

Kotlin OOP Eğitimi

Standart Programlama Yapıları

Kasım ADALAN

Elektronik ve Haberleşme Mühendisi

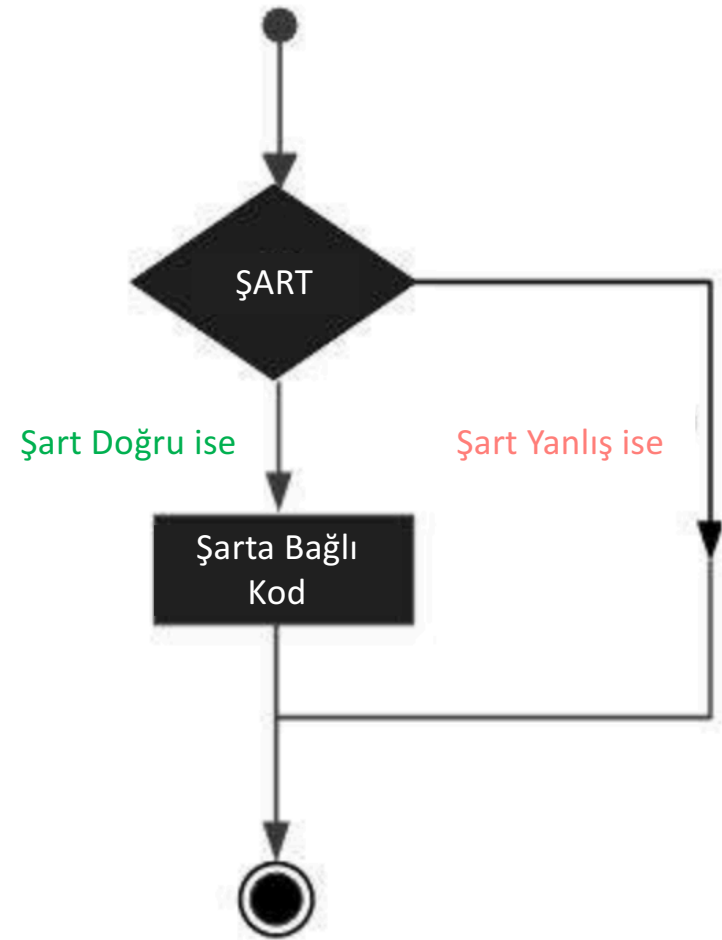
Freelance Software Developer

Eğitim İçeriği

1. Karşılaştırma Operatörleri
2. Kontroller
 - If & When yapısı
3. Döngüler
 - For
 - While
4. Diziler
5. Bazı hazır komutlar

Karar Alma Yapıları (if - when)

- Belirlenen şartlara göre karar alan yapılardır.
- Kararlar, şartın true veya false olmasına göre alınır.



Karşılaştırma Operatörü

- Koşul gereken yerlerde kullanılır. Örn : if , while vb.
 - **Örn** : A = 10 ve B = 20 olsun

Operator	Açıklama	Örnek
==	Eşittir	(A == B) is not true.
!=	Eşit değil	(A != B) is true.
>	Büyüktür	(A > B) is not true.
<	Küçüktür	(A < B) is true.
>=	Büyük Eşittir	(A >= B) is not true.
<=	Küçük Eşittir	(A <= B) is true.

Mantıksal Operatörler

- Mantıksal işlemleri bu operatörler ile yaparız.
- Her iki şart sağlandığında gibi.
- Örn : A = true ve B = false olsun

Operator	Açıklama	Örnek
&&	AND : Her iki koşulda sağlanıyorsa true olur	(A && B) is false.
	OR : İki koşuldandan biri sağlanıyorsa true olur	(A B) is true.
!	NOT : Var olan durumun tersine dönüştürür	!(A && B) is true.

if

if Yapısı

```
if ( ŞART ) {  
  
}  
ŞARTA BAĞLI KOD ( TRUE İSE )
```

```
var varA:Int = 10  
  
if(varA < 20){  
    print("varA 20 den küçüktür.")  
}
```

```
if ( ŞART ) {  
  
}  
else{  
  
}  
ŞARTA BAĞLI KOD ( FALSE İSE )
```

```
ŞARTA BAĞLI KOD ( FALSE İSE )
```

```
var not:Int = 10  
  
if(not >= 50){  
    print("Geçtiniz")  
}else{  
    print("Kaldınız")  
}
```

```
var girdi = "merhaba"  
  
if(girdi == "merhaba"){  
    print("Şart sağlandı")  
}else{  
    print("Şart sağlanmadı")  
}
```

