**Map Interface**

1. Interface
2. Part of the Collection framework
3. Kind of collection which is used to store the value in the form of Key and value pair format.
4. In Map We store the value on the basis of Key and Value pair format. Map<key,value> or Map<k,v>
5. Key is always Unique and Value can be duplicate or not be duplicate on the basis of what kind of Map You are using.
6. Automatically It will be sorted order or non sorted order . Only Unique Elements are allowed in the form of values.
7. There are two classes are available under this particular Map.- a) Hash Table b) HashMap. Both are implementing Map Interface.
8. Hash Table -it is from Map Family.
9. Under the HashMap we have One class called Linked HashMap
10. The Map Child interface is called sortedMap which is extending Map Interface.
11. Under the Sorted Map we have another Interface which is called Navigable Interface . Navigable map is extending sorted Map Interface .Although we never use it
12. We don’t use in automation the LinkedHashMap.
13. Under the Navigable Map we have a class that is called TreeMap. This Tree map is implementing Navigable Map interface.

Map Interface

SortedMap

Navigable Interface

Extending HashMap

1. There are three Interface available in Map Interface – a) Map b) Sorted c) Navigable and
2. There are four classes available in Map Interface – a) HashTable b) Hash Map c) LinkedHashMap and d) Tree Map.
3. We can use HashTable , Hash Map and Sometimes Tree Map we can use.
4. Sorted Map is already in Sorted order.
5. In List object We have to store the value on the basis of Indexes. And There is no key and value pair format.
6. Some Properties are available in Tree Map .
7. Example – Key Value pair-

Let’s see I have some keys

Map<String, String > ---it means key can be String and Value will be String.

Map <String,String> m = Map<String,String>();

It means in this particular map I can add any methods . like

m.put(“name”, “Tom”);

m.put(“dept”,”Admin”);

Q How to get the value?

**// here we have to pass the key**

m.get(“name”)--🡪 it will be return Tom.

m.get(“dept”) -🡪 it will return Admin

**\*\*\*\*\*\***

Hash map , Hash table, Tree Map, everything is based on key and value pair format. But

In list object and set object we don’t store the value on the basis of key and value pair format.

In List object ArrayList we just store the value on the basis of Indexes . like zeroth index we will put some values.

**Q- Difference between HashTable and HashMap? Why HashMap is Important and different features of HashMap.**

HashMap.

Let’s see we are fetching some data from Excel file. so Column will be keys and Value will be your values.

So Column will be First name , LastName, email id , Address, phone number etc all these things will be called Keys and you are entering or adding or putting some values for these keys.

So This is Called hash Map in Java.

Example – If you see all the classes and interfaces Map keyword is there like Map , Hash map , Tree map, LinkedHash map, Sorted map , Navigable map but for Hash Table there is no Map we have written. But Hash table is part of map.

Q -Give me the Best Example of Map- either In your Application or your automation –

Ans – Example of Map –Maps are perfectly designed for Key and Value operation ---

1. Error Codes ---For the particular error like 1.2.1 error is there.So This is the key. And then value will be some fatal error. So Error codes is one of the example that will store in the form of Map.
2. Map of Zip codes—that is state wise or area wise. We can store that the particular area having this particular zip code. So this area having these particular zip code.
3. Map of manager and Employees –The particular can have multiple employees .Each manager is the key. Like

--Ruma is a manger is having multiple List , A particular class teacher is having number of List of students are available.

Error code is very important . In terms of automation also you are talking about whenever you have to store the value test data

1. Test data Storage
2. Configuration – if you want to store some configuration or Environment variables – let’s see username . Password ,Url all these things are available in the form of properties files and same thing you can define in the form of Map also. That username , password
3. Headers in Rest API. – we can store the headers. Like authorization token is this , Content type is this.

