

**T.C.
MİLLÎ EĞİTİM BAKANLIĞI**

BİLİŞİM TEKNOLOJİLERİ ALANI

MOBİL UYGULAMAYA GİRİŞ

Ankara, 2017

- Bu bireysel öğrenme materyali, mesleki ve teknik eğitim okul/kurumlarında uygulanan çerçeve öğretim programlarında yer alan kazanımların gerçekleştirilmesine yönelik öğrencilere rehberlik etmek amacıyla hazırlanmıştır.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	2
1. Mobil Programlama.....	2
1.1. Mobil İşletim Sistemleri.....	2
1.2. Mobil Uygulama Geliştirme Platformları	3
1.2.1. Yerel (Native).....	3
1.2.2. Melez (Hybrid).....	4
1.2.3. Uyumlu (Responsive).....	4
1.2.4. Cross (Çapraz).....	4
1.3. Java Geliştirme Paketi Kurulumu	5
1.3.1. Java Path Ayarları.....	8
1.4. Mobil Uygulama Geliştirme Ortamının Kurulumu.....	10
DEĞERLER ETKİNLİĞİ.....	14
UYGULAMA FAALİYETİ	15
ÖĞRENME FAALİYETİ-2	18
2. Uygulama Geliştirmeye giriş.....	18
2.1 Yeni Proje Oluşturma.....	18
2.2 Emülatör Oluşturma.....	23
2.3 Proje Yapısı.....	28
2.3.1. Manifest.....	28
2.3.2. Java	28
2.3.3. Res	29
2.3.4. GradleScripts	30
2.4 Android Kütüphane Yapısı	30
2.5. Android Projesine Kütüphane Ekleme.....	30
DEĞERLER ETKİNLİĞİ.....	34
UYGULAMA FAALİYETİ	35
ÖLÇME VE DEĞERLENDİRME	36
MODÜL DEĞERLENDİRME	38
CEVAP ANAHTARLARI.....	40
KAYNAKÇA	41

AÇIKLAMALAR

ALAN	Bilişim Teknolojileri
DAL	Veri Tabanı
MODÜLÜN ADI	Mobil Uygulamaya Giriş
MODÜLÜN SÜRESİ	40/8
MODÜLÜN AMACI	Bireye/öğrenciye; iş sağlığı ve güvenliği tedbirleri doğrultusunda uygulama geliştirme programı ile çalışma ile ilgili bilgi ve becerileri kazandırmaktır.
MODÜLÜN KAZANIMLARI	<ol style="list-style-type: none">1. İşletim sistemine uygun uygulama geliştirme ortamını kurabileceksiniz.2. Uygulama geliştirme ortamını kullanarak proje oluşturabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam: Bilişim Teknolojileri laboratuvarı, işletme ortamı Donanım: Bilgisayar
ÖLÇME VE DEĞERLENDİRME	Bireysel öğrenme materyali içinde yer alan ve her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendirebileceksiniz. Öğretmeniniz, bireysel öğrenme materyalinin sonunda, ölçme araçları (uygulamalı faaliyetler, iş ve performans testleri, çoktan seçmeli / doğru-yanlış ve boşluk doldurmalı sorular, vb.) kullanarak kazandığınız bilgi ve becerileri ölçüp değerlendirecektir.

GİRİŞ

Sevgili Öğrencimiz,

Cep telefonları hayata girdiğinden beri sürekli gelişmiş ve bu gelişmenin sonucunda ‘cep’ sıfatının yanına bir de ‘akıllı’ unvanını almıştır. Telefon hayatımıza iletişimden fazlasını sağladığı günden itibaren yeni yazılım ihtiyaçları oluşmuş ve bu yazılımlara platform olacak mobil işletim sistemleri doğmuştur. Mobil teknolojilerin hayatımıza hızlı ve etkili bir şekilde entegre olması, bu sektörde hızla gelişme sağlamaktadır.

Gelişen, değişen, dinamik, mobil dünyada cep telefonlarında uygulama yazma konusunda nitelikli yazılımcı ihtiyacı da büyüyen sektöre paralel olarak artmaktadır.

Bu materyalle mobil uygulama geliştirme platformuna giriş yaparak programlama ortamını kullanmayı öğrenebileceksiniz

ÖĞRENME FAALİYETİ-1

ÖĞRENME KAZANIMI

İşletim sistemine uygun uygulama geliştirme ortamını kurabileceksiniz.

ARAŞTIRMA

- Günümüzde yaygın olarak kullanılan mobil işletim sistemlerini araştırınız.
- Bu mobil işletim sistemlerinin birbirlerine göre artı ve eksi yanlarının neler olduğunu sınıfta tartışınız.
- Mobil işletim sistemlerinde uygulama geliştirmek için kullanılan programlama araçlarının neler olduğunu araştırınız.

1. MOBİL PROGRAMLAMA

1.1. Mobil İşletim Sistemleri

Akıllı telefonlar (smart phones), tablet bilgisayar, PDA (Personal Digital Assistant)'lar gibi taşınabilir cihazlar için üretilen işletim sistemleri, **mobil işletim sistemleri** olarak adlandırılır.

Aralarında farklılıklar olmasına karşın her işletim sistemi farklı tip kullanıcılar tarafından kullanılmaktadır. Dünya üzerinde en çok kullanılan işletim sistemleri şunlardır:

- Android
- Apple iOS
- Blackberry OS
- Windows Phone OS (WP)
- BADA
- Palm OS (Garnet OS)
- Open WebOS
- Symbian
- Maemo
- MeeGo
- VerdAnict

Yukarıda belirtilen işletim sistemlerinden Google firmasının geliştirdiği Android işletim sistemi ve Apple firmasının geliştirdiği Apple iOS işletim sistemi dünya genelindeki cihazların %99'undan fazlasında kullanılmaktadır. Pazar paylarının yüksek olmasından dolayı anlatımlarda bu işletim sistemleri üzerinde durulacaktır.