Çoklu şart

```
var varA:Int = 100
var varB:Int = 200

if(varA == 100 && varB == 100){
    print("Her iki şartta sağlandı..")
}
```

```
var varA:Int = 100
var varB:Int = 200

if(varA == 100 || varB == 100){
    print("Şartlardan biri sağlandı..")
}
```


else if

```
var varA:Int = 100
```

```
if(varA == 20){
```

```
    print("varA , 20'e eşittir.")
```

```
} else if(varA == 50){
```

```
    print("varA , 50'e eşittir.")
```

```
} else {
```

```
    print("varA , Hiçbir şarta uygun değildir.")
```

```
}
```

Kısaltma

- İf yapısının pratik halidir.
- **İf (Koşul) Doğru Sonuç else Yanlış Sonuç**

```
var a = 10  
var b = 20  
  
val sonuc = if (a > b) a else b  
print(sonuc)
```

20

Örnek

Kullanıcıdan alınan seçime göre hesap yapan programı yazınız.

```
Dikdörtgen Alanı (1)
Çember alanı (2)
1
Seçiminiz : 1
Kısa kenar giriniz
10
Uzun kenar giriniz
20
Dikdörtgen Alanı
Sonuç : 200
```

```
Dikdörtgen Alanı (1)
Çember alanı (2)
2
Seçiminiz : 2
Yarı çap giriniz
20
Çember Alanı
Sonuc : 1256.0
```

Örnek

Kullanıcıdan alınan seçime göre hesap yapan programı yazınız.

```
Toplama (1)
Çıkarma (2)
Çarpma (3)
Bölme (4)
1
Birinci sayıyı gir
20
İkinci sayıyı gir
30
Toplama : 50
```

```
Toplama (1)
Çıkarma (2)
Çarpma (3)
Bölme (4)
2
Birinci sayıyı gir
20
İkinci sayıyı gir
30
Çıkarma : -10
```

```
Toplama (1)
Çıkarma (2)
Çarpma (3)
Bölme (4)
3
Birinci sayıyı gir
20
İkinci sayıyı gir
30
Çarpma : 600
```

```
Toplama (1)
Çıkarma (2)
Çarpma (3)
Bölme (4)
4
Birinci sayıyı gir
100
İkinci sayıyı gir
2
Bölme : 50
```

When

When

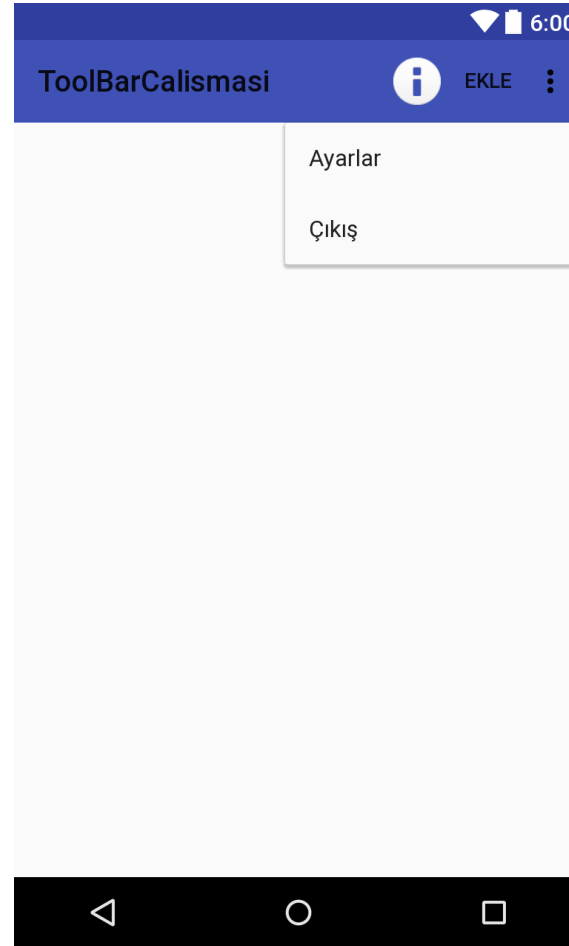
- Switch yapısının adı değişmiş halidir.
- else if yapısının daha pratik kullanımıdır.
- Case denilen durumlar sağlanırsa kod çalışır.

```
var x:Int=5

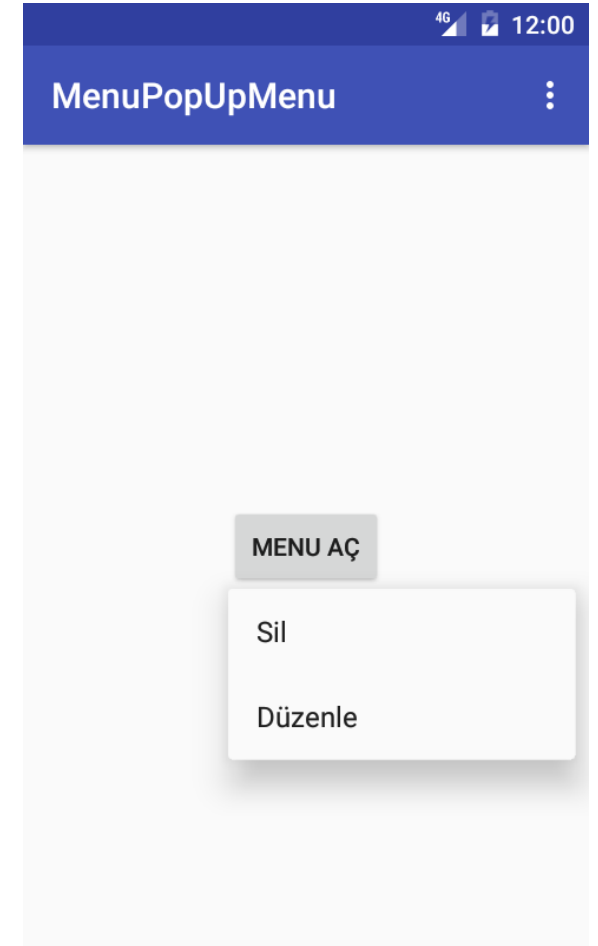
when (x) {
  1 -> print("x , 1'e eşittir")
  2 -> print("x , 2'e eşittir")
  else -> {
    print("x , 1 ve 2'ye eşit değildir")
  }
}
```

