**ANDROİD PROGRAMLAMA 2. DERS**

#Layout: arayüz sayfası.

Layout yapıları, bir nesneyi istediğin yere sürükleyip bırakamazsın.

#Generate layout file’ı seçmek xml ekle demek.

#Main activity java kodları yazacağımız kısım

#firefox developer edition kullan!!!!

**#Metodlar ve değişkenler küçük harfle, class isimleri büyük harfle başlar javada.**

**+Public:**

-erişim seviyesi (diğerleri private, protected)

-her yerden erişim

**+private:** kısıtlı erişim

**+Protected:** üst sınıftan gelen değeri alt sınıfta erişim sağlanması için

**+Package:**

-proje paketidir

-alt proje klasörlerini temsil edebilir.

**+Override:** üst sınftan çağrılan bir metodun davranışının değiştirilip yeniden nasıl davranması gerektiğini tanımlamak ve üst sınıftan çağrılan bir metot olduğunu belirtmek için

**+İmport:** proje paketlerini belirtip ilgili sınıfı bulunduğumuz sınıfta çağırmak

**+Class:** sınıf

**+Extends:**

-miras almak. bundan sonra sadece bir tane sınıf ismi belirtebilirsin

-Bir sınıfın atası yalnızca bir üst sınıftan olabilir.

**+Implements:**

-interfaceleri kullanmak için bu keyword kullanılır.

-İmplements view.OnClickListener

🡪OnClickListener interfaceleri temsil eder, birden fazla interface virgün eklenerek eklenebilir.

**+Void:** geri değer döndürmeyecek bir metot, geri dönüş tipi olarak void yazılır. (return kullanılamaz void ile, void yoksa return kullanılır)

**+Super:** üst sınıftan bir metod çağırmak için

**ANDROİD LAYOUT TÜRLERİ**

-linearlayout

-relativelayout

-constraintlayout

-coordinatorlayout

-Framelayout

-tablelayout

3kalvyede veri girişi-edittext

#Layoutlar belirli düzende nesnelerin sayfa içerisine yerleştirilebilmesi için gerekli bir kalıptır.

Layout sayesinde nesneleri hangi düzende ya da hangi ilişkide hizalandırılabileceğini seçebiliriz.

**+ linearlayout:**

-orientation özelliği sayesinde nesneler vertical ve horizantal olarak hizalandırılabilir.

-Alt alta ya da yan yana eklenebilir sadece.

-weight özelliği sayesinde nesnenin birim olarak kapladığı alanı değiştirebiliriz. (weightsum)

**+constraint layout:**

-android projesinde varsayılan olarak karşımıza gelen layout türüdür.

-Nesnelerin yerleri toplu şekilde taşınabilir.

-nesneler bağlanmazsa sürekli 0 noktasındadır. Yani sol üst köşede kalacaktır.

-nesneler tek seferde infer constraint ile bağlanabilir, clear constraint ile çözülebilir.

-sürükle bırak ile nesne ilgili yerde olması sağlanabilir.

**+relativelayout:**

-nesneler sürükle bırak ile istenilen yerde bulunabilir.

-nesneler birbirleriyle ilişkili şekilde taşınabilir.

**UYGULAMA KAYNAKLARI (RESOURCES)**

+R sınıfı

-resources kelimesinin ilk harfinden gelir.

-Uygulama kaynaklarını temsil eder.

-her üretilen resources dosyaları, xml layout içerisindeki nesne id’leri R sınıfı içerisinde referans adresi olarak saklanır.

+R sınıfının paketleri:

-anim

-attr

-bool

-color

-dimen

-drawable

-id

-integer

-layout

-mipmap

-string

-style

-styable

##Ctrl boşluk—seçenek ekranını getirir

RESOURCES KLASÖRLERİ (Android > app > res)

+drawable:

-içerisine XML(çizim),PNG,JPG dosyaları atılabilir

-resim dosyalarının saklanacağı klasördür.

+layout:

-activity sayfalarının görünen arayüzleri bulunur. XML dosyalarıdır.

+mipmap: Farklı boyutlarda uygulama logolarının saklanabileceği klasörüdür.

+values: String ifadeleri XML dosyaları içeirisnde değişken olarak saklayabileceğimiz klasördür. Strings.xml, color.xml, styles.xml, dimens.xml

+anim: xml olarak animasyonları tanımlayabileceğimiz dosyaları temsil eder

+animator

+color

+font

+interpolator

+menü:menü tanımlamaları yapılır

+navigation

+raw: ses dosyaları, video dosyalarının depolanabileceği klasör

+transition: xml olarak sayfa geçişleri

+xml:kelime işaretleme dili

Layout\_width: nesnein genişliği

Layout\_heigth: nesnenin yüksekliği

Match\_parent:tüm boşlukları kapla

Wrap\_content: nesne boyutu kadar davran

Margib-dış boşluk

Padding-iç boşluk