

Programación Orientada a Objetos

Práctica N° 2 – 2024

Implementar en JAVA cada uno de los siguientes problemas

1. *Persona*

Implementar una clase llamada Persona con las siguientes características:

Atributos: nombre, edad, fecha de nacimiento, DNI, sexo, peso y altura
Valores por defecto:

- El DNI es un valor obligatorio, no posee valor por defecto.
- En caso de la fecha de nacimiento será el 1 de enero de 2000.
- Sexo será Femenino por defecto.
- El nombre por defecto es N y el apellido es N.
- El peso y la altura son 1 por defecto.

Acorde a lo anterior se deben crear los constructores. Con DNI, con DNI nombre y apellido, con DNI nombre apellido y Fecha de nacimiento.....

La **responsabilidad** de la clase es la siguiente:

- Cálculo del índice de masa Corporal. Una persona sabe calcular cuál es su índice de masa corporal, el cual es $\text{peso} / \text{altura}^2$
- Saber si está en forma. Está en forma si el índice de masa corporal se encuentra entre 18,5 y 25 (esto es solo un ejemplo para probar la funcionalidad).
- Saber si está cumpliendo años.
- Saber si es mayor de edad. Una persona es mayor de edad si es mayor de 18 años
- Métodos set de cada parámetro, excepto de DNI.
- Saber si puede votar. Es necesario ser mayor de 16 años.
- Saber si es coherente. Que la fecha de nacimiento coincida con la edad.
- Mostrar toda la información del objeto. Es decir devolver un String con la información del mismo.

2. *Electrodomésticos*

Crear una clase Electrodoméstico con las siguientes características:

Atributos: nombre, precio base, color, consumo energético y peso.

Valores por defecto:

- El color por defecto es gris plata.
- El consumo energético 10 Kw.
- Precio base 100 pesos.
- El peso es 2 kg.

Implementar todos los constructores.

La **funcionalidad** de la clase es la siguiente:

- Todos los métodos get y set.
- Comprobar si el electrodoméstico es de bajo consumo (menor que 45 Kw)
- Calcular el balance, el mismo es el resultado de dividir el costo por el peso
- Indicar si un producto es de alta gama. En un producto de alta gama el balance es mayor que 3

Programación Orientada a Objetos

Práctica N° 2 – 2024

3. Series

Una Serie está formada por un conjunto de temporadas, cada una de las cuales tiene una cantidad de episodios. Cada episodio posee un título, una descripción, un flag indicando si el usuario ya vio el episodio y una calificación dada por el usuario (con valores de 1 a 5). Si el usuario no vio un episodio particular, la calificación dada será un valor negativo.

Las series poseen como atributos (además de los episodios correspondientes) un título, una descripción, un creador y un género.

Implementar en Java las clases involucradas, determinar qué clase es responsable de responder los siguientes servicios e implementarlos en Java.

- Ingresar la calificación de un episodio. Si el valor ingresado como calificación no es correcto imprimir un mensaje por pantalla y no cambiar el valor anterior.
- Obtener el total de episodios vistos de una temporada particular.
- Obtener el total de episodios vistos de una serie.
- Obtener el promedio de las calificaciones dadas por el usuario para una temporada particular.
- Obtener el promedio de las calificaciones dadas por el usuario para una serie.
- Determinar si el usuario ya vio todos los episodios de la serie.

Nota. Para calcular los promedios, tener sólo en cuenta los episodios vistos por el usuario.

4. Estructuras de datos

Plantee la solución identificando objetos que intervienen, su estado y su comportamiento para las siguientes estructuras de datos (todas las estructuras son con enteros) No es necesario implementarlas, solo reconocer el estado y comportamiento:

Matriz de enteros, es una matriz que posee 2 dimensiones y permite guardar y obtener elementos

Cola de cartas, es una estructura que al agregar incorpora al final de la cola pero al pedir un elemento retorna el tope (el primero) y lo elimina. Permite averiguar la cantidad de elementos

Fila de enteros, es una estructura que al agregar incorpora al inicio de la fila y al pedir un elemento retorna el tope (el primero) y lo elimina. Permite averiguar la cantidad de elementos

Conjunto de Strings, es una estructura que permite almacenar objetos del tipo String, no permite repetidos. Permite preguntar por si el conjunto tiene un String dado, permite borrar un elemento, cuando se le pide una elemento retorna cualquiera del conjunto (un elemento aleatorio). Permite obtener la cantidad de elementos.