Examen final de Programación (Parte 2)
Curso 2024-2025

Alomancia

Después de la caída del imperio final, el dios Sazed se encontraba extremadamente aburrido. Para entretenerse, decidió empezar a experimentar con la magia de la alomancia. Hasta entonces, algunos humanos (llamados brumosos) tenían ciertas habilidades asociadas a distintos metales. Por ejemplo, los brumosos de hierro (Fe) podían atraer objetos metálicos hacia ellos, mientras que los brumosos de estaño (Sn) podían aumentar sus sentidos como la visión y el oído.

También estaban otros humanos especiales llamados nacidos de la bruma que poseían todos los poderes brumosos al mismo tiempo. Sazed pensó que estos eran demasiado tramposos y los sacó de la ecuación. Sin embargo, controlar un solo metal por persona era muy aburrido.

Como Sazed era un dios, claramente sabía programar, así que creó la siguiente interfaz:

```
public interface IMetal
{
    public string Name { get; set; }
    public List<IMetal> children { get; set; }

    public void AddChild(IMetal child);

    public void Union(IMetal other);
    public void Intersect(IMetal other);
    public void Difference(IMetal other);
    public void Filter(Func<IMetal, bool> predicate);
   public List<IMetal> Flatten(Func<IMetal, IMetal, int>
       comparison);
}
```

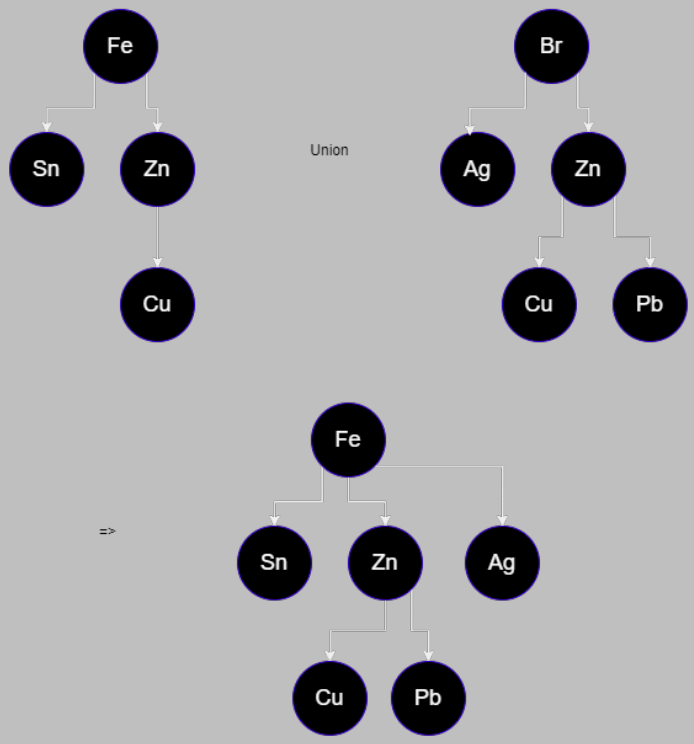
Decidió que todos los metales tendrían un nombre asociado y una lista de otros metales hijos. Si una persona tenía el poder alomántico de un metal, también podía usar los poderes de todos sus descendientes (sus hijos, hijos de sus hijos, hijos de los hijos de sus hijos...). El método **AddChild** recibiría un metal y se lo colocaría como hijo a la instancia actual.

Pero más allá de eso, quería ser capaz de experimentar con las jerarquías de metales para conseguir efectos interesantes. Tu tarea consiste en crear una clase **Metal** dentro del archivo `Solution.cs` que implemente la interfaz **IMetal** y cumpla con las siguientes especificaciones.

Union

Si a un metal a se le invoca el método **Union** con otro metal b, debe quedar en la lista de hijos de a tanto los hijos de a como los de b. En caso de haber metales con el mismo nombre que sean hijos de a y de b, estos deben fusionarse en un sólo metal y el proceso se repite para sus hijos resultantes.