1.2. Mobil Uygulama Geliştirme Platformları

Mobil uygulama geliştirmeye nereden başlamak gerektiği yazılımcı olan olmayan herkeste zaman zaman sıkıntı yaratabilir.

Geliştirilecek mobil uygulamanın etkili bir şekilde kullanılması ve gereksinimlere cevap vermesi için aşağıda verilen ölçütlere uygun olmalıdır. **Bu ölçütler:**

- Performans
- Anlaşılabilirlik
- Görünüm ve etkileşim (kullanıcı dostu arayüzler)
- Uygulama güvenliği
- Stabilité
- Geliştirme, test ve bakım süreçleri
- Yeni geliştirmelere destek
- Kullanılacak platform ve eklenti desteğidir.

Günümüzde mobil uygulamalar genel olarak 3 farklı kategoride geliştirilmektedir.

- Mobil Cihazlar için web uygulamaları
- Oyun uygulamaları
- Cihazdan çalıştırılabilir çevrim içi ya da çevrim dışı uygulamalar.

Yukarıda belirtilen kategorilerde mobil uygulama geliştirilmek istendiğinde projenin yapısına göre aşağıdaki mobil uygulama platformları kullanılabilir.

- Yerel (Native)
- Melez (Hybrid)
- Uyumlu (Responsive)
- Çapraz (Cross)

1.2.1. Yerel (Native)

Native dille yazılmış bir uygulama, belli bir platforma özel, genellikle platform sağlayıcısının belirttiği ya da ürettiği platform SDK'sı, araçlar ve programlama dili yardımıyla geliştirilmiştir. Örneğin, Objective-C programlama dilini kullanarak iOS mobil işletim sistemine özel ya da Java programlama dili ile Android işletim sistemine özel geliştirilen bir uygulama Native uygulamadır.

Mobil platformlara göre bilinmesi gereken programlama dilleri aşağıdaki tabloda listelenmiştir.

Mobil Platformlar	Programlama Dilleri
Apple IOS	C, Objective C
Google Android	Java
Windows Mobile	.NET, C#
Windows 7 Phone	.NET, C#
Symbian	C, C++, Python, HTML/CSS/JS

1.2.2. Melez (Hybrid)

Melez (Hybrid) mobil uygulama geliştirme, tek bir kodlama yapısı kullanılarak geliştirilen uygulamanın birden fazla mobil platformda (Apple IOS, Google Android vb.) kullanılmasıdır. Hybrid mobil uygulama geliştirmek için genellikle HTML5, CSS3 ve JavaScript dilleri ile geliştirilen uygulama çözümleriyle karşılaşılır. Hybrid olarak geliştirilen mobil uygulamalar genellikle mobil cihaz içindeki gömülü web tarayıcısı (browser) üzerinden çalışır. Hybrid mobil uygulama geliştirmenin maliyetinin Native uygulama geliştirmeye göre daha düşük olması avantajlı olarak görülmekle beraber birçok dezavantajı da vardır. Mobil cihazların; kaynaklarını etkin kullanamaması, gömülü web tarayıcının yeni sürümlerinde tüm mobil cihazlar için tek tek yazılmış kod parçalarının testinin yapılmasının gerekliliği ve performans gerektiren karmaşık projelerde kullanımında ortaya çıkan sorunlar hybrid mobil uygulama geliştirmenin önündeki engeller olarak görülmektedir.

1.2.3. Uyumlu (Responsive)

Geliştirilen web sayfalarının, farklı ekran boyutlarına sahip mobil cihazlar üzerinde (cep telefonu, tablet vb.) bir web tarayıcısı aracılığı ile düzgün bir şekilde görüntülenebilmesi için oluşturulmuş tasarım tekniğidir. Responsive tasarım HTML5, JavaScript, CSS gibi standart web teknolojileri kullanılarak geliştirilmektedir.

1.2.4. Cross (Çapraz)

Cross Platform, birden fazla platformda çalışacak uygulamaların geliştirilmesidir. Android için yazılan kodlar iOS ya da Windows Phone platformlarında çalışmaz. Her platform için ayrı ayrı kod (native kod) yazmak yerine Cross Platformun avantajlarından yararlanır. Cross mobil platform geliştirme araçları incelendiğinde, yapılacak projeye göre farklı platformlar olduğu görülür. Bunlardan bazıları:

- **Unity3D:** Mobil oyun ya da etkileşimli oyun tabanlı eğitsel içerikler geliştirmek için tercih edilen oyun motoru olarak tanımlanır.
- **Xamarin:** .NET/C# programlama dili kullanılarak Android, IOS ve WP'de mobil uygulama geliştirilebilen oldukça güçlü bir Cross Mobil Platform geliştirme aracıdır.

- **PhoneGap:** Adobe firmasının desteklediği Css, Html ve JavaScript geliştiricilerinin kullanabileceği OpenSouce kütüphanedir.
- **Sencha:** Sencha; Android, IOS, WP ve BlackBerry marketleri için tek bir kod parçacığı ile uygulama geliştirilen Cross Mobil Platformdur.

