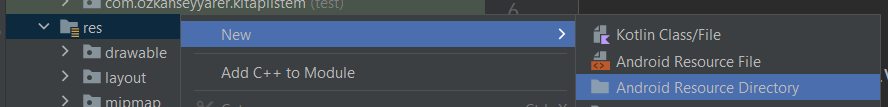
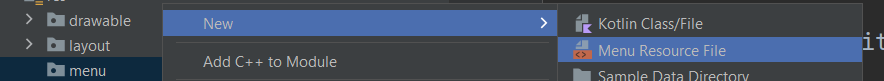
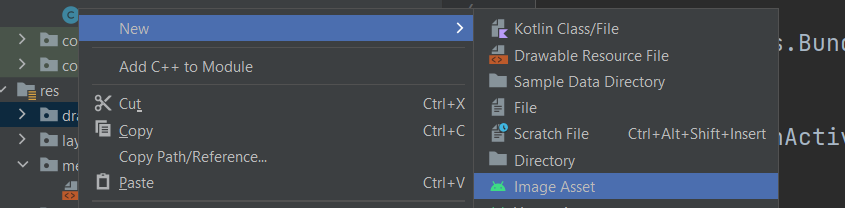
1-Menü ekliyoruz



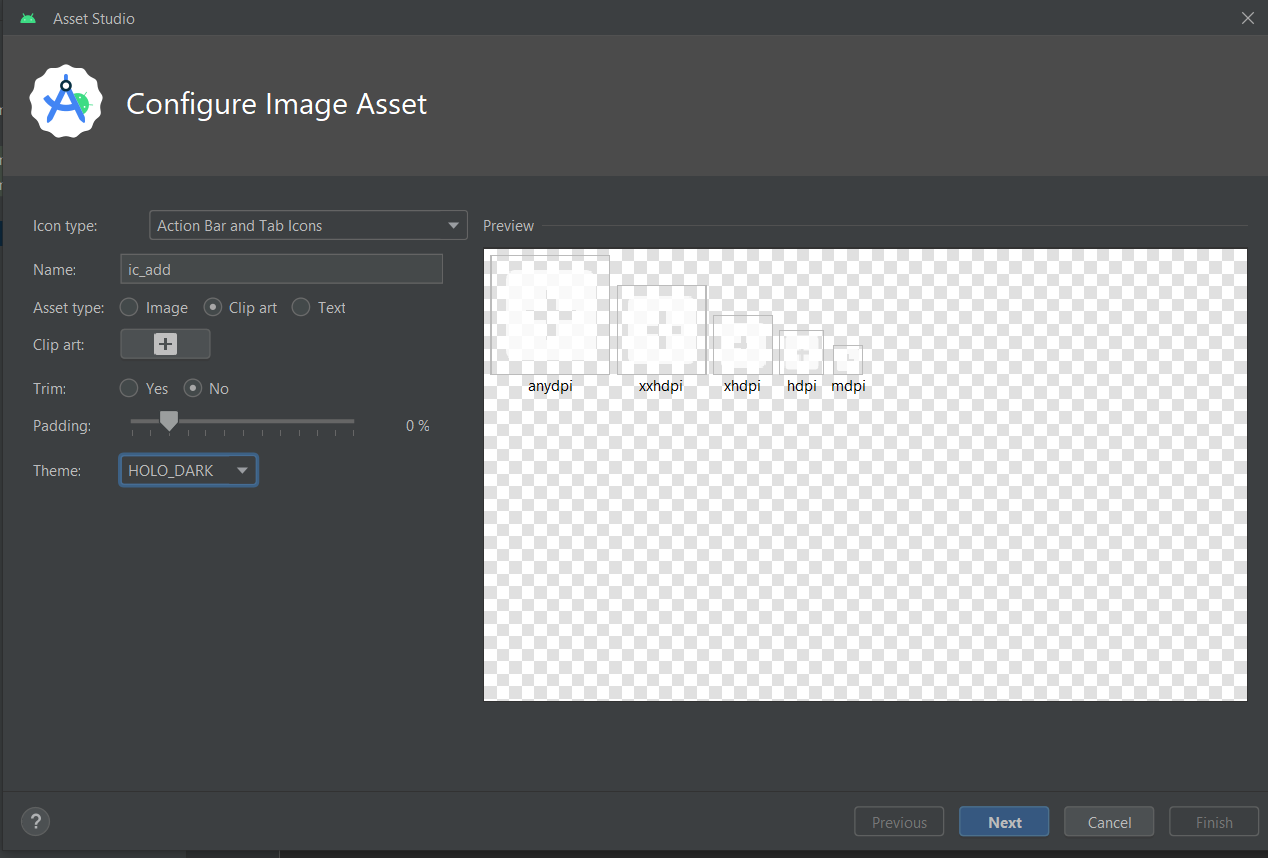
2- menüye bir tane dosya ekliyoruz



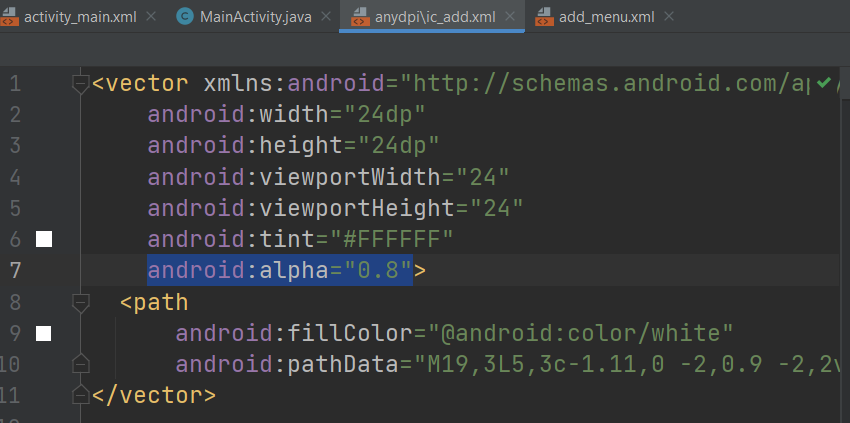
3- ekleme butonumuz için drawble klasörümüzden bir tane icon ekleyeceğiz