Örnek

Android menu
item'larını
yakalama yapısı.



Kasım ADALAN



15

Menu Olayının Yakalanması

```
override fun onOptionsItemSelected(item: MenuItem): Boolean {  
    when (item.itemId) {  
        R.id.action_bilgi -> {  
            Toast.makeText(applicationContext, text: "Bilgi Tıklandı", Toast.LENGTH_SHORT).show()  
            return true  
        }  
        R.id.action_ayarlar -> {  
            Toast.makeText(applicationContext, text: "Ayarlar Tıklandı", Toast.LENGTH_SHORT).show()  
            return true  
        }  
        R.id.action_cikis -> {  
            Toast.makeText(applicationContext, text: "Çıkış Tıklandı", Toast.LENGTH_SHORT).show()  
            return true  
        }  
        R.id.action_ekle -> {  
            Toast.makeText(applicationContext, text: "Ekle Tıklandı", Toast.LENGTH_SHORT).show()  
            return true  
        }  
        else -> return super.onOptionsItemSelected(item)  
    }  
}
```


Döngüler

Döngüler

- Döngüler belirli kodları tekrarlı çalıştırma amaçlı yapılardır.
 - Örneğin ; veri tabanından gelen verileri işlemek gibi.
- Döngü türleri ;
 - for in
 - while

Aralık Operatörleri

- Belirli değişken aralıklarını belirtir.
- **A..B** : 1..5 eşittir 1,2,3,4,5

for in döngüsü

```
for( veri in veri kümesi ) {  
    Tekralanan kod yapısı  
}
```

Örnekler

```
for (index in 1..5) {  
    println(index)  
}
```

```
1  
2  
3  
4  
5
```

```
val dizi = listOf(10, 20, 30, 40)  
for ( deger in dizi ){  
    println("Sonuç : $deger")  
}
```

```
Sonuç : 10  
Sonuç : 20  
Sonuç : 30  
Sonuç : 40
```

for döngüsü until kavramı

```
val test = ArrayList<String>()
```

```
test.add("elma")
```

```
test.add("armut")
```

```
test.add("çilek")
```

//until otomatik olarak dizilerdeki indeks değerinizi 1 azaltarak alır.

//normalde 0,1,2,3 beklenir ama untilden sonra geleni 1 eksik sayar

//ve 0,1,2 şeklinde sayar.

```
for( t in 0 until test.size){  
    Log.e( tag: "test",t.toString())  
}
```

for in döngüsü sınır değerleri

```
for (index in 0..10 step 2) {  
    println(index)  
}
```

İstediğimiz adım sayısında artış yapabiliriz.

