

# 2023-2024 Bilgisayar Mühendisliği

## Veri Yapıları ve Algoritmalar Dersi

### Ara Sınav Projesi

```
public class DataCube<T> {
```

```
// sınıfına, jenerik bir parametre olan T verdik. Bunun nedeni, veri küpünün farklı tiplerdeki  
değerleri kabul edebilmesi içindir.
```

```
private int dimensions;
```

```
// kısmı, veri küpünün kaç boyuta sahip olduğuyla ilgili bilgileri tutar. Ve bu bilgiler, veri  
küpü oluşturulduğunda, boyut değerleri dizisinin uzunluğuna göre belirlenir.
```

```
private int[] sizes;
```

```
// kısmı, her boyutun kaç elemana sahip olduğu bilgisini tutan bir integer dizisidir. Bu bilgiler,  
veri küpüne yeni bir boyut değeri eklendiğinde güncellenir.
```

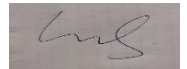
```
private T[][] dimensionValues;
```

```
// kısmı, her boyutun değer bilgilerini tutan bir T tipinde iki boyutlu bir dizidir. Bu dizi, veri  
küpü oluşturulduğunda, boyut değerleri dizisi ile başlatılır ve daha sonra yeni değerler  
eklenebilir.
```

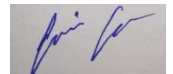
```
private Object[][][] cube;
```

```
// kısmı, veri küpünün değerleri atadığımız asıl kısımdır. Bu, üç boyutlu bir nesne dizisidir ve  
her hücrede bir değer saklar. Bu, veri küpü oluşturulduğunda, varsayılan olarak 1x1x1  
boyutunda bir dizi olarak oluşturulur ve daha sonra boyut değerleri eklendikçe değiştirilir.
```

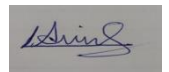
Mehmet Ali ŞAHİN



Emir Can BULDU



Abdulkadir UÇAR



**public DataCube(T[]... dimensionValues)** { // Kurucu metot tanımladık. Bu metot, boyut değerleri dizisi alır.Üç nokta birden fazla argümanla çalışabileceğimizi gösterir.

**this.dimensions = dimensionValues.length;** // Boyut sayısını, boyut değerleri dizisinin uzunluğuna eşitler

**this.sizes = new int[dimensions];** // Boyut sayısına göre bir dizi oluşturur ve sizes alanına atar.

**this.dimensionValues = dimensionValues;** // Boyut değerleri dizisini, dimensionValues alanına atar.

**this.cube = new Object[1][1][1];** // 1x1x1 boyutunda bir dizi oluşturur ve cube alanına atar

**for (int i = 0; i < dimensions; i++) {**

**for (T dimensionValue : dimensionValues[i]) {**  
// Her boyutun değerleri üzerinde bir döngü başlatır.İki noktayı java dilinin bir özelliği olan for-each için kullandık.

**addDimensionValue(i, dimensionValue);** // Her boyut değerini, veri küpüne eklemek için addDimensionValue metodunu çağırdık.

**}**  
**}**  
**}**

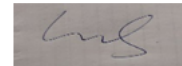
**public void addDimensionValue(int dimensionIndex, T dimensionValue)** {  
// Veri küpüne yeni bir boyut değeri eklemek için bir metot tanımladık. Ve bu metota, boyut indeksi ve boyut değeri parametre olarak belirledik.

**if (dimensionIndex < 0 || dimensionIndex >= dimensions)** { // Boyut indeksinin geçerli olup olmadığını kontrol edelim.

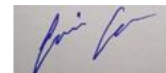
**throw new IllegalArgumentException("Geçersiz boyut endeksi");** // Geçersiz ise, bir istisna fırlatır.  
**}**

**int newDimensionSize = sizes[dimensionIndex] + 1;** // Yeni boyutun eleman sayısını, eski eleman sayısına bir ekleyerek hesaplar.

