



Programación Orientada a Objetos Docente:

PhD(c) Walter Arboleda

Estudiante: Andres Felipe Lemus Victoria

[Lemus2901/Actividad-3-POO-2024: 1er semestre \(github.com\)](#)

Actividad 3: Individual– Valor 10%

Fecha de entrega archivo PDF 4 de Abril

Correo de envío de la actividad ia.walterarboleda@gmail.com

Parte 1: Esta parte consta de 8 ejercicios en java los cuales deber ser realizados usando interfaz gráfica y creación de clases, métodos y atributos.

18)

	Notas	Creditos
Materia 1	<input type="text"/>	<input type="text"/>
Materia 2	<input type="text"/>	<input type="text"/>
Materia 3	<input type="text"/>	<input type="text"/>
Materia 4	<input type="text"/>	<input type="text"/>
Total	<input type="text"/>	<input type="text"/>

```
/**
 *
 * @author pipe
 */
You, 8 minutes ago | 1 author (You)
public class EstudianteManager {
    You, 8 minutes ago | You, 8 minutes ago | 1 author (You) | 1 author (You)
    public static class Result {
        public double totalCredits;
        public double creditAverage;

        public Result(double totalCredits, double creditAverage) {
            this.totalCredits = totalCredits;
            this.creditAverage = creditAverage;
        }
    }

    public Result procesarEstudiantes(int studentCode, double credits1, double credits2, double credits3, double credits4,
                                     double grade1, double grade2, double grade3, double grade4) {
        double totalCredits = credits1 + credits2 + credits3 + credits4;
        double creditAverage = (grade1 * credits1 + grade2 * credits2 + grade3 * credits3 + grade4 * credits4) / totalCredits;
        return new Result(totalCredits, creditAverage);
    }
}
```

19)

```
package com.mycompany.app19;

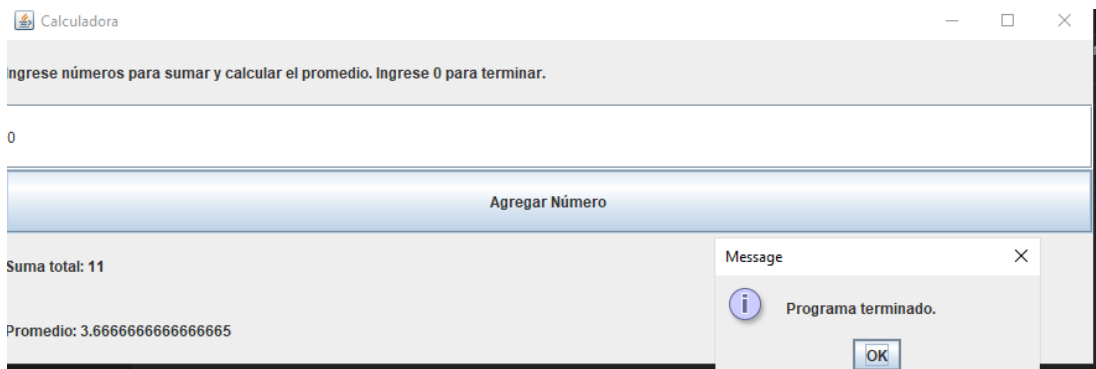
You, 9 minutes ago | 1 author (You)

class Calculadora {
    private int suma = 0;
    private int contador = 0;

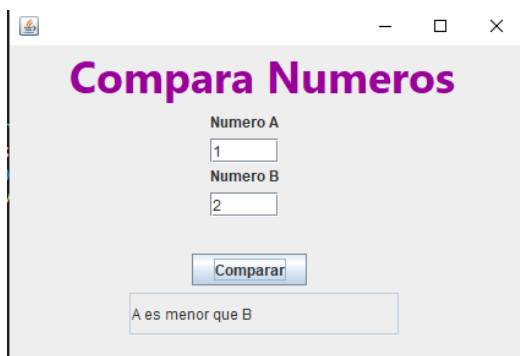
    public void agregarNumero(int numero) {
        suma += numero;
        contador++;
    }

    public int obtenerSuma() {
        return suma;
    }

    public double calcularPromedio() {
        if (contador > 0) {
            return (double) suma / contador;
        } else {
            return 0;
        }
    }
}
```



7)



You, 14 minutes ago | 1 author (You)

```
package com.mycompany.app7;

public class ComparadorNumeros {
    public static String comparar(int A, int B) {
        String resultado;
        if (A > B) {
            resultado = "A es mayor que B";
        } else if (A < B) {
            resultado = "A es menor que B";
        } else {
            resultado = "A es igual a B";
        }
        return resultado;
    }
}
```

