

CURSO JAVA 1

Programación Orientada a Objetos

Ejercicios

CLASE SERVICIO

Ejercicios de aprendizaje

A partir de esta guía vamos a realizar todos los ejercicios, utilizando la clase servicio. Recomendamos ver los videos y leer la teoría las veces que sean necesarias, además, apoyarnos en nuestros compañeros si hay un concepto que no entendemos.



VIDEOS: Te sugerimos ver los videos relacionados con este tema, antes de empezar los ejercicios, los podrás encontrar en tu aula virtual o en nuestro canal de YouTube.

1. Realizar una clase llamada CuentaBancaria en el paquete Entidades con los siguientes atributos: numeroCuenta(entero), dniCliente(entero largo), saldoActual. Agregar constructor vacío, con parámetros, getters y setters.

Agregar la clase CuentaBancariaServicio en el paquete Servicios que contenga:

- a) Método para crear cuenta pidiéndole los datos al usuario.
- b) Método ingresar(double): recibe una cantidad de dinero a ingresar y se le sumará al saldo actual.
- c) Método retirar(double): recibe una cantidad de dinero a retirar y se le restará al saldo actual. Si la cuenta no tiene la cantidad de dinero a retirar se retirará el máximo posible hasta dejar la cuenta en 0.
- d) Método extraccionRapida: le permitirá sacar solo un 20% de su saldo. Validar que el usuario no saque más del 20%.
- e) Método consultarSaldo: permitirá consultar el saldo disponible en la cuenta.
- f) Método consultarDatos: permitirá mostrar todos los datos de la cuenta.

2. Programa Nespresso. Desarrolle una clase **Cafetera** en el paquete Entidades con los atributos capacidadMáxima (la cantidad máxima de café que puede contener la cafetera) y cantidadActual (la cantidad actual de café que hay en la cafetera). Agregar constructor vacío y con parámetros así como setters y getters. Crear clase **CafeteraServicio** en el paquete Servicios con los siguiente:

- a) Método llenarCafetera(): hace que la cantidad actual sea igual a la capacidad máxima.
- b) 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.
- c) Método vaciarCafetera(): pone la cantidad de café actual en cero.
- d) 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.

1. Realizar una clase llamada Persona en el paquete de entidades que tengan los siguientes atributos: nombre, edad, sexo ('H' para hombre, 'M' para mujer, 'O' para otro), peso y altura. Si desea añadir algún otro atributo, puede hacerlo. Agregar constructores, getters y setters.

En el paquete Servicios crear PersonaServicio con los siguientes 3 métodos:

- a) Método esMayorDeEdad(): indica si la persona es mayor de edad. La función devuelve un booleano.
- b) 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, H, M o O. Si no es correcto se deberá mostrar un mensaje

c) 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 -1. 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 0. Finalmente, si el resultado de la fórmula es un valor mayor que 25 significa que la persona tiene sobrepeso, y la función devuelve un 1.

A continuación, en la clase `main` hacer lo siguiente:

- Crear **4 objetos de tipo Persona** con distintos valores, a continuación, llamaremos todos los métodos para cada objeto, deberá comprobar si la persona está en su peso ideal, tiene sobrepeso o está por debajo de su peso ideal e indicar para cada objeto si la persona es mayor de edad.
- Por último, guardaremos los resultados de los métodos `calcularIMC` y `esMayorDeEdad` en distintas variables(arrays), para después calcular un porcentaje de esas 4 personas cuantas están por debajo de su peso, cuantas en su peso ideal y cuantos, por encima, y también calcularemos un porcentaje de cuantos son mayores de edad y cuantos menores. Para esto, podemos crear unos métodos adicionales.