## ÖĞRENCİ BİLGİLERİ



**Bölüm Adı : Bilgisayar Teknolojileri**

**Program Adı : Bilgisayar Programcılıgı**

**Ders Adı : Android Programlamaya Giriş**

**Proje Adı : Sudoku Uygulması**

**HAZIRLAYAN-SUNAN**

**Ad-Soyad : HAYRETTİN KOÇ**

**Okul No : 18MY03022**

**Öğr.Gör : Nilgün İncereis**

**İÇİNDEKİLER**

ÖĞRENCİ BİLGİLERİ 1

ANDROİD STUDİO NEDİR? 3

Android Studio Artıları ; 3

Android Studio Eksileri ; 3

Android Studio Temel Özellikleri ; 3

KOTLİN NEDİR ? 4

KOTLİN DİLİNİN ÖZELLİKLERİ VE ORTAYA ÇIKIŞ AMACI 5

Kotlin Arayüz Nesneleri Ve Özellikleri 6

Sınıf(Class) 7

Nesne(Object) 7

Nokta(.) Operatörü 7

this Anahtar Sözcüğü 7

Özellikler(properties) 7

Metotlar(Methods) 7

Arayüz(Interface) 7

Kalıtım(Inheritance) 8

Soyutlama(Abstract) 8

Sınıf Oluşturmak 8

Değişken ve Metot Oluşturmak 8

Nesne Oluşturmak 9

PROJE BİLGİLERİ 10

AKIŞ DİAGRAMI 11

PROJENİN PROGRAMLANMASI KOD AÇIKLAMALARI 12

PROJE KODLARI 1; 13

PROJE KODLARI 2; 14

PROJE KODLARI 3; 15

PROJE KODLARI 4; 16

PROJE KODLARI 5; 16

KAYNAKÇA 17

## ANDROİD STUDİO NEDİR?

Android Studio, Android uygulama geliştiricileri için tasarlanan oldukça geniş kapsamlı ve ücretsiz programdır. Program beraberinde bir çok Android geliştirici aracıyla birlikte gelmektedir.  
  
Karışık problemleri kolayca çözmenizi sağlayabilecek olan Android Studio ile sadece uygulama geliştirme değil aynı zamanda varolan uygulamalardaki sorunları çözme işlemleride gerçekleştirebiliyorsunuz.  
  
Büyük ve geniş kapsamlı bir program olan Android Studio, Android geliştiricilerinin işlemlerini kolaylaştırmak ve onlara karşılaştıkları sorunlarda yardım etmek amacıyla Google tarafından hazırlanan başarılı ve etkileyici bir program.  
  
Android uygulamaları geliştiriyorsanız veya geliştirmek istiyorsanız Android Studio bu alanda en büyük yardımcılarınızdan biri olabilir. Ücretsiz olarak indirerek hemen kullanmaya başlayabilirsiniz.

## Android Studio Artıları ;

JetBrains IntelliJ Idea üzerine kurulmuş,  
Java geliştirmede en rahat idelerden biri hatta en iyileri(Netbeans kullanırım genelde Java geliştirmek için ama bu da çok iyi),  
Google bunu tamamen elden geçirip bir android geliştirme ortamı sağlamış. Plugin mantığı gibi görünsede değil,  
SDK yönetimi ve diğer işlemler çok iyi. Gereksiz menülerle gezinmiyorsun,  
IDE sayesinde Refactor, reflection vb işlemleri çok rahat yapabiliyorsunuz. IDE bütün olanakları sağlamış sana,  
Kod yazımı çok rahat. IntelliSense yi istersen hızlandırabiliyor ve her şeyi tamamlar hale getirebiliyorsun,  
Gradle ile istediğiniz dependency - library eklemek çok rahat. Composer - Nuget - Maven kullanmışlar için çok faydalı hale gelebiliyor.  
Debugger ve Publish çok rahat,  
Android içinde olup biten çok rahat izleniyor.(Eclipse burda bazen kitleniyor),  
Designer çok rahat.

## Android Studio Eksileri ;

Gradle bazen sapıtıyor veya çok yavaş çalışıyor. Bekletmesi en sinir olduğum yer,  
Cihaza yada emulatore deploy ederken beklemek çok can sıkıcı. Eclipse de başlattığınız anda emulator e yada cihaza gönderiyor direkt,  
Ram kullanımı bazen kontrolden çıkıyor(10 gb ram var makinada ama yaklaşık 3gb ram kullandığını gözlemledim),  
Arada designer sapıtıyor.

