

Ejercicios adicionales UT6 (II) –

Colecciones de tamaño flexible: ArrayList. Otras colecciones.

Ejercicio 3.

La clase **Secuencia** guarda una secuencia de números enteros ordenados que se representa como un **ArrayList** de objetos **Integer**.

El constructor recibe un parámetro **ArrayList** e inicializa a partir de él el atributo *lista*. Si el parámetro *otra* es *null* se lanza una excepción `throw new IllegalArgumentException("Colección vacía");`. Para añadir todos los objetos **Integer** del parámetro recibido utiliza el método `addAll()` de la clase **ArrayList**. Después ordena la colección utilizando un método de la clase **Collections**.

El accesor `getSecuencia()` devuelve la colección de enteros (devuelve una copia, no el propio original. Para ello crea una nueva colección de enteros y vuelve a utilizar `addAll()`).

`public ArrayList<Integer> fusionarCon(ArrayList<Integer> otra)` - Obtiene la fusión (la unión) de dos listas de números ordenadas, la representada por el atributo *lista* y la representada por el parámetro *otra*. Si el parámetro *otra* es *null* se lanza una excepción `throw new IllegalArgumentException("Colección vacía");`. Hazlo utilizando `addAll()` de **ArrayList** y `sort()` de **Collections**

`public ArrayList<Integer> interseccionCon(ArrayList<Integer> otra)` - Obtiene la intersección de *lista* y el parámetro *otra*, es decir, una nueva lista con los elementos comunes y no repetidos de ambas (en orden). Si el parámetro *otra* es *null* se lanza una excepción `throw new IllegalArgumentException("Colección vacía");`. Hazlo usando el método `retainAll()` (`lista1.retainAll(lista2)` borra los elementos de *lista1* que no están contenidos en *lista2*). Como pueden quedar al final elementos repetidos en la intersección elimina los duplicados con ayuda del método `private ArrayList<Integer> eliminarDuplicados(ArrayList<Integer> lista)` (ayúdate de un **HashSet** para eliminar los repetidos)

La fusión de dos listas es la unión de las dos, es decir, la lista nueva obtenida tiene todos los elementos de ambas. La nueva lista está también en orden creciente. La intersección es una nueva lista con los elementos comunes.

[3, 4, 4, 5, 7, 8, 9, 9, 14]
[1, 3, 3, 4, 12]

[1, 3, 3, 3, 4, 4, 4, 5, 7, 8, 9, 9, 12, 14] Fusión
[3, 4] Intersección

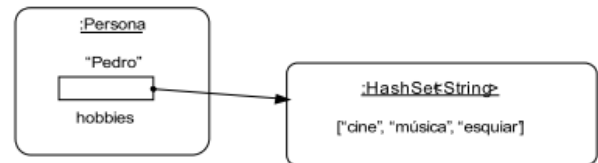
Añade en esta clase el método `main()` para probar todos sus métodos.

Ejercicio 4 .

Un club tiene inscritas una serie de personas cada una de las cuales tienen unos determinados hobbies.

Completa en el proyecto que se te ha proporcionado:

- ➔ la clase **Persona**, que registra el nombre de una persona y el conjunto de hobbies que tiene. Por ejemplo, un objeto de esta clase puede ser:



- el constructor con un parámetro (el nombre de la persona). El constructor crea el conjunto de hobbies inicialmente vacío
 - `public void addHobbie(String h)` – añade el nuevo hobby de la persona (se añade siempre en minúsculas)
 - `public boolean tieneHobbie(String h)` – devuelve *true* o *false* si tiene o no la persona ese hobby
 - `public String hobbiesToString()` - devuelve el conjunto de hobbies como un `String`
 - `public String toString()` - devuelve una representación textual del objeto `Persona` (nombre y conjunto de hobbies)
- ➔ una clase **Club** que guarda en una colección `ArrayList` una serie de personas. Incluye en la clase:
- el constructor que crea la colección vacía
 - `public void addPersona(String nombre, HashSet<String> hob)` – añade una nueva persona a la colección dados su nombre y su conjunto de hobbies
 - `public int personasConHobbie(String cual)`– devuelve cuántas personas en el club tienen un hobby determinado (utiliza un iterador)
 - `public void printClub()` - muestra la relación de personas en el club (utiliza un `for` mejorado)
 - `public HashSet<String> borrarPersonasConHobbie(String cual)` – borra de la lista las personas con un determinado hobby y añade el nombre de las borradas a un conjunto que se devuelve
- añade en esta clase el método `main()` para probar todos sus métodos. El resultado que has de obtener es el siguiente:

```
Personas en el club y sus hobbies
Nombre: Juan
Hobbies: bailar leer deporte caminar nadar cantar
Nombre: Elena
Hobbies: bailar leer cantar
Nombre: Isabel
Hobbies: deporte caminar nadar
Nombre: Amaia
Hobbies: bailar deporte caminar cantar

Personas con hobbie 'nadar'
2
Borrar pesonas con hobbie 'nadar'
Se han borrado [Isabel, Juan]
```