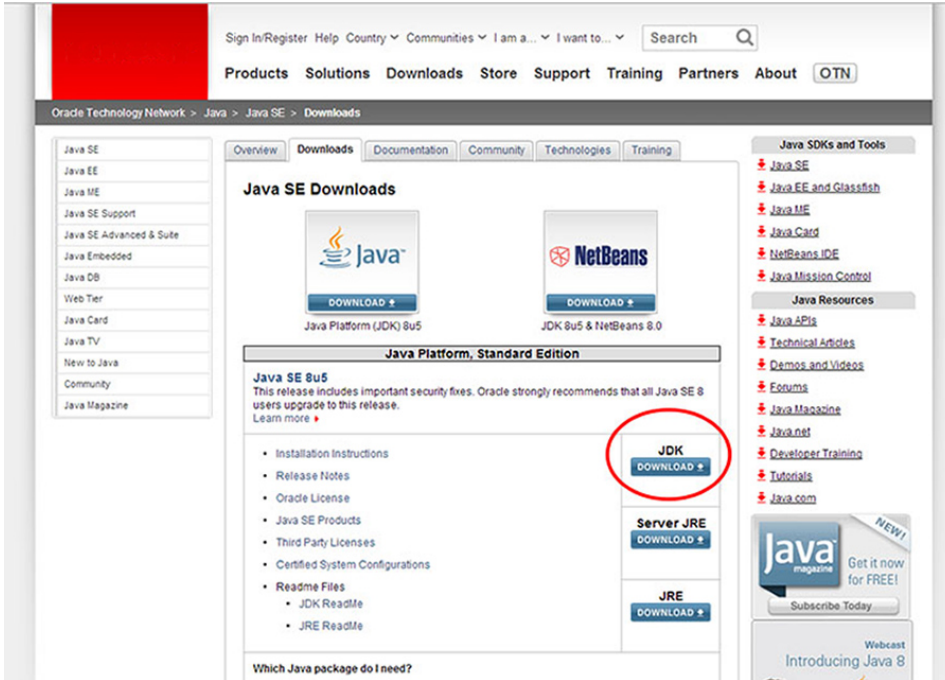
1.3. Java Geliştirme Paketi Kurulumu

Bu ve diğer materyallerde native dille uygulama geliştirme anlatılacaktır. Kullanılacak Native dil ile uygulama geliştirmek için bilgisayarınızda Java Geliştirme Paketi JDK'nin kurulu olması gerekmektedir.

Java Geliştirme Paketi (JDK) ise Java uygulamalarını çalıştırmak için gereken Java Çalışma Zamanı Ortamını (Java Runtime Environment - JRE) içinde barındırır. JRE içinde Java Sanal Makinesi (JVM) ve açık kaynak kodlu Java API' leri yer alır.

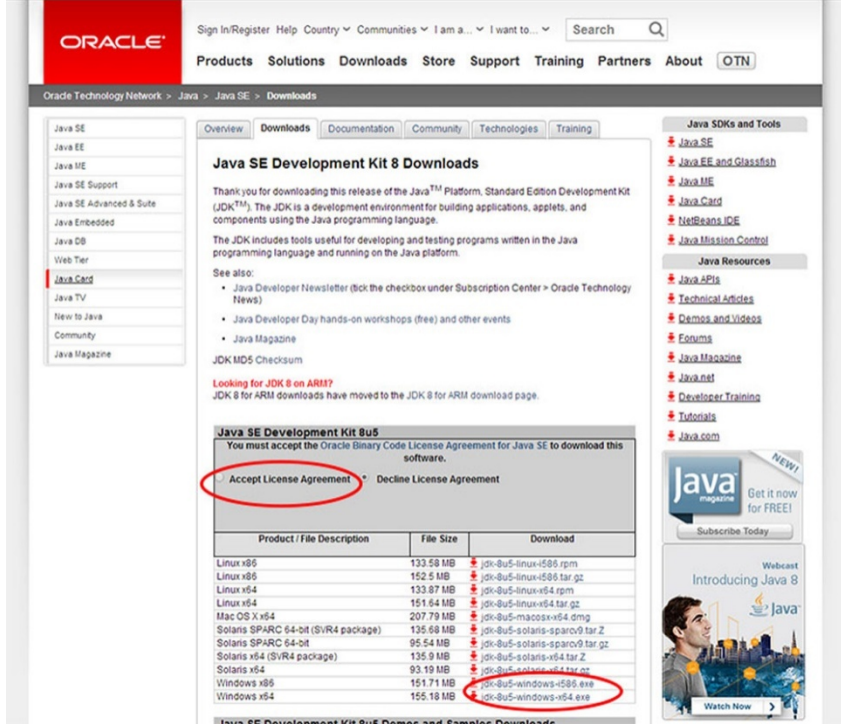
- **Adım 1:** Arama motorunda Java JDK diye arama yapıldığında en üstteki bağlantıda son versiyon olan oracle download web sayfasına ulaşılır. Burada birden fazla java versiyonu olduğu görülür. Java SE, JavaEE, Java ME gibi.

Standart Edition, Java SE' nin kurulumu;



Fotoğraf 1.1: Java download sayfası

- **Adım 2:** Aşağıdaki fotoğrafta gösterilen lisans anlaşması kabul edilir, uygun işletim sistemi seçilip paket bilgisayarına indirilir.



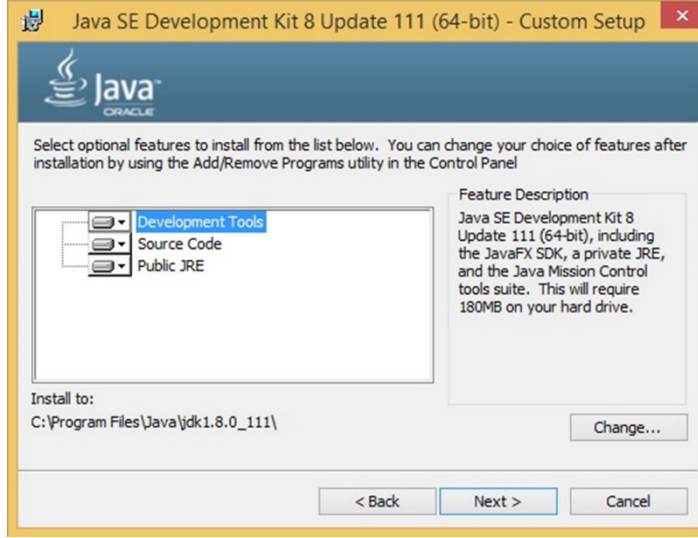
Fotoğraf 1.2: Java License Sözleşmesi ekranı

- **Adım 3:** İndirilen Java paketine tıklandığında Java kurulum sihirbazı ekranı gelir bu ekranda next tuşu ile ilerlenir.



