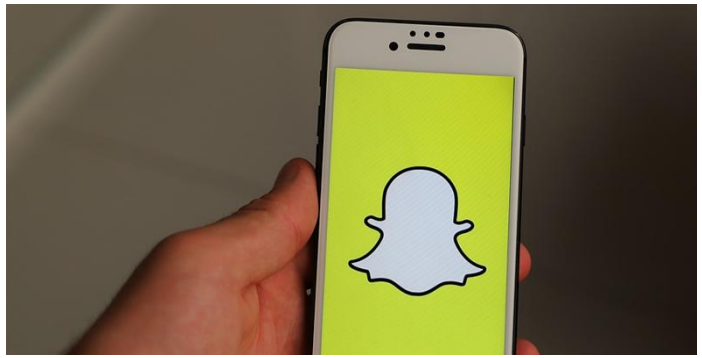
Mobil Uygulama Geliştirme

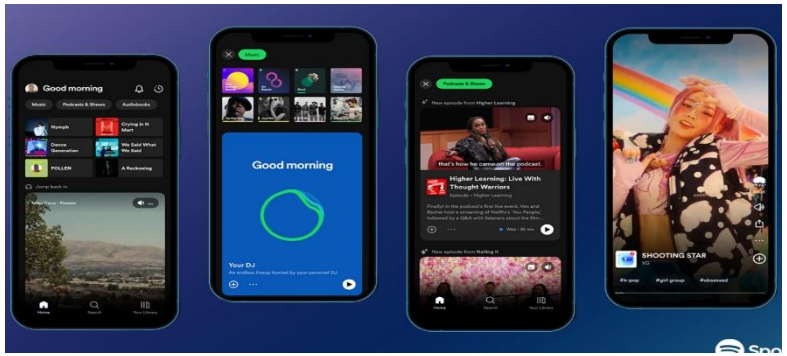
Öğr. Gör. Zafer SERİN

MOBİL UYGULAMA NEDİR?

- Mobil uygulama, akıllı telefonlar, tabletler ve diğer mobil cihazlar üzerinde çalışan yazılım uygulamalarıdır. Bu uygulamalar, kullanıcıların belirli görevleri yerine getirmesine, bilgi almasına, eğlenmesine veya işlerini yürütmesine yardımcı olur. Mobil uygulamalar, genellikle App Store (iOS için) veya Google Play Store (Android için) gibi platformlardan indirilebilir ve yüklenebilir.

MOBİL UYGULAMA NEDİR?





NATİVE(YEREL) MOBİL UYGULAMA NEDİR?

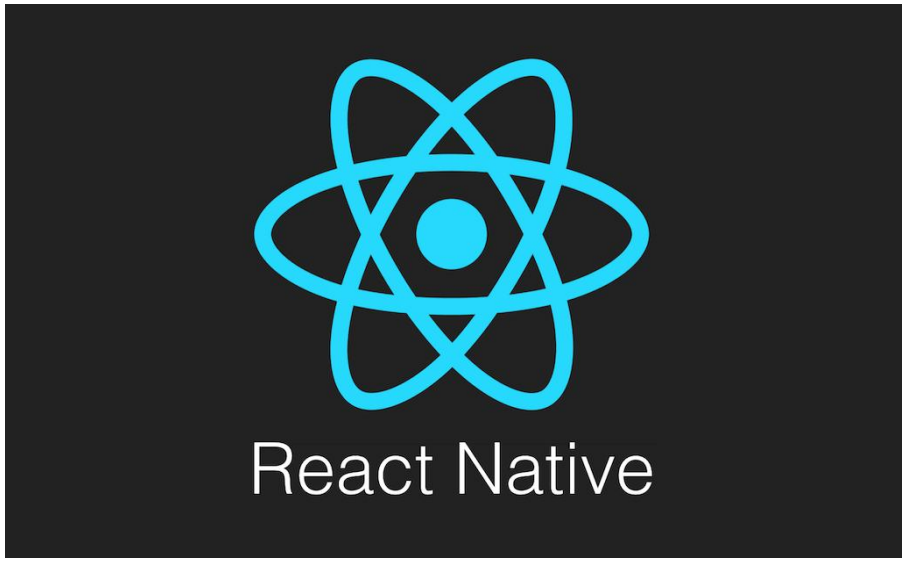
- Yerel uygulamalar, belirli bir işletim sistemi için özel olarak geliştirilen uygulamalardır. Bu uygulamalar, cihazın donanım ve yazılım özelliklerinden tam olarak yararlanabilir.
- **iOS Uygulama Geliştirme**: Swift veya Objective-C kullanılarak geliştirilir ve iOS cihazlarında çalışır.
- **Android Uygulama Geliştirme**: Kotlin veya Java kullanılarak geliştirilir ve Android cihazlarında çalışır.

