**HashMap (using hashtables)**

Implement Data structure that works the same as HashMaps (maps Keys and Values). Note that Keys must be unique. You will need to implement **MyHashMap<K, V>** and **Entry<K, V>** classes. Implement HashMap as an **array** of entries, where each array cell is an entry that can have next entries (it will be explained in the video).

**MyHashMap** class must have fields:

* double loadFactor
* int capacity.

Capacity is the size of an array created to hold entries.

About loadFactor, assume your capacity is 16 and loadFactor is 0.5 (in percentage: 50%). Then if an array is full for 50% or more(50% of 16 = 8 or more array cells) you need to make array bigger size and do rehash operation. (Also explained in the video).

MyHashMap must have constructor with **capacity** and **loadFactor** fields.

**Entry** class holds: key, value and next entry reference.

**MyHashMap** class has next methods:

* void clear() – removes all entries from the map
* void put(K key, V value) – adds and entry into the map
* V get(K key) – returns the value that matches
* Set<K> keySet() – returns the set of all keys from map (java.util.Set, can use HashSet class)
* Set<V> values() – returns the set of all values from map (java.util.Set, can use HashSet class)
* boolean isEmpty() – checks if the map is empty
* void remove(K key) – removes an entry for the specified key
* void rehash() – does the rehashing operation (after array size changes or some array cell entry is removed) – will explain in the video.
* void print() – prints all entries. If one array cell has more than one entries – show them in one row.
* If you are adding and entry with key that exists in a map – replace the old value with the new one.
* Note: combine 2 ways of hashing (**double hashing** and **chaining methods**). By default use chaining method.
* But if the size of list in any array cell is >= than **1.0/(1.0-loadFactor)** then use **double hashing** method to insert an entry.

After finishing the task check the task by inserting and deleting entries.