\*\*\*\*\*Fotoğraflar eklemek faydalı ve daha okunaklı hale getirecektir\*\*\*\*\*

Kaynak dosyaların github linkelri kısmını düzenle bir ders dosyalarından hepsine ulaşım mevcut ek olarak da ders sonrası tekrarlardan spesifik olarak hangisine ulaşabileceklerini belirtmek güzel olur.

Bu yazımda genel olarak Android Studio platformunu üzerinde Kotlin programlama dilini kullanrak mobil uygulama geliştirme adımlarını anlatacağım.

Öncelikle parça parça gitmek adına Kotlin dilinin temelleri ve Anrodis Studio üzerinden proje oluşturmasından başlayacağım sonrasında …

Android studio uygulamasını <https://developer.android.com/studio?hl=tr> sayfasından indirme işlemini yaptıktan sonra kurulum kısmını tamamlıyoruz(gayet vasit bir şekilde ileri-next adımları ile kurulumu yapabiliriz). Sonrasında uygulamayı açtığımızda karşımıza çıkan sayfadan

New project -> Empty views activity ->

Name -> Name of project

Package name -> Package kısmını bir klasör gibi düşünebiliriz. Burada mümkün mertebe küçük harf kullanılmalı

Language -> Kotlin

Minumum Sdk -> Bu kısımda genelde mevcut olarak önerilen hangisi ise o kullanılabilir. Seçtiğimiz sürüm ve sonrasına uygun olacak şekilde programımızın çalışacağını ayarlamak içindir bu kısım. Buradaki yüzdelik ise yaklaşık olarak cihazların kaçına uygun olabileceğini göstermek içindir.

Build configuration language -> Değiştirmeye gerek yoktur.

Bu adımlardan sonra konfigurasyon işlemi yapıldıktan sonra açılan dosyalar üzerinde yapılacak işlemler;

app -> kotlin+java -> proje oluştururken verdiğimiz paket ismine gelip sağ tık ardından New -> kotlin Class/file -> file -> file name(Class ve file isimleri büyük harfle başlar). Bu adımlardan sonra elimizde verdiğimiz dosya ismiyle uyumlu .kt uzantılı bir dosya oluşmuş olması lazım. Buraya kodlarımızı yazıp çalıştırma işlemlerini yapmaya başlayabiliriz.

fun main(){ } bunu yazdıktan sonra (main yazıp tab tuşuna basarak direkt kendisi tamalar) kodlarımızı yazmaya başlayabiliriz

Kotlin değişkenler;

-Kotlinde biz genel olarak String, integer, double ve diğer değişkenlelri tutabilmek için var ifasesini kullanırız(variable) Örneğin ;

var number = 15

var name = “HBA” veya var name : String = “HBA”

Bu şekilde kendisi otomatik olarak hafıza da yer tutma işlemlerini ayarlamış olur. var yerine val ifadesini kullanarak değişken oluşturusak bu sefer bu ifadenin bir sabit olduğunu yani sonrasında değiştirmemize müsade etmediği anlamına gelir. Eğer bir ifade de değişkenlik yoksa val kullanmak hafıza kullanmı açısından çok daha avantajlıdır. Çünkü boyut sabit olarak ayarlanmıştır fakat var ifadelerde ileride bunun boyutu değiştirilebileceği için hafıza da yer ayırma işlemlerinde daha fazla yer tutup hafıza performansı açısından daha işlevsizdir.

Değişkenlerimizi veya bir ifadeyi terminale vermek için

println() fonksiyonu kullanılır. Burada kendi yazmak istediğimiz ifadeleri çift tırnak içinde yazarız ve bir değişkeni yazdırmak istersek eğer bir string ifade yoksa println içinde sadece değişken ismini vermemiz yeterli olacaktır. Ama bir string ifade ile beraber yazmak içinse şu şekilde bir yol izlenir;

println(“Name of student is : $studentName”) //buradaki $ işaretinden sonra değişken ismini yazabiliriz.

Yorum satırı için hem çift slaş // hemde b,rden fazla satırı yorum satırına alabilmek için /\* ile başlangıç /\* ile bitiş şeklinde yapılabilir.