## Android Studio Temel Özellikleri ;

* Gelişmiş kod tamamlama, uygulamanın iç yapısını yeniden düzenleyerek geliştirme (refactoring) ve kod analizi yapabilen **akıllı kod düzenleyici**.
* **Anında Çalıştırma** kod ve kaynak değişikliklerinin bir cihaz veya emülatör üzerinde çalışan uygulamanıza hızlı bir şekilde yansıtılmasını sağlayarak düzenleme, derleme ve çalıştırma süreçlerini hızlandırır.
* Sanal ivme ölçer, ortam sıcaklığı, manyetometre ve diğer sensörlerle **zengin özelliklere sahip ve hızlı Android Emülatör**.
* **Tüm Android platformları desteklenir:** Android telefonlar ve tabletler, Android Wear, Android Auto ve Android TV.
* **Gradle tabanlı esnek yapılı derleme sistemi** derleme otomasyonu, bağımlılık yönetimi ve özelleştirilebilir APK oluşturma yapılandırmaları sunar.
* Yaygın olarak kullanılan uygulama özelliklerini oluşturmanıza yardımcı olacak **kod şablonları**.
* Uygulamaları daha sezgisel bir şekilde oluşturmak için plan modu ve sürükleyip bırakarak yerleşim düzenlemeyi destekleyen **Zengin Yerleşim Düzenleyici**.
* Büyük, karmaşık düzenlerin düz ve sadeleştirilmiş bir hiyerarşiye göre tasarlanması için **Yeni Sınırlama Düzeni yöneticisi** (Android API seviye 9'dan önceki sürümlerle uyumludur).
* Performans, kullanışlılık, sürüm uyumluluğu ve uygulama kodunuzdaki diğer sorunları tespit etmek için **Lint araçları**.
* Uygulamanızda hızlıca JNI bileşenleri oluşturabilmeniz için **C/C++** kod düzenleme ve LLDB tabanlı hata ayıklama desteği.
* Firebase SDK, Firebase Test Lab, Firebase Uygulama Dizinine Ekleme ve Google Cloud Platform için **yerleşik destek**.
* Her bileşenin boyuta olan katkısını anlayabilmeniz için APK'larınızın içeriğini incelemenize olanak tanıyan **APK Analizcisi**.
* Uygulamanızın etkileşimlerini kaydederek kullanıcı arayüzü testleri oluşturan ve ardından kullanıcı arayüzü test kodunu çıkaran **Espresso Test Kaydedici (Beta)**.
* Çalışma zamanında uygulamalarınızın görünüm hiyerarşisini incelemek için **Düzen Denetleyici**.
* Android cihazınızda OpenGL ES komut akışını yakalayan ve analiz için bunu Android Studio'nun içinde yeniden oynatan **GPU Hata Ayıklayıcısı (beta)**.

## KOTLİN NEDİR ?

* Kotlin JVM, yani (Java Virtual Machine) üstünde işlem gösteren bir programlama dili olarak ifade edilebilir. Buna ek olarak Javascript koduna da derlenebilmesi mümkün. Kotlin yazılım dilinin 2011 senesinde Jet Brains firması aracılığıyla geliştirilmeye başlanmış olduğunun duyurusu yapıldı ve Kotlin ilk sabit sürümünü (v1.0) 2016 senesinde çıkardı.  (JetBrain firması Intellij IDEA IDE’nin geliştiricisi olarak tanıtılabilir. Bundan daha sonra Google ile beraber Intellij IDEA bazlı Android Studio’yu geliştirdiler ve Android’in resmi olarak geliştirici ide’si olmuş oldular. Hem java geliştiricileri hem de android için Eclipse ide yerine Intellj IDEA ve Android Studio alıyor. Yazılım dili olan Kotlin Google I/O 2017 etkinliğin içerisinde Java ve C++ dan sonra olmak kaydıyla Android’in resmi olarak 3. resmi geliştirici dili unvanıyla adını duyurdu. Bu duyuru sonrasında da Kotlin’in tanınırlığının artması da kaçınılması mümkün olmayan bir durum.
* Kotlin yazılım dilinin öğrenilmesiyle birlikte hem Android uygulamaların gelişimleri sağlanabilir hem de server-side uygulamaları geliştirebilir ama bunların da gelişiminin yapılabilmesi için ilk olarak Kotlin Yazılım dilinin temelleri öğrenilebilir. Kotlin’in geliştirebilmesi amacıyla ilk olarak bilgisayara JDK’nin kurulması gerekiyor. Ide olarak da Intellij IDEA, Android Studio , Eclipse(JetBrains’in geliştirmiş olduğu kotlin plugini ile) ya da ide kullanmadan terminal vasıtasıyla Kotlin uygulamalar geliştirebilir.

## KOTLİN DİLİNİN ÖZELLİKLERİ VE ORTAYA ÇIKIŞ AMACI

* Kotlin’i Java platformunda işlem gösteren yeni bir dil olarak açıklamak mümkün. Yaklaşık olarak Java’nın kullanılabildiği bütün alanlarda kullanılabiliyor Kotlin. Buna sunucu programlama, web programlama ve Android programlama örnek olarak verilebilir. Kotlin dili Nesne Yönelimli bir programlama dili olarak da ifade edilebilir. Nesne yönelimi temel alan bütün programlama dillerinin anlaşılabilmesi için ilk olarak ortada yer alan ihtiyacın neden ortaya çıktığının anlaşılması gerekiyor. Örneğin B benzeri nesne yönelimi olmayan prosedürel şekilde olan bir programa dili örnek olarak verilsin ve bu tarz dillerde bu program her ne kadar birbirinden değişik daha farklı durumdaki dosyalara bölünebilse de içinde görevsel olarak ya da alt yordamlar barındırsa dahi temel manada ardı ardına yazılmış olan koskocaman satırlar yığını olarak ifade etmek yanlış olmaz.
* Bu duruma göre de bir programcının bir kodu okuyup okuduktan sonra anlamlandırması oranla daha zor olan bir seçenek. Kotlin benzeri bir nesne yönelimi de bir programlama dilindeyse de bu programı meydana getiren alt yapılar kendiliğinden yeten nesneler olarak bölünüyor ve adeta program içerisinde ufak birer ve birbirinden ayrı programcık şeklinde oluşturuluyor. Nesnelerin her biri uygulamanın farklı bir yerini temsil ediyor. Her nesne sahip olduğu verileri ve mantıksal açıdan işlemlerini barındırıyor ve bu nesneler kendi arasında iletişim halinde bulunuyor. Birçok nesne yönelimine sahip programlama dilinde nesnelerin gruplanabilmesi amacıyla sınıf diye adlandırılan yapılar bulunuyor. Sınıfları bahsi geçen nesnelerin benzer nitelikleri bünyesinde bulunduran ve nesnelerin meydana getirilmesi amacıyla tarif vazifesinde bulunan yapılar olarak da düşünmek mümkün.
* Kotlin dili de bu sınıf yapısıyla beraber çalışan nesne yönelimine sahip bir dil. Nesne yönelimine sahip olan programlama dillerinin geliştirilmesinin temel amacı ise bütün bir şekilde olacak şekilde programlama işini basitleştirmek ve yazılan kodların anlaşılma durumunu somutlaştırmak.