Let’s see any particular value you want to pass . and you want to store and current

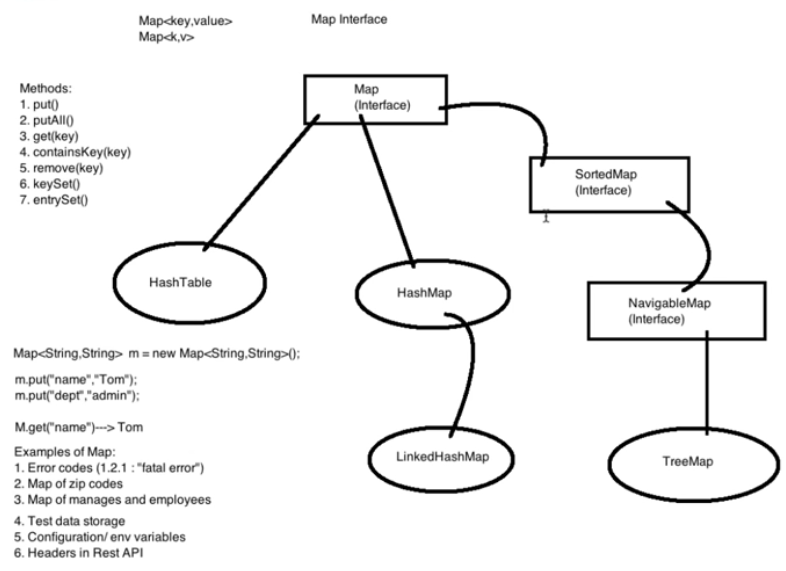
**Methods of Map Interface –**

1. Put() –to add the value
2. putAll() ---which is used to insert the specified Map inside the Map.
3. get() -we just pass the key and then get the value.
4. contains(Key) key- the particular key is available and it will return the values of that particular key. Ad Specific key is available or true.
5. Remove() – remove on the basis of key also.
6. KeySet()- it returns the set view containing all the keys , it is used to iterate the Hash Map or Map object. .

It will give you the complete view what are different keys are available and their values . if you want to iterate you have to use keyset.

1. entrySet()---which returns set view containing all the keys and values.

Difference between KeySet() and entrySet()—



**Q What is HashMap? Why HashMap is important in Java Collection ?**

Hash Map is--

* a class
* Extends / Inherits Abstract Map class and
* implement Map interface .
* a part of Map family. It is stores on the basis of Key and Value pair.
* It contains only Unique Elements
* Stores a values in the form of key and value pair format
* It may have one null key and it is allowed multiple null values that we can store
* It is used in multi threading application where you
* It maintains no order
* HashMap is NonSyschronize. It means if it is used in multi threading Environment or multi Threading concept then more than one thread can access and process the HashMap Simultaneously.
* It means There are multiple thread are running on and then HashMap object can be access by all the threads. So Performance of the Hash Map will be increased automatically , because Whenever you have to use passed performance system, In that case you have to use HashMap.

**----we don’t use Hash Table** because it is synchronized , it means thread safe. It means Hash Table at a time only one thread can access Hash table object.

But Hash Map can be accessible by multiple threads.

**Internal Difference between hash Map and hash table---**

1. Generally in Hash Map Let’s see <String , String >. Here one String is associated with key and it may be Integer or any class object as a key .
2. Hash Table stores the value on the basis of hash code of the key

So In case hash code the key will be like this 254525 internally in case of hash code in will be like this <254555, “Tom”> .So on that particular hash code value will be associated

**Q – What is the problem with HashMap?** What is non-Synchronize ? Why Hash map is non synchronize and Hash table is synchronize ? What do you mean by Fail-First Condition ? What kind of exception you will get Concurrent Modification Exception ?

[It is removal part . if you remove, it will not shift the key. Key will be removed completely , it will not shift the key and it contains only unique values. So It can store may have one null key and multiple null values .]

Let’s see this is a Hash Map and multiple threads are there , This is a 1st  thread , 2nd thread , 3rd thread and forth thread. And I have got some value , Let’s see 1-A, 2--B, 3-C, and 4-D key and value pair format.

.1 T1 T2 2

3 Thread

4 44 Thread

So These particular thread is accessing this object and this particular thread is accessing the same Hash Map object. So at a time all the threads is accessed this particular object.

