

Tarea 1 - Clase - Método y atributos o campos

1. Finaliza la clase Coche explicada en teoría con todos sus métodos getters y setters. Recuerda que sus atributos o campos son:

```
public class Coche {  
  
    private int puertas;  
    private double peso;  
    private double precio;  
    private String motor;  
    private String modelo;  
    private String color;  
    private String marca;
```

```
public class Coche {  
    private int puertas;  
    private double peso;  
    private double precio;  
    private String motor;  
    private String modelo;  
    private String color;  
    private String marca;  
  
    public int getPuertas() {  
        return puertas;  
    }  
  
    public void setPuertas(int puertas) {  
        this.puertas = puertas;  
    }  
  
    public double getPeso() {  
        return peso;  
    }  
  
    public void setPeso(double peso) {  
        this.peso = peso;  
    }  
  
    public double getPrecio() {  
        return precio;  
    }  
}
```

```
public void setPrecio(double precio) {
    this.precio = precio;
}

public String getMotor() {
    return motor;
}

public void setMotor(String motor) {
    this.motor = motor;
}

public String getModelo() {
    return modelo;
}

public void setModelo(String modelo) {
    this.modelo = modelo;
}

public String getColor() {
    return color;
}

public void setColor(String color) {
    this.color = color;
}

public String getMarca() {
    return marca;
}

public void setMarca(String marca) {
    this.marca = marca;
}
}
```

2. Escribe una clase con el nombre Calculadora. La clase debe tener dos campos o atributos (variable miembro) con los nombres primerNumero y segundoNumero, ambos de tipo double. Codifica los siguientes métodos:

Getters:

- Método llamado getPrimerNumero sin ningún parámetro, debe devolver el valor del campo primerNumero.
- Método llamado getSegundoNumero sin ningún parámetro, debe devolver el valor del campo segundoNumero.

Setters:

- c. Método llamado setPrimerNumero con un parámetro de tipo double, debe establecer el valor del campo primerNumero.
- d. Método llamado setSegundoNumero con un parámetro de tipo double, debe establecer el valor del campo segundoNumero.

Otros métodos:

- e. Método llamado getResultadoSuma sin ningún parámetro, debe devolver el resultado de sumar los valores de campo de primerNumero y segundoNumero.
 - f. Método llamado getResultadoResta sin ningún parámetro, debe devolver el resultado de restar los valores de campo de segundoNumero del primerNumero.
 - g. Método llamado getMultiplicationResult sin ningún parámetro, debe devolver el resultado de multiplicar los valores de campo de primerNumero y segundoNumero.
 - h. Método llamado getDivisionResult sin ningún parámetro, debe devolver el resultado de dividir los valores de campo de primerNumero por segundoNumero. En caso de que el valor de segundoNumero sea 0, devuelve 0.
- TEST, código para probar la clase Calculadora. Escribe este código en una clase Main, dentro del método main.

```
Calculadora calculator = new Calculadora ();
calculator.setPrimerNumero(5.0);
calculator.setSegundoNumero(4);
System.out.println("suma= " + calculator.getResultadoSuma());
System.out.println("resta= " + calculator.getResultadoResta());
calculator.setPrimerNumero(5.25);
calculator.setSegundoNumero(0);
System.out.println("multiplicación= " +
calculator.getMultiplicationResult());
System.out.println("división= " + calculator.getDivisionResult());
OUTPUT
suma= 9.0
resta= 1.0
multiplicación= 0.0
división= 0.0
```

Comprueba que realiza correctamente todas las operaciones.

