Flutter ile Uygulama Geliştirme Kursu | Android & IOS

Collections

Kasım ADALAN
Elektronik ve Haberleşme Mühendisi
Android - IOS Developer and Trainer

Eğitim İçeriği

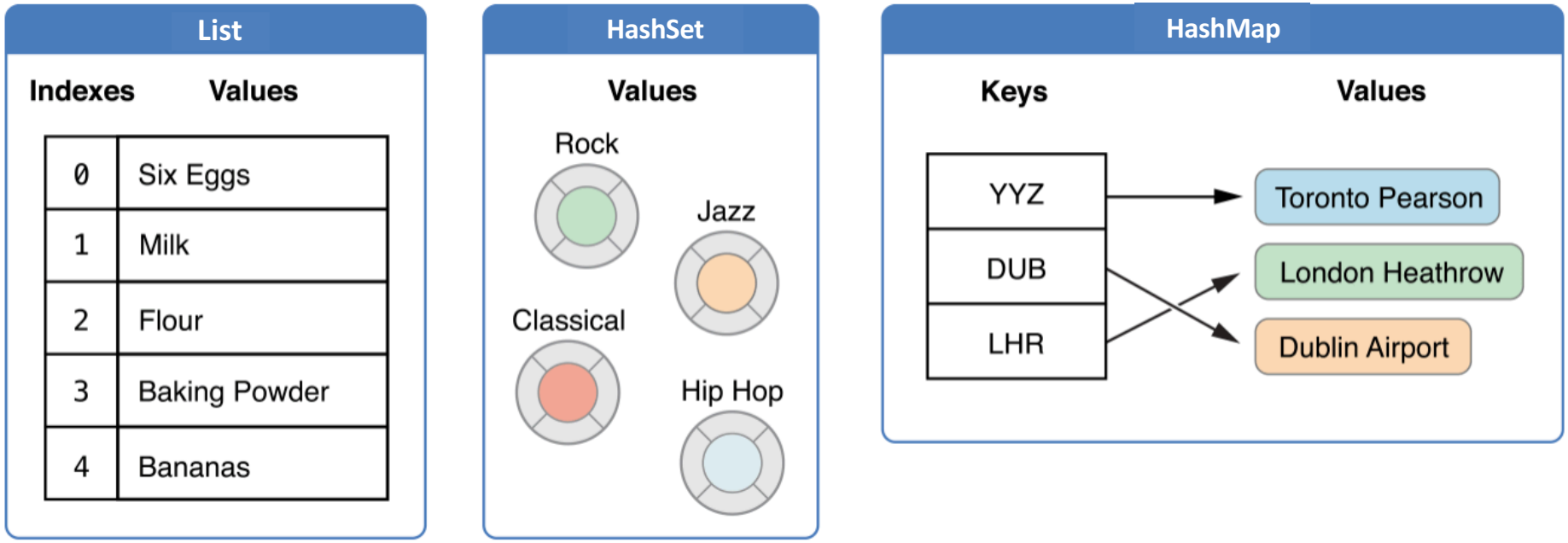
1. List
2. HashSet
3. HashMap

Sabit Boyutlu List İşlemleri

```dart
var dizi = List<int>.filled(5, 0);

dizi[0] = 10 ;//Veri ekleme
dizi[1] = 20 ;
dizi[2] = 30 ;
dizi[3] = 40 ;
dizi[4] = 50 ;

print(dizi);

//Veri okuma
print(dizi[1]);//20

dizi[3] = 99 ; //Güncelleme
print(dizi);
```

```dart
//Döngü ile veri okuma

for(var d in dizi){
  print("Sonuç : $d");
}

for(var i=0;i<dizi.length;i++){
  print("$i. indeksteki veri : ${dizi[i]}");
}
```

Collection Types



List

List

- Aynı türde verileri bir arada tutar.
- İndeks numaraları 0 dan başlar.
- Veri eklendikçe otomatik olarak boyutu artar.

```
var meyveler = <String>[];

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
meyveler.add("Kiraz");
```

meyveler

İndeks	Değer
0	Çilek
1	Muz
2	Elma
3	Kivi
4	Kiraz

List Tanımlama Yöntemleri

```
var iller = <String>[];

var plakalar = [16,34,6];
```