Tipler arası dönüşüm;

3 çeşit tip dçnüşümü vardır;

1)Number to number;

Burada bir sayıyı başka türde sayıya dönüştürürüz örneğin double bir sayıyı interger dönüştümek gibi;

val doubleNum = 12.3

val intNum = doubleNum.toInt()// intNum şu anda 12 oldu

Burada toInt, toFloat … gibi çeşit çeşit ifade mevcuttur

2)Number to string;

Burada amaç bir sayıyı string ifadeye çevirmektir. Genelde string daha kapsayıcı bir tür olduğu için tercih edilir. Özellikle bir output verileceği zaman kullanılır(sayılar için)

val num = 12.3

val numString = num.toString()

3)String to number

Burada işler biraz karışır çünkü amaç bir stringden sayıya çevirmedir ve bunu yaarken eğer string içinde yanlışlıkla girilmiş bir ifade varsa(sayıya çevrilemeyecek) işlerin karışmaması için direkt olarak .toInt gibi bir ifade kullanmamız programın çökmesine sebebiyet verecektir. Bunu önlemek adına .toIntOrNull() fonksiyonu kullanılır böylece eğer çevrilemeyek bir ifade varsa null ifadesi atanmış olur değişkene.(1.ders sonu yazısı 23.04.2025/17.34)

Standart prgoramlama yapabilmek için kullandığımız bazı yapılar bulunur bunlardan birincisi koşullarla işlem yapmaktır. Herhangi bir ifadenin değişkenin değerine göre yapacağımız işler değişebilir ve bunun kontrolünü 2 yapı ile kontrol edeibliriz. Bunların ilki;

If yapısı;

if yapısı temelde içerideki ifade doğru olduğu takdirde(true) içerdiği kod bloğuna girer ve orada ne ypaılması isteniyorsa onu yapar yanlış(false) olması durumnda ise koşul bloğuna girmeden devam eder.Birden fazla koşulu yönlendirmek için if-else if-else kullanılır.Burada 1 tane if 1 tane else ifademiz olur arada ise isteiğimiz kadar else if kullanabiliriz. Yukarıdan aşşağı doğru sırayla tüm koşullar denenir ve ilk true ifadesine ulaşılunca o bloktaki kod çalıştırılır ve alt tardındaki diğer koşullar incelenmeye gerek duyulmaz.

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

When yapısı;

When yapısı genellikle menü vb işlemlerin gerektiği durumlarda kullanılan tek bir ifadenin(değişkenin) aldığı değerlere göre ilgili kod parçasının çalışmasına olanak veren bir yapıdır.Çalışması yine if gibidir, eşleşme olduktan sonra alt tarafın bakmaz.

metin, ekran görüntüsü, ekran, görüntüleme, yazılım içeren bir resim

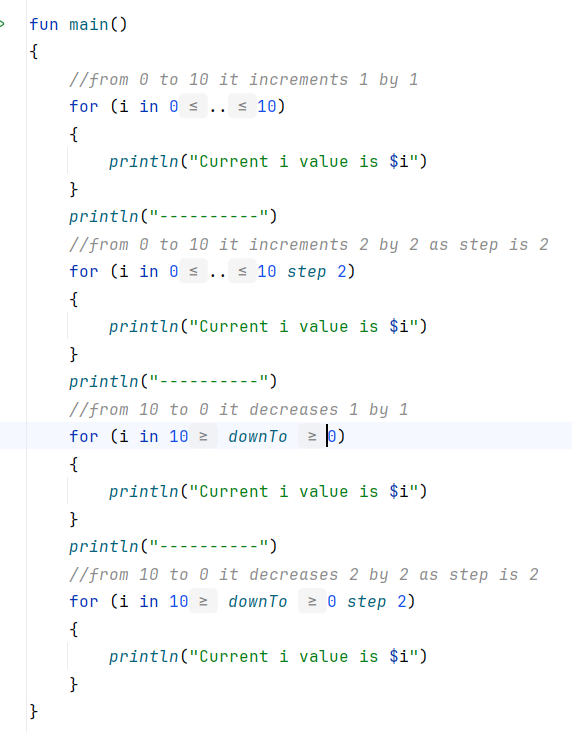
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Standart programlama kaynak dosyalarını yine github linki üzerinden My Application kısmındaki standard\_programing paketinin aldından bulabilirsiniz

Şimdide standart programlamanın bir diğer önemli yapısı olan döngülere(loops) giriş yapalım. Burada da genel olarak biz 2 tane yapı kullanırız.Bunlardan birincisi ;

For kullanımı;

For yapısı parantez içerisine yazılan bir ifade doğrultusunda ilgili aralıkta ifadenin isteiğimiz şekilde ilerlemesi(artış-azalış) durumuna göre istenilen kodun tekrar tekrar bir döngü halinde çalıştırılmasına olanak sağlar.

metin, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Bir diğer döngü yapımız ise while döngüsüdür.Burada da while(condition) buradaki condition kısmı true oldığı müddetçe döngü sürekli çalışır

