

SET

La interface SET en Un conjunto en Java es una colección de elementos que, como el conjunto en matemáticas, no permite elementos duplicados dentro de ella y no tiene orden entre sus elementos. Además nos obliga a implementar determinados métodos, por ejemplo el método `hashset`, `hashtree` entre otros.

Se debe de considerar el uso de una colección Set cuando desea almacenar elementos distintamente sin duplicación, o elementos únicos, sin importar el orden de los elementos.

Por ejemplo, puede utilizar un conjunto para almacenar números enteros únicos; usted puede utilizar un sistema para almacenar tarjetas aleatoriamente en un juego de tarjeta; puede utilizar un conjunto para almacenar números en orden aleatorio, etc.

Dentro de la interfaz **Set** existen varios tipos de implementaciones realizadas dentro de la plataforma

- Clase HashSet

Un HashSet es una estructura de datos que contiene un conjunto de objetos. Es una implementación de Set y permite buscar un objeto dentro del conjunto de forma rápida y fácil. Internamente gestiona un array y guarda los objetos utilizando un índice calculado con un código hash del objeto.

El nombre HASH, hace referencia a una técnica de organización de archivos llamada hashing o "dispersion" en el cual se almacenan registros en una dirección del archivo que es generada por una función que se aplica sobre la llave del registro.

Todos los objetos de Java pueden producir un código de tipo hash, y `hashCode()` es un método de la clase raíz `Object`. Un HashSet toma un `hashCode()` del objeto y lo utiliza para localizar rápidamente la clave. Esto redundará en una mejora dramática de rendimiento.

- Los elementos de un HashSet no están ordenados
- Para añadir un elemento al HashSet se utiliza el método **`add(Object obj)`**.
- Para borrar un elemento se utiliza **`remove(Object obj)`**.
- Para borrar todos los elementos se utiliza **`clear()`**.
- El tamaño del HashSet se puede obtener con la función **`size()`**

```
Set<String> hs = new HashSet<String>();
```

```
String [] nombres = {"Jorge","Pepe","Juan","Ale"};
```

```

        for (int i = 0; i < nombres.length; i++) {
            hs.add(nombres[i]);
        }

        System.out.println("HashSet");
        System.out.println(hs.toString());
        System.out.println("");

        System.out.println("-----Metodo Remove-----");
        hs.remove("Pepe");
        System.out.println(hs.toString());
        System.out.println("");

        System.out.println("-----Metodo Clear-----");
        hs.clear();
        System.out.println(hs.toString());

```

La interfaz SortedSet

Es una subinterfaz de la interfaz Set, es un conjunto de elementos ordenados. Algunos métodos importantes con los que permite trabajar son:

- Object first() à Devuelve el primer valor de la lista
- Object last() à Devuelve el ultimo valor de la lista
- SortedSet **subSet**(Object ElementoDesde, Object ElementoHasta) à Devuelve una sub SortedSet desde el elemento marcado como principio hasta el elemento marcado como fin.
- SortedSet **headSet**(Object ElementoHasta) à Devuelve una sub SortedSet desde el principio hasta el elemento marcado como fin.
- SortedSet **tailSet**(Object ElementoDesde) à Devuelve una sub SortedSet desde el fin hasta el element marcado como principio.

Clase TreeSet

Un TreeSet es una implementacion de SortedSet, mantiene los objetos ordenados en lo que se conoce como un *red-black tree*, es decir, en un árbol binario balanceado (cada padre tiene como máximo 2 hijos, y cuando se inserta una entrada se autobalancea de forma que quede un árbol binario simétrico).

Un TreeSet permite hacer búsquedas rápidas. No tanto como un HashSet, pero el TreeSet tiene la ventaja de estar ordenado por clave.

```

        SortedSet<String> arbol = new TreeSet();
        SortedSet<String> arbol2 = new TreeSet();

        String [] nombres = {"Jorge","Pepe","Juan","Ale"};

```

```
    for (int i = 0; i < nombres.length; i++) {  
        arbol.add(nombres[i]);  
    }
```

```
System.out.println("Arbol");  
System.out.println(arbol.toString());  
System.out.println("");
```

```
System.out.println("-----Metodos first y last-----");  
System.out.println("Primer elemento del arbol: " + arbol.first());  
System.out.println("Ultimo elemento del arbol: " + arbol.last());  
System.out.println("");
```

```
System.out.println("-----Metodo HeadSet-----");  
arbol2 = arbol.headSet("Pepe");  
System.out.println(arbol2.toString());  
System.out.println("");
```

```
System.out.println("-----Metodo TailSet-----");  
arbol2 = arbol.tailSet("Jorge");  
System.out.println(arbol2.toString());
```