Ayuda. En total debes tener 8 métodos en la clase Calculadora. Los métodos NO deben ser public static, solo public.

```

public class Calculadora {

    private double primerNumero;
    private double segundoNumero;

    public double getPrimerNumero() {
        return primerNumero;
    }

    public void setPrimerNumero(double primerNumero) {
        this.primerNumero = primerNumero;
    }

    public double getSegundoNumero() {
        return segundoNumero;
    }

    public void setSegundoNumero(double segundoNumero) {
        this.segundoNumero = segundoNumero;
    }

    public double getResultadoSuma(){
        return primerNumero+segundoNumero;
    }

    public double getResultadoResta(){
        return primerNumero-segundoNumero;
    }

    public double getMultiplicationResult(){
        return primerNumero*segundoNumero;
    }

    public double getDivisionResult(){
        if(segundoNumero==0){
            return 0;
        }else{
            return primerNumero/segundoNumero;
        }
    }
}

```

3. Codifica una clase Persona. La clase debe tener tres campos con los nombres nombre, apellido de tipo String y edad de tipo entero. Escribe los siguientes métodos:

Getters:

- a. Método llamado `getNombre` sin ningún parámetro, debe devolver el valor del campo nombre.
- b. Método llamado `getApellido` sin ningún parámetro, debe devolver el valor del campo apellido.
- c. Método llamado `getEdad` sin ningún parámetro, debe devolver el valor del campo de edad.

Setters:

- d. Método llamado `setNombre` con un parámetro de tipo `String`, debe establecer el valor del campo nombre.
- e. Método llamado `setApellido` con un parámetro de tipo `String`, debe establecer el valor del campo apellido.
- f. Método llamado `setEdad` con un parámetro de tipo `int`, debe establecer el valor del campo de edad. Si el parámetro es menor que 0 o mayor que 100, establece el valor del campo de edad en 0.

Otros métodos:

- g. El método llamado `esAdolescente` sin ningún parámetro, debe devolver verdadero si el valor del campo de edad es mayor que 12 y menor que 20; de lo contrario, devuelve falso.
- h. Método llamado `getFullName` sin ningún parámetro, debe devolver el nombre completo de la persona.
 - i. En caso de que los campos nombre y apellido estén vacíos, las cadenas devuelven una cadena vacía.
 - ii. En caso de que apellido sea una cadena vacía, devuelva nombre.
 - iii. En caso de que nombre sea una cadena vacía, devuelve apellido.

Para verificar si un `String` está vacío, usa el método `isEmpty` de la clase `String`. Por ejemplo, `nombre.isEmpty()` devuelve verdadero si el `String` está vacío o en otras palabras, cuando el `String` no contiene ningún carácter.

TEST CODE (Probar en la clase `Main` dentro del método `main`)

```
Persona persona = new Persona();
persona.setNombre(""); // nombre is set to empty string
persona.setApellido(""); // apellido is set to empty string
```

```
persona.setEdad(10);
System.out.println("fullName= " + persona.getFullName());
System.out.println("adolescente= " + persona.esAdolescente());
persona.setFirstName("Patri"); // nombre is set to Patri
persona.setAge(18);
System.out.println("fullName= " + persona.getFullName());
System.out.println("adolescente = " + persona.esAdolescente());
persona.setLastName("Guerrero"); // se establece apellido a
Guerrero
System.out.println("fullName= " + persona.getFullName());
OUTPUT
```

```
fullName=
adolescente = false
fullName= Patri
adolescente = true
fullName= Patri Guerrero
```

```
public class Persona {

    private String nombre;
    private String apellido;
    private int edad;

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        if(edad < 0 || edad > 100) {
            this.edad = 0;
        } else {
            this.edad = edad;
        }
    }
}
```

```
public boolean esAdolescente(){
    if(edad >12 && edad <20){
        return true;
    }return false;
}

public String getFullName(){
    if(nombre.isEmpty()&&apellido.isEmpty()){
        return " ";
    }else if(nombre.isEmpty()){
        return apellido;
    }else if (apellido.isEmpty()){
        return nombre;
    }return nombre + " " + apellido;
}

}
```

4. ¡Hazlo tú! Crea una clase con sus campos (estado) y métodos (comportamiento) que representen algo en el mundo real. Dentro de los métodos crea diferentes tipos, tanto setters como getters y otros métodos de interés para la clase. Incluye código de prueba para comprobar que funciona correctamente como hemos hecho en los ejercicios anteriores. La clase debe contener más de 1 campo o atributo.