**Set Yapısı**

Set, benzersiz öğelerin koleksiyonunu temsil eden bir arayüzdür. Set yapısı, bir koleksiyonun tekrarlayan öğeler içermediği ve öğelerin sırasız olduğu bir veri yapısıdır.

**Set Yapısının Özellikleri:**

**Benzersiz Elemanlar:** Set yapısı, her elemanın yalnızca bir kez bulunabileceği bir yapıdadır. Yani aynı elemanı birden fazla kez içeremez.

**Sırasız Elemanlar:** Set yapısı elemanlarını sıralı bir şekilde tutmaz. Elemanlar eklenme sırasına göre saklanmazlar.

**Eleman Ekleme:** Yeni bir eleman Set'e eklendiğinde, eğer bu eleman zaten Set'te varsa, Set bu elemanı ikinci kez eklemez. Yani Set yapısı benzersiz elemanları korur.

**Ana Fonksiyonlar:**

**add(element):** Kümeye bir eleman ekler. Eğer eleman zaten kümeye aitse ekleme işlemi gerçekleşmez.

**remove(element):** Belirtilen bir elemanı kümeden kaldırır. Eğer eleman kümede yoksa, hiçbir şey yapılmaz.

**contains(element):** Kümede belirtilen bir elemanın olup olmadığını kontrol eder. Eğer eleman kümede varsa true, yoksa false döndürür.

**size():** Kümedeki eleman sayısını döndürür.

**Yardımcı Fonksiyonlar:**

isEmpty(): Kümenin boş olup olmadığını kontrol eder. Eğer küme boşsa true, değilse false döndürür.

**Java Kodu:**

public class Data\_Structures\_Set {

public static void main(String[] args) {

Set<Integer> numbers = new HashSet<>(); // Set oluşturma

numbers.add(1); // Eleman ekleme

numbers.add(2);

numbers.add(3);

numbers.add(4);

System.out.println("Boyut: " + numbers.size()); // Boyutunu gösterme

System.out.println("Set'te '2' var mi? " + numbers.contains(2)); // Belirli bir elemanin olup olmadiğini kontrol etme

numbers.remove(3); // Eleman Çıkarma

System.out.println("'3' Set'ten kaldirildi.");

System.out.println("Set bos mu? " + numbers.isEmpty()); // Boş olup olmadiğini kontrol etme

System.out.println("Set icindeki elemanlar:"); // Set içindeki elemanlari yazdirma

for (Integer number : numbers) {

System.out.println(number);

}

}

}

**Map Yapısı**

Map, anahtar-değer çiftlerini depolamak için kullanılan bir arayüzdür. Map yapısı, her bir anahtarın yalnızca bir kez bulunabileceği ve her anahtarın bir değerle ilişkilendirildiği veri yapısıdır.

**Map Yapısının Özellikleri:**

**Benzersiz Anahtarlar:** Map yapısında her anahtar yalnızca bir kez bulunabilir. Aynı anahtarla birden fazla değer ilişkilendirilemez.

**Anahtarlar ve Değerler:** Map, her anahtara karşılık gelen bir değeri saklar. Bu değerler herhangi bir nesne olabilir ve null değeri de alabilir.

**Çeşitli Implementasyonlar:** Java'da farklı Map implementasyonları bulunur. En sık kullanılanlar arasında HashMap, TreeMap ve LinkedHashMap bulunur. Her biri farklı performans ve sıralama özellikleri sunar.

**Ana Fonksiyonlar:**

**put(K key, V value):** Belirtilen anahtarın verilen değerle eşlenmesini sağlar. Eğer anahtar zaten varsa, değeri günceller. Eğer anahtar haritada yoksa, yeni bir eşleme oluşturur.

**remove(Object key):** Belirtilen anahtara sahip eşlemeyi haritadan kaldırır. Eğer bu anahtar haritada bulunmuyorsa, işlem yapmaz.