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

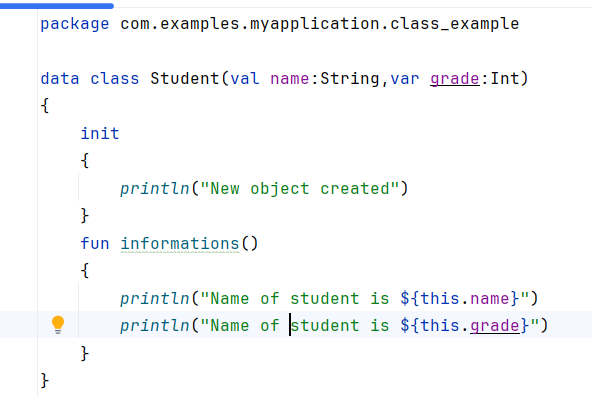
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

metin, yazı tipi, ekran görüntüsü, beyaz içeren bir resim

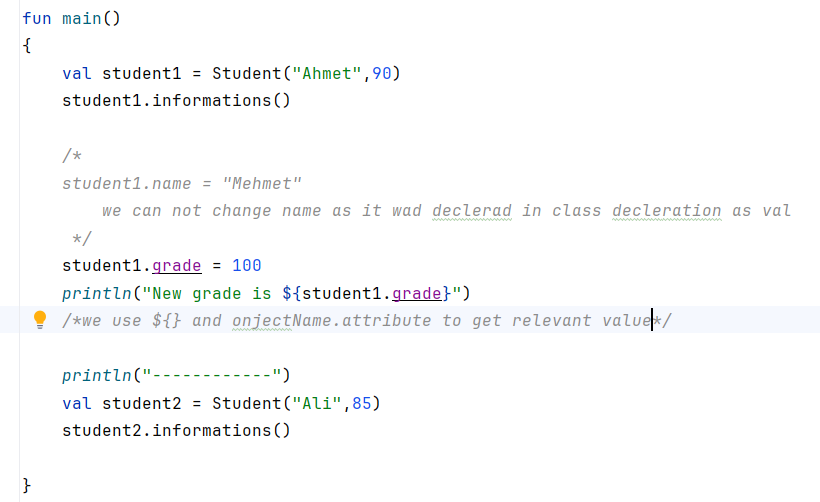
Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Genel olarak bir program yazarken bu yapıları(koşullar ve döngüler) sürekli olarak kullandığmı yapılardır.

Şimdide programlamanın en önemli olaylarından bir tanesine nesneye yönelik programlama(OOP)ya giriş yapalım. OOP kısaca nesneyi baz alarak nesne temelli uygulamalar yaptığımız bir yaklaşım, kod yazma biçimidir. OOP de en önemli unsurlardan biri doğru sınıf(class) oluşturabilmektir. Sınıf yapısını kabaca belli başlı özellikleri(attributes-data field) ve işlevleri(metod-fonksiyon) bulunan bir yapıyı toplu bir şekilde ele alabilmemizi(tutabilmemizi) sağlayan yapı olarak düşünebilriz. Ve biz sınıflardan nesneler oluşturarak onları kullanabiliriz. Bunu en iyi anlamanın yollarından birisi bir film-dizi izleme platformunu düşünelim. Karşımıza herbir film için görsel-isim-süre-puan vb bilgiler geliyor. Biz içinde bu bilgileri saklayan bir class oluşturabiliriz daha sonrasında her bir film için bu sınıftan bir nesne oluşturup onun özelliklerini atayabiliriz(görsel-isim…). Bu tip bir yapı kullanmak bize benzer özellikteki(yapıdaki) nesnelerin bilgilerini daha rahat tutabilmemiz için bir kapalı kutu(sınıf) sağladığı için ve bize herbirine ait bilgiileri atama kısmı kaldığı için işimizi daha düzenli anlaşılır yapmamızı sağlamış olacaktır.Sınıf yapısını biraz da kod üzerinde görelim. Bunun için öncelikle yeni bit Kotlin class/file kısmından bir class olşturamız gerekiyor(\*\*Unutmayınız class isimleri de file isimleri gibi büyük harfle başlamaludur.) Daha sonrasında alttaki gibi bir yapı yapmamız ilk başta yeterli olacaktır.



Buradaki data ifadesini kullanaran hem veritabanları ile kullanım da kolaylık sağlamış oluyor(ileride göreceğiz) hem de otomatik olarak getter-setter oluşturmuş gibi olduk. Kotlin modern bir dil olduğu için diğer dillerdeki(C++,Java…) constructor oluşturmadan burada görüldüğü gibi direkt parametre alarak nesne oluşturmamıza olanak saplıyor aynı zamanda eğer nesne oluşturulur oluşturulmaz(constrcutor) yapılmasını istediğimzi bir şey varsa bunu init (comes from initialize) ile sağlayabiliriz. Sınıf içinde this anahtar sözcüğünü kullanarak bulunduğnuz sınıftaki özelliğe işaret etmiş oluyoruz. Bu classın bir main içinde kullanımı ise şu şekildedir;



Burada görüldüğü gibi direkt class ismi ğzerinden bir nesne oluşturduk ve sonrasında bunun bilgilerini sınıfın içinde bulunana metodlardan birini çağırarak yazdırdık. Burada bizim name parametremiz sınıf tanımlamasında val oldığı için değiştirelmez fakat frade var olduğu için değiştirilebilir. Aynı şekilde nesnelerimizi val yaptığmız için nesneleriöiz değiştirilemez(atama-eşitleme yapılamaz).