NATİVE(YEREL) MOBİL UYGULAMA NEDİR?

CROSS PLATFORM MOBİL UYGULAMA NEDİR?

- Cross-platform uygulama geliştirme, tek bir kod tabanı ile birden fazla platformda (iOS, Android, Windows Phone vb.) çalışan uygulamalar geliştirmeyi amaçlar. Bu yaklaşım, geliştirme sürecini hızlandırır ve maliyetleri düşürür. **Örnekler:**
- **React Native**: Facebook tarafından geliştirilen ve JavaScript ile cross-platform uygulamalar geliştirmeyi sağlayan bir framework.
- **Flutter**: Google tarafından geliştirilen ve Dart dili ile cross-platform uygulamalar geliştirmeyi sağlayan bir framework.
- **MAUI**: Microsoft tarafından geliştirilen ve C# ile cross-platform uygulamalar geliştirmeyi sağlayan bir framework.

CROSS PLATFORM MOBİL UYGULAMA NEDİR?



HİBRİT MOBİL UYGULAMA NEDİR?

- Hybrid uygulama geliştirme, web teknolojileri (HTML, CSS, JavaScript) kullanarak mobil uygulamalar geliştirmeyi amaçlar. Bu uygulamalar, web görünümleri içerir ve genellikle bir webview içinde çalışır. **Örnekler:**
- **Ionic**: HTML, CSS ve JavaScript kullanarak mobil uygulamalar geliştirmeyi sağlayan bir framework.
- **Cordova (PhoneGap)**: Apache tarafından geliştirilen ve web teknolojileri ile mobil uygulamalar geliştirmeyi sağlayan bir platform.

HİBRİT MOBİL UYGULAMA NEDİR?

MOBİL OYUN GELİŞTİRME

- Oyun uygulamaları, özel olarak oyun geliştirme platformları ve motorları kullanılarak geliştirilir. Bu uygulamalar, grafikler, sesler ve oyun mekaniği gibi özellikleri içerir. **Örnekler:**
- **Unity**: C# dili ile 2D ve 3D oyunlar geliştirmeyi sağlar.
- **Unreal Engine**: C++ dili ile yüksek kaliteli oyunlar geliştirmeyi sağlar.

ANDROİD NEDİR?

- Android, Google tarafından geliştirilen ve açık kaynak kodlu bir mobil işletim sistemidir. İlk olarak 2008 yılında piyasaya sürülen Android, şu anda dünyadaki en yaygın kullanılan mobil işletim sistemlerinden biridir. Android, Linux çekirdeği üzerine inşa edilmiştir ve çeşitli donanım platformlarında çalışabilir. Android uygulama uzantıları genelde .apk(android package kit) ile ifade edilir.

Android

iOS NEDİR?

- iOS, Apple tarafından geliştirilen ve iPhone, iPad, iPod Touch ve diğer Apple cihazlarında kullanılan mobil işletim sistemidir. İlk olarak 2007 yılında iPhone ile birlikte piyasaya sürülen iOS, Apple'ın mobil cihazlar için özel olarak tasarlanmış bir işletim sistemidir. iOS, kullanıcı dostu arayüzü, güçlü güvenlik özellikleri ve zengin uygulama ekosistemiyle bilinir. iOS uygulama uzantıları genelde .ipa(ios package archive) ile ifade edilir.

KOTLİN NEDİR?

- Kotlin, JetBrains tarafından geliştirilen modern bir programlama dilidir. İlk olarak 2011 yılında duyurulan Kotlin, Java Virtual Machine (JVM) üzerinde çalışabilen statik olarak yazılmış bir dildir. Kotlin, Java ile tamamen uyumlu olması ve modern programlama özellikleri sunması nedeniyle hızla popülerlik kazanmıştır.

KOTLİN NEDİR?

- Google, 2017 yılında Kotlin'i Android uygulama geliştirme için resmi dil olarak desteklemeye başladı ve bu da Kotlin'in daha da yaygınlaşmasına katkıda bulundu.