## Kotlin Arayüz Nesneleri Ve Özellikleri

### **Sınıf(Class)**

Sınıflar nesne özelliklerine sahip değişken ve metodların bulunduğu yapılardır. Sınıflar nesne tabanlı programlama kullanan her programlama dilinde kullanılabilir. Sınıf ile nesne aynı yapılar değildir. Sınıf nesne özelliklerini tutan soyut (abstract) ifadelerdir.  Sınıf ve nesne birbirine bağlıdır ama aynı kavramlar değildir.

### Nesne(Object**)**

Sınıf ile nesne arasında bir ilişki bulunduğunu yukarıda belirtmiştik. Sınıflar soyut ifadelerdir. Gündelik hayatta evlerin planlarına benzemektedir. Sınıflar gündelik hayatta planlar ise nesneler ise o planları takip ederek inşa edilen eve benzemektedir. Birden fazla ev yapmak ise bir sınıfın birden fazla nesnesi olacağı anlamına gelmektedir.

### **Nokta(.) Operatörü**

Programın içerisinde bulunan sınıflara ait özellikleri ve metotları kullanabilmek için bir sınıftan bir nesne oluşturmamız gerekmektedir. Nesne ile sınıfın özellik ve metodlarına erişebilmek için nokta(.) operatörü kullanılmaktadır. Nokta operatörü nesne ile birlikte kullanılır.

### **this Anahtar Sözcüğü**

this anahtar sözcüğü Javada da kullanılmaktadır. İşaret etmek anlamında kullanılır. Sınıf içerisinde kullanılır ve nesneyi oluşturduktan sonra oluşan nesneyi ifade etmeye yarar.

### **Özellikler(properties)**

Sınıf ve nesne ile ilişkili olan bir kavramdır. Nesnelerin özelliklerini ifade etmek için kullanılmaktadır. Bir sınıf oluşturulduğunda: Değişkenler, Sabitler, Metotlar ve Sınıfların içinde tanımlanmış sınıflar yani İnner Class oluşturulur. Saymış olduğumuz bu dört kavramda nesne özelliğine aittir.

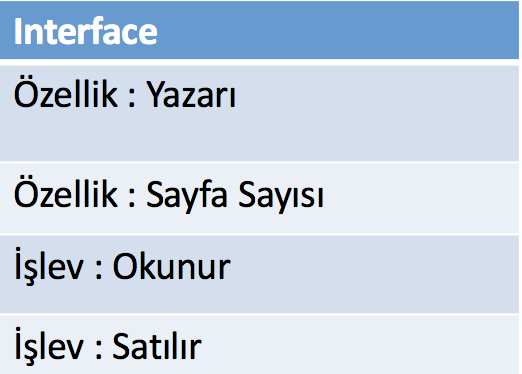
### **Metotlar(Methods)**

Metodlar uygulamalarda belirlenmiş işlevleri ve görevleri yapmak için dizayn edilmiş programcıklardır. Metod tanımlamak için fun anahtar sözcüğü kullanılır. Nesnelerin işlevlerini belirlemek ve nasıl kullanılacaklarını göstermek için metodlar kullanılır.

### **Arayüz(Interface)**

Nesne yönelimli programlama dillerinde en soyut kavramlar arayüzlerdir diyebiliriz.  Sınıflar soyuttur fakat metodlar belirli bir işi yerine getirmek üzere tasarlandıkları için soyut değildir. Arayüzlerde ise nesnelerin özellikleri ve işlevleri sadece liste olarak tanımlayabildiğimiz kavramlardır. Kısacası Arayüzlerde nesnelere hangi özellik ve işlevler verilecekse onların sadece sınıf isimleri belirtilir.

Örnek olarak kitap nesnesi için bir arayüz aşağıdaki gibidir :



### **Kalıtım(Inheritance)**

Kalıtım gündelik hayattaki kullanımdan esinlenerek oluşturulmuş bir yapıdır. Nesneler kalıtım özelliğini kullanarak özelliklerini başka nesnelere aktarabilirler. Kalıtım kullanılarak oluşturulan bir sınıftan yeni sınıflar türetilebilir. Kotlin dilinde tüm sınıflar Any sınıfından türetilerek elde edilmiştir. Kalıtım kullanılarak kod karmaşasından kurtularak daha verimli geliştirme yapılabilir.

