

Koleksiyonlar - Collections

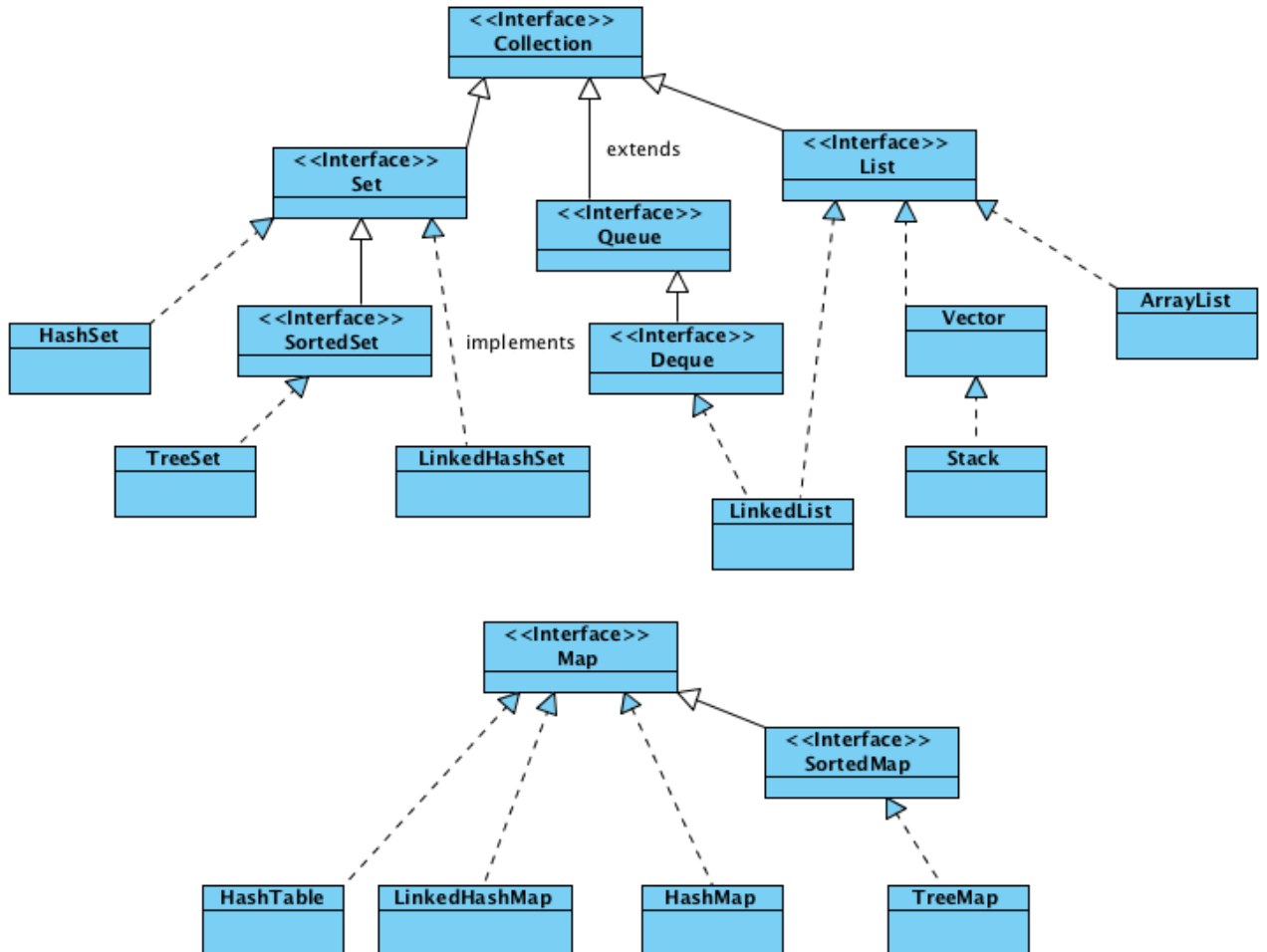
İçerisinde birden fazla nesne barındırabilen ve gerektiğinde nesne ekleme-silme gibi işlemlere imkan sağlayan bir başka nesnedir. Koleksiyonun içerisinde barındırdığı nesnelere koleksiyonun elemanları denir. Java koleksiyon çatısı altında 3 ana koleksiyon tipi sunmaktadır. **SET, LIST, MAP** ve yine bunlara ait aynı isimde 3 interface bulunmaktadır. Bu interface arabirimlerin nesneleri şunlardır:

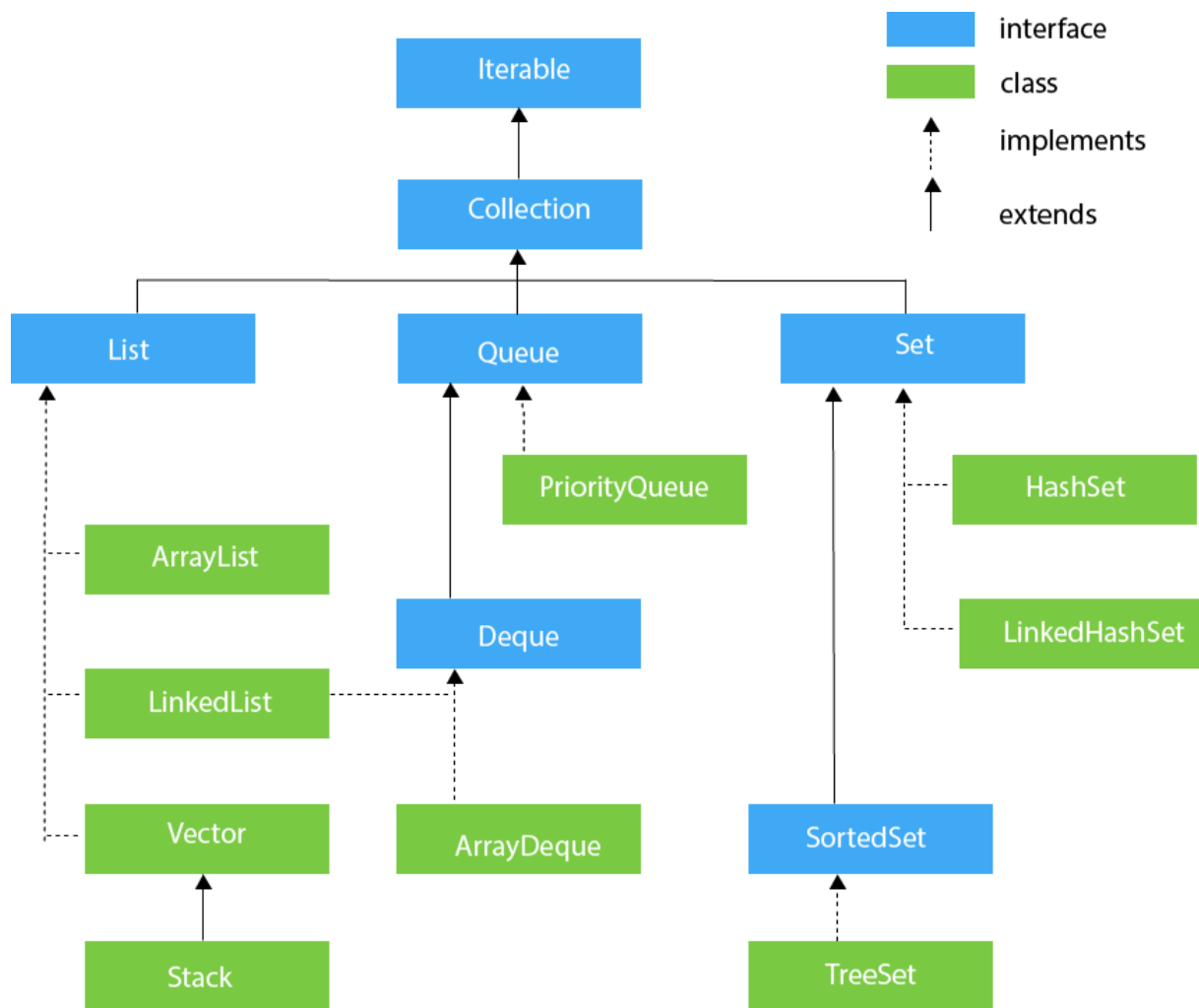
Set Nesnesi: Kendisine verilen elemanların her birinde sadece bir tanesini tutar. Kopya ya da tekrarlanan elemanları barındırmaz.

List Nesnesi: Kendisine verilen elemanları sıralı şekilde tutar. Tekrarlanan elemanları barındırabilir.