0
2
4
6
8
10

```
for (index in 10 downTo 0 step 2) {  
    println(index)  
}
```

Tersten sayma işlemi yapabiliriz

10
8
6
4
2
0

Index ve Değeri aynı anda alma

```
val dizi = listOf(10, 20, 30, 40)

for ((index, deger) in dizi.withIndex()){
    println("$index. indeksin değeri : $deger")
}
```

```
0. indeksin değeri : 10
1. indeksin değeri : 20
2. indeksin değeri : 30
3. indeksin değeri : 40
```

While Döngüsü

```
while ( Şart ) {  
    Tekrarlanan ifade  
}
```

```
var index=0
```

```
while ( index < 5 ){  
    println("Sonuç : $index")  
    index = index + 1  
}
```

Sonuç : 0

Sonuç : 1

Sonuç : 2

Sonuç : 3

Sonuç : 4

Break : İşlemi bitirir

```
var index=0

while ( index < 5 ){
    if (index == 3) {
        break;
    }
    println("Sonuç : $index")
    index = index + 1
}
```

```
Sonuç : 0
Sonuç : 1
Sonuç : 2
```

Continue : İşlemi pas geçer.

```
for ( index in 0..5 ) {
    if (index == 3) {
        continue;
    }
    println("Sonuç : $index")
}
```

```
Sonuç : 0
Sonuç : 1
Sonuç : 2
Sonuç : 4
Sonuç : 5
```

Örnek

Konsoldan girilen adınızı konsoldan girdiğiniz tekrar sayısı kadar yazdıran programı yazınız.

```
İsim giriniz  
ahmet  
Tekrar sayısı giriniz  
5  
1. ahmet  
2. ahmet  
3. ahmet  
4. ahmet  
5. ahmet
```

Örnek

Konsoldan girilen işlenecek veri sayısını işleyerek ekrana yazdırınız.

```
İşlenecek veri miktarı giriniz
```

```
3
```

```
3. veri
```

```
2. veri
```

```
1. veri
```

Örnek

Konsoldan girilen sayının tek mi çift mi olduğunu bulan programı yazınız.
(Sonsuz Döngü)

```
Sayı giriniz
10
Çift sayıdır
Çıkmak için (1) - Devam etmek için diğer sayılar
2
Sayı giriniz
5
Tek sayıdır
Çıkmak için (1) - Devam etmek için diğer sayılar
1
Çıkış yapıldı
```

Örnek

Rasgele 10 tane sayı oluşturunuz

Diziler

Diziler

- Çoklu değişken oluşturmak gibi düşünebilirsiniz.
- Sabit boyutlu bir yapıdır. Baştan ne kadar boyutu olduğu bellidir
- İndeks numaralarına göre veriler eklenir.
- İlk indeks numarası 0'dan başlar.
- Mutable yapıdadır.
- Hem okunabilir hem üzerinde işlem yapılabilir.

```
val meyveler = arrayOf("Çilek", "Muz", "Elma", "Kivi", "Kiraz")
```

```
val meyveler = arrayOf<String>("Çilek", "Muz", "Elma", "Kivi", "Kiraz")
```

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

Array Tanımlama Yöntemleri

```
var dizi1 = Array<Int>(5){0} //5 elemanlı boş bir dizi  
var dizi2 = arrayOf(1,10,4,6,15)  
var dizi3 = arrayOf<Int>(1,10,4,6,15)  
val dizi4 = arrayOf<String>("Çilek","Muz","Elma","Kivi","Kiraz")  
var dizi5 = arrayOf(1,10,4, "Elma","Bursa")
```


Array Verilerine Erişim

```
val meyveler = arrayOf<String>("Çilek","Muz","Elma","Kivi","Kiraz")  
  
val str1 = meyveler[2] //Elma  
  
val str2 = meyveler.get(3) //Kivi  
  
println(meyveler.get(1)) //Muz  
println(meyveler[4]) //Kiraz
```

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

Array Üzerine Veri Ekleme

```
val meyveler = arrayOf<String>("Çilek", "Muz", "Elma", "Kivi", "Kiraz")  
meyveler.set(1, "Yeni Muz") // ["Çilek", "Yeni Muz", "Elma", "Kivi", "Kiraz"]  
meyveler[2] = "Yeni Elma" // ["Çilek", "Yeni Muz", "Yeni Elma", "Kivi", "Kiraz"]
```