### **Soyutlama(Abstract)**

Kalıtım yoluyla alınan özelliklerin türetilen sınıf içinde override edilerek farklı özellikler kazandırılmasıdır. Kotlin’de soyutlama iki farklı şekilde yapılmaktadır: Abstract sınıflar ve Interface kullanılarak soyutlama işlemi yapılabilir.

### Sınıf Oluşturmak

Sınıflar, nesneleri örneklendirebileceğimiz bir yapı planlarına benzemektedir. Java, C++ ve C# dillerinde olduğu gibi Kotlin’de de sınıf oluşturmak için class anahtar sözcüğü kullanılır.

class sinifAdi{

// Kodlar

}

## Değişken ve Metot Oluşturmak

Değişkenler nesne ile ilgili verileri tutacağımız alanlardır. Metotlar nesnelerin işlevlerini belirlediğimiz alanlardır. Aşağıdaki örnekte sınıf içerisinde 3 adet değişken ve 1 adet metot kullanılmıştır. Değişkenler nesnenin isim, yaş, numara bilgilerini tutarken, metod ise bu değerleri ekrana yazdırmak için kullanılır.

class ogrenciSinifi{

    var name : String = "Halil Özel"

    var number: Int= 1234

    var age: Int = 21

    fun bilgileriGoster(){

        println("name : "+name)

        println("number : "+number)

        println("age : "+age)

    }

}

ogrenciSinifi adında bir sınıf oluşturduk. Sınıfın içerisine name, number ve age özellikleri ile birlikte bilgileriGoster() metodu tanımlanmıştır.

## Nesne Oluşturmak

Sınıfların soyut kavramlar olduğunu daha önceden belirtmiştik. Sınıfların kodlarını etkili bir hale getirmek için nesne oluşturmalıyız. Kotlin dilinde nesne oluşturmak için aşağıdaki yapı kullanılır :

|  |  |
| --- | --- |
|  | var nesneAdi=SinifAdi() |

Öğrendiklerimizi koda dökelim.

var ogrenci1=ogrenciSinifi()

ogrenci1 adında bir nesne oluşturduk. Bu işlemden sonra ogrenci1 nesnesi ogrenciSinifi icerisinde bulunan metot ve özelliklere erişilebilir hale getirdik. Nesnenin sınıf içerisindeki özelliklerine erişilebilmesi için nokta operatörü kullanılır.

ogrenci1.age

 ogrenci1.name

 ogrenci1.number

Son olarak tüm kodları toplayı bir hale getiriyoruz.

class ogrenciSinifi{

    var name : String = "Halil Özel"

    var number: Int= 1234

    var age: Int = 21

    fun bilgileriGoster(){

        println("name : "+name)

        println("number : "+number)

        println("age : "+age)

    }

}

fun main(args: Array<String>) {

    var ogrenci1=ogrenciSinifi()

    ogrenci1.age

    ogrenci1.name

    ogrenci1.number

    ogrenci1.bilgileriGoster()

}

Kodumuzun çıktısı ise aşağıdaki gibidir

|  |  |
| --- | --- |
|  | name : Halil Özel  number : 1234  age : 21 |

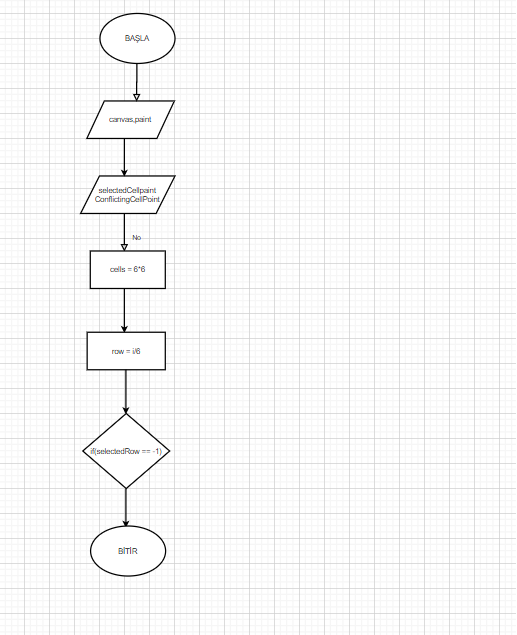
## PROJE BİLGİLERİ

PROJE ADI : Sudoku Uygulması

PROJE KONUSU : İnsanların kağıt üzerinde oyması yerine Android telefonlarda oynamasını sağlamak bu sayede istedikleri zaman istedikleri yerde oynayabileceklerdir.

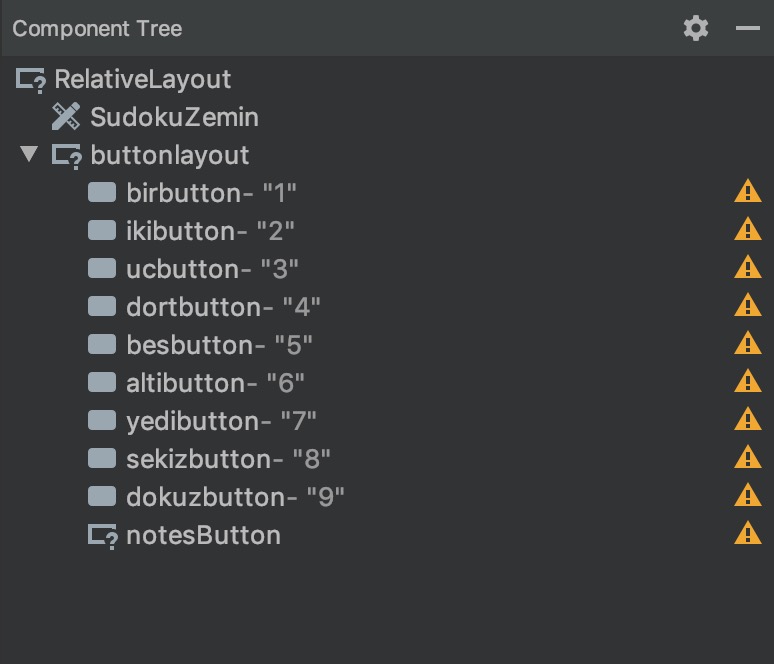
ALGORİTMA PLANLANMASI : Sudoku oyununda bildiğimiz üzere 3x3 olacak şekilde toplamda 9 adet kutu bulunmaktadır. Her satır ve sütun olarak düşünürsek her yatay ve dikey kısımlarda 1 den 9'a kadar olan sayılardan yalnız 1 tane olması gerekmektedir ve bu şekilde geliştirilmiştir.