Map Nesnesi: Her biri birbirinden farklı anahtarlar ile eşleştirilen nesnelerden oluşur.

Koleksiyonlar Collection Interface'ini temel alırlar. Collection Interface'ne bundan sonra Koleksiyon arabirimi diyeceğiz. Bu arabirimi List ve Set arabirimleri genişletmektedir. Bunlar java tarafından sağlanan ve nesneler üzerinde işlemler yapmamızı sağlayan en tepe arabirimlerdir. En tepede Koleksiyon Arabirimi ve bu arabirimden türeyen List, Set, ve Set Arabiriminden türeyen SortedSet arabirimi bulunmaktadır.





https://www.tutorialspoint.com/java/java_generics.htm

<https://medium.com/gokhanyavas/javada-generics-43546e77bd8f>

<https://www.geeksforgeeks.org/collections-in-java-2/>

List Arabirimi

List arabirimindeki soyut sınıflar;

AbstractList Sınıfı list arabirimini uygulamakta ve AbstractCollection soyut sınıfından türemektedir.

Bu sınıftan ise AbstractSequentialList soyut sınıfı türetilmiştir.

List arabirimindeki soyut sınıflar;

LinkedList sınıfı, AbstractSequentialList soyut sınıfından türetilmiştir. AbstractList soyut sınıfından Vector ve ArrayList somut sınıfları, Vector sınıfından ise Stack somut sınıfı türetilmiştir.

Set Arabirimi

Set arabiriminden SortedSet arabirimi üretilmiştir.

Set arabirimindeki soyut sınıflar;

AbstractSet sınıfı set arabirimini uygulamakta ve AbstractCollection soyut sınıfından türemektedir.

Set arabirimindeki somut sınıflar;

AbstractSet soyut sınıfından HashSet ile TreeSet somut sınıfları HashSet sınıfından LinkedHashSet sınıfı türetilmiştir. TreeSet somut sınıfı aynı zamanda SortedSet arabirimini de uygulamaktadır.

Map Arabirimi

List ve Set arabirimlerinin aksine Koleksiyon Arabiriminden türetilmemiştir. Çünkü bu arabirimden oluşan sınıflar elemanları-dataları bir liste ya da küme içersinde tutarken, map arabirimindeki sınıflar aynı elemanları birbirinden farklı her eleman bir tane olan anahtarlarla eşleştirmektedir. Map arabiriminden SortedMap arabirimi türetilmiştir.

Map arabirimindeki soyut sınıflar;

Map arabiriminden AbstractMap soyut sınıfı türetilmiştir.

Map arabirimindeki somut sınıflar;

AbstractMap soyut sınıfından map arabirimini uygulayan HashMap ve TreeMap somut

sınıfları ile HashMap somut sınıfından LinkedHashMap somut sınıfı türetilmiştir. TreeMap sınıfı ise aynı zamanda SortedMap arabirimini uygulamaktadır.

Iterator(): Koleksiyonların elemanları için bir interator nesnesi döner. Bu nesne ile koleksiyon elemanları üzerinde dolaşabilir, işlem yapılabilir ve koleksiyonun elemanlarını okuyabiliriz. Burada hasNext() metodu ile koleksiyondaki yeni elemanlar kontrol edilir. next() metodu ile koleksiyonun elemanları sırası ile alınır. remove() metodu ile elemanlar koleksiyondan çıkarılır.

Set Arabirimi

Koleksiyon arabiriminden türeyen bu arabirim kümeleri temsil eder. Koleksiyon arabirimindeki tüm metotları aynen kullanmaktadır.

HashSet sınıfı: Bu sınıf elemanlarını bir hash tablosunda tutar. Hash tablousu elemanlarına erişimi

kolaylaştırmak için elemanlarına anahtarlar vasıtasıyla ulaşılmasını sağlar.

LinkedHashSet Sınıfı: HashSet sınıfı ile aynı işi görmektedir. Bu sınıf elemanları koleksiyona eklenme sıralarına göre tutulmaktadır.

TreeSet Sınıfı: Bu sınıf set arabiriminden türeyen sortedSet arabirimini uygulamaktadır. SortedSet arabirimi koleksiyona ait elemanları sıralı olacak şekilde tutmasını sağlar.