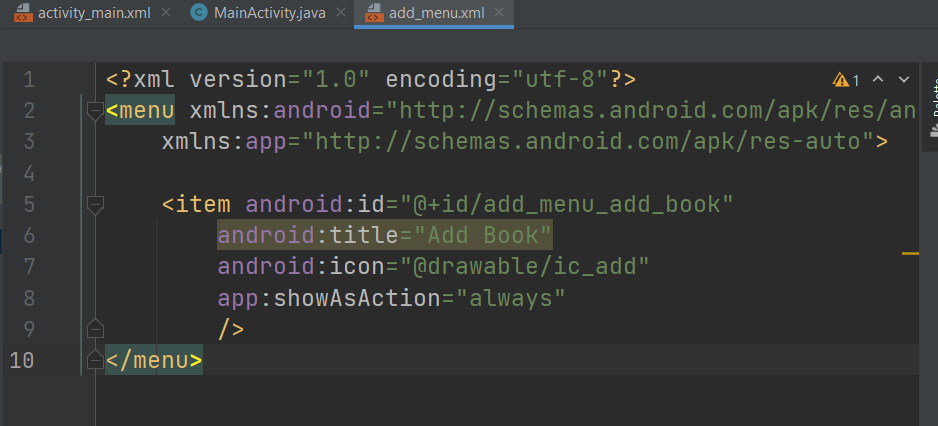
4- bu şekilde ayarlarımızı yapıyoruz



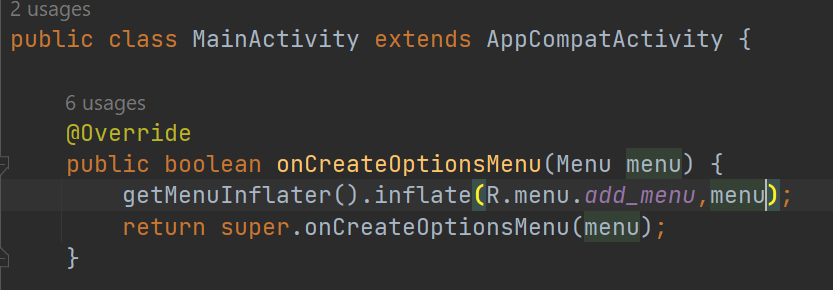
5- drawable klasörğ altındaki ic\_add klasörüne çift tıkladıktan sonra alpha değerini silmemiz gerekiyor simgemizin gözükmesi için



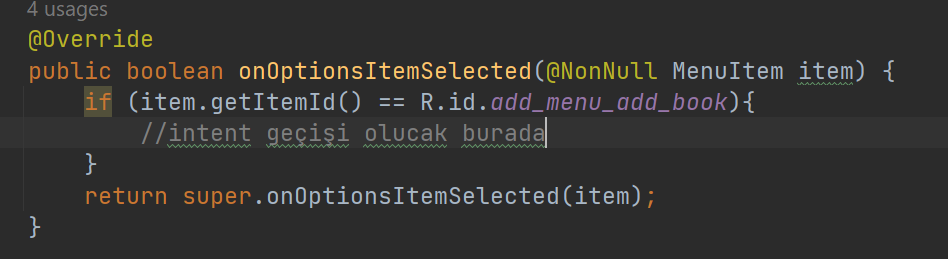
6- ikonumuzu menümüze ekledik



7- menüyü projemize bağlıyoruz



8 - menüye tıkladığımız zaman kitap ekleme sayfasına gitmemiz gerekiyor

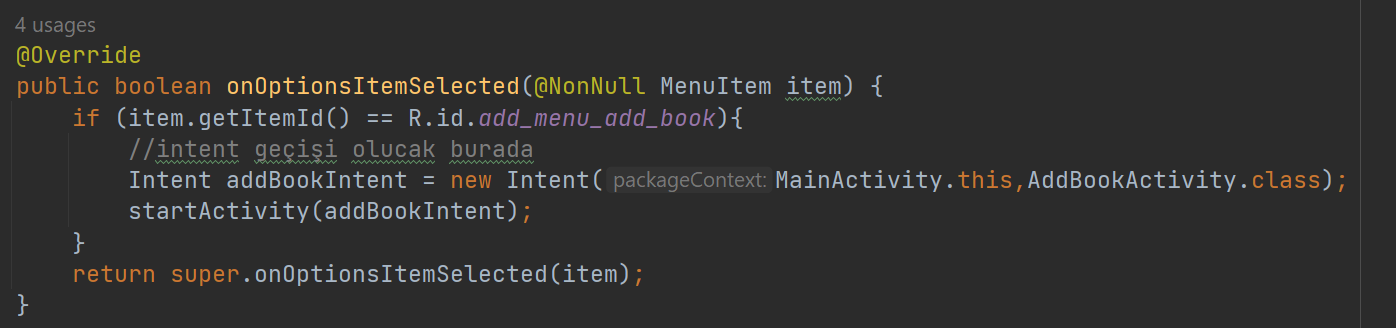


9- kitap ekleme sayfası olan “AddBookActivity” Sayfasını ekliyoruz.

10- activty\_add\_book sayfamıza ekleyeceğimiz şeylerin hepsi alt alta olacağı için “ScrolView” süzende yazacağız xml dosyamızı

11- activty\_add\_book.xml dosyasının içeriklerini yazdık

12- bu işçerikleri yazdıktan sonra ıntent kısmını doldurduk

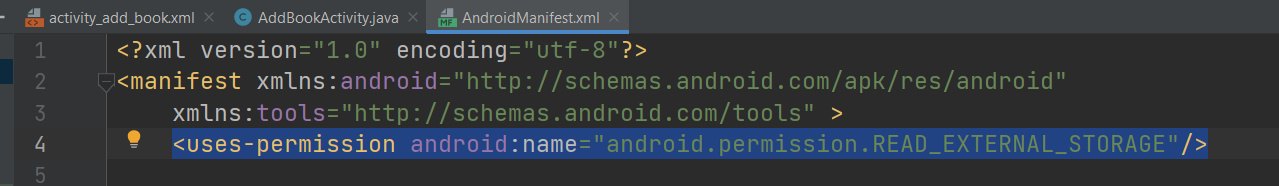


13- Kitabı kaydetmek için eklediğimiz butonumuzu add\_boook\_activity dosyamıza ekliyoruz.

