**Trabajo Práctico 3**

1. ¿Qué es una relación involutiva en java?

Una relación involutiva es una asociación especial entre dos clases en la que una instancia de una clase puede estar relacionada con otra instancia de la misma clase. En otras palabras, una relación involutiva es reflexiva. Un objeto puede tener atributos que son instancias de la misma clase que el objeto en sí.

1. De un ejemplo de relación involutiva.

Ejemplo: la clase Persona tiene un atributo madre que es otra instancia de la clase Persona.

public class Persona{

private String nombre;

private Persona madre;

}

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

Métodos comunes de la interfaz Collection:

e: elemento

c: colección

* **boolean**add(E e): Añade un nuevo elemento al final de la lista.
* **boolean** remove(E e): Elimina la primera ocurrencia del elemento indicado.
* **boolean** contains(E e): Comprueba si el elemento especificado está en la colección.
* **void** clear(): Elimina todos los elementos de la colección.
* **int** size(): Devuelve el número de elementos en la colección.
* **boolean** isEmpty(Collection<?> c): Comprueba si la colección está vacía.

Los siguientes métodos combinan dos colecciones:

* **boolean**addAll(Collection<?> c):  Añade todos los elementos de la colección c.
* **boolean** removeAll(Collection<?> c): Elimina  todos los elementos de la colección c.
* **boolean** containsAll(Collection<?> c): Comprueba si coinciden las colecciones.
* **boolean** retainAll(Collection<?> c): Elimina todos los elementos a no ser que estén en c. (obtiene la  intersección).

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

La interfaz Iterable en Java es una interfaz genérica que proporciona la capacidad de iterar sobre los elementos de una colección. Su propósito principal es permitir que una clase sea capaz de ser recorrida por un bucle foreach (for-each) o un iterador, lo que facilita el acceso a los elementos contenidos en la colección.

La interfaz Iterable define un único método:

Iterator<T> iterator(): Este método devuelve un iterador sobre los elementos de la colección. Un iterador es un objeto que permite recorrer los elementos de una colección, uno por uno, y realizar operaciones como la eliminación de elementos.

Una clase que implementa la interfaz Iterable debe proporcionar una implementación del método iterator() que devuelve un objeto que implementa la interfaz Iterator, el cual puede ser utilizado para recorrer los elementos de la colección.

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

La ventaja de usar la interfaz Iterable es que una clase puede proporcionar una forma estándar de iterar sobre sus elementos, lo que facilita su uso en algoritmos genéricos y bibliotecas estándar de Java.

**Ejercicio 1**

Responda: ¿En qué se diferencian una asociación de una composición y de una agregación?

La asociación establece una conexión entre clases que puede ser unidireccional o bidireccional. La composición indica una relación más fuerte donde una clase es parte de otra clase y su existencia depende de ella. Y la agregación es una relación más débil donde una clase puede contener a otras, pero estas pueden existir independientemente.