KOTLİN ÖZELLİKLERİ

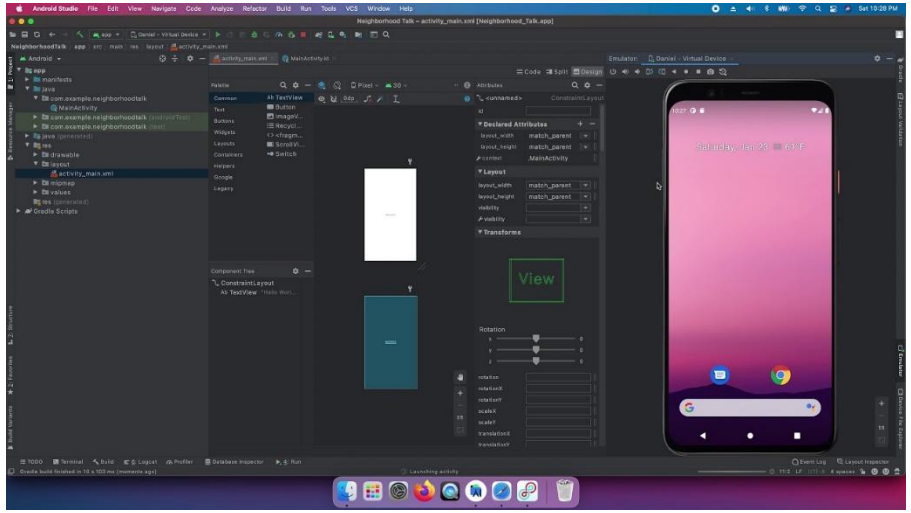
- **Java Uyumluluğu**
- **Null Güvenliği**
- **Veri Sınıfları**

KULLANILABİLECEK IDELER

- **Android Studio(Ücretsiz)**
- **IntelliJ(Ücretli)**
- **https://play.kotlinlang.org/ (Online ama yetersiz)**

ANDROİD STUDİO

- Android Studio, Google tarafından geliştirilen ve Android uygulamaları geliştirmek için kullanılan resmi entegre geliştirme ortamıdır (IDE). Android Studio, Java, Kotlin ve C++ dillerinde Android uygulamaları oluşturmak için kullanılır.



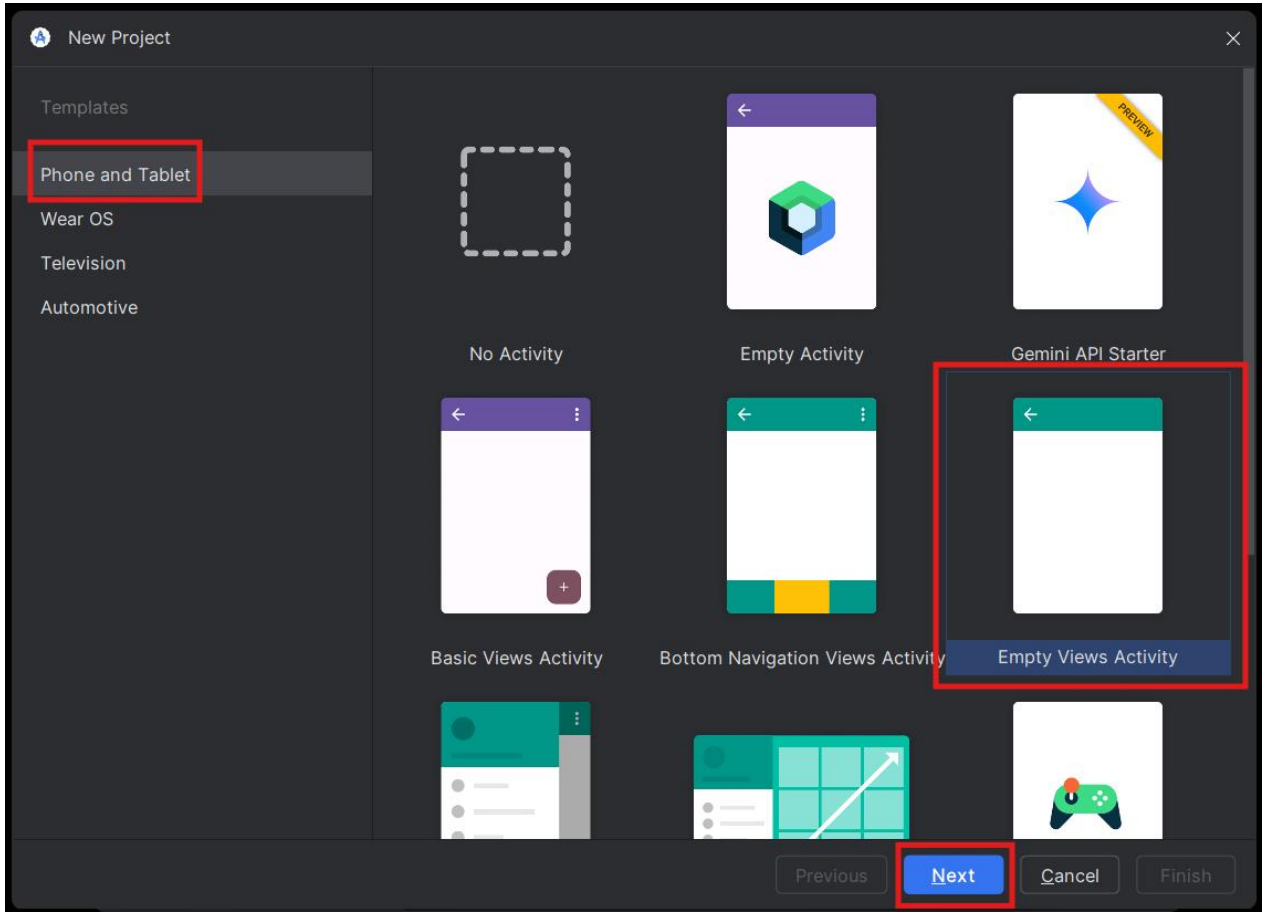
ANDROİD SDK NEDİR?