14- içerikleri init ettim addBook.java içerisinde

15- resme tıklanıncada bir tür işlemler olsun diye onada onclick tanımladım

16- burada önce izin isteme işlemleri olacak

17- manifest dosyasına gidip kullanıcının depolaa alanına erişim izni istememiz gerekiyor. Bu mavili olan satır bu depolama alanına erişmemiz için gerenkli olan izni bize sağlayacaktır. 

18- bundan sonre cihazdan resim almak için kullanıcı resme tıklayacak bu işlem için önce bir izinin verilip verilmediini gösteren kodu yazıyoruz

imgKitapResim.setOnClickListener(view -> {  
  
 //kullanıcının izin verip vermediğini kontrol ediyoruz  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*READ\_EXTERNAL\_STORAGE* )!= PackageManager.*PERMISSION\_GRANTED*){  
 //kullanıcı izin vermediyse bu kısma girecek ve kullanıcıdan tekrar izin isteyecek  
 ActivityCompat.*requestPermissions*(this,new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*},izinAlmaKodu);  
 }  
});

Bu kod bloğu, bir Android uygulamasının "Dış Depo Okuma İzni" (READ\_EXTERNAL\_STORAGE) gereksinimini kontrol eden ve kullanıcıdan bu izni isteyen bir işlemi içerir. İşte bu kodun ayrıntılı açıklaması:

imgKitapResim.setOnClickListener(view -> { ... }); kısmı: Bu kod, bir ImageView bileşeni olan imgKitapResim üzerine bir tıklama olayı dinleyicisi ekler. Yani, kullanıcı bu görüntüye tıkladığında bu kod bloğu çalışacaktır.

Kodun içeriği:

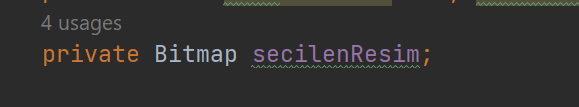
ContextCompat.checkSelfPermission(this, Manifest.permission.READ\_EXTERNAL\_STORAGE) != PackageManager.PERMISSION\_GRANTED ifadesi, uygulamanın "Dış Depo Okuma" iznini kontrol eder. ContextCompat.checkSelfPermission işlevi, belirli bir iznin durumunu kontrol eder. Eğer bu izin zaten verilmişse, PackageManager.PERMISSION\_GRANTED (İzin verildi) döner ve işlem devam eder. Ancak, izin verilmemişse, bu kod bloğuna girecektir.

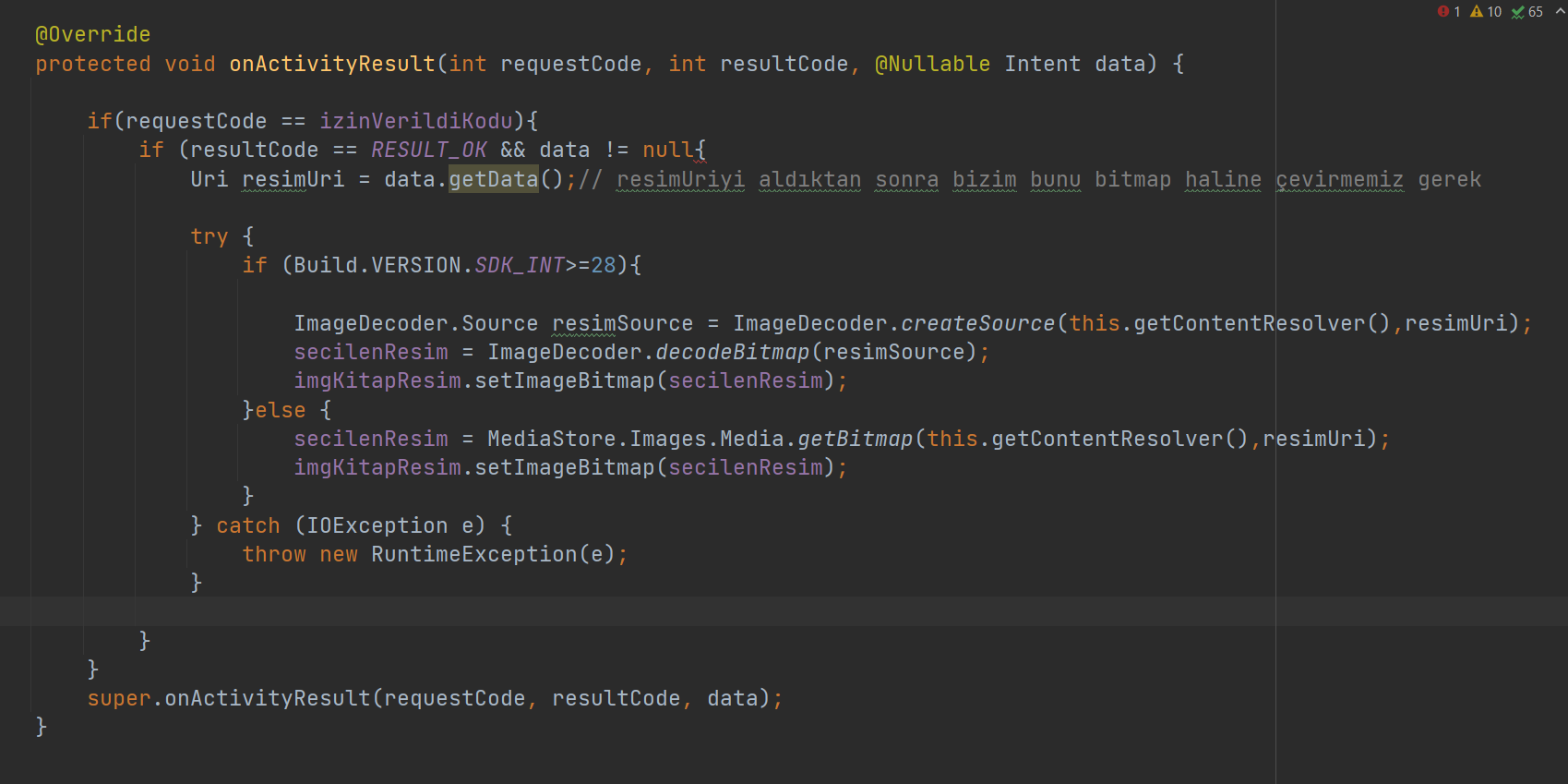
ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE}, izinAlmaKodu) ifadesi, izin almak için bir istekte bulunur. ActivityCompat.requestPermissions işlevi, kullanıcıdan izinler istemek için kullanılır. İlk parametre olarak, izin istenen etkinliği (activity) veririz (this kullanıldığı için bu etkinlikte çalışır). İkinci parametre olarak, istenen izinleri içeren bir dizi veririz (new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE}). Üçüncü parametre ise bir istek kodudur (izinAlmaKodu). Bu istek kodu, izin isteği sonuçlarını işlerken kullanılabilir.