But the problem is that Let’s see This particular object is accessed this particular object and Simultaneously what exactly the thread does , it will remove this particular 1,A and then make it 1,C and now The value of key number 1 is C.

Simultaneously like thread T2 is running and thread T1 is also attacking the same value . So T1 thread is updating the value of 1,A from A to C . So A is gone undated by Thread T1 . T2 is trying to access this particular 1,A . so what will happen? It is expecting 1,A but T2 will get 1,C now .

Simultaneously you are getting updated value. If any particular threads updated such specific value. so thread number T2 will get updated value, . This is the biggest problem in Hash Map .Then that conditions is called failed first condition.

Fail Fast is whenever you have to iterate the hash map is a failed first condition. It means Concurrent modification exception will be there.

Fail - Fast condition means whenever you are calling any particular Hash Map and any modification is happing in between by some other thread.

**Concurrent modification exception** will come into the picture, when one thread is adding or removing the value and at the same time concurrently another thread is trying to access that particular value but the particular value is gone from the object, immediately it will give you concurrent modification exception .

But if you are trying to access, Simple get the value. if you want to get the value T2 is accessing the same without adding or removing . So all the concurrency will happen that all the thread can attack the same object and can get the value . that’s why it is called non synchronize.

**Non synchronize –** means all the threads Without the problem of synchronization can fit or can access the value from that particular object. So that is the difference between Hash Table and Hash Map.

**Synchronize means** one by one , if T1 is attacking a particular object , it means once a T1 is not released, T2 and T3 can not access it , that is called synchronization.

Concurrently means simultaneously all together if any thread modify the Map Structure by may be adding or removing , Let’s see T2 add one more value 3, C or removing 2, B. So T3 thread is trying to access 2,B but 2,B is not available because it got deleted by thread number T2. That is called failed first condition.

In that case it will give you concurrent modification exception. So Any thread modify map Structure or add or remove the element on that particular Hash Map object , Specially in case of remove , This is called failed first condition or Concurrent Modification Exception you will get.

…………………………….

Q.How to Implements the Hash Map ?

1. First create the object of HashMap
2. In HashMap you have to define and you have to write that what exactly key and value pair format.
3. Generic you have to define
4. Use methods –
5. put(key, Value)
6. to get the value to specific index use get()
7. if you want to print the values of HashMap. Use for each loop with entry and object.entrySet();

**Q Why we use Entry and Entry Set()?**

I want to print the all values .It doesn’t stores on the basis of indexes. It’s not like that 0,1,2.

That ‘s why we have use Entry and entrySet() which is representing Map.entry.

**for**(Entry<Integer,String> m : hm.entrySet()) {

System.***out***.println(m.getKey() + " "+ m.getValue());

Here one reference I have created and the basis of reference hm.entrySet() will store the complete values which are available in this particular HashMAp. So It will iterate one by one .

m.getKey() 🡪 it means that with the help of this particular it will use these other keys and it will iterate .

It doesn’t maintain the order ---How –Using the code ----

// To remove something .|it does not maintain the order|

hm.remove(3);

System.***out***.println(hm);

**Out put**

**Here I wanted to remove 3rd key . then we have like that** {1=Selenium, 2=QTP, 4=TestNg} , 1st key, 2nd key and 4th key .

If you removed 3rd key . it means that 4th key will be shifted to three . 3rd particular value will be blank now. It doesn’t store any value.

Q Can we store some java object in hash map?

Yes .

Step –

1. Create a class any name like HashMapEmployee class. Here n employee are there
2. Create a HashMap object
3. In the Hash Map object I put value of Employee instead of String word.

Like that

HashMap<Integer,String> hm1 = **new** HashMap<Integer, String >(); // before

HashMap<Integer,**HashMapEmployee>** hm1 = **new** HashMap<Integer, **HashMapEmployee** >(); // After . So I have to put all information or value from Employee class.