Fotoğraf 1.3: Java kurulum ekranı

- **Adım 4:** JDK' nin kurulumunun nereye yapılacağı gösterilmektedir. Change seçeneğine basılıp JDK' nin kurulacağı dizin değiştirilir. Next seçeneği ile kurulum başlatılır.



Fotoğraf 1. 4: Java kurulum ekranı

- **Adım 5:** Kurulum işlemleri tamamlandıktan sonra close seçeneği ile kurulum tamamlanır.

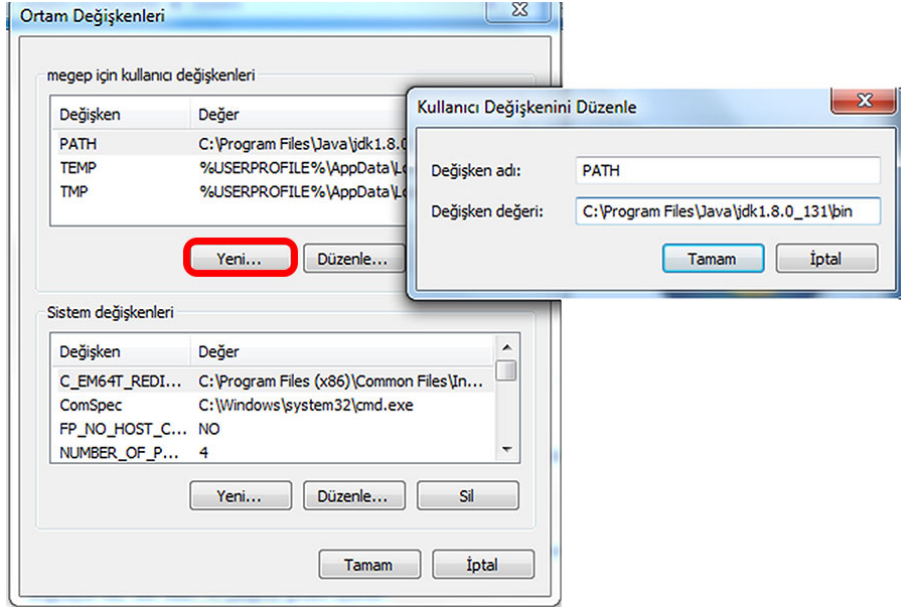


Fotoğraf 1. 5: Java kurulumunun başarı ile tamamlandığını gösteren ekran

1.3.1. Java Path Ayarları

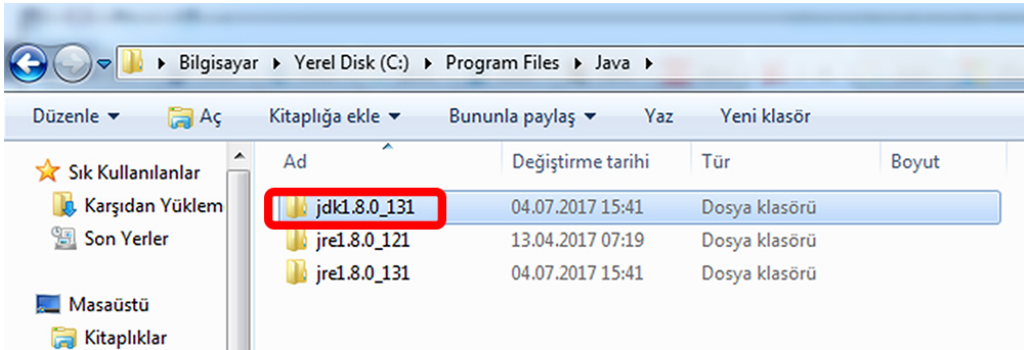
Java kurulumu tamamlandığında Windows işletim sistemi kullanılıyorsa birkaç ayarlanmanın el ile yapılması gerekir. Bilgisayarım→ Sistem Özellikleri→ Gelişmiş Sistem Özellikleri Sekmesi→ Ortam Değişkenleri penceresi açılır.

- **Adım 6:** Öncelikle Java'nın hangi klasör altında çalıştığını gösteren PATH Değişkeni eklenir.



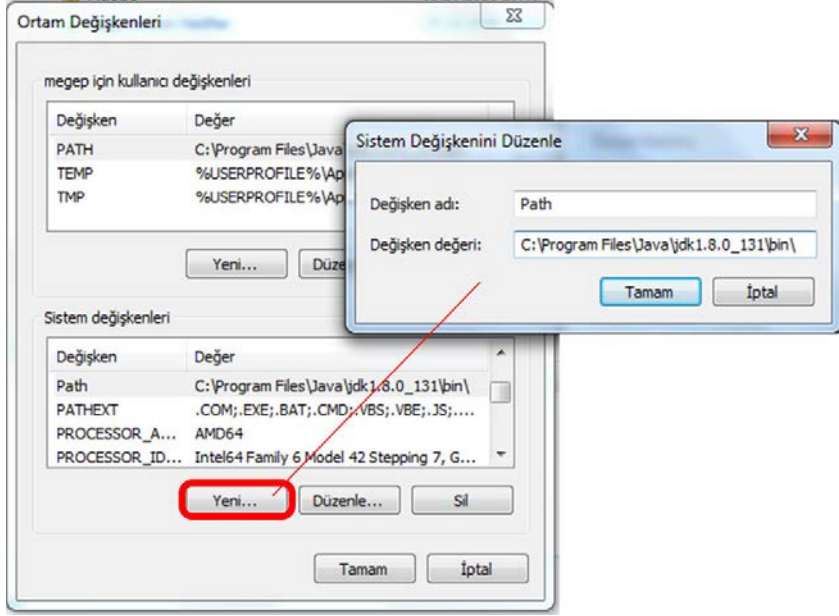
Fotoğraf 1.6: Java path ayarları ekranı

Burada dikkat edilmesi gereken kısım javajdk versiyonunun doğru olarak yazılmasıdır. Hangi versiyonun yüklendiği, Java'nın yüklü olduğu dizin içinden bulunabilir. Aşağıdaki ekran görüntüsünü inceleyiniz. Birden fazla jdk versiyonu varsa en son versiyonu seçiniz.