## AKIŞ DİAGRAMI

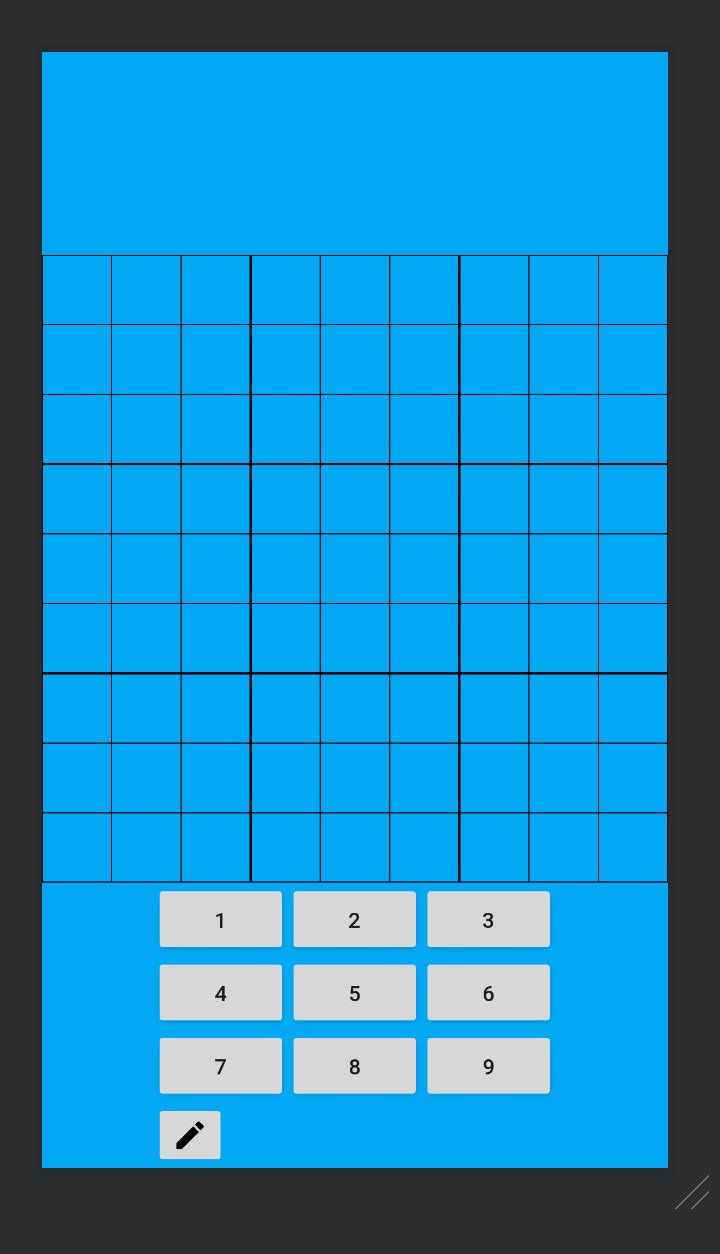


PROJE ARAYÜZ TASARIMLARI

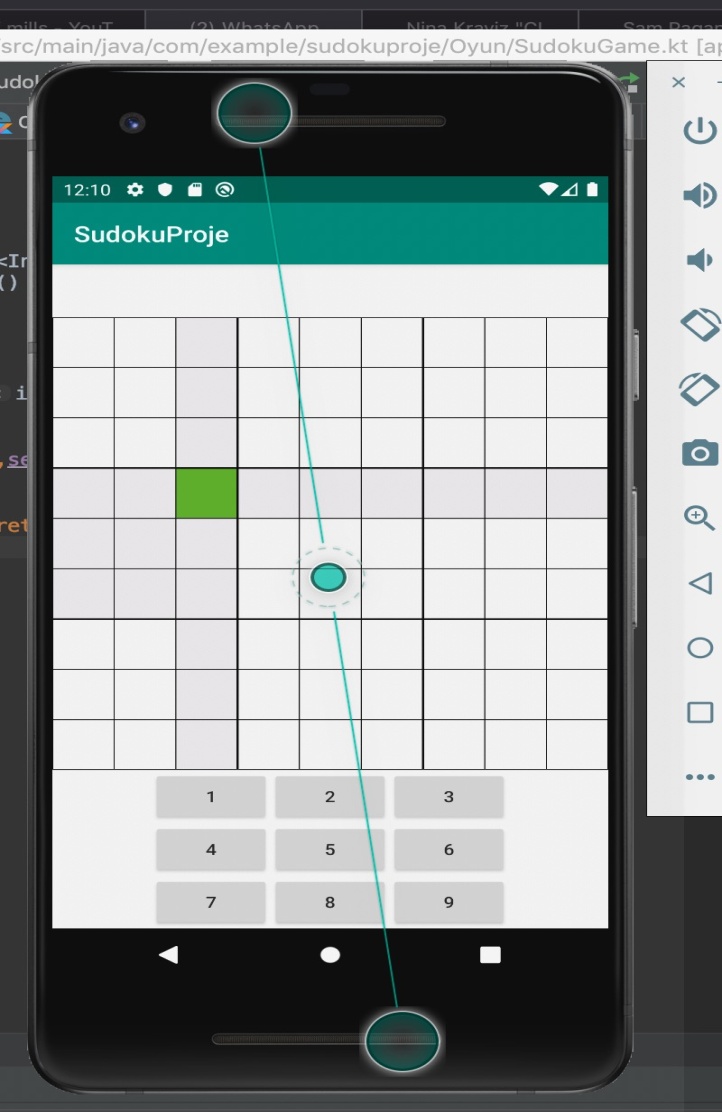
Buttonlara verilen isimler ;