List'e Veri Ekleme ve Okuma

```
var meyveler = <String>[];

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
meyveler.add("Kiraz");

meyveler.add("Mandalina");//Sonuna ekleme
print(meyveler);//[Çilek, Muz, Elma, Kivi, Kiraz, Mandalina]

meyveler[2] = "Ananas"; //Güncelleme
print(meyveler);//[Çilek, Muz, Ananas, Kivi, Kiraz, Mandalina]

meyveler.insert(3, "Portakal");//List içine veriyi ekleme ve diğer verileri öteleme
print(meyveler);//[Çilek, Muz, Ananas, Portakal, Kivi, Kiraz, Mandalina]

//Veri Okuma
String str = meyveler[0] ; //Çilek
print(str);

print(meyveler[2]);//Ananas
```

meyveler

İndeks	Değer
0	Çilek
1	Muz
2	Elma
3	Kivi
4	Kiraz

List İşlemleri

```dart
var meyveler = <String>[];

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
meyveler.add("Kiraz");

print(meyveler.isEmpty);//Dolu mu Boş mu kontrolü : false
print(meyveler.length);//Boyutu : 5
print(meyveler.first);//Listenin ilk elemanı : Çilek
print(meyveler.last);//Listenin son elemanı : Kiraz

print(meyveler.contains("Kiraz"));//İçeriği listede arar : true

var liste = meyveler.reversed;//Listeyi terse çevirir.
print(liste);//(Kiraz, Kivi, Elma, Muz, Çilek)

meyveler.sort();//Listeyi sıralar
print(meyveler);//[Elma, Kiraz, Kivi, Muz, Çilek]

meyveler.removeAt(2);//İstenilen indeksteki veriyi siler.
print(meyveler);//[Elma, Kiraz, Muz, Çilek]

meyveler.remove("Kiraz");//İstenilen elemanı siler.
print(meyveler);//[Elma, Muz, Çilek]

meyveler.clear();//Bütün listeyi siler.
print(meyveler);//[]
```

List Iterating – Döngüler ile Veri Çekme

```
var meyveler = <String>[];

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
meyveler.add("Kiraz");

for(var m in meyveler){
  print("Sonuç : $m");
}
```

```
Sonuç : Çilek
Sonuç : Muz
Sonuç : Elma
Sonuç : Kivi
Sonuç : Kiraz
```

```
for(var i=0;i<meyveler.length;i++){
  print("$i. indeksteki veri : ${meyveler[i]}");
}
```

```
0. indeksteki veri : Çilek
1. indeksteki veri : Muz
2. indeksteki veri : Elma
3. indeksteki veri : Kivi
4. indeksteki veri : Kiraz
```

Nesne Tabanlı - List

```
class Ogrenciler{
  int no;
  String ad;
  String sinif;

  Ogrenciler(this.no, this.ad, this.sinif);
}
```

```
var o1 = Ogrenciler(100,"Ahmet","10F");
var o2 = Ogrenciler(200,"Mehmet","12A");
var o3 = Ogrenciler(300,"Zeynep","9C");

var ogrenciler = <Ogrenciler>[];

ogrenciler.add(o1);
ogrenciler.add(o2);
ogrenciler.add(o3);

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

List Sıralama

List Sıralama

Küçükten Büyüğe Sıralama

```
Comparator<Ogrenciler> siralama1 = (x,y) => x.no.compareTo(y.no);
ogrenciler.sort(siralama1);
```

Büyükten Küçüğe Sıralama

```
Comparator<Ogrenciler> siralama2 = (y,x) => x.no.compareTo(y.no);
ogrenciler.sort(siralama2);
```

Sıralamayı terse çevirmek için
parametrelerin yerini ters çeviriyoruz.

Örnek

```
class Ogrenciler{
  int no;
  String ad;
  String sinif;

  Ogrenciler(this.no, this.ad, this.sinif);
}
```

```
İlk Hali
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
**********
Öğrenci no : 300
Öğrenci ad : Zeynep
Öğrenci sınıf : 9C
```

```
var o1 = Ogrenciler(100,"Ahmet","10F");
var o2 = Ogrenciler(200,"Mehmet","12A");
var o3 = Ogrenciler(300,"Zeynep","9C");

var ogrenciler = <Ogrenciler>[];

ogrenciler.add(o1);
ogrenciler.add(o2);
ogrenciler.add(o3);

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

```
Sayısal Küçükten Büyüğe
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
**********
Öğrenci no : 300
Öğrenci ad : Zeynep
Öğrenci sınıf : 9C
```

```
Comparator<Ogrenciler> siralama1 = (x,y) => x.no.compareTo(y.no);
ogrenciler.sort(siralama1);

print("Sayısal Küçükten Büyüğe");

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}

Comparator<Ogrenciler> siralama2 = (y,x) => x.no.compareTo(y.no);
ogrenciler.sort(siralama2);

print("Sayısal Büyükten Küçüğe");

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

```
Sayısal Büyükten Küçüğe
**********
Öğrenci no : 300
Öğrenci ad : Zeynep
Öğrenci sınıf : 9C
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
```

```
Metinsel Küçükten Büyüğe
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
**********
Öğrenci no : 300
Öğrenci ad : Zeynep
Öğrenci sınıf : 9C
```

