Difference between HashMap and HashSet in Java

HashMap	HashSet
• Map interface. • Key-value pair. • It allows duplicate values but not key. • It can only contain only one null key and multiple null values. • HashMap is faster than HashSet. • HashMap is preferred when we do not need to keep uniqueness. • Example: {a->1,b->2,c->3} here a,b and c are keys and 1,2,3 are values.	• Set interface. • Key only. • Duplication of key is not allowed. • It can contain a single null value. • HashSet is slower because member object is used to calculate the hashcode. • HasgSet is used when we need to keep uniqueness. • Example: {a,b,c,d,e} it denotes a set.

- Programming example of HashMap

```
import java.util.HashMap;

import java.util.Map;


public class HashMapExample {


public static void main(String[] args) {

// Creating a HashMap

Map<String, Integer> hashMap = new HashMap<>();


// Adding key-value pairs to the HashMap

hashMap.put("John", 25);

hashMap.put("Jane", 30);

hashMap.put("Doe", 22);


// Accessing values using keys

System.out.println("Age of John: " + hashMap.get("John"));


// Iterating through the HashMap

System.out.println("\nIterating through the HashMap:");

for (Map.Entry<String, Integer> entry : hashMap.entrySet()) {

System.out.println("Name: " + entry.getKey() + ", Age: " + entry.getValue());

}
```

```
// Checking if a key exists
String searchKey = "Jane";
if (hashMap.containsKey(searchKey)) {
System.out.println("\n" + searchKey + " found in the HashMap. Age: " + hashMap.get(searchKey));
} else {
System.out.println("\n" + searchKey + " not found in the HashMap.");
}

// Removing a key-value pair
String removeKey = "Doe";
hashMap.remove(removeKey);
System.out.println("\nAfter removing " + removeKey + ":");
for (Map.Entry<String, Integer> entry : hashMap.entrySet()) {
System.out.println("Name: " + entry.getKey() + ", Age: " + entry.getValue());
}
}
}
```

- ## Programming example of HashMap

```
import java.util.HashSet;

public class HashSetExample {
    public static void main(String[] args) {
        // Creating a HashSet
        HashSet<String> fruitSet = new HashSet<>();

        // Adding elements to the HashSet
        fruitSet.add("Apple");
        fruitSet.add("Banana");
        fruitSet.add("Orange");
        fruitSet.add("Grapes");
        fruitSet.add("Apple"); // Duplicate element, won't be added
```

```
        // Displaying the elements of the HashSet
        System.out.println("HashSet elements: " + fruitSet);

        // Checking if an element is present in the HashSet
        String searchElement = "Banana";
        if (fruitSet.contains(searchElement)) {
            System.out.println(searchElement + " is present in the HashSet.");
        } else {
            System.out.println(searchElement + " is not present in the HashSet.");
        }

        // Removing an element from the HashSet
        String elementToRemove = "Orange";
        fruitSet.remove(elementToRemove);
        System.out.println("After removing " + elementToRemove + ": " + fruitSet);

        // Iterating through the HashSet
        System.out.println("Iterating through the HashSet:");
        for (String fruit : fruitSet) {
            System.out.println(fruit);
        }

        // Clearing all elements from the HashSet
        fruitSet.clear();
        System.out.println("HashSet after clearing all elements: " + fruitSet);
    }
}
```

Name: Manoj M

E-Mail: manoj1999gtr@gmail.com

GitHub: skylinedude