

Programación Orientada a Objetos - Prefinal 7 de Julio de 2023

Defina las clases para implementar una solución orientada a objetos para el siguiente problema e implemente en Java.

Un coleccionista de objetos diversos desea diseñar un sistema que le permita mantener el orden en su colección. Para ello, desea poder agrupar los objetos en diferentes sub-colecciones. Cada colección, agrupa un conjunto de objetos únicos de la colección completa y puede agrupar también un conjunto de colecciones más específicas. Así, por ejemplo, la colección



"juguetes antiguos" puede contener los objetos "yo-yo de coca-cola de 1996", "bolitas de vidrio de 1972" y la colección "Muñecos" que contiene los objetos "He-man articulado de 1987", y la colección "Thundercats" que contiene el objeto "León-O articulado de 1988", y el objeto "pantro articulado 1990" etc. Incluso puede ser que la colección "juguetes antiguos" esté contenida dentro de una colección más general llamada "Juegos y Juguetes".

De cada objeto de la colección se guarda un nombre, el año en el que fue creado, un valor estimado y un peso.

Las colecciones poseen un nombre descriptivo y el año se calcula como el promedio de los años de los objetos que contiene. El valor estimado se calcula como la suma de los valores de sus elementos, más un 0.1% por cada objeto contenido (en cualquier nivel). Considerar que el porcentaje puede ser diferente para cada colección. El peso de una colección es la suma de los pesos de los elementos contenidos.

A la hora de agregar un elemento a una colección, se debe controlar que el mismo no se encuentre ya agregado a ella, o a cualquier subcolección que contenga. Se considera que dos elementos son iguales si su nombre, año de creación, valor y peso coinciden.

El sistema debe proveer los siguientes servicios:

1. Chequear si un elemento dado ya pertenece a una colección (en cualquier nivel de agrupamiento)
2. Obtener el objeto preferido del coleccionista. El objeto preferido es el objeto más antiguo, si hay dos o más con la misma antigüedad se retorna el más caro de ellos y en el caso de que varios tengan misma antigüedad y precio estimado se retorna el más pesado. El objeto se debe buscar en cualquier nivel de agrupamiento. El criterio para decidir el preferido puede cambiar según lo desee el coleccionista, por ejemplo puede ser el objeto más caro, y en caso de igualdad el más antiguo.
3. Obtener una copia de la colección, restringida según lo solicite el usuario, por ejemplo:
 - o Que incluya solo los objetos creados antes del año 2000
 - o Que incluya solo los objetos cuyo peso no supere los 2kg
 - o Que incluya solo los objetos cuyo valor sea superior a \$5.000.000
 - o Combinaciones lógicas de las restricciones anteriores

NOTA: En el caso de las colecciones, si luego de aplicar las restricciones quedasen sin elementos, las mismas NO se retornan en la copia.

Se desea incorporar un tipo de colección "exclusiva", la cual es una colección que nunca se incluye en una copia, y cuando se le pregunta si contiene un elemento siempre dice que no. También se desea incorporar un tipo de objeto controlado, que tiene la particularidad que cuando se copia concatena un número de copia al nombre original. Por ejemplo, si el objeto tiene por nombre "Donatello", su primera copia se llamará "Donatello (1)", la segunda "Donatello (2)", y así por cada vez que se lo copie.