Sonuç olarak, bu kod bloğu, kullanıcının "Dış Depo Okuma" iznini verip vermediğini kontrol eder. Eğer izin verilmemişse, kullanıcıdan bu izni istemek için bir izin isteği (permission request) başlatır. Kullanıcı izni verdiğinde veya reddettiğinde, uygun işlemi gerçekleştirmek için bu izin isteği sonuçlarını işlemek gerekecektir.

19- şimdi kullanıcı izin verdikten sonra galeriye gidip seçilen resmin değerini almamız gerekiyor

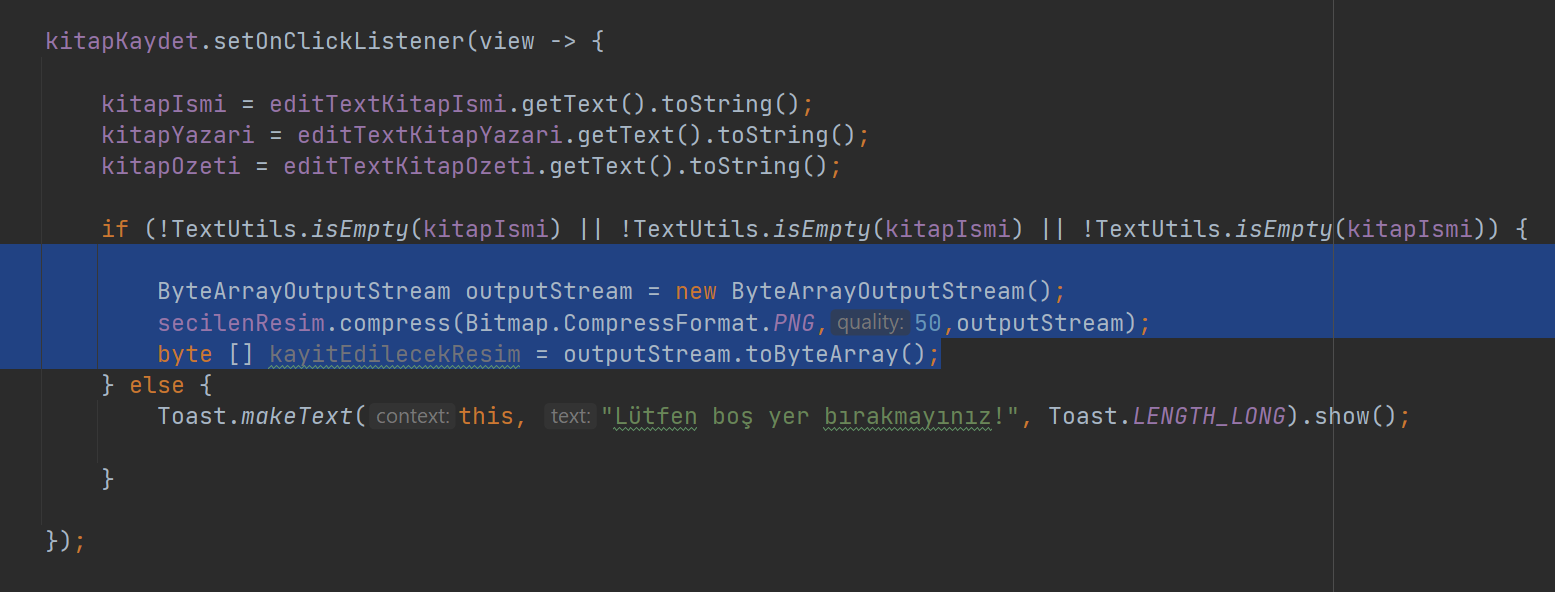
Önce uygulamamızın en başına gidit bir tane Bitmap tanımlıyoruz





20- resmi aldıktan sonra o resmi kaydetmemiz gerekiyor. Resmi kaydetmemiz için bizim o resmi byte haline dönüştürmemiz gerekiyor. Bizim resmimiz “secilenResim” nesnesi içerisinde dururuyor. Bizim o nesneyi byte haline getirmemiz gerekiyor. Bir ezme işlemi yapacağız bu ezme işlemini “compress ile yapıcağız.”

Aşağıdaki şekilde resmi ezmiş olduk. Bunu byta dönüştürmüş olduk.Bizim bu resme birde boyutsal olarak bir değişiklik yapmamız gerekiyor ki. Bu resimler orada donmaya sebep olmasın ve SqlLite’ı şişirmesin.

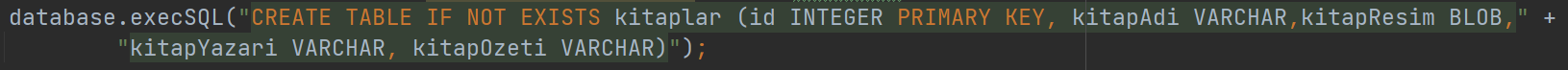


21 - veri tabanını yapalım

Aşağıdaki kod bir veri tabanı varsa açar yoksa oluşturur.



Aşağıdaki kod satırı bize eğer oluşturlmamışsa bir tane tablo oluşturacak bu tablonun sütunlarından biri id olacak bu id kısmını INTEGER PRİMARY KEY olarak yaparsak bu aydi otomatik olarak artacak ardından bir başka sütun olan kitapYazari kısmı VARCHAR olarak tanımladım. Bunların dışında bşr de resim tanımlamamız gerekiyor, gelen bu resmin sütununa kitapResim adını verdim bunu da BLOB olarak tanımladım



Şimdi veriyi eklememiz gerekiyor. Tablonun sütunlarına değerleri atamak için işlemler yapıyoruz. Değerleri kullanıdan almak istiyorsak, kaç tane kullanmak istediğimiz sütun varsa o kadar oraya soru işareti bırakmamız gerekiyor.

Yaptığımız “statemen.bindString(index,value)” kod satırları ilk olarak index alıyor olacak bu indexler kolon indexi olacak yalnız dikkat edilmesi gereken bir nokta var burada koloon indexleri 0 dan başlamıyor 1’den başlıyor. Ardından value yerinede alacağı yani kaydedilcek değeri yazıyoruz.