- Android SDK (Software Development Kit), Android uygulamaları geliştirmek için gereken araçların ve kütüphanelerin bir koleksiyonudur. Bu SDK, Android platformunda uygulama geliştirmek isteyen geliştiricilere, uygulamalarını oluşturmak, test etmek ve hata ayıklamak için gereken her şeyi sağlar.

PROJE OLUŞTURMA

- Android Studio'da 'New Project' seçilerek yeni bir proje oluşturulabilir. Burada hangi donanım(telefon, tablet, akıllı saat, televizyon, otomobil) için geliştirme yapılacaksa ona uygun templateler(şablonlar) görüntülenecektir. Başlangıç için telefon seçilerek 'Empty Views Activity' seçilebilir. Daha sonra 'Next' butonuna tıklanmalıdır.

PROJE OLUŞTURMA



PROJE OLUŞTURMA

- Sonraki aşamada proje ismimizi(Name) belirlememiz gerekiyor. Proje isminde **kesinlikle Türkçe karakter[ç, ğ, ö, ş, ü ve bunların büyük halleri ayrıca büyük i(İ) ve küçük ı'da kullanılmamalıdır!]**
- Paket adı(Package Name) bir uygulama için id gibi düşünülebilir. Yani o uygulama için özel tanımlanmış bir isimdir. 'com.isim.uygulamaadi' şeklinde kullanılabilir. Profesyonel dünyada 'com.sirketadi.uygulamaadi' olarak kullanılır. Uygulama play store gibi uygulama marketlerine yüklenirken bu paket adı geliştiriciden istenecektir ve tek olması gerekir.

PROJE OLUŞTURMA

- Kayıt Yeri(Save Location), uygulamanın bilgisayarınızda nereye kaydedileceğini belirtir. Burada da Türkçe karakter olmamasına dikkat edilmelidir.
- Dil(Language), kullanılacak Programlama Dili'ni belirtmektedir. Kotlin seçilmelidir.
- Minimum SDK, geliştirilen uygulamanın minimum hangi Android cihazlarda çalışacağını belirtir. Sonradan değiştirilebilir. **Burada minimum sürümün seçildiği unutulmamalıdır. Güncel olsun diye en son Android sürümü seçilirse uygulama çok çok az cihazda çalışabilir!**