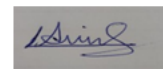
Mehmet Ali ŞAHİN



Emir Can BULDU



Abdulkadir UÇAR



```
Object[][][] newCube = new
Object[newDimensionSize][sizes.length][sizes[dimensionIndex]];
// Yeni boyutun eleman sayısına göre yeni bir nesne dizisi oluşturur.
```

```
if (dimensions > 1) {
    for (int i = 0; i < newDimensionSize; i++) {
        for (int j = 0; j < sizes.length; j++) {
            newCube[i][j] = new Object[sizes[dimensionIndex]];
        }
    }
}
// Yeni küpün her hücresine yeni bir nesne dizisi atar
```

```
if (i < sizes[dimensionIndex]) {
    int length = Math.min(cube[i][j].length, sizes[dimensionIndex]);
}
// Eski küpün ve yeni küpün hücrelerinin uzunluklarının minimumunu alır. Math burada
matematiksel işlemler yapmamıza yardımcı oldu.
```

```
System.arraycopy(cube[i][j], 0, newCube[i][j], 0, length);
// Eski küpün hücrelerini yeni küpün hücrelerine kopyaladık. Bu işlemle, veri küpüne yeni bir
boyut eklendiğinde, eski veri küpünün elemanlarını yeni veri küpüne aktarmak için kullandık.
}
}
}
}
```

```
sizes[dimensionIndex] = newDimensionSize;
cube = newCube; // Yeni küpü, cube alanına atar.
```

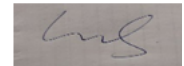
```
dimensionValues[dimensionIndex][newDimensionSize - 1] = dimensionValue;
// Yeni boyut değerini, dimensionValues dizisine atar.
}
```

```
public void setValue(T dimension1, T dimension2, T dimension3, Object value) {
// Veri küpünün bir hücresine herhangi bir değeri atamak için bir metod tanımladık. Bu metod,
üç boyut değeri ve bir nesne değeri alır.
```

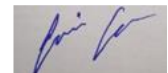
```
int index1 = getIndex(dimension1, 0); // Birinci boyut değerinin indeksini, getIndex
metodunu çağırarak bulur.
```

```
int index2 = getIndex(dimension2, 1);
int index3 = getIndex(dimension3, 2);
```

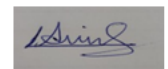
Mehmet Ali ŞAHİN



Emir Can BULDU



Abdulkadir UÇAR



**if (index1 == -1 || index2 == -1 || index3 == -1) {** // Herhangi bir indeksin bulunup bulunamadığını kontrol etmek için bir sorgu oluşturduk.

**throw new IllegalArgumentException("Bir boyut küp içinde mevcut değil.");**  
// Bulunamadıysa, bir istisna fırlatır.

**} else if (index1 >= sizes[0] || index2 >= sizes[1] || index3 >= sizes[2]) {** // Herhangi bir indeksin dizi sınırlarını aştığını kontrol eder.

**throw new ArrayIndexOutOfBoundsException("Dizi sınırları dışında bir endeksleme yapılmış.");** // Aştıysa, bir istisna fırlatır.

**} else {**  
**cube[index1][index2][index3] = value;** // Küpün ilgili hücreğine değeri atar.  
**}**  
**}**

**private int getIndex(T dimensionValue, int dimensionIndex) {**  
// Bir boyut değerinin indeksini bulmak için bir metot tanımladık. Bu metot, parametre olarak bir boyut değeri ve bir boyut indeksi alır.

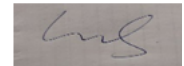
**for (int i = 0; i < dimensionValues[dimensionIndex].length; i++) {**  
**if (dimensionValue.equals(dimensionValues[dimensionIndex][i])) {**  
// Boyut değerinin, dimensionValues dizisindeki bir değere eşit olup olmadığını kontrol ettik.

