**Sets in Java**

- A set keeps unique elements

- Provides methods for adding, removing and searching elements

- Offers very fast performance

(Sometimes two converted strings with Hash Function have the same value. The chance is 0.001%)

* HashSet<E>
* The elements are randomly ordered;

Initialization: HashSet<String> hash = new HashSet<>();

* TreeSet<E>
* The elements are ordered alphabetically

Initialization: TreeSet<String> tree = new TreeSet<>();

* LinkedHashSet<E>
* The order of appearance is preserved (saved)

Initialization: LinkedHashSet<String> linkedHash = new LinkedHashSet<>();

* **Sets Methods**
* .size();
* .isEmpty();
* add();
* remove();

**Associative Arrays ( HashMap<Key, Value> )**

* Associative arrays are arrays indexed by keys

( Not by numbers )

* Hold a set of pairs <key, value>

Key Value

“JOHN” “0878 774 954”

“Goshi” “0888 774 224”

* Initialization

HashMap<String, Integer> hash = new HashMap<>();

* Methods : .size(); .isEmpty(); .put(); .remove();
* **TreeMap<Key,Value> - ordering alphabetically by key**
* **Utility methods in maps:**
* **Size();**
* **keySet() : returns a set of unique keys**
* **values(): a collection of all values**

**Basic operations: put(), remove(), clear();**

**Boolean methods:**

* **containsKey(): checks if a key is present in the dictionary**
* **containsValue(): checks if a value is present in the dictionary**