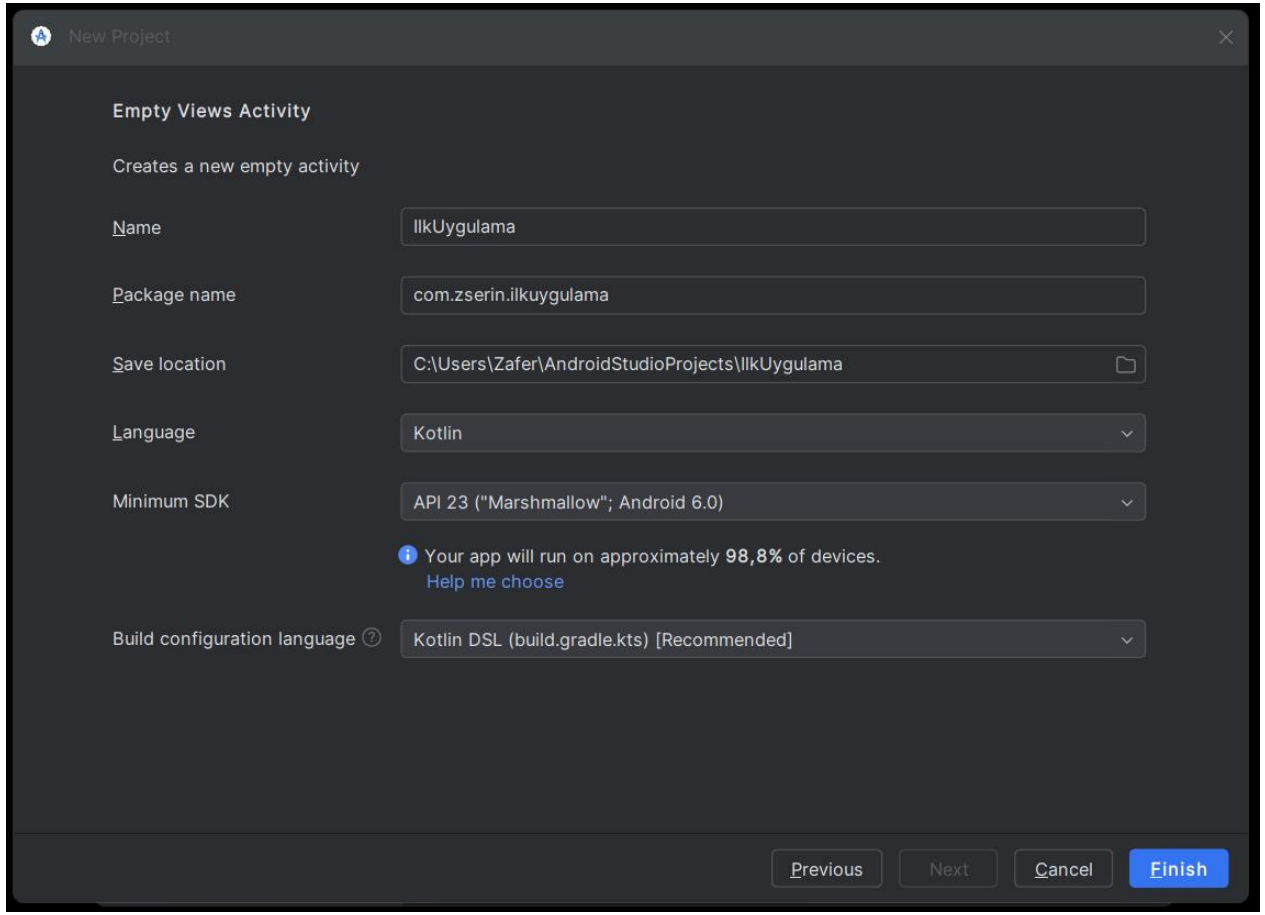
PROJE OLUŞTURMA

- **Minimum SDK, çok düşük seçilirse bu durumda neredeyse her cihazda çalışan bir uygulama elde edilir ama bu durumda yeni sürümlerde gelen özellikler kullanılamayabilir.** Sektörde Android 5, 6 ve 7 genel olarak iyi bir seçimdir.
- Build Configuration Language, yazdığımız kodları inşa eden derleyen toparlayan kısım olarak düşünülebilir. Kotlin DSL seçilebilir.
- Tüm bu seçimlerden sonra ‘Finish’ butonuna tıklanarak proje oluşturulabilir.

