Наталья, проверил вашу работу. Задача на этот раз немного посложнее, вижу применение интерфейса Iterable, Iterator, реализовали соответствующие методы hasNext, next, получается наделили нашу структуру данных способностью быть перечисляемой. Но я хочу вам продемонстрировать и свой вариант решение данной задачи, основанный на интерфейсе Iterable:

public class HashMap<K, V> implements Iterable<HashMap.Entity>

Тут, элементом коллекции HashMap выступает Entity, который содержит ключ + значение, имплементируем интерфейс:

public class HashMap<K, V> implements Iterable<HashMap.Entity> {

private static final int INIT\_BUCKET\_COUNT = 16;

private Bucket[] buckets;

private Bucket.Node currentNode;

private int currentIndex;

@Override

public Iterator<HashMap.Entity> iterator() {

return new HashMapIterator();

}

Создаем класс HashMapIterator, имплементирующий интерфейс Iterator<HashMap.Entity>:

class HashMapIterator implements Iterator<HashMap.Entity>{

@Override

public boolean hasNext() {

if (currentNode == null){

for (int i = 0; i < buckets.length; i++)

if (buckets[i] != null && buckets[i].head != null){

currentIndex = i;

currentNode = buckets[i].head;

return true;

}

return false;

}

else{

if (get((K)currentNode.value.key) == null)

{

currentNode = null;

currentIndex = 0;

return hasNext();

}

else{

HashMap.Bucket.Node node = currentNode;

currentIndex = calculateBucketIndex((K)node.value.key);

if (node.next != null){

currentNode = node.next;

return true;

}

for (int i = ++currentIndex; i < buckets.length; i++){

if (buckets[i] != null && buckets[i].head != null){

currentIndex = i;

currentNode = buckets[i].head;

return true;

}

}

currentNode = null;

currentIndex = 0;

return false;

}

}

}

@Override

public Entity next() {

if (currentNode == null){

for (int i = 0; i < buckets.length; i++)

if (buckets[i] != null && buckets[i].head != null){

currentIndex = i;

currentNode = buckets[i].head;

return currentNode.value;

}

return null;

}

else if (get((K)currentNode.value.key) == null){

currentNode = null;

currentIndex = 0;

return null;

}

else

return currentNode.value;

}

}

И вот теперь можно протестирвать перебор всех элементов коллекции HashMap:

for (HashMap.Entity entity: hashMap1) {

System.out.println(entity);

}

Основной плюс подобного подхода, в рамках перечисления, через Entity, вы получаете доступ к ключу и значению очередного элемента.