Çizim ekranı yani oyunun oynanacağı kısım ;



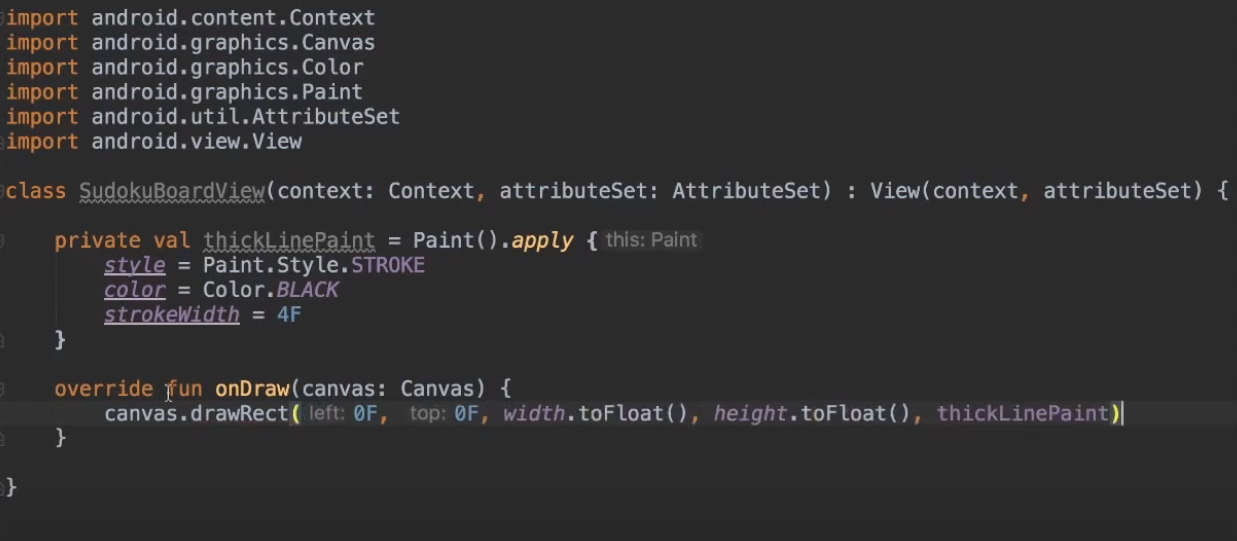
## PROJENİN PROGRAMLANMASI KOD AÇIKLAMALARI



Canvas sınıfını kullanarak çizim emirlerini,komutlarını tutup board game oluşturdum ve xml kısmından butonlarını oluşturup layout\_row,layout\_column kısımlarının değerlerini belirttim.

### 

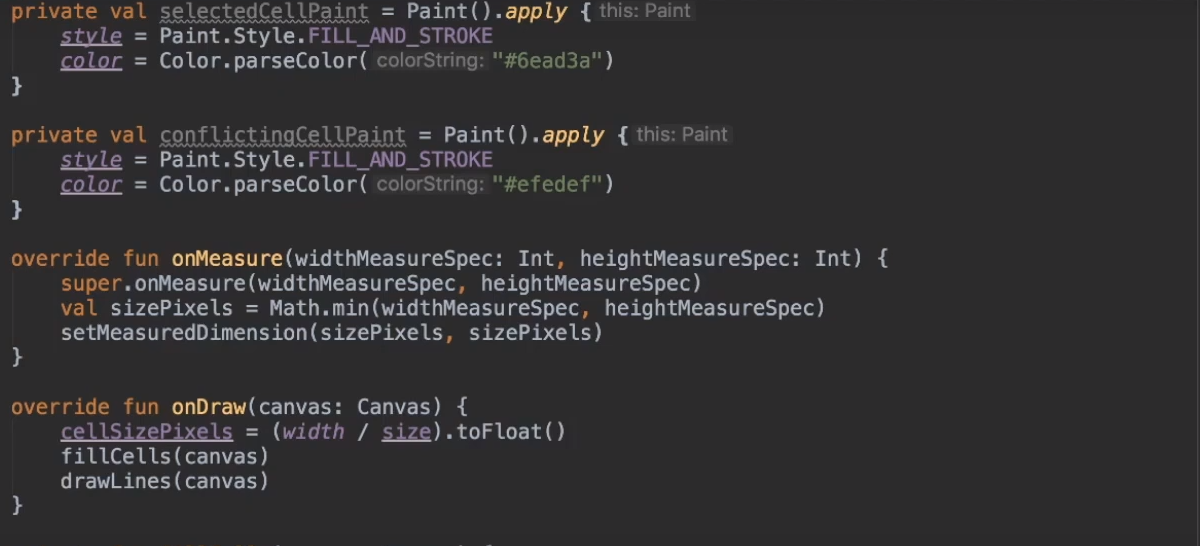
### PROJE KODLARI 1;