1. Create Multiple Employees / **HashMapEmployee by creating employee class object and initialize the constructor.**

HashMapEmployee e1 = **new** HashMapEmployee("Asma", 30, "Accounts");

1. **With the help of the particular hm1 , hm1.put () I will be using.**

**Hm1.put(key, value ) . here what exactly the key like employee 1, and what is the value? object name of Employee class. Ie we want infor / value from the employees. The first value e1 and 2nd e2 like that ----------------**

1. **Iterate to print the every thing in the console or I want to traverse using Entry and hashmap object name . entrySet()**

**for**(Entry<Integer, HashMapEmployee> m : hm1.entrySet()) {

**}**

1. First – Get the Key . {How to get the key} m.getkey(); and it will return integer. Because the entry is representing m. m is part of <Integer, Employee>
2. Then m.getValue(); this is a value and it will return Employee and stores it with Employee ref.
3. Then I will print this particular key with one String Employee Info.
4. Then I will print the value one by one , how to ? because The entire get value will be stored in the particular e. and this m.getValue() is representing all the Employees [4 Employee object]. Because I have written emp.entrySet() and it will store in the particular m. and This m.getValue() will be stored in the particular e Employee e is the ref.

This particular e is pointing to all these 4 Employee object |HashMapEmployee| .

System.***out***.println(e.name+""+e.age+""+e.dept);

Here e.name+""+e.age+""+e.dept it will pick e1 first of all and then 2nd time loop will come here and print e2…. Like this.

\*\*\* So every time I will fetching the value from Employee and The Entry set we are using .

1. Entry is an Interface which is used to traverse the HashMap.
2. So Class object you have to use in Hash Map .

………………….

Q. Why Hash Map is no synchronize ?

====

package NaveenCollections;

import java.util.HashMap;

import java.util.Map.Entry;

public class HashMapConcept {

public static void main(String[] args) {

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

hm.put(1, "Selenium");

hm.put(2, "QTP");

hm.put(3, "Java");

hm.put(4, "TestNg");

System.out.println(hm.get(2));

System.out.println(hm.get(4)); // it will return null, bcoz there is no 4th index.

System.out.println(hm.size());

for(Entry<Integer,String> m : hm.entrySet()) {

System.out.println(m.getKey() + " "+ m.getValue());

}

/\*

\* Output- 1 Selenium 2 QTP 3 Java

\*/

// To remove something .|it does not maintain the order|

hm.remove(3);

System.out.println(hm);

HashMap<Integer,HashMapEmployee> emp = new HashMap<Integer, HashMapEmployee >();

//Employee class object

HashMapEmployee e1 = new HashMapEmployee("Asma", 30,"Accounts");

HashMapEmployee e2 = new HashMapEmployee("Rubina",33,"Accounts");

HashMapEmployee e3 = new HashMapEmployee("Rahima", 32, "Accounts");

HashMapEmployee e4 = new HashMapEmployee("Jorina", 31,"Accounts");

emp.put(1, e1);

emp.put(2, e2);

emp.put(3, e3);

emp.put(4, e4);

//Iterate or traverse the HashMap

for(Entry<Integer, HashMapEmployee> m : emp.entrySet()) {

int key = m.getKey();

m.getValue();

HashMapEmployee e = m.getValue(); //It returns Employee

//System.out.println("Employee "+key +" info : " );

//or

System.out.println(key +" info : " );

System.out.println(e.name+" "+e.age+" "+e.dept);

}}}

/\*

1) Example of put() -- Let's see key will be 1, value is selenium

2) If you want to face the value from this particular HashMap , you have to use one concept

that is called entrySet(); and entry\*/

**package** NaveenCollections;

**public** **class** HashMapEmployee {

String name;

**int** age;

String dept;

HashMapEmployee(String name , **int** age, String dept){

**this**.name = name;

**this**.age = age;

**this**.dept = dept;

}}

**Hash Table**

**Q -Differnce between Hash Table and Hash Map ?**

**Hash Map**

* **It is Non-Synchronized.**
* **It means it’s not thread safe.**
* **Multiple threads at a time can access Hash Map object.**
* **It stores a value on the basis of key and value pair.**