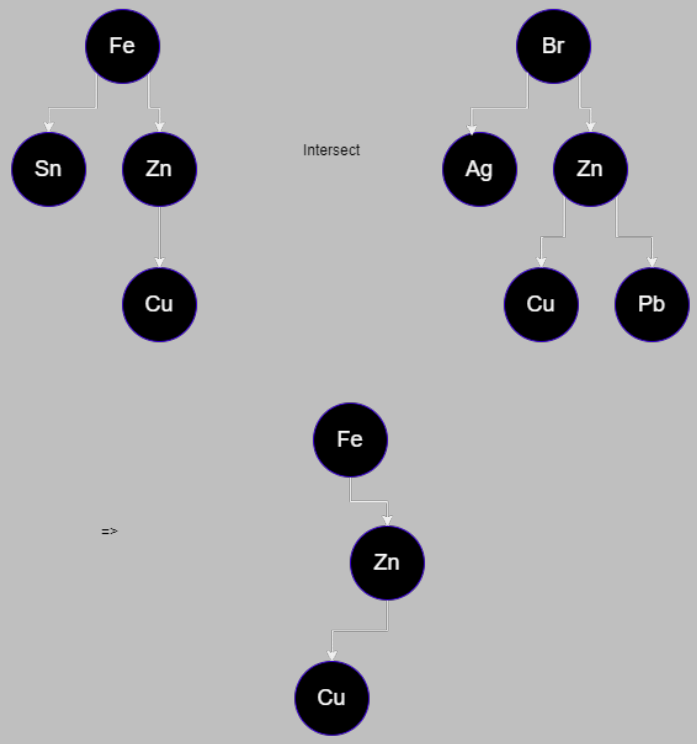
Ejemplo



Intersect

Si un metal *a* se le invoca el método **Intersect** con otro metal *b*, debe quedar en la lista de hijos de *a* sólo aquellos metales que tengan el mismo nombre y estén tanto en *a* como en *b*. Todos estos metales con igual nombre deben fusionarse en un sólo metal y el proceso se repite para sus hijos resultantes.

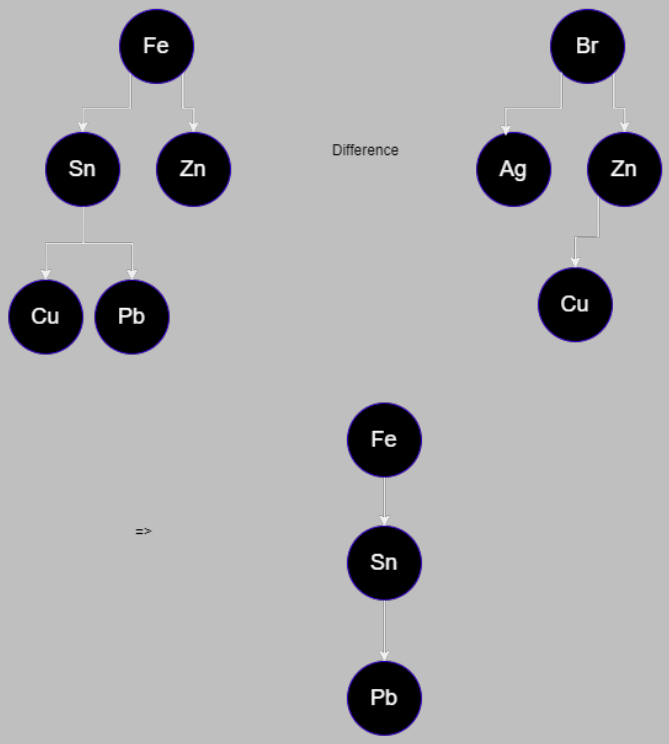
Ejemplo



Difference

Si un metal *a* se le invoca el método **Difference** con otro metal *b*, debe quedar en la lista de hijos de *a* sólo aquellos metales que estén en *a* pero no compartan nombre con *b* ni con ninguno de sus descendientes. Luego, a todos los hijos de *a* resultantes debe aplicársele también la diferencia con *b*.

