CSCI 2120:
Software Design & Development II

UNIT 2: Collections Framework & Generics

WeakHashMap

Overview

1. Introduction
2. Hierarchy of WeakHashMap
3. WeakHashMap class Declaration
4. Features of WeakHashMap
5. Internal Workings of WeakHashMap
6. Constructors of TreeMap class
7. TreeMap methods
8. When to use TreeMap

Introduction

- **WeakHashMap in Java** is a class that provides another Map implementation based on weak keys. It was added in JDK 1.2 version and present in java.util.WeakHashMap.

- Java WeakHashMap extends AbstractMap class to use HashTable with weak keys. The key of WeakHashMap has a weak reference that allows an entry in a map to be garbage-collected when its key is no longer in use.

- In simple words, an entry in a WeakHashMap will be automatically removed from the map by the garbage-collector when its key is unused.

- Garbage-collector clears all weak references to the keys(inside and outside of the map) when keys are no longer in ordinary use.

Hierarchy of WeakHashMap class

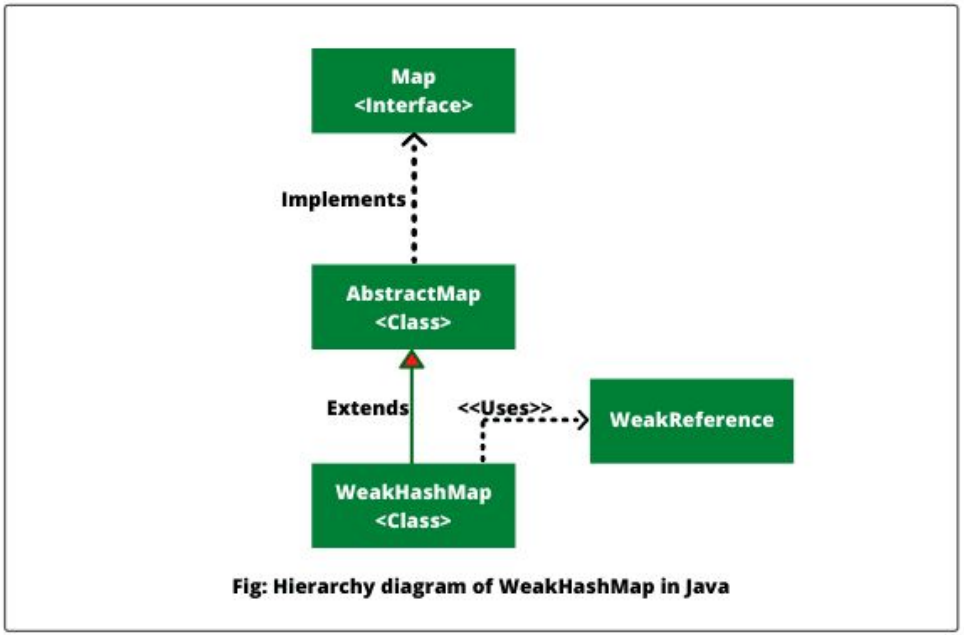

Fig: Hierarchy diagram of WeakHashMap in Java

WeakHashMap class extends AbstractMap class that implements Map interface.

The hierarchy diagram of WeakHashMap class is shown in the below figure.

WeakHashMap class Declaration

WeakHashMap is a generic class that is declared as follows:

```
public class WeakHashMap<K,V>
       extends AbstractMap<K,V>
       implements Map<K,V>
```

Here, K defines the type of Keys and V defines the type of Values.

Features of WeakHashMap

There are several features of `WeakHashMap` class that you need to keep in mind:

1. Garbage collector may remove keys and their associated values from the WeakHashMap at any time in java.

2. It enables us to store data in form of key-value pairs in java.

3. Java WeakHashMap does not allow storing duplicate keys.

4. It allows to insert only one null key in the map but several null values can be added.

5. WeakHashMap does not maintain insertion order in java.

6. WeakHashMap in Java is not synchronized. So, it is not thread-safe. Two or more threads on the same WeakHashMap object can access at the same time in java.

Synchronize WeakHashMap

We can synchronize weakHashMap by using Collections' class synchronizedList() method.

```
Map synchronizedMap = Collections.synchronizedMap(weakHashMap);
```

Now, no two threads can access the same weakhashmap object simultaneously.

Internal working of WeakHashMap

In this section, we will learn how WeakHashMap works internally in java. WeakHashMap uses a weak reference to hold keys. An object can have both weak and strong (normal) references. If an object has a strong reference, it can never be garbage collected.

A weak reference is a reference that by itself does not prevent an object from being garbage collected. If an object has only weak references or no references then it is collected by garbage collector.

WeakHashMap periodically checks objects that have weak references. A weak reference signifies that the key is no longer used by anyone and is collected by garbage collector. Then WeakHashMap removes the associated entry from the map.

Constructors of WeakHashMap class

WeakHashMap class defines four constructors that are as:

1. WeakHashMap(): This form of constructor creates a new, empty WeakHashMap with the default initial capacity (16) and load factor (0.75).

2. WeakHashMap(int initialCapacity): This form of constructor creates a new, empty WeakHashMap with the specified initial capacity and default load factor (0.75).

3. WeakHashMap(int initialCapacity, float loadFactor): This constructor is used to create a new, empty WeakHashMap with the given initial capacity and the given load factor.

4. WeakHashMap(Map m): It constructs a new WeakHashMap with the same mappings as the specified map.

WeakHashMap methods

The methods of WeakHashMap is exactly the same as HashMap methods. To know more detail about HashMap methods, check out the slides for HashMap.

1. Methods inherited from class `java.util.AbstractMap`

   ```
   clone(), equals(), hashCode(), toString()
   ```

2. Methods inherited from class `java.lang.Object`

   ```
   finalize(), getClass(), notify(), notifyAll(), wait()
   ```

3. Methods inherited from interface `java.util.Map`

   ```
   compute(), computeIfAbsent(), computeIfPresent(), equals(), hashCode(),
   merge(), putIfAbsent(), remove(), replace()
   ```

When to use WeakHashMap in Java?

WeakHashMap can be used when:

1. We want to remove keys and their associated values automatically by garbage-collector when keys are no longer in use.

2. We want to store data in key-value pair form on the map.

3. We want to store only one null key.

4. We don’t care about insertion order in java.

5. It can be used when we are not working in a multithreading environment in java.

END