```
Comparator<Ogrenciler> siralama3 = (x,y) => x.ad.compareTo(y.ad);
ogrenciler.sort(siralama3);

print("Metinsel Küçükten Büyüğe");

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}

Comparator<Ogrenciler> siralama4 = (y,x) => x.ad.compareTo(y.ad);
ogrenciler.sort(siralama4);

print("Metinsel Büyükten Küçüğe");

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

```
Metinsel Büyükten Küçüğe
**********
Öğrenci no : 300
Öğrenci ad : Zeynep
Öğrenci sınıf : 9C
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
```

List Filtreleme

Filtreleme

- Listeyi filtrelemek için koşul yazılmalıdır.
- Iterable sınıfından yaralanılır.

```
Iterable<Ogrenciler> filtrelenListe = ogrenciler.where((ogrenci) {
  return ogrenci.no > 100 ;//Öğrenci no'su 100 den büyük olanları al.
});
```

Şart

```
ogrenciler = filtrelenListe.toList();
```

Filtrelenen sonuç listeye aktarılır.

```
for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

Filtreleme Örnek

```
class Ogrenciler{
  int no;
  String ad;
  String sinif;

  Ogrenciler(this.no, this.ad, this.sinif);
}
```

```
var o1 = Ogrenciler(100,"Ahmet","10F");
var o2 = Ogrenciler(200,"Mehmet","12A");
var o3 = Ogrenciler(300,"Zeynep","9C");

var ogrenciler = <Ogrenciler>[];

ogrenciler.add(o1);
ogrenciler.add(o2);
ogrenciler.add(o3);

Iterable<Ogrenciler> filtrelenListe = ogrenciler.where((ogrenci) {
  return ogrenci.ad.contains("t") ;//Öğrenci isimlerinde t harfi içerenleri al.
});

ogrenciler = filtrelenListe.toList();

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

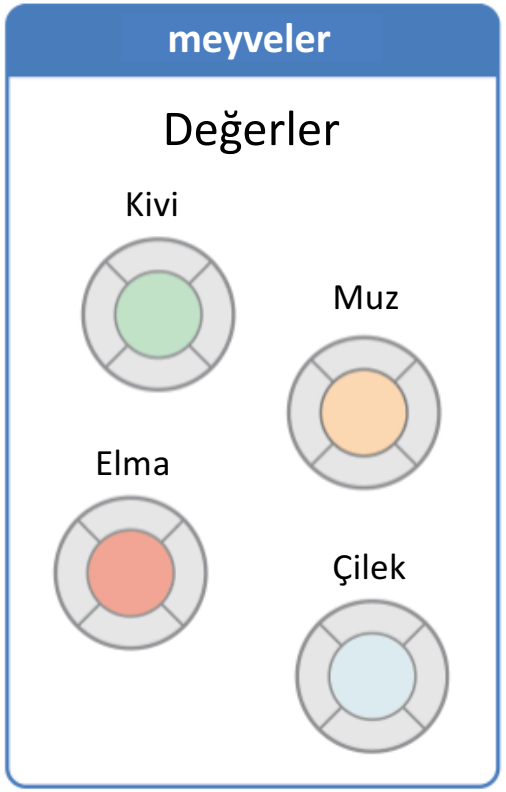
```
**********
Öğrenci no : 100
Öğrenci ad : Ahmet
Öğrenci sınıf : 10F
**********
Öğrenci no : 200
Öğrenci ad : Mehmet
Öğrenci sınıf : 12A
```

HashSet

HashSet

- List ile aynı özelliklere sahiptir.
- İçine eklenen veriler düzensiz rasgele yerleştirilir.
- Aynı veriden tekrar kayıt edilemez.

```
var meyveler = Set<String>();

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
```



HashSet Tanımlama Yöntemleri

```
var sayilar = HashSet<int>();

var isimler = HashSet.from(["Ahmet","Mehmet"]);
```

HashSet İşlemleri

```dart
var sayilar = HashSet<int>();

var isimler = HashSet.from(["Ahmet","Mehmet"]);

var meyveler = HashSet<String>();

meyveler.add("Çilek");
meyveler.add("Muz");
meyveler.add("Elma");
meyveler.add("Kivi");
print(meyveler);//{Kivi, Muz, Elma, Çilek}

meyveler.add("Elma");//Aynı veri tekrar kayıt edilemez.
print(meyveler);//{Kivi, Muz, Elma, Çilek}

print(meyveler.elementAt(1));//İstenilen indeksteki elemanu verir.

print(meyveler.length);//Boyutu verir : 4
print(meyveler.isEmpty);//Dolu mu boş mu kontrolü yapar : false

print(meyveler.contains("Muz"));//Liste içinde belirtilen eleman var mı : true
```

```dart
for(var m in meyveler){
  print("Sonuç : $m");
}

for(var i=0;i<meyveler.length;i++){
  print("$i. indeksteki veri : ${meyveler.elementAt(i)}");
}

meyveler.remove("Kivi");//İstenilen elemanı siler
print(meyveler);//{Muz, Elma, Çilek}

meyveler.clear();//Bütün verileri siler
print(meyveler);//{}
```