Ejemplo

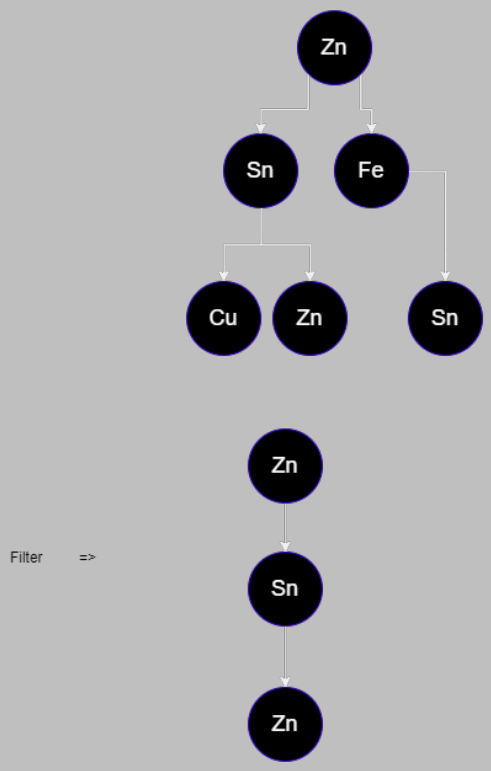


Filter

Si a un metal *a* se le invoca el método **Filter** con un predicado, debe quedar la descendencia de *a* filtrada por el predicado. Es decir, sólo deben quedar aquellos metales que cumplan con el predicado.

Ejemplo

Supongo que el filtro es todos aquellos metales que contengan la letra `n` en su nombre.



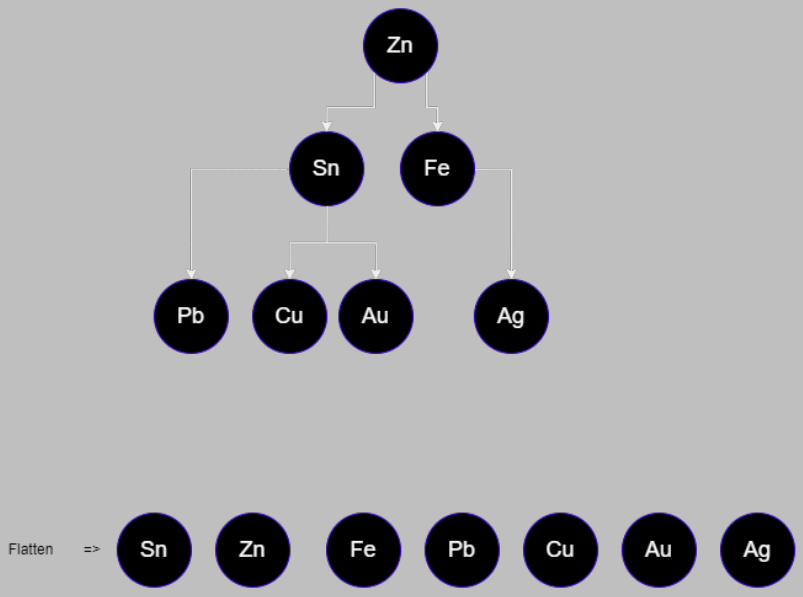
Note que aunque el *Sn* hifo de *Fe* cumple con el filtro, al su padre ser eliminado, este también desaparece.

Flatten

Si a un metal *a* se le invoca el método **Flatten**, se debe retornar una lista con *a* y todos sus descendientes, ordenados según el criterio de comparación que se le pase como parámetro. El criterio de comparación recibe dos metales y debe retornar un número negativo si el primer metal es menor que el segundo, cero si son iguales y un número positivo si el primer metal es mayor que el segundo. Si el criterio de comparación da 0 para dos metales, estos deben aparecer en la lista en orden de una búsqueda en profundidad por la izquierda.

Ejemplo

Suponga que el criterio de comparación es la cantidad de hijos de cada metal.



Note que como Sn tiene 3 hijos, este es el primero, luego van Zn con 2 hijos, fe con un hijo y el resto que no tienen hijos están ordenados según su aparición en una búsqueda en profundidad por el lado izquierdo del árbol.