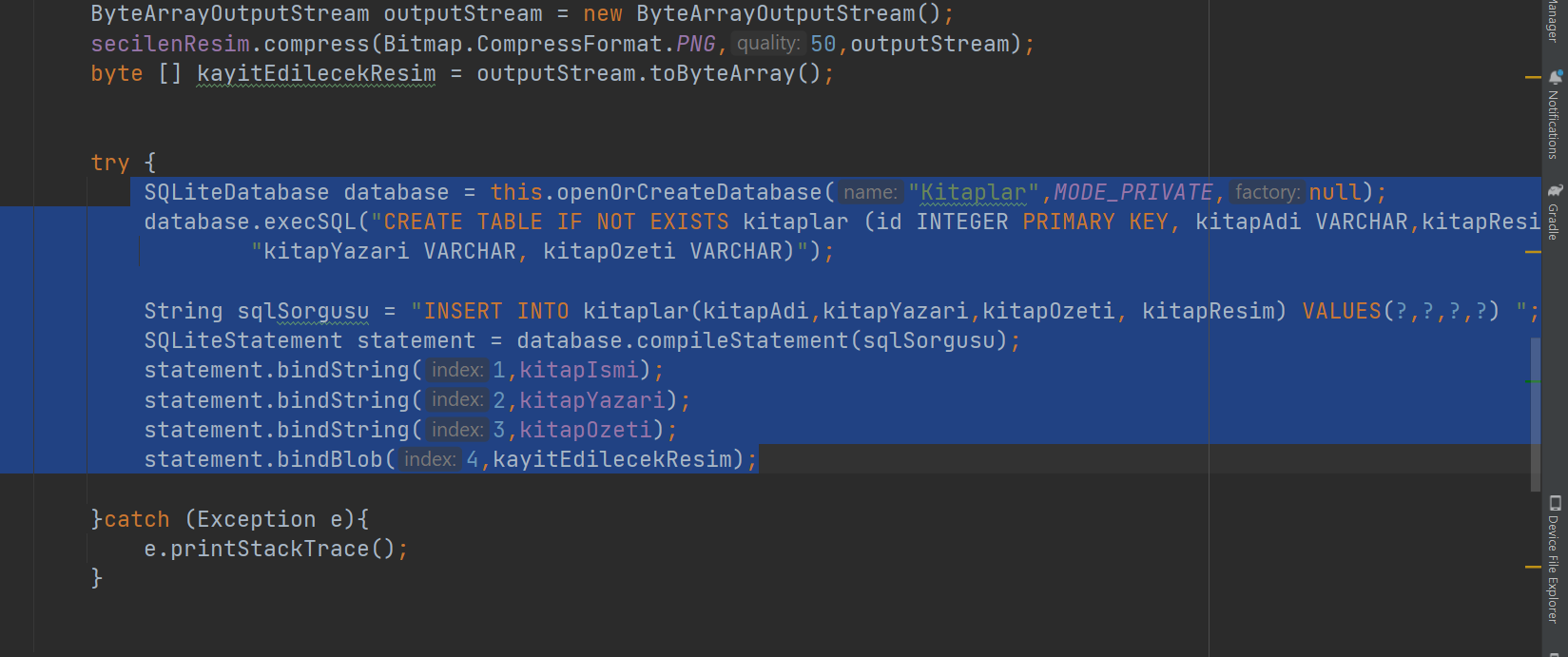
Bunarın dışında resmim kaydetme olayı biraz daha farklı gerçekleşiyor. Byte dizisi olarak kaydettiğimiz resmi daha sonra sqlLite tablosuna eklemek için aşağıdaki şekilde kullanıyoruz (bindBlob)



Bunların hepsini yaptıktan sonra verilerin ekleme işleminin gerçelkşetirilebilemesi için “statement.execute()” yazıyoruz.

Kaydın eklendiğini kullanıcıya belli etmek için Toast mesajı verebiliriz

Aşağıda bu işlemlerinin hepsinin yapılmış hali bulunmakta.

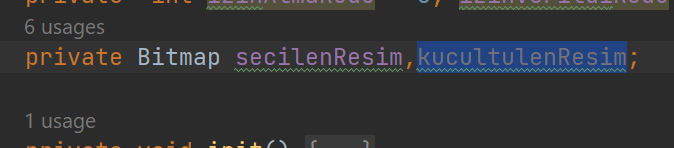


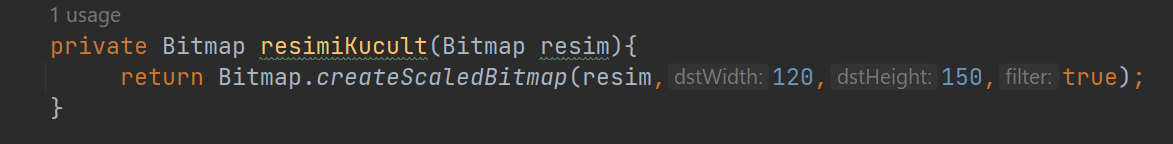
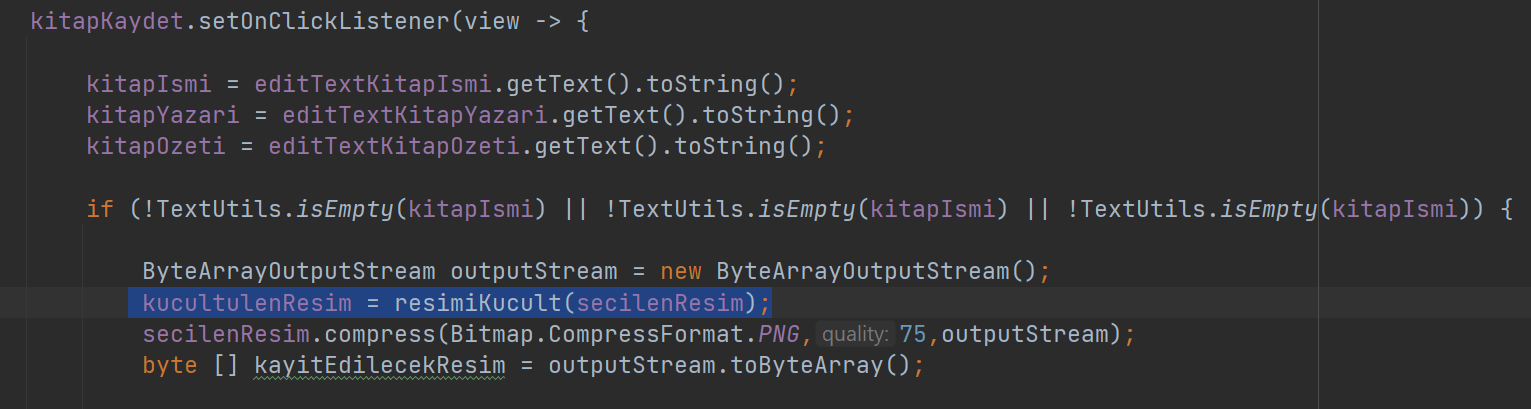
22- kaydedilen resim eğer çok büyükse sqliteımızı şişirip telefonu dondurabilir bu yüzden bunu önlemek için bizim resmi yeniden boyutlandırmamız gerekiyor. recycleView deki görünüleri aktivitedeki görünümün yarısı ile oranlayalım. Mesela biz yüksekliği 300dp genişliği ise 240 dp olarak ayarlamıştık öyleyse biz bunların yarısına oranlayalım. Bunu bir metot halinde yapıyoruz.

