1. **¿Qué es una relación involutiva en java?**

En Java, una relación involutiva es una relación entre dos entidades en la que ambas pueden ser consideradas como el "origen" o "destino" de la relación. Es decir, la relación es simétrica y reflexiva, lo que significa que si un objeto A está relacionado con un objeto B, entonces B también está relacionado con A, y cualquier objeto está relacionado consigo mismo.

1. **De un ejemplo de relación involutiva.**

Ejemplo de relación involutiva:

En una red social, la relación de "amistad" entre usuarios podría ser una relación involutiva. Si el usuario A es amigo del usuario B, entonces el usuario B también es amigo del usuario A, y cada usuario es amigo de sí mismo.

1. **¿Qué métodos comunes se encuentran en la interfaz Collection y qué funcionalidades proporcionan?**

La interfaz Collection en Java define una serie de métodos comunes que proporcionan funcionalidades para manipular colecciones de objetos. Algunos de los métodos comunes incluyen:

add(E e): Agrega un elemento a la colección.

remove(Object o): Elimina un elemento de la colección.

contains(Object o): Verifica si la colección contiene un elemento específico.

isEmpty(): Verifica si la colección está vacía.

size(): Devuelve el número de elementos en la colección.

iterator(): Devuelve un iterador sobre los elementos en la colección, permitiendo recorrer la colección de manera secuencial.

1. **Explica el propósito de la interfaz Iterable en Java y cómo se utiliza.**

La interfaz Iterable en Java es una interfaz funcional que permite a un objeto ser "iterable", es decir, que se puede recorrer secuencialmente utilizando un iterador. El propósito de la interfaz Iterable es proporcionar una manera estándar de iterar sobre los elementos de una colección o secuencia de elementos.

1. **¿Qué ventajas ofrece el uso de la interfaz Iterable en comparación con simplemente iterar sobre una colección utilizando un bucle for estándar?**

Las ventajas del uso de la interfaz Iterable en comparación con simplemente iterar sobre una colección utilizando un bucle for estándar incluyen:

Abstracción de la iteración: Permite que la lógica de iteración sea independiente de la implementación subyacente de la colección, lo que facilita la reutilización del código y la flexibilidad en el diseño.

Utilización de foreach: Al implementar la interfaz Iterable, se puede utilizar la sintaxis de foreach de Java para iterar sobre los elementos de la colección, lo que hace que el código sea más legible y conciso.

Soporte para diferentes implementaciones de colecciones: Al devolver un iterador a través del método iterator(), la interfaz Iterable permite que diferentes implementaciones de colecciones proporcionen iteradores específicos para recorrer sus elementos de manera eficiente.

**Codigos en github: https://github.com/MarianoSanti01/TUP-2DO-Semestre/tree/master/2do%20sem/Programacion/TP3**