



GUIA DE EJERCICIOS

**POO**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**Es importante tener en cuenta que entender la programación orientada a objetos lleva sobre todo PRÁCTICA, no desesperarse, con cada ejercicio vamos a ir entendiendo un poco más cómo aplicar este paradigma.**

1-Crear una clase llamada Libro que contenga los siguientes atributos: Título, Autor, número de páginas, un constructor con todos los atributos pasados por parámetro y un constructor vacío.

Crear un método para cargar un libro pidiendo los datos al usuario y luego informar mediante otro método el título, el autor del libro y el numero de páginas.

2-Desarrollar una clase Cancion con los siguientes atributos: titulo y autor.

Se deberá definir además dos constructores: uno vacío que inicializa el titulo y el autor y otro que reciba como parámetros el titulo y el autor de la canción.

Se deberán además definir los métodos getters y setters correspondientes.

3- Crear una clase llamada Operación que tenga como atributos privados numero1 y numero2. Crear los siguientes métodos:

- a) Método constructor con todos los atributos pasados por parámetro.
- b) Método constructor vacío.
- c) Métodos get y set.
- d) Método para crearOperacion(): que le pide al usuario los dos números y los guarda en los atributos del objeto.
- e) Método sumar(): calcular la suma de los números y devolver el resultado al main.
- f) Método restar(): calcular la resta de los números y devolver el resultado al main
- g) Método multiplicar(): primero valida que no se haga una multiplicación por cero, si fuera a multiplicar por cero, el método devuelve 0 y se le informa al usuario el error. Si no, se hace la multiplicación y se devuelve el resultado al main
- h) Método dividir(): primero valida que no se haga una división por cero, si fuera a pasar una división por cero, el método devuelve 0 y se le informa al usuario el error. Si no, se hace la división y se devuelve el resultado al main.

4- Crear una clase Rectángulo que contenga atributos privados de base y altura.

La clase incluirá un método para crear el rectángulo con los datos del Rectángulo dados por el usuario.

También incluirá un método para calcular el área del rectángulo.

Se deberán además definir los métodos getters, setters y constructores correspondientes.

$\text{Área} = \text{base} * \text{altura}.$

5. Realizar una clase llamada Cuenta (bancaria) que debe tener como mínimo los atributos: numeroCuenta (entero), el DNI del cliente (entero largo), el saldo actual.

Los métodos a realizar son:

- Constructor por defecto(vacío) y constructor con DNI, saldo, número de cuenta e interés.
- Agregar los métodos getters y setters correspondientes
- Método para crear un objeto Cuenta, pidiéndole los datos al usuario.
- Método ingresar(double ingreso): el método recibe una cantidad de dinero a ingresar y se la sumara a saldo actual.

- Método retirar(double retiro): el método recibe una cantidad de dinero a retirar y se la restará al saldo actual. Si la cuenta no tiene la cantidad de dinero a retirar, se pondrá el saldo actual en 0.
- Método extraccionRapida(): le permitirá sacar solo un 20% de su saldo. Validar que el usuario no saque más del 20%.
- Método consultarSaldo(): permitirá consultar el saldo disponible en la cuenta.
- Método consultarDatos(): permitirá mostrar todos los datos de la cuenta.

6- Programa Nespresso. Desarrolle una clase Cafetera con los atributos capacidadMaxima (la cantidad máxima de café que puede contener la cafetera) y cantidadActual (la cantidad actual de café que hay en la cafetera).

Implemente los siguientes métodos:

- Constructor predeterminado o vacío
- Constructor con la capacidad máxima y la cantidad actual
- Métodos getters y setters.
- Método llenarCafetera(): hace que la cantidad actual sea igual a la capacidad máxima.
- Método servirTaza(int): se pide el tamaño de una taza vacía, el método recibe el tamaño de la taza y simula la acción de servir la taza con la capacidad indicada. Si la cantidad actual de café “no alcanza” para llenar la taza, se sirve lo que quede.

El método le informará al usuario si se llenó o no la taza, y de no haberse llenado en cuanto quedó la taza.

- Método vaciarCafetera(): pone la cantidad de café actual en cero.
- Método agregarCafe(int): se le pide al usuario una cantidad de café, el método lo recibe y se añade a la cafetera la cantidad de café indicada.

7-Realizar una clase llamada Persona que tenga los siguientes atributos: nombre, edad, sexo ('M' hombre, 'F' mujer, 'O' otro), peso y altura. Los métodos que se implementarán son:

- Un constructor por defecto y un constructor con todos los atributos como parámetro.
- Métodos getters y setters de cada atributo.
- Metodo crearPersona(): el método crear persona, le pide los valores de los atributos al usuario y después se le asignan a sus respectivos atributos para llenar el objeto Persona. Además, comprueba que el sexo introducido sea correcto, es decir, M, F o O. Si no es correcto se deberá mostrar un mensaje.
- Método calcularIMC(): calculara si la persona está en su peso ideal ( $\text{peso en kg}/(\text{altura}^2 \text{ en mt}^2)$ ).

Si esta fórmula da por resultado un valor menor que 20, significa que la persona está por debajo de su peso ideal y la función devuelve un mensaje advirtiéndolo.

Si la fórmula da por resultado un número entre 20 y 25 (incluidos), significa que la persona está en su peso ideal y la función devuelve un mensaje de felicitaciones.

Finalmente, si el resultado de la fórmula es un valor mayor que 25 significa que la persona tiene sobrepeso, y la función devuelve otra advertencia.

- Método esMayorDeEdad(): indica si la persona es mayor de edad. La función devuelve un booleano.