**get(Object key):** Belirtilen anahtara sahip değeri döndürür. Eğer anahtar bulunamazsa, null değeri döner.

**Yardımcı Fonksiyonlar:**

**size():** Haritadaki eşleme sayısını verir.

**containsKey(Object key):** Belirtilen anahtarın haritada olup olmadığını kontrol eder. Eğer varsa true, yoksa false döner.

**keySet():** Haritadaki tüm anahtarların bir listesini döndürür.

**entrySet():** Haritadaki tüm girişlerin bir listesini döndürür.

**Java Kodu:**

public class Data\_Structures\_Map {

public static void main(String[] args)

{

Map<Integer,String> names = new HashMap<>();

names.put(1,"Emre"); // Eleman Ekleme

names.put(2,"Hakan");

names.put(3,"Yavuz");

names.put(4,"Baran");

System.out.println(names); // Hepsini Yazdırma

names.remove(2); // Eleman Silme

System.out.println(names.size()); // Boyutu Yazdırma

System.out.println(names.get(1)); // Belirli Bir Anahtarı Yazdırma

System.out.println(names.containsKey(5)); // Anahtar Var mı Kontrol Etme

System.out.println( names.keySet()); // Bütün Anahtarları Yazdırma

System.out.println(names.entrySet()); // Bütün Girişleri Yazdırma

}

}

**Kuyruk Java Kodu:**

public class Data\_Structures\_Queue {

public static void main(String[] args)

{

Queue<Integer> nums = new LinkedList<>(); //Queue oluşturma

nums.add(1); // Eleman Ekleme

nums.add(2);

nums.add(3);

nums.add(4);

System.out.println("Sira Boş mu: "+nums.isEmpty()); // Dolumu Kontrol Etme

System.out.println(nums.size()); // Queue Boyutunu Gösterme

System.out.println(nums.peek()); // Sıradaki Elemana Çıkarmadan Bakma

System.out.println(nums.poll()); // Sıradaki Elemanı Çıkarma

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

System.out.println(nums.peek());

}

}

**Yığın Java Kodu:**

public class Data\_Structures\_Stack {

public static void main(String[] args)

{

Stack<Integer> nums = new Stack<>(); //Stack Oluşturma

nums.push(1); // Eleman Ekleme

nums.push(2);

nums.push(3);

nums.push(4);

System.out.println("En Ustteki Eleman: "+nums.peek()); // Üstteki Elemana Çıkarmadan Bakma

System.out.println("Boyut: "+nums.size()); // Stack Boyutunu Gösterme

System.out.println("Cikarlian Eleman: "+nums.pop()); // Eleman Çıkarma

System.out.println("Boyut: "+nums.size());

System.out.println("Stack Bosluk Durumu: "+nums.empty()); // Stack'in boş olup olmadiğini kontrol etme

}

}

**Liste Java Kodu:**

public class Data\_Structures\_List {

public static void main(String[] args)

{

List<Integer> nums = new ArrayList<>(); // List oluşturma

nums.add(0);

nums.add(1);

nums.add(2);

for (int num : nums) //List Yazdırma

{

System.out.println(num);

}

System.out.println(nums.size()); // Listenin boyutunu alma

nums.add(3, 3); // Belirli bir indekse eleman ekleme

nums.get(3); // Belirli bir indeksteki elemanı alma

nums.remove(2); // Belirli bir indeksi silme

}

}

**Diziler Java Kodu:**

public class Data\_Structures\_Arrays {

public static void main(String[] args)

{

int[] nums = new int[5];

nums[0] = 1;

nums[1] = 2;

nums[2] = 3;

nums[3] = 4;

System.out.println(Arrays.toString(nums)); // Hepsini yazdırır.

nums[2] = 5; //Eleman Değiştirme

System.out.println("2 indeksli eleman degisti: " + nums[2]);

System.out.println(Arrays.toString(nums));

Arrays.sort(nums); // Diziyi Sıralama

System.out.println("Dizinin siralanmis butun elemanlari:");

for (int i = 0; i < nums.length; i++) // Dizinin Boyutunu Alma

{

System.out.println(nums[i]);

}

}

}