcula el area de un triangulo rectangulo
    public double calcularArea() {
        return (base * altura / 2);
    }
    // Metodo que calcula el perimetro de un triangulo rectangulo
    public double calcularPerimetro() {
        return (base + altura + calcularHipotenusa()); /* Invoca al metodo calcular hipotenusa */
    }
}

```

```

// Metodo que calcula la hipotenusa de un triangulo rectangulo
public double calcularHipotenusa() {
    return Math.pow(base*base + altura*altura, 0.5);
}
// Metodo que determina el tipo de triangulo rectangulo
public String determinarTipoTriangulo() {
    String tipoTriangulo;
    if ((base == altura) && (base == calcularHipotenusa()) && (altura == calcularHipotenusa()))
        tipoTriangulo = "Es un triángulo equilátero"; // Todos sus lados son iguales
    else if ((base != altura) && (base != calcularHipotenusa()) && (altura != calcularHipotenusa()))
        tipoTriangulo = "Es un triángulo escaleno"; // Todos sus lados son diferentes
    else
        tipoTriangulo = "Es un triángulo isósceles"; // De otra manera, es isosceles;
    return tipoTriangulo;
}

```

Interfaz:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    float valorRadio = Float.parseFloat(radio.getText());  
    float valorLado = Float.parseFloat(lado.getText());  
    float valorBaseRectangulo = Float.parseFloat(baseRectangulo.getText());  
    float valorAlturaRectangulo = Float.parseFloat(alturaRectangulo.getText());  
    float valorBaseTriangulo = Float.parseFloat(baseTriangulo.getText());  
    float valorAlturaTriangulo = Float.parseFloat(alturaTriangulo.getText());  
  
    Circulo figura1 = new Circulo(valorRadio);  
    Cuadrado figura2 = new Cuadrado(valorLado);  
    Rectangulo figura3 = new Rectangulo(valorBaseRectangulo, valorAlturaRectangulo);  
    TrianguloRectangulo figura4 = new TrianguloRectangulo(valorBaseTriangulo, valorAlturaTriangulo);  
  
    double areaCirculo = figura1.calcularArea();  
    double periCirculo = figura1.calcularPerimetro();  
    double areaCuadrado = figura2.calcularArea();  
    double periCuadrado = figura2.calcularPerimetro();  
    double areaRectangulo = figura3.calcularArea();  
    double periRectangulo = figura3.calcularPerimetro();  
    double areaTriangulo = figura4.calcularArea();  
    double periTriangulo = figura4.calcularPerimetro();  
  
    viAreaCirculo.setText(String.format("%.2f", areaCirculo));  
    viPeriCirculo.setText(String.format("%.2f", periCirculo));  
    viAreaCuadrado.setText(String.format("%.2f", areaCuadrado));  
    viPeriCuadrado.setText(String.format("%.2f", periCuadrado));  
    viAreaRectangulo.setText(String.format("%.2f", areaRectangulo));  
    viPeriRectangulo.setText(String.format("%.2f", periRectangulo));  
    viAreaTriangulo.setText(String.format("%.2f", areaTriangulo));  
    viPeriTriangulo.setText(String.format("%.2f", periTriangulo));  
    viTipoTriangulo.setText(figura4.determinarTipoTriangulo());  
}
```