PROJE KLASÖRLEME YAPISI

- Manifest içerisinde projenin çok genel ayarları bulunmaktadır. Uygulama ikonu, başlığı, uygulama ilk açıldığında hangi ekran görünecek gibi…
- Kotlin+java klasörü içerisinde test yazan klasörler uygulamanın test edilmesi ile ilgili durumları içerir. Uygulamanın sadece paket adı yazan kısım ise asıl kodların yazılacağı kısımdır.
- Res klasörü ise resources(kaynak) ın kısaltmasıdır. Görsel, video, ses gibi dosyaları saklamak ve kullanmak için kullanabileceğimiz klasördür. Özellikle res klasörü altındaki layout bölümü uygulamanın tasarımı için temel bir dosyadır.

PROJE OLUŞTURMA



main Fonksiyonu

- Kotlin programlama dilinde bir programın başlangıç noktası olan main fonksiyonu aşağıdaki şekilde oluşturulabilir.

```
fun main()
{

}
```

print() ve println() Fonksiyonları

- Kotlin programlama dilinde ekrana çıktı vermek için print() ve println() fonksiyonları kullanılabilir. println() fonksiyonu bir alt satıra geçer. Kotlin programlama dilinde “;” kullanmak zorunlu değildir!

```
print(“Merhaba”)

println(“Nasilsin?”)
```

readln() Fonksiyonu

- Kotlin programlama dilinde kullanıcıdan bir girdi almak için readln() fonksiyonu kullanılabilir.

```
print("Lutfen bir deger giriniz..:")
var metin = readln()
print(metin)
```

Değişkenler

- Kotlin programlama dilinde aşağıdaki koşullara uyarak değişken ataması gerçekleştirilebilir.

degisken_belirteci degisken_adi = degisken_degeri

var yas = 30

Tür Belirterek Değişken Oluşturma

- Kotlin programlama dilinde aşağıdaki şekilde değişken tipi belirtilerek değişken tanımlanabilir.

var yas:Int = 30

Veri Tipleri

- Tam Sayılar: Long, Int, Short, Byte
- Ondalıklı Sayılar: Double, Float
- Metinsel İfadeler: String:yazılar, Char:Harfler-Karakterler
- Mantıksal İfadeler: Boolean: True veya False

Literals - Değerlerin Yazılma Kuralları

- Literals değişkenler için kullanılan değerlerin nasıl yazılması gerektiğini temsil eder.

“Zafer” // String (Metinsel ifade)

‘a’ // Char (Harfsel ifade)

18 // Tamsayı

1.78 // Double (Ondalıklı Sayı)

1.78f // Float (Ondalıklı Sayı)

Değişkenlere İsim Verme Kuralları

- Case sensitive'dir. Büyük küçük harf farkı vardır.
- Rakamla başlayamaz.
- _ hariç @, $ ve % gibi özel karakterler değişken ismi içerisinde kullanılamaz.
- Bazı örnekler: Azad, zara, abc, move_name, a_123, myname50, _temp, j, a23b9, retVal

Örnek

- Bir öğrencinin adını, yaşını, boyunu, başharfini ve devam durumunu içeren değişkenleri oluşturunuz. Aşağıda ki şartlara uyunuz.

ogrenci_ad(String)

ogrenci_yas(Int)

ogrenci_boy(Float)

ogrenci_bas_harf(Char)

ogrenci_devam_durumu(Boolean)

Örnek - 2

- Aşağıdaki veritabanı kaydını Kotlin'de değişkenler ile tutunuz. Türleri belirtiniz!

urun_id	urun_adi	urun_adet	urun_fiyat	urun_tedarikci
3416	Macbook Pro	100	42999	Apple

$ İşaretinin Kullanımı

- Kotlin programlama dilinde ‘$’ işareti ile string içerisinde biçimlendirme yapılabilir.

```
var isim = “Zafer”
var selamla = “Merhaba ${isim} hosgeldiniz”
print(selamla)
```

Yorum Satırı

- Kotlin programlama dilinde “//” ile tek satırda ve “/* */” ile çok satırda yorum satırı yapılabilir.

```
// Tek satırlı yorum satiri, bilgilendirme, aciklama
/*
Cok satirli
yorum satiri
aciklama
*/
```

Sabitler

- Kotlin programlama dilinde ‘var’ anahtar sözcüğü ile tanımlanan bir değişkenin değeri sonradan değiştirilebilirken ‘val’ ile tanımlanan bir değişkenin değeri sonradan değiştirilemez. ‘val’ anahtar sözcüğü derleme zamanında değişmez bir değer oluşturmak için kullanılır.

```
var yas = 30
yas = 50 // Bir problem olmaz

val pi_sayisi = 3.14
pi_sayisi = 3.15 // HATA! val ile tanımlanan değişkenin degeri degişemez
```

