Java – Colecciones

Ejercicio 1

1. Crea la clase Alumno con los atributos nombre (String) y nota (double).
2. En el main, crea una List de Alumno (elije la implementación ArrayList)
3. Añade 10 estudiantes
4. Muestra los estudiantes por pantalla ordenados descendentemente por nota.
5. Muestra sólo los estudiantes que estén por encima de la nota media.

Ejercicio 2

Vamos a ver las colecciones de tipo Map<V, K>. Su implementación más importante es HashMap<V,K>. Al contrario que Set y List, tenemos que definir dos clases (V y K) para utilizar un Map. V es la clave y K es el valor. La clase V debe implementar hashCode y equals para que se pueda indexar correctamente (String, Integer, String, Float y similares ya lo implementan). No hay claves repetidas. Comprobar si existe una entrada con una clave (map.get(clave) o map.containsKey(clave)) es generalmente muy eficiente (coste O(1)). Añadir elementos también es eficiente (map.put(clave, valor)). Haz lo siguiente:

1. Crea un Map<String, Long> (utiliza la implementación HashMap).
2. Añade cinco Nombres y números (de teléfono) distintos.

3. Recorre el Map mostrando todas sus entradas (Nombre -> Teléfono). Puedes utilizar for(String nombre : map.keySet()). ¿Salen en orden?
4. Añade otro teléfono para un nombre que ya hayas insertado previamente. ¿Qué sucede al mostrar el contenido del map?
5. Recorre y muestra los valores contenidos en el Map (usa map.values())

Ejercicio 3

Implementa un programa que lea líneas del teclado y las procese de la siguiente forma:

```
Entrada                          Salida
----------------------------     ------------------
[1, 2, 3] + [3, 5, 7]            [1, 2, 3, 5, 7]
[10, 9, 8, 7] * [2, 4, 6, 8]     [8]
[5, 10, 15, 20] - [0, 10, 20]    [5, 15]
```

Los números entre [] representan conjuntos. "+" representa la unión de conjuntos, "*" la intersección y "-" la resta. Cuando se introduce una línea vacía, el programa termina.

Para los conjuntos podemos usar Set<Integer> con la implementación de TreeSet o HashSet**. ¿Qué diferencia crees que hay entre usar TreeSet o HashSet?.** Set tiene métodos como addAll, retainAll, removeAll. **¿Por qué usamos Set y no List?.**

Para parsear la cadena de entrada te recomendamos utilizar los métodos de String replaceAll, indexOf, substring, y split (también podemos utilizar la clase StringTokenizer). Así como Integer.parseInt para pasar de String a int o Integer.