You, 14 minutes ago | 1 author (You)

10)

```
public class AppMatricula {
    private int numeroInscripcion;
    private String nombres;
    private double patrimonio;
    private int estratoSocial;
    private double pagoMatricula;

    // Constructor
    public AppMatricula(int numeroInscripcion, String nombres, double patrimonio, int estratoSocial, double pagoMatricula) {
        this.numeroInscripcion = numeroInscripcion;
        this.nombres = nombres;
        this.patrimonio = patrimonio;
        this.estratoSocial = estratoSocial;
        this.pagoMatricula = pagoMatricula;

        // Lógica para calcular el pago de matrícula
        if (this.patrimonio > 2000000 && this.estratoSocial > 3) {
            this.pagoMatricula += this.pagoMatricula * 0.10; // Incrementa en 10%
        }
    }

    // Getters and Setters (optional, if needed)
    public int getNumeroInscripcion() {
        return numeroInscripcion;
    }

    public void setNumeroInscripcion(int numeroInscripcion) {
        this.numeroInscripcion = numeroInscripcion;
    }

    public String getNombres() {
        return nombres;
    }

    public void setNombres(String nombres) {
        this.nombres = nombres;
    }

    public double getPatrimonio() {
        return patrimonio;
    }

    public void setPatrimonio(double patrimonio) {
        this.patrimonio = patrimonio;
    }

    public int getEstratoSocial() {
        return estratoSocial;
    }

    public void setEstratoSocial(int estratoSocial) {
        this.estratoSocial = estratoSocial;
    }

    public double getPagoMatricula() {
        return pagoMatricula;
    }

    public void setPagoMatricula(double pagoMatricula) {
        this.pagoMatricula = pagoMatricula;
    }
}
```

numero Inscripcion

2

Nombres

Andres

Patrimonio

20000000

Estracto Social

3

Resultado

Pago Matricula

50000.0

22)

```
You, 18 minutes ago | 1 author (You)
class Empleado {
    String nombre;
    double salarioPorHora;
    int horasTrabajadas;

    public Empleado(String nombre, double salarioPorHora, int horasTrabajadas) {
        this.nombre = nombre;
        this.salarioPorHora = salarioPorHora;
        this.horasTrabajadas = horasTrabajadas;
    }

    public double calcularSalarioMensual() {
        return salarioPorHora * horasTrabajadas;
    }

    public void mostrarInformacion() {
        double salarioMensual = calcularSalarioMensual();
        if (salarioMensual > 450000) {
            System.out.println("Nombre: " + nombre + ", Salario Mensual: $" + salarioMensual);
        } else {
            System.out.println("Nombre: " + nombre);
        }
    }
}
```

Empleado

Nombre del empleado:	Andres
Salario por hora:	3
Horas trabajadas en el mes:	2
Calcular Salario	Salario: 6.0Nombre: Andres

23)

```
You, 22 minutes ago | 1 author (You)
package com.mycompany.app23;
You, 22 minutes ago * trabajo finalizado

You, 22 minutes ago | 1 author (You)
public class Solucion {

    public String resolverEcuacionSegundoGrado(double A, double B, double C) {
        double discriminante = B * B - 4 * A * C;
        double x1, x2;
        String resultado;

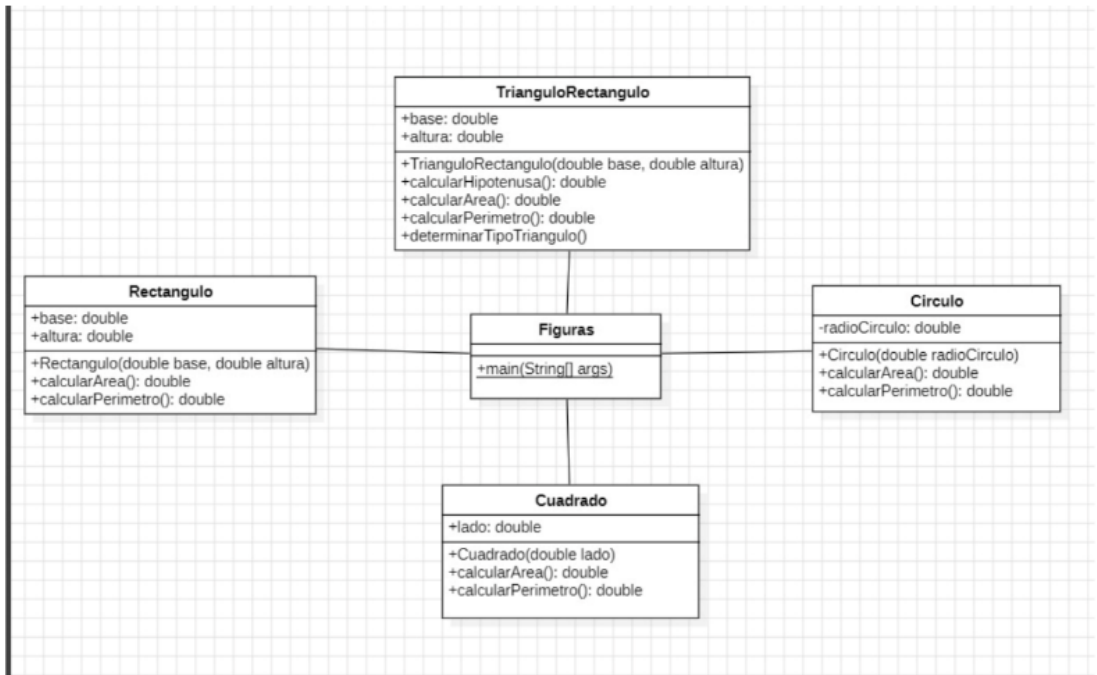
        if (discriminante > 0) {
            x1 = (-B + Math.sqrt(discriminante)) / (2 * A);
            x2 = (-B - Math.sqrt(discriminante)) / (2 * A);
            resultado = "La ecuación tiene dos soluciones reales y distintas: x1 = " + x1 + ", x2 = " + x2;
        } else if (discriminante == 0) {
            x1 = -B / (2 * A);
            resultado = "La ecuación tiene una única solución real: x = " + x1;
        } else {
            resultado = "La ecuación no tiene soluciones reales.";
        }

        return resultado;
    }
}
```