Nesne Tabanlı - HashSet

Nesne Tabanlı - HashSet

- Set yapı itibari ile içine insert edilen verileri rasgele sıralamaktadır.
- Bu rasgele sıralama ınt,string içeren set gibi ifadelerde kolaylıkla yapılabiliyor.
- Fakat set içine nesne yerleştirildiğinde nesne içindeki hangi değişkene göre bu rasgele sıralamayı yapacağını belirtmemiz gerekiyor.
- Örn : Öğrencinin nosuna göre mi ? adına göre mi ? sınıfına göre mi ? sıralama yapılacak

Örnek

```
class Ogrenciler{
  int no;
  String ad;
  String sinif;

  Ogrenciler(this.no, this.ad, this.sinif);

  @override
  int get hashCode => this.no;

  @override
  bool operator ==(other) {

    if(no == (other as Ogrenciler).no){
      return true;
    }else{
      return false;
    }

  }

}
```

```
var o1 = Ogrenciler(100,"Ahmet","10F");
var o2 = Ogrenciler(200,"Mehmet","12A");
var o3 = Ogrenciler(300,"Zeynep","9C");
var o4 = Ogrenciler(300,"Zeynep","9C");
//Aynı no daha önce kullanıldığı için hashsete eklenmez.

var ogrenciler = HashSet<Ogrenciler>();

ogrenciler.add(o1);
ogrenciler.add(o2);
ogrenciler.add(o3);
ogrenciler.add(o4);

for(var o in ogrenciler){
  print("**********");
  print("Öğrenci no : ${o.no}");
  print("Öğrenci ad : ${o.ad}");
  print("Öğrenci sınıf : ${o.sinif}");
}
```

Hash Map

HashMap

- Key ve value ilişkisi ile çalışır.
- Key ile verilere erişiriz.

```
var sayilar = HashMap<int,String>();

var oranlar = HashMap<double,String>();
```

Veri Ekleme – Güncelleme - Okuma

```dart
var iller = HashMap<int,String>();

iller[16] = "BURSA";//Ekleme
iller[34] = "İSTANBUL";

print(iller);//{16: BURSA, 34: İSTANBUL}

iller[16] = "YENİ BURSA";//Güncelleme
print(iller);//{16: YENİ BURSA, 34: İSTANBUL}

print(iller[34]);//Veri okuma
```

HashMap İşlemleri

```dart
var iller = HashMap<int,String>();

iller[16] = "BURSA";//Ekleme
iller[34] = "İSTANBUL";

print(iller);//{16: BURSA, 34: İSTANBUL}

iller[16] = "YENİ BURSA";//Güncelleme
print(iller);//{16: YENİ BURSA, 34: İSTANBUL}

print(iller[34]);//Veri okuma

print(iller.length);//Boyutu : 2

print(iller.isEmpty);//Boş mu dolu mu : false
print(iller.containsKey(17));//Belirtilen key var mı yok mu : false
print(iller.containsValue("İSTANBULX"));////Belirtilen değer var mı yok mu : false
```

```dart
var anahtarlar = iller.keys;

for(var a in anahtarlar){
  print("Sonuç : ${iller[a]}");
}

iller.remove(16);//Belirtilen key ile silme işlemi
print(iller);

iller.clear();//Bütün verileri silme
print(iller);
```

Nesne Tabanlı - HashMap

Örnek

```
class Ogrenciler{
  int no;
  String ad;
  String sinif;

  Ogrenciler(this.no, this.ad, this.sinif);
}
```

```
var o1 = Ogrenciler(100,"Ahmet","10F");
var o2 = Ogrenciler(200,"Mehmet","12A");
var o3 = Ogrenciler(300,"Zeynep","9C");

var ogrenciler = HashMap<int,Ogrenciler>();

ogrenciler[o1.no] = o1;
ogrenciler[o2.no] = o2;
ogrenciler[o3.no] = o3;

var anahtarlar = ogrenciler.keys;

for(var a in anahtarlar){

  var o = ogrenciler[a];

  if (o != null){
    print("**********");
    print("Öğrenci no : ${o.no}");
    print("Öğrenci ad : ${o.ad}");
    print("Öğrenci sınıf : ${o.sinif}");
  }

}
```

Teşekkürler...


kasım-adalan


kasimadalan@gmail.com


kasimadalan