Sabitler

- Kotlin programlama dilinde ‘var’ anahtar sözcüğü ile tanımlanan bir değişkenin değeri sonradan değiştirilebilirken ‘val’ ile tanımlanan bir değişkenin değeri sonradan değiştirilemez. ‘val’ anahtar sözcüğü derleme zamanında değişmez bir değer oluşturmak için kullanılır.

```
var yas = 30
yas = 50 // Bir problem olmaz

val pi_sayisi = 3.14
pi_sayisi = 3.15 // HATA! val ile tanımlanan değişkenin degeri degişemez
```

Tür Dönüşümleri

- Sayısaldan sayısala dönüşüm
- Sayısaldan metine dönüşüm
- Metinden sayısala dönüşüm
- toDouble(), toFloat(), toLong(), toInt(), toShort(), toByte(), toChar(), toString()

Sayısaldan Sayısala Dönüşüm

Sayısal türlerin kendi arasında dönüştürülmesini ifade eder.

```
val i = 11
val d = 43.55

val sonuc1 = i.toDouble()
val sonuc2 = d.toInt()

println(sonuc1)
println(sonuc2)
```

Sayısaldan Metine Dönüşüm

Sayısal bir ifadenin metine dönüştürülmesini ifade eder.

```
val i = 11
val d = 43.55

val sonuc3 = i.toString()
val sonuc4 = d.toString()

println(sonuc3)
println(sonuc4)
```

Metinden Sayısala Dönüşüm

Metinsel bir ifadenin sayısal bir türe dönüştürülmesini ifade eder

```
val yazi = “3.14”

val sonuc5 = yazi.toDouble()
println(sonuc5)
```

Metinden Sayısala Dönüşüm

Metinsel bir ifade sayısal bir değeri dönüştürülürken dönüştürme başarılı olursa sonuç alınır; ancak dönüştürmesi başarısız olursa hata ile karşılaşılacaktır. Bu durumda toDouble() fonksiyonu yerine toDoubleOrNull() fonksiyonu kullanılabilir. Bu durumda değer dönüştürülemezse hata verilmesi yerine null değer alınacaktır.

```
val yazi = “3.14t”

val sonuc6 = yazi.toDoubleOrNull()
println(sonuc6) // null
```

Örnek 3

Kullanıcıdan kilo ve boy değerlerini alarak Vücut Kitle İndeksini(VKİ) hesaplayan Kotlin kodunu yazınız.

VKİ = Kilo / (Boy * Boy)

Örnek 4

Kullanıcıdan doğum yılını ve güncel yılı alarak yaşını hesaplayan Kotlin kodunu yazınız.

Aritmetik Operatörler

Operatör	Anlamı	Örnek	Sonuç
+	Toplama	5 + 3	8
-	Çıkarma	5-3	2
*	Çarpma	5*3	15
/	Bölme	10/2	5
%	Mod Alma(Kalma)	10 % 3	1

Not: + Operatörü metinleri(String) birleştirmek için de kullanılır.
“Merhaba” + “Dünya” sonucu “Merhaba Dünya” olur.

Atama Operatörleri

Operatör	Örnek	Açıklama(Eşdeğeri)
=	a = 5	a'ya 5 değerini ata
+=	a += 5	a = a + 5
-=	a -= 5	a = a - 5
*=	a *= 5	a = a * 5
/=	a /= 5	a = a / 5
%=	a %= 5	a = a % 5

Karşılaştırma ve Eşitlik Operatörleri

Operatör	Anlamı	Örnek	Sonuç
==	Eşit mi?	5 == 5	true
!=	Eşit değil mi?	5 != 3	true
>	Büyük mü?	5 > 3	true
<	Kücük mü?	5 < 3	false
>=	Büyük veya eşit mi?	5 >= 5	true
<=	Küçük veya eşit mi?	5 <= 3	false

Mantıksal Operatörler

Operatör	Anlamı	Açıklama
&&	VE (AND)	Her iki koşul da true ise true döner
||	VEYA (OR)	Koşullardan en az biri true ise true döner
!	DEĞİL (NOT)	true ise false, false ise true döner