List Arabirimi

Bu arabirim sayesinde koleksiyonumuzda tekrarlanan ya da tekrarlanmayan elemanları sıralı olarak tutabiliriz. Aynı zamanda elemanlara indeks yoluyla erişebiliriz.

List Arabiriminin metotları:

boolean add(indeks, eleman): Listeye, parametrede indeks verilen elemanın o indekse eklenmesini sağlar.

boolean addAll(indeks, koleksiyon): Listeye, parametrede verilen indeksten başlayarak parametrede verilen koleksiyonun eklenmesini sağlar.

int get(indeks): Listenin, parametre indeks verilen elemanı döner.

int indexOf(eleman): Parametrede verilen elemanın listedeki ilk indeksini döner.

int lastIndexOf(eleman): Parametrede verilen elemanın listedeki son geçtiği indeksini döner.

boolean remove(index): Parametrede verilen indeks ait elemanı çıkarır.

boolean set(index, eleman): Parametrede verilen indekse verilen elemanı atar.

subList(baslangic index, bitis index): Parametrede verilen indeks aralığına ait bir liste döner.

listIterator ve bazı özellikleri;

listIterator(): Bu metot sayesinde list koleksiyonumuza ait elemanlar arasında istenilen yönde ilerleyebilmekteyiz. Yani baştan sona, sondan başa listeleme yapabiliriz.

boolean hasNext(): Sıradaki elemanı döner. Sırada bir eleman varsa true döner. Yoksa NoSuchElementException hatası döner.

boolean hasPrevious(): Bir önceki elemanı döner. Eleman varsa true döner. Yoksa NoSuchElementException hatası döner.

int previousIndex(): Bir önceki elemanın indeksini döner.

int nextIndex(): Bir sonraki elemanın indeksini döner.

ArrayList Sınıfı

Bu sınıf elemanlarını boyutu değişen dinamik dizilerde tutmaktadır. Yani bu sınıfa ait dizilerin boyutları azaltılabilir ya da arttırılabilir. Bir başka ifade ile bir dizinin boyutunu değiştinde ilk dizideki elemanlar yeni diziye kopyalanır.

LinkedList Sınıfı

Bağlı listeler, Bu sınıf C dilindeki bağlı listelerin javadaki görüntüsüdür. Bu tip listelerde elemanlar kendisinden sonra gelen elemanların bilgilerini ya da adreslerini tutarlar. Bu sınıf herhangi bir listeye eleman ekleme konusunda ArrayList sınıfından daha etkilidir.

LinkedList Sınıfına ait metotlar;

boolean addFirst(eleman): Parametrede verilen elemanı listenin başına ekler.

boolean addLast(eleman): Parametrede verilen elemanı listenin sonuna ekler.

boolean removeFirst(eleman): Listenin başındaki elemanı döner ve listeden çıkarır.

boolean removeLast(eleman): Listenin sonundaki elemanı döner ve listeden çıkarır.

void getFirst(): Listenin başındaki elemanı döner.

void getLast(): Listenin sonundaki elemanı döner.

Vektör Sınıfı

Boyutları dinamik olarak değişebilen dizilerdir. ArrayList'e benzetilebilir.

Stack Sınıfı

List arabirimini uygulayan bu sınıf vektör sınıfından türetilmiştir. Bu sınıfın özelliği en son hangi eleman kaydedilmişse sorguda ilk onu dönmesidir. (LIFO — last in first out). Kendisine ait bazı metotları;

empty(): stack boş ise true, değilse false döner.

peek(): Stack'in son eklenen elemanını döner.

pop(): En son elemanı döner ve listeden çıkarır.

push(eleman): Parametrede verilen elemanı stack'in sonuna ekler.

search(): Parametrede verilen elemanın stack'teki yerini döner.

Queue (kuyruk) Sınıfı

Bu sınıf stack sınıfından farklı olarak özel durumlar dışında ilk giren ilk çıkar mantığı ile çalışmaktadır.

FIFO (first in first out)

Bu sınıfa ait metotlar;

add(eleman): Parametrede verilen elemanı kuyruğa ekler. İşlemin başarısız olması durumunda hata fırlatır.

offer(eleman): Parametrede verilen elemanı kuyruğa ekler. İşlemin başarısız olması durumunda null döner.

poll(): Kuyruğun başındaki elemanı kuyruktan çıkartır.

peek(): Kuyrukta sıradaki elemana ulaşmak için kullanılır.

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(anahtar,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.