Ecuación de Segundo Grado

Valor de A:	1
Valor de B:	-5
Valor de C:	6
Calcular	La ecuación tiene dos soluciones reales y distintas: x1 = 3.0, x2 = 2.0

Parte 2: Esta parte consta de 2 ejercicios (aplicación con interfaz gráfica y diagrama de clases en StarUML <https://staruml.io/>)



Calculadora de Figuras

Radio del círculo:	4
Base del rectángulo:	2
Altura del rectángulo:	5
Lado del cuadrado:	6
Base del triángulo:	2
Altura del triángulo:	3

Calcular

El área del círculo es = 50.26548245743669

El perímetro del círculo es = 25.132741228718345

El área del rectángulo es = 10.0

```

3  * Click nbfs://nbhost/SystemFileSystem/Templa
4  */
5  package com.mycompany.figuras;
6
7  /**
8   *
9   * @author pipe
10  */
11  public class Circulo {
12      private double radio;
13      public Circulo(double radio){
14          this.radio = radio;
15      }
16      double calcularArea() {
17          return Math.PI*Math.pow(radio,2);
18      }
19      double calcularPerimetro() {
20          return 2*Math.PI*radio;
21      }
22  }

```

```

7
package com.mycompany.figuras;

/**
 *
 * @author Administrator
 */
public class TrianguloRectangulo {
    public double base;
    public double altura;

    public TrianguloRectangulo(double base, double altura) {
        this.base = base;
        this.altura = altura;
    }

    public double calcularArea() {
        return (base * altura / 2);
    }

    public double calcularPerimetro() {
        return (base + altura + calcularHipotenusa());
    }

    public double calcularHipotenusa() {
        return Math.sqrt(base * base + altura * altura);
    }

    public void determinarTipoTriangulo() {
        if (base == altura && base == calcularHipotenusa()) && altura == calcularHipotenusa()
            System.out.println("Es un triángulo equilateral");
        else if (base != altura && base != calcularHipotenusa() && altura != calcularHipotenusa())
            System.out.println("Es un triángulo escaleno");
        else
            System.out.println("Es un triángulo isósceles");
    }
}

```

```

2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes
4  */
5  package com.mycompany.figuras;
6
7
8  public class Rectangulo {
9      public double base; // Atributo que define la base de
10     public double altura; // Atributo que define la altura
11
12     public Rectangulo(double base, double altura) {
13         this.base = base;
14         this.altura = altura;
15     }
16
17     public double calcularArea() {
18         return base * altura;
19     }
20
21     public double calcularPerimetro() {
22         return (2 * base) + (2 * altura);
23     }
24 }

```

```

1  package com.mycompany.figuras;
2
3  /**
4   * Clase que representa un cuadrado.
5   *
6   */
7
8  public class Cuadrado {
9      public double lado;
10
11     public Cuadrado(double lado) {
12         this.lado = lado;
13     }
14
15     double calcularArea() {
16         return lado * lado;
17     }
18
19     double calcularPerimetro() {
20         return (4 * lado);
21     }
22 }

```