Burada da canvas sınıfını kullanıp çizim emirlerini belirttim onun yanında da onDraw komutunu kullandım çünkü oda çizim için görünüm oluşturmamı sağladı.

Ve point komutunu kullanıyoruz o da çizeceğimiz ortamı belirlemek için kullanılıyor içinde de style,color,srtokeWidth komutlarını kullanıyoruz

## PROJE KODLARI 2;



SelectedCellPaint methodunu tanımlayarak seçilen hücreyi belirtip tıkladığımızda olacak rengini belirtiyoruz

Conflicting methodunu tanımlayarak boş olan hücrelerin beyaz renkte olmasını sağlıyoruz.

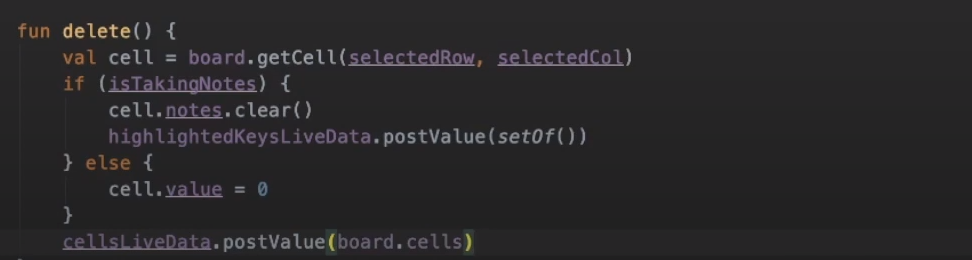
Ayrıca burada onMeasure fonksiyonunu kullandım çünkü görünüm genişliğini ve yüksekliğini görmemiz için olanak sağlar.

Ek olarak super.OnMeasure kodunu kullanmayıp bunun yerine setMeasureDimension kodunu çağırıp genişlik yükseklik öğelerini kullandım.

Bu sayfada da drawLines komutunu kullandım çünkü birden fazla çizgi çekmek istiyoruz.

FiilCells methodunu da kullanarak dosya hücrelerini çizim dosyasıyla bağlıyoruz.

## PROJE KODLARI 3;



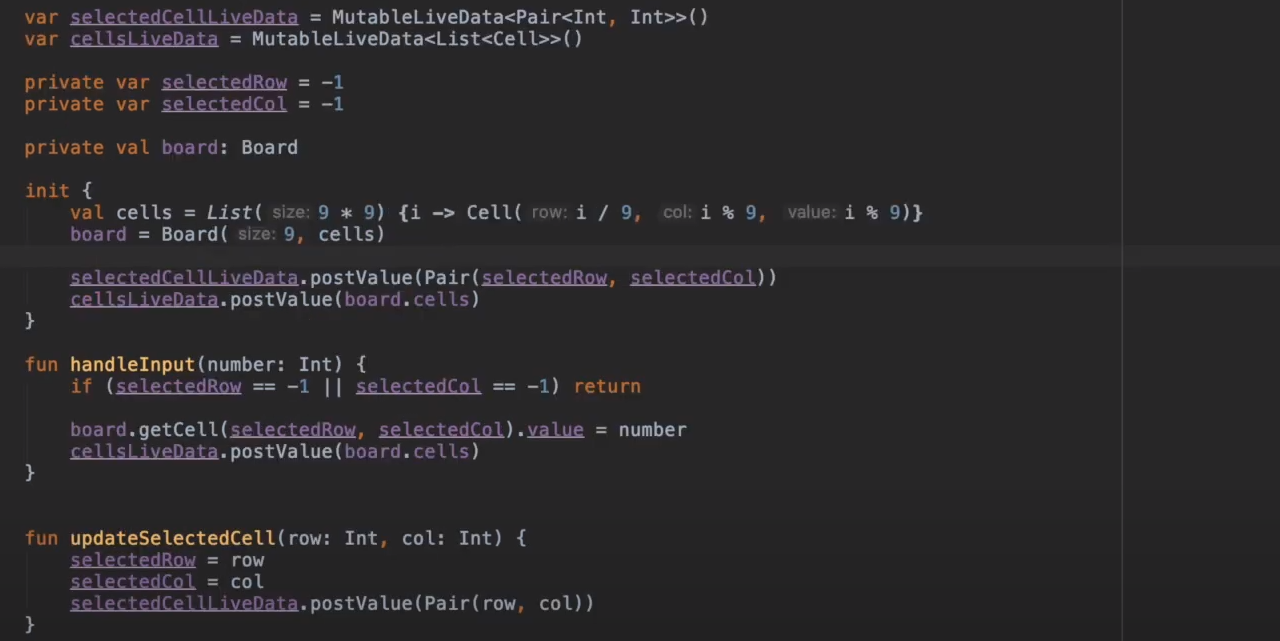
Burada silme işlemini gerçekleştiyoruz yazı tahtasından seçilen satır (Selected row) seçilen sütun(Selected col) çıkarmamızı,silmemizi sağlıyor.

