PARTE 3: Ejercicio de programación JAVA

Este ejercicio comprende **3** partes:

1. Desarrollo de la funcionalidad especificada más abajo
2. Desarrollo del programa ejecutable con datos de ejemplo provistos y entrega de todo el proyecto Netbeans en la tarea correspondiente de la webasignatura.
3. Desarrollo de los casos de test para las funcionalidades requeridas

ESCENARIO

Una casa de venta de repuestos automotores tiene un sistema de gestión de su depósito, en el que almacena los datos de todas las piezas mecánicas que vende, en una estructura de árbol binario de búsqueda. Cada nodo de este árbol referencia a una "pieza", utilizando como clave única el código de pieza.
Cada pieza está representada por un código único de pieza, un código de catálogo (compuesto por rubro, sub-rubro e ítem, separados por ".") al que pertenece, descripción de la pieza, la cantidad existente y el valor unitario de la misma.
Ejemplo:

KS01000396, 17.4.2, Pumps BX HCV,2, 1017

Se desea contar con funcionalidades que permitan:

1. Conocer la cantidad total de piezas contenidas en el depósito (árbol), junto con el valor total del stock.
2. Dado un cierto rubro (el primer campo del código de catálogo), obtener otro árbol binario con las piezas de ese rubro, insertadas por **código de catálogo**

PARTE 1: Funcionalidad a desarrollar (vale 35%):

En el TDA **ArbolPiezas** (que desciende de TArbolBB):

- **cantYvalorStock() : devuelve 2 valores enteros**// la posición 0 contiene la cantidad total de piezas en el stock y la posición 1 el valor total del stock.
- **piezasPorRubro(unRubro)** : devuelve otro **Arbol de Piezas** con las piezas del rubro pasado como parármetro insertadas por **código de catálogo**.

clase **Pieza:**

atributos:

- Codigo de pieza: **entero**
- Codigo de catálogo: **string** de la forma: **rr.ss.ii** (rubro, sub-rubro, ítem)
- Cantidad **entero**
- PrecioUnitario **entero**

PARTE 2: PROGRAMA (vale 35%)

La clase principal se denomina "**Parcial2**", y tiene su correspondiente método "**main**". En éste, implementa lo necesario para aplicar los TDA y métodos desarrollados.

1. Se provee una clase "Pieza" para representar a las piezas contenidas en el depósito
2. Se provee interfaz TArbolPiezas y clase ArbolPieza para incluir los métodos solicitados.

3. Instanciar un ArbolPiezas y cargar los datos del archivo "piezas.txt", en donde la clave primaria será el Código único de pieza.
4. Invocar al método **cantYvalorStock()** y emitir por consola la cantidad total de piezas en el depósito y el valor total de stock del mismo
5. Emitir un archivo de salida "deposito.txt" conteniendo todas las piezas existentes, ordenadas crecientemente por código de pieza.
6. Dado un rubro que se indicará en el pizarrón, crear un nuevo árbol de piezas que contenga solamente las piezas existentes en el depósito correspondientes a ese rubro, insertadas por código de catálogo.
7. Emitir un archivo de salida "porrubro.txt" conteniendo las piezas contenidas en este nuevo árbol, ordenadas crecientemente por código de catálogo.
8. Invocar al método **cantYvalorStock()** para este nuevo árbol y emitir por consola la cantidad total de piezas existentes del rubro indicado, y el valor total de stock de las mismas

PARTE 3: TEST CASES (vale 30%).

Implementa el o los **Casos de Prueba** necesarios para verificar el correcto funcionamiento de los métodos desarrollados.

NOTA IMPORTANTE:

Se proveen las interfases y clases necesarias. Deben implementarse los métodos de TElementoAB<T> necesarios (insertar, buscar, inorden, etc.). NO SE DEBEN ALTERAR LAS INTERFASES DE TArbolBB ni TElementoAB provistas, ni agregar otros métodos que los requeridos en las interfases.

NO DE DEBEN CREAR NUEVAS CLASES Y NO SE DEBE INCLUIR NINGUN METODO NO SOLICITADO O INNECESARIO.

ENTREGA: Debes entregar TODO el proyecto Netbeans y los archivos de salida solicitados, en un archivo comprimido "**Parcial2.zip**" en la tarea "**PARCIAL2-PARTE3**" publicada en la webasignatura, hasta la hora indicada.

NOTAS SOBRE ARCHIVOS:

El departamento de soporte de TI ha provisto un archivo con el listado completo de las piezas, con el siguiente formato (campos separados por comas):

CODIGO PIEZA, CODIGO CATALOGO, DESCRIPCION, CANTIDAD, VALOR UNITARIO

CODIGO PIEZA // string

CODIGO CATALOGO // string

DESCRIPCION // string

CANTIDAD // entero

VALOR UNITARIO // entero