Fotoğraf 1.7: Java kurulum dizini

- **Adım 7:** Path Değeri tanımlandıktan sonra aşağıdaki fotoğrafta verilen sırada sistem değişkenleri alanından Path değeri bulunur. Buradaki değer sonuna noktalı virgül bırakılarak daha önce kopyalanmış olan Java dizin yolu buraya yapıştırılır.



Fotoğraf 1.8: Java path ayarlarının yapıldığı ekran

JDK kurulumu yapıldıktan sonra çalışıp çalışmadığını anlamak için aşağıdaki adımlar takip edilmelidir. Öncelikle **Başlat > Çalıştır** denir, **cmd** yazılır, aşağıdaki gibi komut sistemi ekranı gelir.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Sürüm 6.1.7601]
Telif Hakkı (c) 2009 Microsoft Corporation. Tüm hakları saklıdır.
C:\Users\megep>javac
```

Fotoğraf 1.9: Komut ekranı

Bu ekrana javac yazılır ve enter tuşuna basılır. Aşağıdaki ekran görüldüğünde Java JDK başarıyla kurulmuştur.

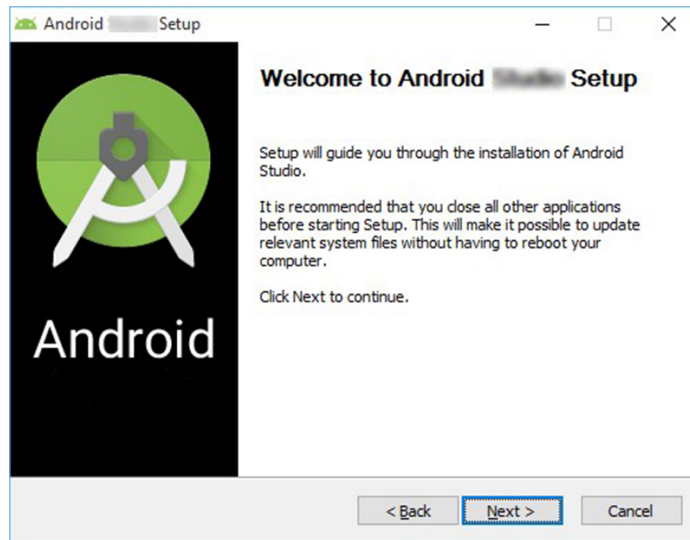
```
C:\Windows\system32\cmd.exe
-processorpath <path> Specify where to find annotation processors
-parameters Generate metadata for reflection on method parameters
-rs
-d <directory> Specify where to place generated class files
-s <directory> Specify where to place generated source files
-h <directory> Specify where to place generated native header files
-s
-implicit:<none,class> Specify whether or not to generate class files for
implicitly referenced files
-encoding <encoding> Specify character encoding used by source files
-source <release> Provide source compatibility with specified release
-target <release> Generate class files for specific VM version
-profile <profile> Check that API used is available in the specified profile
-profile
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system
-Werror Terminate compilation if warnings occur
@<filename> Read options and filenames from file
C:\Users\megep>
```

Fotoğraf 1.10: Java kurulumu test ekranı

1.4. Mobil Uygulama Geliştirme Ortamının Kurulumu

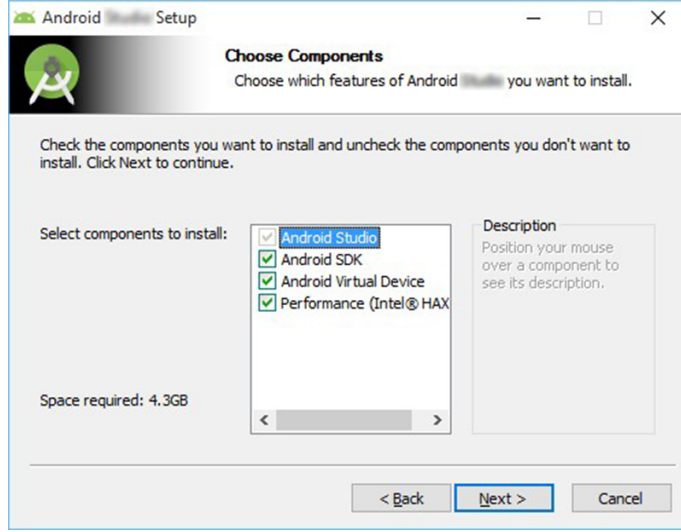
Mobil uygulama geliştirme ortamı olarak Android İşletim Sistemi için Java programlama dilini kullanan Native uygulama geliştirme ortamının kurulumu anlatılacaktır.

- **Adım 1:** İndirilen program tıklandığında cihaz üzerinde kurulu yönetimsel izinler istenir. Yönetici izni verildikten sonra, Android Studio kurulumu için ilk adım görülür. Next düğmesine basılır, kurulum işlemine başlanır.



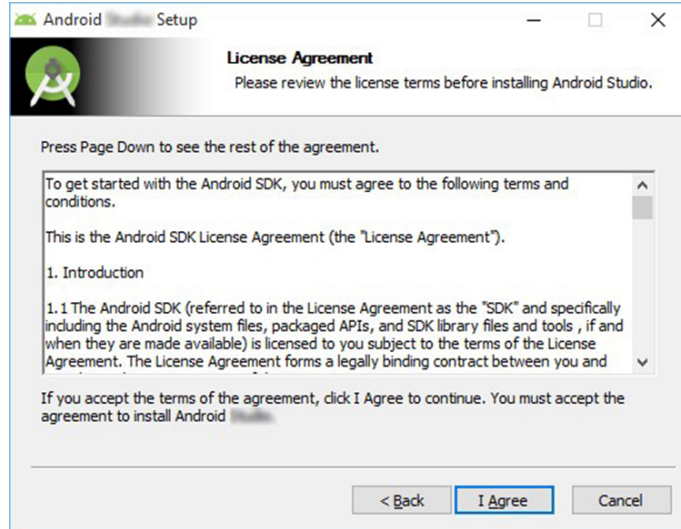