**Hash Table**

1. **It is similar to Hash Map but it is synchronized. It stores the value on the basis of key and value.**
2. **Although it stores a value on the basis of key and value pair format**
3. **Hash Table is an Implementation of Hash Table**
4. **It is stores of keys on the basis of Hash Code of the object**
5. **Non null object can be used as a key and as a value.**
6. **In Hash Table we specify an object that is used as a key. So Whatever the key is there which will be specified as an object . And each and every object is having one hash code and on that particular hash code the value will be stored ,This is the concept of Hash Table.**
7. **In Hash Table the key is stored in the form of object and each and every object inside the java is having one special unique identifier that is called Hash code.**
8. **Hash code is a java object. Java provides a particular number. Whenever you create any object java provides a particular hash code for that object. So java object is defined by a some number that is called hash code number.**

**Like it’s a 32 bit integer number , sign integer number.**

1. **Example – This is my memory and I am creating an object. So whenever I am creating the object , First of all a class object will be created and my class name is Test1.**

**So I am creating object which is referred by Test t1 = new Test();**

**JVM will give number that number is called Hash code number. And this number is 32-bit Integer number. So Allows an object to be managed by hash based Data Structure.**

**By using the hash code we can access that particular object and if you want to check on the basis of this particular unique number and you want to store or manipulate the object or you want to define the memories for these objects in that particular data structure we can use hash code concept.**

**32-bit Integer number**

**255545 255545**

**=**

**<255545, “Tom”>**

**Test t1 = new Test();**

**Memory**

**Exception of Hash Table concept --**

**As per the concept we know that it’s a unique number but actually speaking that hash code is not a unique number for an object. if two objects are equal then these two objects should return same hash code.**

**It means I have created equal object.**

**Example – I have done some cloning. I have created equal object.**

**These object are exactly equal and hash code number is something like this for this particular object 255545 and the same hash code number will be given to this particular number .**

**So in case of cloning or in case that two objects are exactly same , in that case the hash code number will not be unique. The same hash code will be given to these two objects .**

**And if you want to access these hash code you have to implement Hash code method from the object class. If two objects are equal in that case it will return same hash code for the different object. In that case you have to override the method from the object class and you need to override equals methods .**

**So whenever we are comparing in two object to get the hash code values we have override equals method also .**

**That is a different concept of Hash code in term of Hash Table.**

**Hash Table is specify any object that is used as key.If any object you are storing as a key then the key is storing in the form of object .**

**Object is always having one hash code and for that particular hash code one value will be associated to that particular hash code.**

Internal Difference between hash Map and hash table---

1. Generally in Hash Map Let’s see <String , String >. Here one String is associated with key and it may be Integer or any class object as a key .
2. Hash Table stores the value on the basis of hash code of the key

So In case hash code the key will be like this 254525 internally in case of hash code in will be like this <254555, “Tom”> .So on that particular hash code value will be associated

Hash table stores the value on the basis of hash code of the key.It is synchronized, it means thread safe and it means at a time only one object

** T1**

** T2**

**T3**



**T4**

Object OoO

O

T4

Let’s see this is the object and I have multiple threads are available , thread number 1,2,3,4,5….. If this thread T1 is accessing this particular object. All these three thread can not this object. Because this is thread safe. Hash Table is always Synchronized. It means One by one . Once this T1 is done then T2 is done [can access the object], then T3 is done and then T4 can access this Object .

It’s not like a Hash Map , In Hash Map at a time all these threads [4 or 5 or multiple threads ] can access Hash Map Object. but In Hash table , it always synchronized.

This is basic difference between Hash Map and Hash table.

……………………………

**Q How to Create Hash Table ?**

1. **Hash Table is a Class**

**package** NaveenCollections;

**import** java.util.Hashtable;

**public** **class** HashTableConcept {

**public** **static** **void** main(String[] args) {

// Create a Hash Table

// Object of HAsh TAble

Hashtable h1 = **new** Hashtable ();

//how to put the values

h1.put(1, "Tomket");

h1.put(2, "java");

h1.put(3, "Tobako");

//How to create a clone copy/ shallow copy of this particular hash table

//for this purpose I will create one more hash table over here with an object .