Artırma ve Azaltma Operatörleri

Operatör	Anlamı	Örnek	Açıklama
++	1 artır	a++	a'nın değerini 1 artır
--	1 azalt	a--	a'nın değerini 1 azalt

Koşul Blokları

Kotlin programlama dilinde klasik if-else if-else yapısı ile koşul blokları oluşturulabilir.

```
if(kosul1)
{
// kosul1 saglanirsa yapilacaklar
} else if(kosul2)
{
// kosul1 saglanmiyor ama kosul2 saglaniyorsa yapilacaklar
} else{//kosul1 ve kosul2 saglanmiyorsa yapilacaklar}
```

Koşul Blokları

Kotlin programlama dilinde when yapısı ile koşul blokları oluşturulabilir.

```
val gun = 3
    when(gun)
    {
        1-> print("Pazartesi")
        2-> print("Sali")
        3-> print("Carsamba")
        4-> print("Persembe")
        5-> print("Cuma")
        6-> print("Cumartesi")
        7-> print("Pazar")
        else -> print("Hicbiri")
    }
```

Döngüler

Kotlin programlama dilinde tekrarlayan işlemler için döngüler kullanılabilir. En sık kullanılan döngü tipleri for ve while döngüleridir.

```
for (i in 1..3)
{
print(i)
}
```

Bu kod ekrana 1,2 ve 3 değerlerini yazdıracaktır.

Döngüler

```
for (i in 1..30 step 5)
{
print(i)
}
```

Bu kod ekrana 1,6, 11,16,21 ve 26 değerlerini yazdıracaktır.

Döngüler

```
for (i in 30 downTo 2 step 5)
{
print(i)
}
```

Bu kod ekrana 30, 25, 20, 15, 10 ve 5 değerlerini yazdıracaktır.

Döngüler

```
var i = 1
    while (i < 10)
    {
        println(i)
        i++
    }
```

Bu kod ekrana 1,2,3,4,5,6,7,8 ve 9 değerlerini yazdıracaktır.

Döngü Kontrol Mekanizmaları

break anahtar sözcüğü ile döngü sonlandırılabilir.

continue anahtar sözcüğü ile o an ki işlem sonlandırılıp döngünün bir sonraki aşamaya geçmesi sağlanabilir.

Örnek 5

Bilgisayarın 1 ile 100 arasında rastgele bir sayı tuttuğu bir "Sayı Tahmin Oyunu" programı yazmanız istenmektedir. Program başladığında kullanıcıdan bu sayıyı tahmin etmesini istemelisiniz. Kullanıcının girdiği her tahminden sonra, program kullanıcıya "Daha büyük bir sayı gir." veya "Daha küçük bir sayı gir." şeklinde ipuçları vererek onu doğru sayıya yönlendirmelidir. Kullanıcı doğru sayıyı bulana kadar tahmin isteme işlemi bir döngü içinde devam etmelidir. Kullanıcı doğru sayıyı bulduğunda ise "Tebrikler! [X] denemede doğru sayıyı buldun." şeklinde bir mesajla program sonlanmalıdır. Ayrıca, kullanıcının sayı yerine harf gibi geçersiz bir giriş yapma ihtimaline karşı basit bir kontrol ekleyerek "Lütfen sadece sayı giriniz." uyarısı vermelisiniz. (1..100).random() ile 1-100 aralığında rastgele sayı üretebilirsiniz.

Örnek 6

Belirli bir başlangıç bakiyesi (örneğin 1000 TL) olan bir ATM programı yazmanız isteniyor. Program, kullanıcı "Çıkış" seçeneğini seçene kadar sürekli çalışmalıdır. Her döngüde kullanıcıya "1- Bakiye Görüntüle", "2- Para Yatır", "3- Para Çek", "4- Çıkış" gibi seçenekler sunulmalıdır. Kullanıcının seçimine göre ilgili işlemi yapmalısınız. Para çekerken, kullanıcının çekmek istediği tutarın mevcut bakiyeden fazla olup olmadığını kontrol etmeli ve eğer bakiye yetersizse "Yetersiz bakiye!" uyarısı vermelisiniz. Para yatırma ve çekme işlemlerinde kullanıcıdan ne kadar işlem yapmak istediği sorulmalı ve bakiye güncellenmelidir. Kullanıcı 1, 2, 3, 4 dışında geçersiz bir seçenek girerse "Geçersiz işlem!" uyarısı gösterilmelidir.