Fotoğraf 1.11: Uygulama ortamı kurulumu başlangıç ekranı

- **Adım 2:** Gelen ekranda mobil işletim sisteminin SDK'lerinin yükleneceği ve mobil işletim sistemi test ortamı için sanal makinenin kurulacağı seçenekler yer alır. Intel Donanım Hızlandırıcı desteği bulunan bilgisayarlarda üçüncü bir seçenek olarak performans hızlandırıcı yer alır. Eğer bilgisayarda bu özellik yoksa bu seçenek yer almaz. Next düğmesine basılarak bu ekran geçilir.



Fotoğraf 1.12: Uygulama ortamına bileşen ve sanal makinenin kurulum seçenekleri

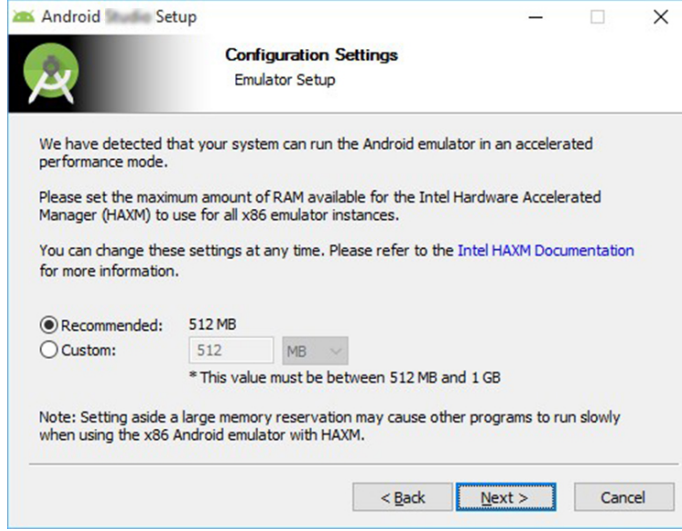
- **Adım 3:** Burada lisans sözleşmesi ekrana gelir. Sözleşme okunduktan sonra eğer kabul edilirse "I Agree" düğmesine basılır. Daha sonra next' e basılarak diğer ekrana geçilir.



Fotoğraf 1.13: Lisans sözleşmesi ekranı

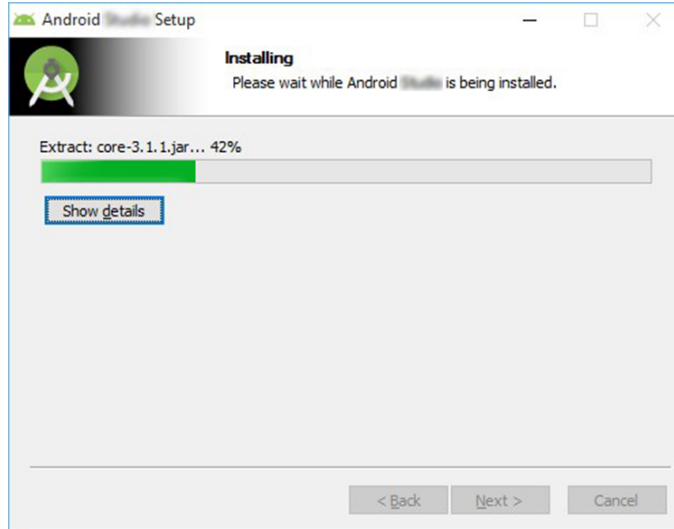
- **Adım 4:** Eğer Intel Donanım Hızlandırıcı varsa, bir sonraki adımda Android uygulaması için kullanılacak emülatörün kurulum ayarlarının bulunduğu ekran yer alır. Buradan emülatörün kullanabileceği bellek miktarı ayarlanabilir.

Intel Donanım Hızlandırıcı desteği bulunmayan bilgisayarlarda, bu pencere görüntülenmeyip, doğrudan bir sonraki adıma geçilir.



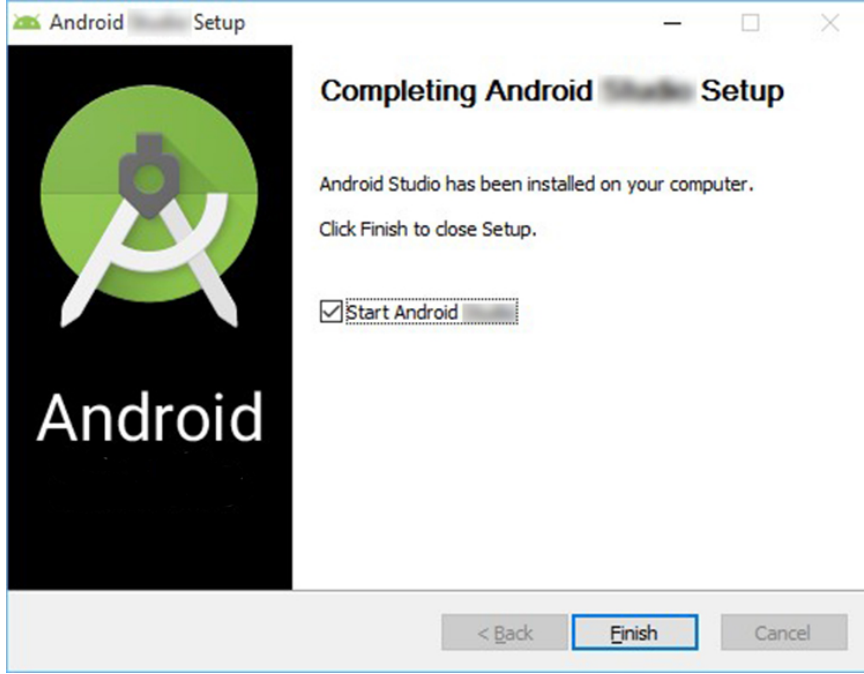
Fotoğraf 1.14: Donanım hızlandırıcılar için emülatör yapılandırma ekranı

- **Adım 5:** Kurulum işleminin başladığını gösteren bileşenlerin yüklenme ekranı gelir.



Fotoğraf 1.15: Bileşenlerin ve programın yüklenmesi

- **Adım 6:** Kurulum başarı ile tamamlandığında aşağıdaki ekran gelir.



Fotoğraf 1.16: Kurulum başarıyla tamamlandığı ekranı

DEĞERLER ETKİNLİĞİ-1

Aşağıdaki hikâyeyi okuyunuz. Hikâyeden çıkardığınız sonucu aşağıda boş bırakılan bölüme yazınız.

SIFIRIN DEĞERİ

Öğrenciler, sınıfta, öğretmenin gelmesini beklemektedir.

Az sonra, öğretmen, tüm ciddiyetiyle kapıda belirir; sınıfa şöyle bir göz atar, tahtanın önüne geçer; bir tebeşir alarak, tahtaya, kocaman bir 1 yazar.

- Bakın, bu 1 kişiliktir; erdemli bir kişilik... Hayatta sahip olabileceğiniz en değerli şey, der.

Sonra 1'in yanına 0 koyar.

- Bu, başarıdır. Başarılı bir kişilik, 1'i 10 yapar.

Bir 0 daha...

-Bu da tecrübedir. Ona sahip olduğunuzda 10'ken, 100 olursunuz, der.

Sıfırlar, böyle uzayıp gider; yetenek, disiplin, sevgi, tevazu... Eklenen her yeni 0'ın kişiliği, 10 kat zenginleştirdiğini anlatır öğretmen.

