

Diseño e Implementación De Estructuras de Datos

Guía 07 – Práctica Integradora

Contenido

Guía 07	1
Catálogo.....	1
Registro de Usuarios	1
Facturación.....	2
Reportes	2

Guía 07

Catálogo

1. Implementar el método "obtenerAutor" de la clase Catalogo para que retorne el autor existente o uno nuevo.
2. Resolver el método comparador para la clase Material para que compare por titulo y fecha de publicación.
3. Probar dicho método
4. Implementar en la clase Catalogo el siguiente método

```
private Comparator<Material> getCriterio(CriterioOrdenamiento criterio) {}
```

El objetivo de este método es poder obtener comparadores del material de disintos campos según los criterios definidos en el enum CriterioOrdenamiento.

5. Implementar los siguientes métodos de búsqueda de material
 - a. Por Titulo, ordenado por titulo
 - b. Por Rango de fecha de publicación, ordenado por fecha
 - c. Por tipo de material y Autor ordenado por Autor
 - d. Por Rango de Calificación. Ordenado por calificación

Todas las búsquedas deben poder limitarse por cantidad

Registro de Usuarios

1. Se deben poder agregar al sistema usuarios de los 3 perfiles (gratuitos, base y premium).
 - a. La lógica acerca de como calcular el precio del material y si es posible adquirir el material o no debe realizarse con Interfaces Funcionales del paquete java.util.function de acuerdo a cada perfil de usuario.

2. Implementar en las subclases de Material, los métodos "puedeAdquirir" y "costo"
3. Implementar en la clase Usuario el método adquirir.
 - a. Si el usuario no puede adquirir el material lanzar una excepción personalizada "AdquisicionException"
4. Implementar en la clase registro el método "adquirir" el cual recibe como argumento un material y un usuario, e intenta registrar la adquisición o lanza una excepción en caso contrario.
5. Agregar a la clase usuario, un atributo double que indica el máximo de gasto que puede tener sin facturar un usuario. Si el usuario quiere agregar un material y el costo del mismo supera el máximo limite de gasto sin facturar lanzar una excepción personalizada "LimiteMaximoGastoException"

Facturación

1. Implementar un algoritmo para que se le pague la comisión acordada a cada Autor. Para esto se recorre las listas de las adquisiciones de los materiales del autor, y si la adquisición no ha sido ya facturada, se la marca como facturada y se agrega el porcentaje correspondiente a las ganancias del autor.
2. Implementar un algoritmo para que se le debiten de la tarjeta los gastos pendientes a todos los usuarios.

Reportes

1. Implementar los métodos de la clase Reporte