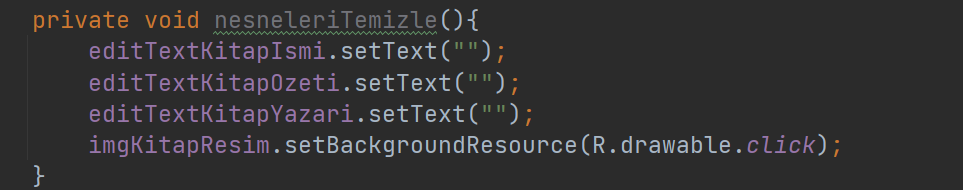
Tanımladığımız bu metot bir tane bitmap halinde bir resim alıcak ve buraya gelen resim parametresi üzerinde bir küçültme uygulayacağız. Bu metot bir tane metot döndürecek döndürülen bu etot “Bitmap.crateScaledBitmap”

Bu metot dört tane değer alıyor bu değerlerden biri resmin kaynağı yani metotla beraber gelen resim olacak. Ardından genişlik değeri ve yükseklik değerini alıyor.

Buna değeri kaydet butonuna tıklama işleminin gerçekleştiği metot içinde byt dizisi oluşturmadan hemen önce gönderip küçülttüceğiz.





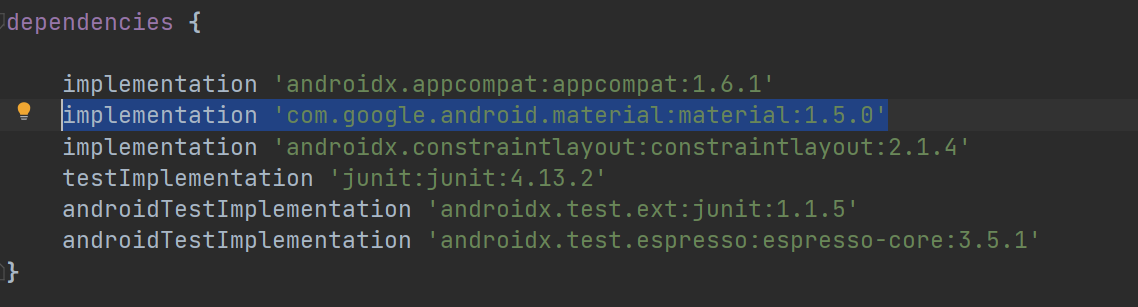
23- kullanııcı en son “kaydet” butonuna bastıktan sonra doldurduğu içerikleri sıfırlamamız lazım ki kullanıcı tekrar silmekle uğraşmasın onuda bir metot aracılığıyla yapıyoruz. Bu metotdumuzun adı “nesneleriTemizle”. 

Ardından yazdığımız bu metodu kaydet butonunun metodu içinde çağırmamız gerekiyor.

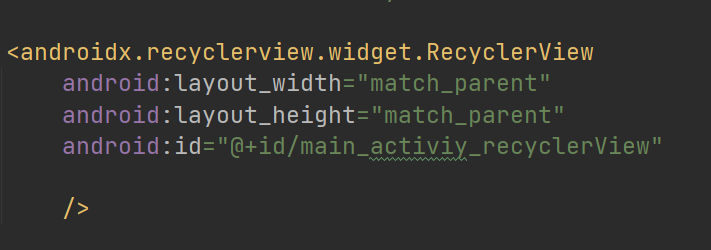
24- Main activity.xml dosyasına recyclerView oluşturmak için düzenleme yapıyoruz

Bu dosyanın düzenini RelativeLayoutdüzende yapıyoruz. recycelerView’i kullanabilmek için onun bir kütüphanesi vardı o kütüphaneyi eklememiz gerekiyor. Bunun için “project > GradleScripts > buil.gradle (Module : app)”

Buraya geldikten sonra “dependencies” altına o kütüphaneyi ekliyoruz.(eski sürümlerinde bu kütüphane yüklü bir şekilde gelmiyordu o yüzden eğitmen bunu nasıl yüklememiz gerektiğini anlatıyor fakat yeni sürümlerde zaten yüklü)



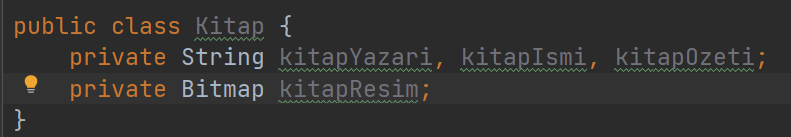
25-Main\_activty.xml dosyamızı düzenleyelim içine “recyclerView” eklememiz gerekiyor



26- bir tane class oluşturacağız bu klass veriyi hem almak hemde veriyi göndermek için kullanacağımız bir class olacak.

Ayrıca biz bu classta verileri dolduracağız. Bu classın adı “kitap” olacak

Bizim kitap yazarı, kitap ismi, kitap özeti, kitap resmi olmak üzere toplamda dört tane verimiz var. Bu yüzden burada buna göre tanımlamalar yapmamız gerekiyor.