Array İşlemleri

```
val meyveler = arrayOf<String>("Çilek", "Muz", "Elma", "Kivi", "Kiraz")

//Boş mu dolu mu kontrol : false
println(meyveler.isEmpty())

//Dizi boyutunu verir :)
println(meyveler.count())

//Dizinin ilk elemanı verir : Çilek
println(meyveler.first())

//Dizinin son elemanı verir : Kiraz
println(meyveler.last())

//Elemanın bulunduğu indeks numarasını verir : 1
println(meyveler.indexOf("Muz"))

//Dizi içinde bu eleman var mı : true
println(meyveler.contains("Muz"))

//Dizi içindeki en büyük eleman : Çilek
println(meyveler.max())

//Dizi içindeki en küçük eleman : Elma
println(meyveler.min())

//Diziyi sıralar : [Elma, Kiraz, Kivi, Muz, Çilek]
println(meyveler.sort())

println(meyveler.contentToString())

//Diziyi tersten sıralar : [Çilek, Muz, Kivi, Kiraz, Elma]
println(meyveler.reverse())

println(meyveler.contentToString())
```

Array Iterating – Döngüler ile Veri Çekme

```
val meyveler = arrayOf<String>("Çilek","Muz","Elma","Kivi","Kiraz")
```

```
for (meyve in meyveler) {  
    println("Sonuç : $meyve")  
}
```

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

```
for ((indeks,meyve) in meyveler.withIndex() ) {  
    println("Sonuç $indeks : $meyve")  
}
```

Sonuç 0 : Çilek
Sonuç 1 : Muz
Sonuç 2 : Elma
Sonuç 3 : Kivi
Sonuç 4 : Kiraz

Örnek

5 kişinin ismini konsoldan girerek kayıt işlemi yapınız ve girilen isimleri ekranda gösteriniz.

1. isim giriniz.

ahmet

2. isim giriniz.

mehmet

3. isim giriniz.

ece

4. isim giriniz.

sevda

5. isim giriniz.

ali

1. isim : ahmet

2. isim : mehmet

3. isim : ece

4. isim : sevda

5. isim : ali

Örnek

5 kişinin ismini ve telini konsoldan girerek kayıt işlemi yapınız ve girilen isimleri ve telleri ekranda gösteriniz.

1. ismi giriniz.

ahmet

1. teli giriniz.

53712

2. ismi giriniz.

mehmet

2. teli giriniz.

81231

3. ismi giriniz.

ece

3. teli giriniz.

2109

4. ismi giriniz.

ali

4. teli giriniz.

31134

5. ismi giriniz.

kemal

5. teli giriniz.

87134

1. isim :ahmet tel : 53712

2. isim :mehmet tel : 81231

3. isim :ece tel : 2109

4. isim :ali tel : 31134

5. isim :kemal tel : 87134

Örnek

5 adet dersin adı ve notu konsoldan girilecek. Karne şeklinde çıktı alınacak ,ortalama hesaplanacak ve ortalama 50'ye eşit veya üzerindeyse geçti diğer durumda kaldı mesajı verecek.

```
1. dersi giriniz.  
Tarih  
1. notu giriniz.  
40  
2. dersi giriniz.  
Fizik  
2. notu giriniz.  
30  
3. dersi giriniz.  
Kimya  
3. notu giriniz.  
90  
4. dersi giriniz.  
Türkçe  
4. notu giriniz.  
100  
5. dersi giriniz.  
Beden  
5. notu giriniz.  
90  
Tarih : 40  
Fizik : 30  
Kimya : 90  
Türkçe : 100  
Beden : 90  
Ortalama : 70  
Geçtiniz
```


Örnek

Tahmin Oyunu :
Program bir tane rasgele sayı
üretecek.
Kullanıcının 10 tahmin hakkı
olacak.
Tahminlere göre kullanıcıyı
yönlendireceksiniz.

Rasgele sayı : 70
1. tahmini giriniz
50
Tahmini arttır
2. tahmini giriniz
80
Tahmini azalt
3. tahmini giriniz
70
Tebrikler

Bazı Hazır Komutlar

- Rasgele sayı üret

```
val randomValues = Random.nextInt(0, 10) // 0 ile 9
```

- Matematiksel ifadeler

```
//Yukarı Yuvarlama  
val c = ceil(6.5) //7.0  
//Aşağı Yuvarlama  
val f = floor(6.5) //6.0  
//Karakök  
val s = sqrt(4.0) //2.0  
//Mutlak Değer  
val a = abs(-10) //10  
//Maksimum  
val mx = max(100,200) //200  
//Minimum  
val mn = min(100,200) //100  
//Üslü sayı  
val p = 2.0.pow(3.0) //8.0
```

Teşekkürler...



kasım-adalan



kasimadalan@gmail.com



kasimadalan