Hashtable h2 = **new** Hashtable ();

// I want to create a clone of h1 and this h1.clone we can store in h2 and you have typecast into hash table

h2 = (Hashtable)h1.clone();

// print h1 and h2 both

System.***out***.println("The values from h1 : "+ h1);

System.***out***.println("The values from h2 : "+ h2);

}

}

/\* Q- How to create a cloning?

\*

\* `1)Here I have created two object h1 and h2 and in h2 I am not storing any values

\* but I just created h1.clone() and storing into h2 {2nd object } .

\* and now h2 has all these three variables

\* 2) whatever are the values are available in h1 and those values will be available h2 also.

\* 3) So two objects are there and h1 and h2 are ref. In h2 I am not storing any values with the help of h2.

\* I am converting h1 cloning into h2 . So h2 just like a duplicate object will be created

\* So This is how we create the cloning.

\*

\* Out put

\* -------

The values from h1 : {3=Tobako, 2=java, 1=Tomket}

The values from h2 : {3=Tobako, 2=java, 1=Tomket}

This is a concept that we create the clone .

\*/

Q if you remove the value 4 from h1, you can clear the value.

\*

\* h1.clear();

System.out.println("The values from h1 : "+ h1);

System.out.println("The values from h2 : "+ h2);

Out put

------

The values from h1 : {}

The values from h2 : {3=Tobako, 2=java, 1=Tomket}

So after clear h1, it will be deleted all the values from h1 but

h2 will be remain same created.

So First I created h1 and duplicate of h1 to given to h2. I cleared h1 then all the values are gone but h2 will remain same.

//Contains ().

1. I will create a new hash table with a proper values and I have given three values in this particular hash table object . I want to check that whether a value exist or not.
2. I put one conditions {a)What is my object name, b) Contains value method is there }---

Like If contains Value Ruma then print it----

**if**(st.containsValue("Ruma"))

System.***out***.println("Value is found");

1. If Ruma is available, value is found.
2. **If the value is not available it will not give any value . So we can check the value is available or not .**

**Q – How to print all the values ? / How to iterate ?/ How to access all the values from hash table using that concept is called Enumeration ? how to use that ?**

1. **Print all values from hash table using ----Enumeration -----elements().**
2. **So we have a method on this particular hash table , st.elements();**

**We use element() like this ------------**

**Step—**

1. I have a particular hash table and on this particular hash table - st.elements(); is there.
2. So this st.elements () will give you Enumeration class object.
3. It returns an enumeration of the values in this hash table. So all these value will be stored in this particular Enumeration.

Use the Enumeration methods on the return object to fetch the elements sequentially . Sequentially If you want to fetch all the elements will be stored inside this particular enumeration reference and then you can iterate it in the form of sequentially.

**How to do this** ----

🡪Lets see I have a particular hash table like using st object ref. So We have an method on this particular hash table , elements() is there.

Enumeration e = st.elements(); // use Same Hash Table st.

System.***out***.println("Print values from h3 using Enumeration");

🡪simple ---st.elements () and it will return one Enumeration object so I have to store in Enumeration reference

Enumeration e =st.elements () and then print the value .

1. So to get the value one by one I have to iterate it . while (e. **hasmoreelements**) we have more particular methods **hasmoreelements**
2. In this particular Enumeration, values are available as hasmoreelements. Simple use syso

**while** (e.hasMoreElements()) {

System.***out***.println(e.nextElement());//just like Iterator we have e.hasNext to Enumeration we can get the values from this particular hash table .

So through Enumeration also we can get the values.This is a concept of Enumeration.

…………………..

**package** NaveenCollections;

**import** java.util.Enumeration;

**import** java.util.Hashtable;

**public** **class** HashTableConcept {

**public** **static** **void** main(String[] args) {

// Create a Hash Table

// Object of HAsh TAble

//Contains value

Hashtable st = **new** Hashtable();

h3.put("A", "Afsheen");

h3.put("B", "Ruma");

h3.put("C", "Taseen");

**if**(st.containsValue("Ruma"))

System.***out***.println("Value is found");

// Print all the values from hash table using -Enumeration -elements();

Enumeration e = st.elements();

System.***out***.println("Print values from h3 using Enumeration");

**while** (e.hasMoreElements()) {

System.***out***.println(e.nextElement());

}}}

………………

OutPut==

Value is found

Print values from h3

Afsheen

Taseen

Ruma

**EntrySet()-**

1)Get all the values from Hash table using EntrySet()- which will return one set object. The set of Hash table values.