Sonra, eline silgiyi alır, en baştaki 1'i siler; geriye bir sürü 0 kalır ve der ki:

- Erdemli bir kişiliğiniz yoksa gerisi hiçtir.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ İş sağlığı ve güvenliği tedbirlerini alınız.➤ Java Geliştirme Paketi JDK'nin kurulum dosyasını indiriniz.	<ul style="list-style-type: none">➤ Java SE Development Kit 8 paketini browserda aratarak işletim sisteminizin versiyonuna göre indirmelisiniz. Varsa en güncel sürümünü indirmelisiniz.
<ul style="list-style-type: none">➤ Kur (setup) dosyasını açarak kurulumu başlarız.	<ul style="list-style-type: none">➤ Kurulum yaparken dikkatli olmalısınız. Öğrenme faaliyetindeki işlemleri uygulamalısınız.
<ul style="list-style-type: none">➤ Path ayarlarını yapınız.	<ul style="list-style-type: none">➤ Şekil 1.6, 1.7, 1.8 yönergelerini takip etmelisiniz.
<ul style="list-style-type: none">➤ Kurulumun doğru yapıldığını test ediniz.	<ul style="list-style-type: none">➤ Başlat->çalıştır-cmd ile komut ekranına geçmelisiniz.➤ Javac komutu ile kurulumun başarıyla yapılıp yapılmadığını test etmelisiniz.
<ul style="list-style-type: none">➤ Android Mobil İşletim Sistemi Uygulama Geliştirme Programı'nın dosyalarını elde ediniz.	<ul style="list-style-type: none">➤ Bu program ücretsiz olarak dağıtılmaktadır. Varsa programın en güncel sürümünü elde etmelisiniz.
<ul style="list-style-type: none">➤ Kur (setup) dosyasını açarak kurulumu başlarız.	<ul style="list-style-type: none">➤ Mobil İşletim Sistemi bileşenlerini ve sanal makine kurulumunu yapmalısınız Şekil 1.11 ve şekil 1.12'ye bakmalısınız.

KONTROL LİSTESİ

Aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. İş sağlığı ve güvenliği tedbirlerini aldınız mı?		
2. Java SE Development Kit paketini İnternet'ten bulup indirebildiniz mi?		
3. Java SE kurulumunu yapabildiniz mi?		
4. Java Path ayarlarını yapabildiniz mi?		
5. Java'nın doğru olarak kurulduğunun testini yapabildiniz mi?		
6. Android geliştirme ortamının kurulumunu yapabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda **Hayır** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **Evet** ise bir sonraki öğrenme faaliyetine geçiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatle okuyarak doğru seçeneği işaretleyiniz.

1. Geliştirdiğiniz mobil uygulamanın gereksinimlere cevap verebilmesi için aşağıdaki ölçütlerden hangisine uygun olmasına gerek yoktur?
A) Performans
B) Güvenlik
C) Karışık arayüzler