İf else komutları ise bizim şart ifadelerimizdir.

Not hücresinden girilen sayıyı temizliyor,olmazsa hücrenin sıfırıncı değerini veriyor.

Ek olarak LiveData gözlemlenebilir veri tutma sınıfıdır.

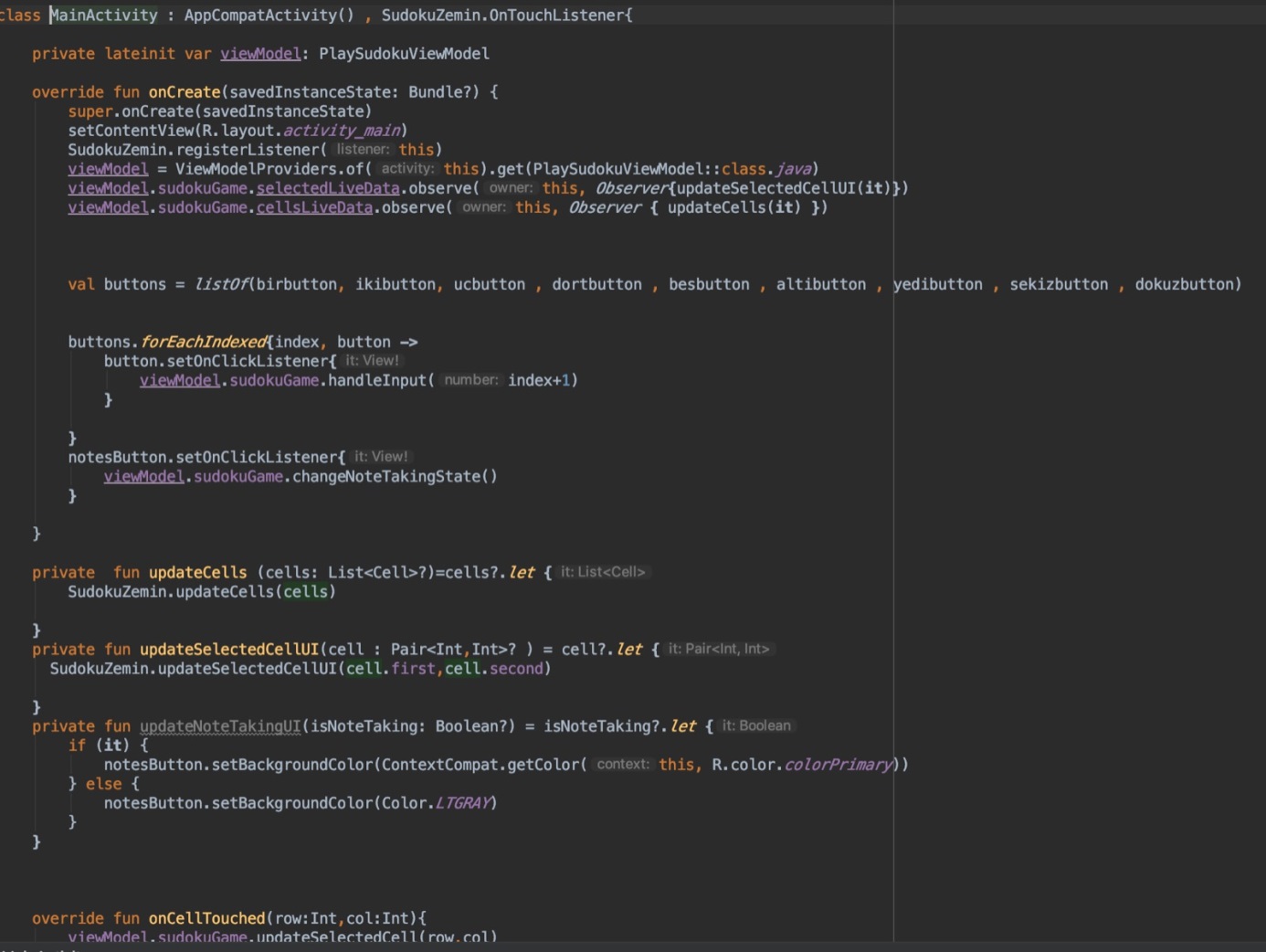
## PROJE KODLARI 4;



Burada hücrelerimin yani çizim ekranımın 9’a 9 liste olmasını oluşturdum.

Row , Col değerlerinin verilmesini oluşturdum ve Cell Dizisini oluşturup sayıları set ettim içine.

## PROJE KODLARI 5;



Buttonları bilist’in içinde topladım (ikibuton,ucbuton vs..). Yukarıdaki viewModel ile sudokuGameclassındanselectedLiveData ve cellsLive data import ettim Observermethodu ile kullanıyoruz. onCellTouched fonksiyonu ile dokunduğumuzda row ve col’daki değişiklikleri sudokuGame sınıfındaki updateSelectedCellfonkisyonu ile yapmasını sağlamış oldum.

## KAYNAKÇA

<https://www.youtube.com/watch?v=o6P05m0E9z4> Youtube üzerinden yapılmış bir projeyi izleyerek yaptım ve kendime göre bazı yerlerini değiştirdim