2) We have seen Hash Map also set of Hash table values .

So there are ways to get the values either you can get it through a)Enumeration and We can get the values through 2)entryset also.

1. Its not like a ArrayList because it does not store the value in the form of ordering that 0,1,2,3,4, that you can provide the for loop or you can get it like this, NO.

Either you have to use Enumeration or you have to use entrySet.

Q How to do that to get the value using Entryset ?

Step –

1. Create a Hash table object and create a Hash table and using that object ref
2. And it will return one set reference
3. Print
4. Return a set off view of the Mapping contains in this Map or in this particular hash table.
5. Out put ---

Print values from h3 using EntrySet

[A=Afsheen, C=Taseen, B=Ruma]

Q—Equals()

**Equals () which is used to compare a specified object with this Hash Table or not for Equality point of view**

Let’s see we have one Equals() is available, So Let’s see I have this particular Hash Map and Similar one more Hash Map I will create –an object of Hash table like with the same value. and then

1. Create a Hash Table object.
2. Create a Hash Table using that object ref
3. And I want to check both the object checking whether both the hash tables are Equals or not . and Both the Hash Table objects are equal or not.
4. Check both the hash tables are Equal or not :

How to do that ----

1. Use equals()
2. To compare we have to use if condition

Q get() --Get the value from a Key—

Steps –

1. System.***out***.println(h4.get("A"));

Q **Hashcode() –**

**--Which is used to and it returns the hash code value of this particular hash table as for the definition.**

**---Get the HashCode of the Hashtable object.**

System.***out***.println("The Hashcode value of h4 : "+h4.hashCode());

\*\*\*Each and Every object represented by one hashcode value .it is a 32 bit integer number. It’s a random unique number.

Q- One more concept that In hash table it contains only and only Unique elements.

Unique elements means the key and value pair format always unique.

Hashtable st1 = **new** Hashtable();

st1.put("A", "Afsheen");

st1.put("B", "Ruma");

st1.put("C", "Taseen");

st1.put("C", "Taseen");

Like if you create st1.put(“C”,”Taseen”); and again st1.put(“C”,”Taseen”); and if you true to print st1 ,

Syso(st1);

You will get from st2 and it will only print like this—

Values from st2 :

{A=Afsheen, C=Taseen, B=Ruma}

It is not printing C -Taseen two times So it only contains unique elements. If you try to add two time , it will ignore.

It can not have any null keys or null values . If you define any null key , immediately it will return an one Exception that is called null pointer Exception.

So null and null value are not allowed. So never define any null keys or null values.

But In Hash Map it is allowed. Multiple null values and one null keys are allowed in hash map.

But Hash table it’s very restricted. It is sysnchronized , it contains only unique values.

If you want to define some specific to generics , We can define the generic also.

Restriction –

1. In Hash Table we don’t use Multi-threading and
2. If you want to improve the performance in that case we don’t use Hash table because this is synchronized. It’s a thread safe because at a time only one thread can access. So Obviously it is Sequential threading. It means once the one thread is completed then second thread will come and then third thread will come and So obviously the performance of the application will be slow.

But Hash Map is non-synchronized and in that case Multiple threads at a time can access or can attack hash map object. So Hash Map can serve and can be given to the multiple threads. But the problem with if any value is getting changed or removed by thread one in hash Map, it will get the null value or something like that.

package NaveenCollections;

import java.util.Enumeration;

import java.util.Hashtable;

import java.util.Set;

public class HashTableConcept {

public static void main(String[] args) {

// Create a Hash Table

// Object of HAsh TAble

Hashtable h1 = new Hashtable ();

//how to put the values

h1.put(1, "Tomket");

h1.put(2, "java");

h1.put(3, "Tobako");

//How to create a clone copy/ shallow copy of this particular hash table

//for this purpose I will create one more hash table over here with an object .