**return i;**  
**}**  
**}**  
**return -1;** // Eşit değil

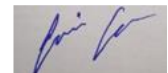
**public Object getValue(T dimension1, T dimension2, T dimension3) {**  
// Veri küpünden bir değeri okumak için bir metot tanımladık. Bu metot, üç boyut değeri alır ve bir nesne değeri döndürür.

**int index1 = getIndex(dimension1, 0);**  
**int index2 = getIndex(dimension2, 1);**  
**int index3 = getIndex(dimension3, 2);**

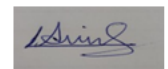
Mehmet Ali ŞAHİN



Emir Can BULDU



Abdulkadir UÇAR



```
    if (index1 != -1 && index2 != -1 && index3 != -1) {
        return cube[index1][index2][index3]; // Bulunduysa, küpün bulunan hücresindeki
        değeri döndürür.
    }
    return null; // Bulunamadıysa, null değeri döndürür.
}

public static void main(String[] args) {

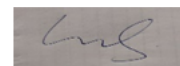
    DataCube<String> dataCube = new DataCube<>(new String[]{"1234567890"}, new
    String[]{"Veri Yapıları ve Algoritmalar", "Nesneye Yönelik Programlama"}, new
    String[]{"2022-2023", "2023-2024"});
    // String tipinde bir nesne oluşturduk ve boyut değerlerini vererek başlattık.

    // Değer atamak için
    dataCube.setValue("1234567890", "Veri Yapıları ve Algoritmalar", "2022-2023",
    40);
    dataCube.setValue("1234567890", "Veri Yapıları ve Algoritmalar", "2022-2023",
    90);
    dataCube.setValue("1234567890", "Nesneye Yönelik Programlama", "2022-2023",
    80);

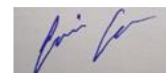
    // Değer okumak için
    Object value = dataCube.getValue("1234567890", "Veri Yapıları ve Algoritmalar",
    "2022-2023");
    // Veri küpünden girdiğimiz verilerin değerini okur. Buradaki değerler atadığımız değerlerle
    aynı olmalı eğer farklı bir verinin değerini istiyorsak bu kısım ona göre yeniden
    oluşturmalıyız.

    System.out.println("Value: " + value); // Okunan değeri ekrana yazdırır.
```

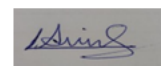
Mehmet Ali ŞAHİN



Emir Can BULDU



Abdulkadir UÇAR



```
public DataCube(T[]... dimensionValues) {  
    return -1;  
}  
  
public Object getValue(T dimension1, T dimension2, T dimension3) {  
    int index1 = getIndex(dimensionValue: dimension1, dimensionIndex: 0);  
    int index2 = getIndex(dimensionValue: dimension2, dimensionIndex: 1);  
    int index3 = getIndex(dimensionValue: dimension3, dimensionIndex: 2);  
  
    if (index1 != -1 && index2 != -1 && index3 != -1) {  
        return cube[index1][index2][index3];  
    }  
    return null;  
}  
  
public static void main(String[] args) {  
    DataCube<String> dataCube = new DataCube<>(new String[]{"1234567890"}, new String[]{"Veri Yapıları ve Algoritmalar",  
    // Değer atama  
    dataCube.setValue(dimension1: "1234567890", dimension2: "Veri Yapıları ve Algoritmalar", dimension3: "2022-2023", value: 40);  
    dataCube.setValue(dimension1: "1234567890", dimension2: "Veri Yapıları ve Algoritmalar", dimension3: "2022-2023", value: 90);  
    dataCube.setValue(dimension1: "1234567890", dimension2: "Kesneye Yönelik Programlama", dimension3: "2022-2023", value: 80);  
    // Değer okuma  
    Object value = dataCube.getValue(dimension1: "1234567890", dimension2: "Veri Yapıları ve Algoritmalar", dimension3: "2022-  
    System.out.println("Value: " + value);  
}
```

Mehmet Ali ŞAHİN

Emir Can BULDU

Abdulkadir UÇAR