Sorted Maps

**Map Arabirimi**

Bu arabirim eşleme tablolarını temsil etmektedir ve koleksiyon arabiminden türetilmemiştir. Map arabirimi elemanlarını benzersiz anahtarlar ile eşleştirerek tablolarda tutmaktadır. Yani eşleme tablosundaki her bir eleman benzersiz bir anahtar ile eşleştirmekte gerektiğinde bu anahtarlar yardımıyla işlem yapmaktadır.

**Map arabirimine ait metotlar;**

* **size():** Eşleme tablosundaki eşleme sayısını döner.
* **values():** Eşleme tablosundaki elemanlardan bir koleksiyon nesnesi döner.
* **clear():** Map’teki tüm eşleşmeleri kaldırır.
* **isEmpty():** Eşleşme tablosu boş ise yani eşleşme tablosunda eleman yoksa true döner.
* **entrySet():** Eşleşme tablosunda bulunan elemanlar için bir set nesnesi-küme döner.
* **keySet():** tabloda bulunan anahtarlar için set nesnesi döner.
* **put(anahar,eleman):** Parametrede verilen anahtar elemanı birbirleri ile eşleştirip, eşleme tablosuna yazar.
* **putAll(eslemeTablosu X):** Parametrede verilen eş. tab. X in tüm elemanlarını ilgili eşleme tablosuna ekler.
* **get(anahtar):** Parametrede verilen anahtar ile eşlesen elemanı döner.
* **containsKey(anahtar):** parametrede verilen anahtar ile eşlesen bir eleman varsa true döner.
* **containsValue(eleman):** parametrede verilen anahtar ile eşlesen bir eleman varsa true döner.
* **remove(anahtar):** parametrede verilen değeri eşleme tablosundan çıkarır.

**SortedMap Arabirimi**

Map arabirimindeki tüm metotları uygulamakla birlikte, map arabirimine ek olarak eşleme tablosundaki eşleme anahtarlarını sıra ile tutar.

**İlgili metotları;**

* firstKey(): Eşleme tablosundaki en küçük elemanı döner.
* lastKey(): Eşleme tablosundaki en büyük elemanı döner.
* HeadMap(anahtar): Eşleme tablosunun, parametrede verilen anahtardan küçük olan anahtarlar bölümünü döner.
* tailMap(eleman): Eşleme tablosunun, parametrede verilen anahtardan büyük olan anahtarlar bölümünü döner.

**AbstractMap Soyut Sınıfı**

Map arabiriminin entrySet() metodu dışındaki tüm metotlarını kullanmaktadır.

**Map arabiriminin somut sınıfları:**

**HashMap Sınıfı:** Karışık eşleme de denilebilir. Eşleme tablosuna eleman ekleme, çıkartma işlemlerinde ve anahtarı verilen elemanı tabloda bulma işlemlerinde oldukça etkilidir.

**LinkedHashMap Sınıfı:** Sıralı eşleme denilebilir. HashMap Sınıfından türemiştir. Bu sınıf elemanları sıralı olarak tutar. Bu sıralama ekleme sırası ya da erişim sırasına göre olabilir.

**TreeMap Sınıfı:** Map arabirimini uygulayan sortedMap arabirimini uygular. Eşleme tablosundaki anahtarları sıra ile tutar. Bu sıra tablodaki anahtarların karşılaştırılması ile oluşur. Yani küçükten büyüğe gelir. 1.2.3 , a.b.c vb.  
**Örnek:**

public class App {  
 public static void main(String[] args) {Map<Integer, String> hashMap = new HashMap<Integer, String>();  
 Map<Integer, String> linkedHashMap = new LinkedHashMap<Integer, String>();  
 Map<Integer, String> treeMap = new TreeMap<Integer, String>();testMap(linkedHashMap);  
 }public static void testMap(Map<Integer, String> map){  
 map.put(9,"fox");  
 map.put(2,"cat");  
 map.put(3,"dog");  
 map.put(4,"swan");  
 map.put(7,"bear");  
 map.put(1,"snake");for (Integer key: map.keySet()){  
 String value = map.get(key);System.out.println(key + " :" + value);  
 }  
 }  
}