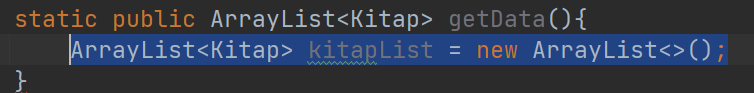


27- bu class içerisinde boş bir constructor oluşturacağız. Ardından bir tanede dolu constructor oluşturacağız.

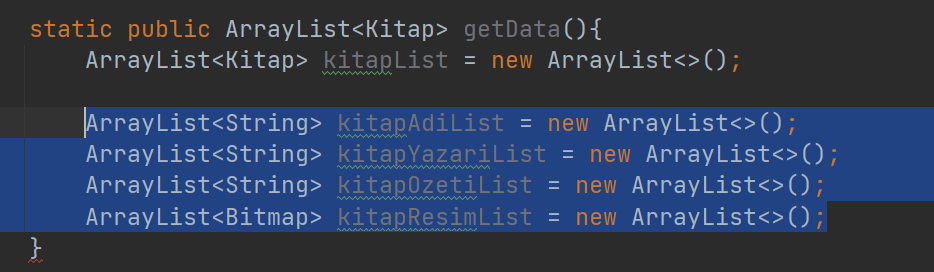
Bunun içierisine getter ve setterlarıda eklememiz gerekiyor. Ardından içerisine bir tane static public metot ekleyeceğiz.

Bu metot geriye eklediğimiz verilere ait bir tane liste döndürmesi gerekiyor. Bu yüzden bu metodun türü ArrayList<> olacak.

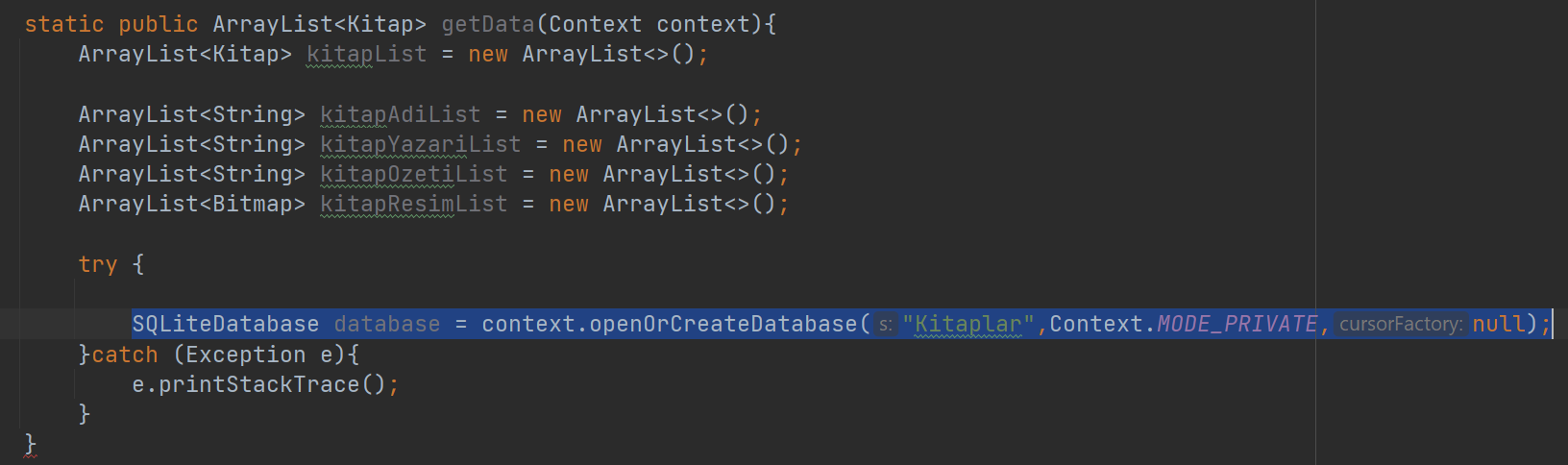
Bu ArrayList ise bizim clasımızın türünde olacak yani “Arraylist<Kitap>” bu metodun adı ise “getData()” olacak.

Bu metotta biz geriye bir tane “list” döndüreceğiz o yüzden başlangıçta bir tane liste oluşturmamız gerekiyor. 

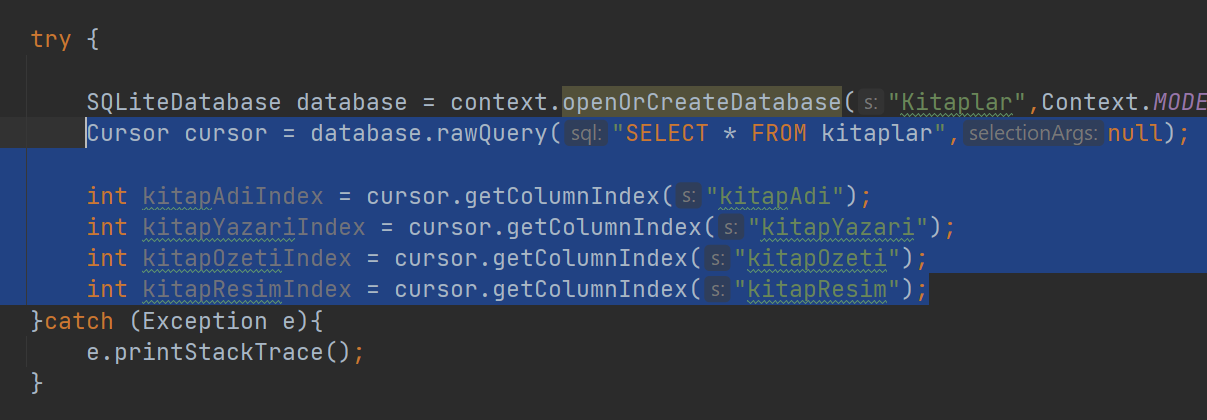
28- verileri veri tabanından almamız gerekiyor ki verileri bu “kitapList” içerisine doldurabilelim

Verileri alacağız, bu yüzden bu veriler içinde birer liste oluşturmamız gerekiyor 

29- bu listelerin altına bir tane try-catch yapısı kurmamız gerekiyor ardından buradaki try içerisinde veri tabanını açmamız lazım.