Buradan hazır metodları işlemişken fonksiyon(class içinde olursa metod diye adlandırılır) nasıl oluşturulur birazda ona bakalım.

Fonksyionlar bizim bir işlemi(ya da işlem grubunu) tekrar tekrar her istediğimiz yerde kullanmamızı önlemek adına tek bir kere tanımlatıp ardından ihticaımız oldukça sadece fonksiyon çapırması ile o kodu çalıştıraiblmemizi sağlamak adına kullandığımız bir yapıdır. Örneğin bizim bir menümüz olsun biz kod içinde sürekli menüyü alt alta yazmaktansa tek bir yerde (fonksiyon tanımında) bir kere yazıp daja sonrasında menü yazdıracağımız zaman sadece fonksyion adı ile çağırma işlemi yapıp aynı sonıca ulaşabiliriz. Ayrıca fonksiyonların bir diğer avantajı ise bir yerde değişiklik yapmak istediğmizde her bir parçayı değiştirmke yerine sadece tanımlamadaki parçayı dğeiştirmemiz yeterli olacaktır.Unutmayınız fonksyion isimleri küçük harfle başlamalıdır.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir. metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

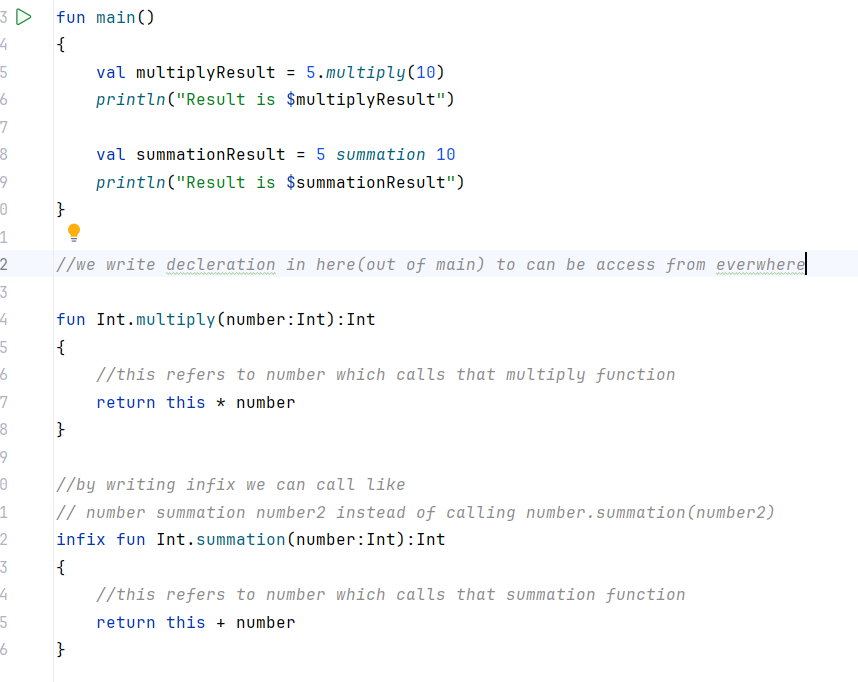
Burada görüldüğü gibi 3 farklı fonksiyon var. Bunların birincisi bir parametre almıyor ve herhangibir değer döndürmüyor. İkincisi ise String tipinde veri döndürüyor(çağrıldığı yerde kullanıcak şekilde). Üçüncüsü ise bir parametre alıyor ve bu parametreyi kullanıyor

İkinci kısımdaki fonksyion ise aynı isimli fakat aldığı paremtre tipi veya sayısı bakımından farklı oldup aynı isimle çağrılıp farklı sayı ve tipte parametre girmemize ve girilenlere göre ilgili fonksyionu çalıştırmamızı sağlayan overloading olayının uygulanmış halidir.

metin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Fonksiyonalrdaki bir diğer konuda Extension olayıdır. Extension genel olarak bizleremevcut sınıfa yeni metodlar eklememizi sağlayan yapıdır bunun nasıl yapıldığını alt taraktaki kod resminde anlatılmıştır.



(2.ders sonu yazısı 25.04.2025/17.06)