Hashtable h2 = new Hashtable ();

// I want to create a clone of h1 and this h1.clone we can store in h2 and you have typecast into hash table

h2 = (Hashtable)h1.clone();

// print h1 and h2 both

System.out.println("The values from h1 : "+ h1);

System.out.println("The values from h2 : "+ h2);

System.out.println("=====================");

h1.clear();

System.out.println("The values from h1 : "+ h1);

System.out.println("The values from h2 : "+ h2);

System.out.println("==================");

//Contains value -I have to check whether a value

Hashtable st = new Hashtable();

st.put("A", "Afsheen");

st.put("B", "Ruma");

st.put("C", "Taseen");

if(st.containsValue("Ruma")) // if contains value "Ruma", simple I have to print like

System.out.println("Value is found");

System.out.println("=======================");

// Print all the values from hash table using -Enumeration -elements();

Enumeration e = st.elements(); // use Same Hash Table st.

System.out.println("Print values from h3 using Enumeration");

while (e.hasMoreElements()) {

System.out.println(e.nextElement());

}

//Get all the values from Hash table using--- EntrySet()-----Set of Hash table values

System.out.println("Print values from st using EntrySet");

Set s =st.entrySet(); // it will return one set ref

System.out.println(s);

// Equals () which is used to compare a specified object with this Hash Table or not. or

// Check both the hash tables are Equal or not :

Hashtable st1 = new Hashtable();

st1.put("A", "Afsheen");

st1.put("B", "Ruma");

st1.put("C", "Taseen");

System.out.println();

if(st.equals(st1))

System.out.println("Both are Equal"); //st and st1 are Exact Hash table are there.---use Equals ()

else {

System.out.println("Both are not same");

}

// get the value from a Key like st1 is there .

System.out.println(st1.get("A"));

// get the HashCode of the Hash table Object :

System.out.println("The Hashcode value st1 : "+st1.hashCode());

Hashtable st2 = new Hashtable();

st2.put("A", "Afsheen");

st2.put("B", "Ruma");

st2.put("C", "Taseen");

st2.put("C", "Taseen"); // it contains only unique value.

//It can not have any null keys or null values and immediately it will give an null pointer exception

//st2.put(null, "Taseen");

//st2.put("D", "null"); // null pointer Exception

System.out.println("Values from st2 : ");

System.out.println(st2);

//Generic

Hashtable <String, String>st3 = new Hashtable <String, String>(); // It means I can store only string key and string values

}}

/\* Q- How to create a cloning?

\*

\* `1) Here I have created two object h1 and h2 and in h2 I am not storing any values

\* but I just created h1.clone() and storing into h2 {2nd object } .

\* and now h2 has all these three variables

\* 2) whatever are the values are available in h1 and those values will be available h2 also.

\* 3) So two objects are there and h1 and h2 are ref. In h2 I am not storing any values with the help of h2.

\* I am converting h1 cloning into h2 . So h2 just like a duplicate object will be created

\* So This is how we create the cloning.

\*

\* Out put

\* -------

\* The values from h1 : {3=Tobako, 2=java, 1=Tomket}

The values from h2 : {3=Tobako, 2=java, 1=Tomket}

\*

\* This is a concept that we create the clone .

\*

\* -------------------

\* Q if you remove the value 4 from h1, you can clear the value.

\*

\* h1.clear();

System.out.println("The values from h1 : "+ h1);

System.out.println("The values from h2 : "+ h2);

Out put

------

The values from h1 : {}

The values from h2 : {3=Tobako, 2=java, 1=Tomket}

So after clear h1, it will be deleted all the values from h1 but

h2 will be remain same created.

So First I created h1 and duplicate of h1 to given to h2. I cleared h1 then all the values are gone

but h2 will remain same. because this is a duplicate object.

Q- Contains() method

1) Create a hash table object

2)Put the value into the

\*/