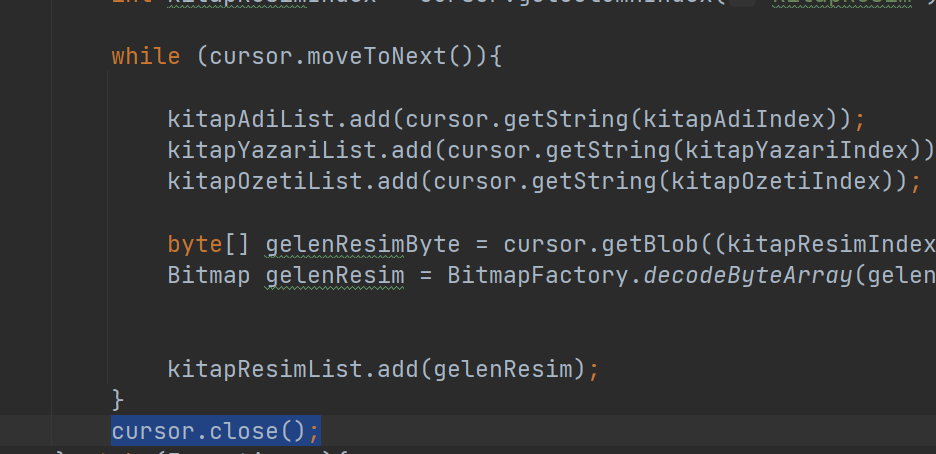
30-Bu satırı yazdıktan sonra hemen aşağısına veri tabanından verileri almak için gereken komutları yazmamız gerekiyor.



31- oluşturduğumuz listeleri while döngüsü sayesinde nesnelerden gelen string değerler ile doluracağız.

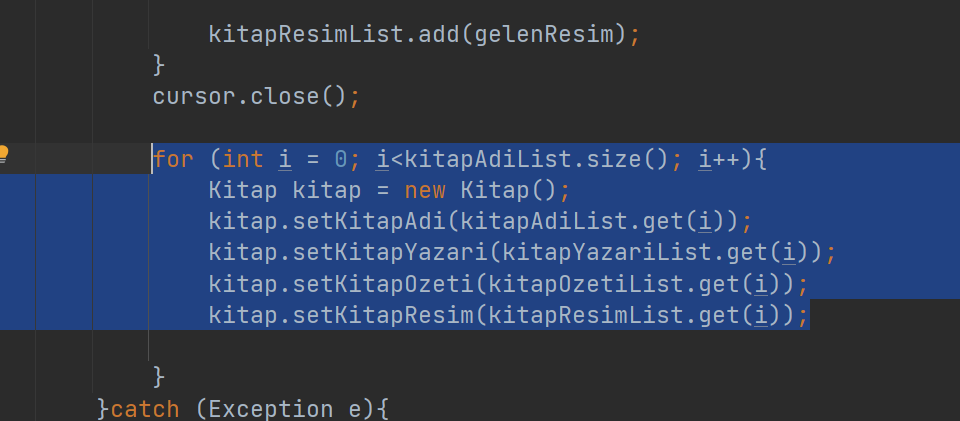


Verilerimizi aldıktan sonra nesneyi kapatmayı unutmuyoruz

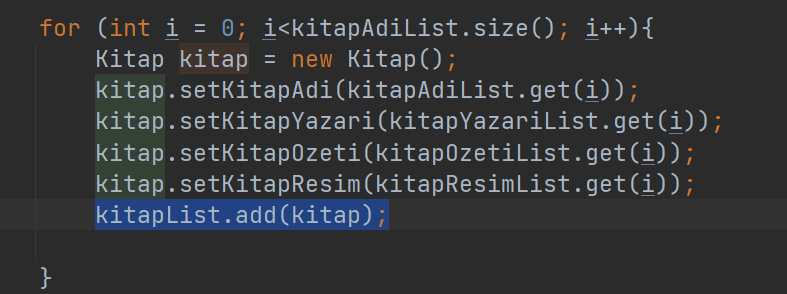


32- kitapList’in içerisini doldurmamız gerekiyor. Bunun için bir tane for döngüsü oluşturmamız gerekiyor. Bu for döngüsünün uzunluğu için daha önce oluşturduğumuz herhangi bir ArrayList’in uzunluğunu baz alabiliriz çünkü hepsinin uzunluğu aynı olacaktır.

Ardından bu for föngüsü içerisinde yeni bir tane kitap sınıfı oluşturmamız gerekiyor.



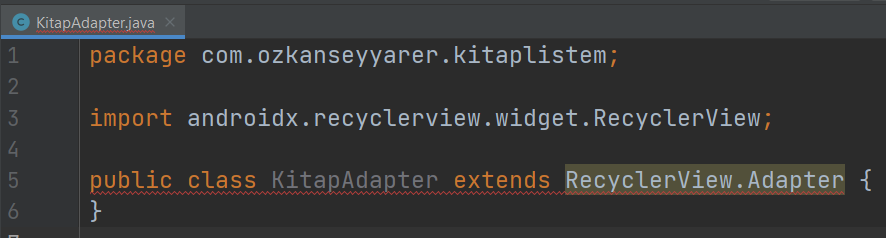
Bunu yaptıktan sonra for döngüsü içerisinde oluşturmuş olduğumuz classımız dolmuş oluyor, bunu kitapList içerisine atmamız gerek şimdi



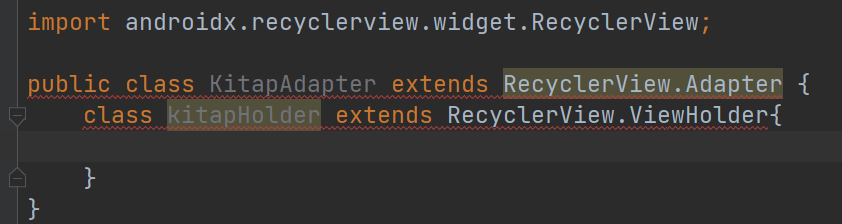
Bunun en sonunda kitapList’i geri döndürüceğiz.

33- bunları bir görünüme atmak için bir tane xml dosyası oluşturduk. “kitap\_item.xml”

34- bir tane adapter oluşturmamız gerek bunun adı KitapAdapter.java olacak. Oluşturdumuz bu adapter RecyclerView.Adapter sınıfından miras alacak.



35- bunun içine bir tane inner class yazmamız gerek



36- bu inner classın conctructorunu oluşturmamız gerek

