

1) Librería

Una librería requiere un sistema que le aconseje qué libros puede ofrecerle a un determinado cliente. Un libro se compone de un nombre, autor (solo uno), un resumen y un listado de géneros literarios que abarca. Para que un libro pueda ser ofrecido a un cliente, alguno de sus géneros literarios (al menos uno) debe estar incluido en los géneros por los que el cliente tiene preferencia. Es decir que el cliente acepta la recomendación siempre y cuando el libro contenga alguno de sus géneros preferidos. Por ejemplo, si a David le gustan los géneros “ciencia ficción”, “aventuras”, y “novelas románticas”, y se le pregunta si acepta el libro “Mujercitas” cuyos géneros son “Novela Romántica”, “Clásico”, y “Drama”, el lo va a aceptar porque incluye uno de sus géneros preferidos (“novela Romantica”)

La librería agrupa a sus clientes en diferentes categorías a las que le aplica un descuento determinado: clientes frecuentes (han comprado más de 5 libros en el último año) reciben un 10% de descuento, clientes con más de 5 años de antigüedad reciben un 15% de descuento y clientes VIP que reciben un 30% de descuento en cualquier libro que adquieran. Los descuentos no son acumulables.

También hay clientes más exigentes que solo aceptan si el autor está en su lista de preferidos y el libro contiene TODOS sus géneros preferidos

Defina las clases para implementar una solución orientada a objetos para el problema e implemente en Java todos los métodos involucrados en responder qué libros se pueden recomendar a un cliente, y qué precio se le debe cobrar a un cliente por un determinado libro.

2) Extienda la solución anterior para incluir el siguiente requerimiento

a) Incorporar Grupos de clientes, los cuales se componen de varios clientes (incluso sub grupos) y aceptan un libro si al menos alguno de sus integrantes acepta el libro.

b) Poder cambiar en tiempo de ejecución la forma en la que el cliente simple decide si acepta o no un libro.

3) Dado el siguiente código responda las consignas

```
public void prefinal(Comparable a, Comparable b){  
    System.out.println( a.compareTo(b) )  
}
```

a) Porque puede dar error el siguiente código? Como lo solucionaría?

```
Persona a = new Persona("Juan");  
Persona b = new Persona("Carlos");  
prefinal(a,b);    // El método prefinal es accesible
```

Solucionado el problema ¿que imprime por consola el llamado?