D) Stabilité
E) Anlaşılabilirlik
2. Aşağıdakilerden hangisi mobil uygulama geliştirme platformlarından biri değildir?
A) Yerel (Native)
B) Melez (Hybrid)
C) Cross (Çapraz)
D) Uyumlu (Responsive)
E) Değişken (Variable)
3. Platform sağlayıcısının ürettiği platform SDK'si, araçları ve programlama dili yardımıyla geliştirilmiş uygulama türü aşağıdakilerden hangisidir?
A) Yerel (Native)
B) Melez (Hybrid)
C) Cross (Çapraz)
D) Uyumlu (Responsive)
E) Değişken (Variable)
4. Aşağıdaki mobil işletim sistemlerinden hangisine Java dili ile uygulama geliştirme yazılabilir?
A) Apple iOS
B) Symbian
C) Windows 7 Phone
D) Google Android
E) Windows Mobile

ÖĞRENME FAALİYETİ-2

ÖĞRENME KAZANIMI

İşletim sistemine uygun uygulama geliştirme ortamı kurabileceksiniz.

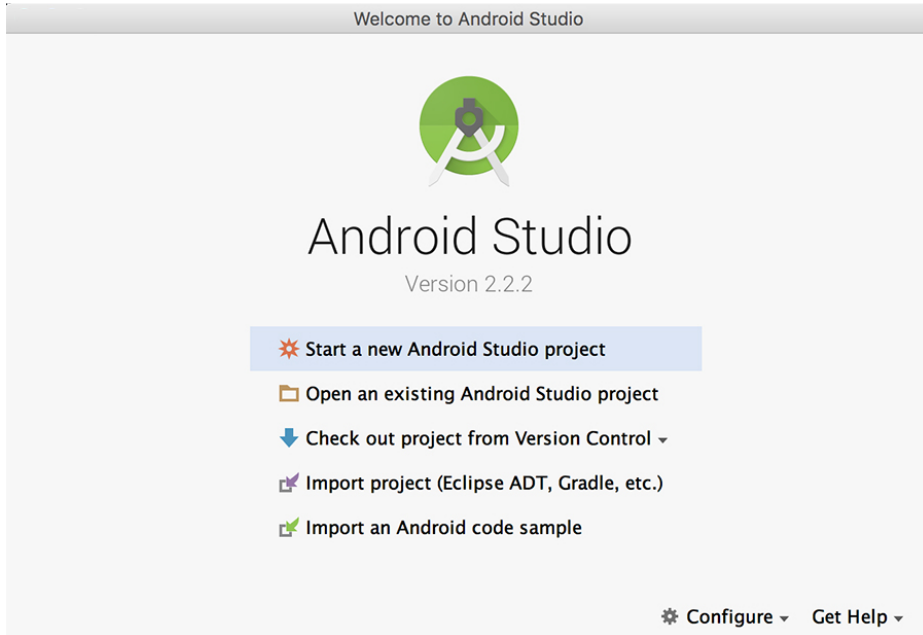
ARAŞTIRMA

- Andorid Play Store inceleyiniz. Android Studio ile yazılmış uygulamalardan örnekler bularak sınıfta bunları tanıtınız.
- Android Kütüphane yapısını araştırınız. En popüler kütüphanelerden 3'ünü seçiniz. Seçtiğiniz kütüphaneleri sınıfta tanıtınız.

2. UYGULAMA GELİŞTİRMEYE GİRİŞ

2.1 Yeni Proje Oluşturma

Uygulama geliştirme programı ilk çalıştırıldığında şekil 2-1'deki karşılama ekranı görülür.



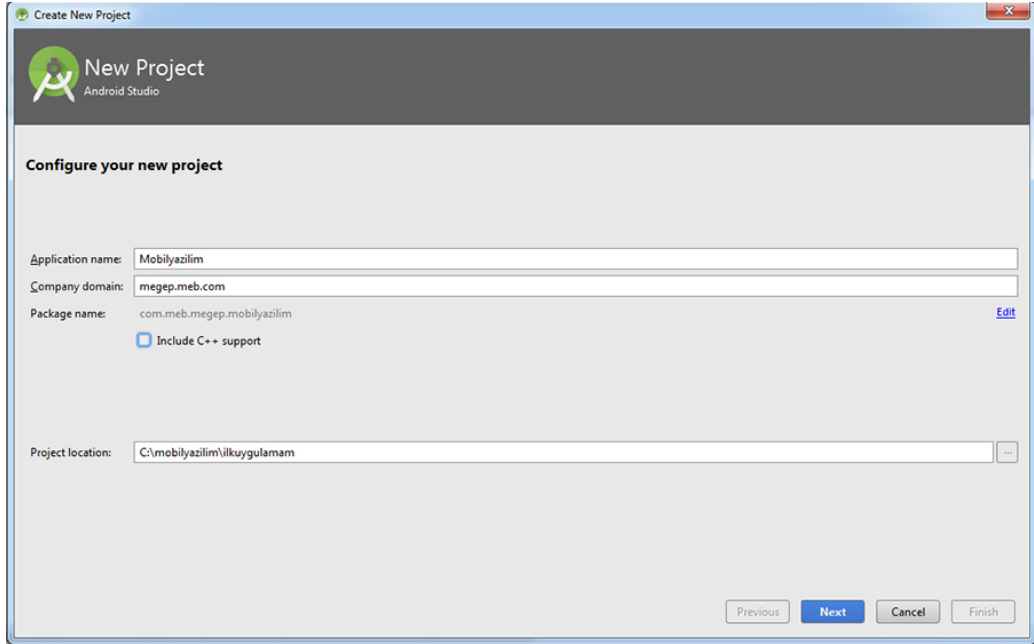
Fotoğraf 2. 1: Uygulama geliştirme programı karşılama ekranı

Ekranda ilk görünen Android Studio giriş ekranıdır. Eğer önceden bir uygulama oluşturulmuşsa bu ekranın sol tarafında önce oluşturulan uygulamalar da listelenir.

Android Studio giriş ekranında yer alan seçenekler şunlardır.

- **Start a new Android Studio project:** Yeni bir Android Studio projeye başlamak için bu seçenekle devam edilir.
- **Open an existing Android Studio project:** Bu menü önceden oluşturulmuş bir Android Studio projesini açmak için kullanılır.
- **Check out Project from Version Control:** Git, svn gibi herhangi bir versiyon kontrol sisteminden çekilecekse bu seçenekle devam edilir.
- **Import project (Eclipse ADT, Gradle,etc.):** Farklı bir ide ya da farklı bir build sistemi ile yazılmış bir projeyi Android Studio projesine çevirerek açmak için kullanılır. Özellikle önceden yazılmış bir eclipse projesini açmak için kullanılabilir.
- **Import an Android code sample:** Android'in kullanıcıya sunduğu hazır kodları açmak için bu menü kullanılır.

Start a new Android Studio project ile yeni proje oluşturma ekranına geçilir.

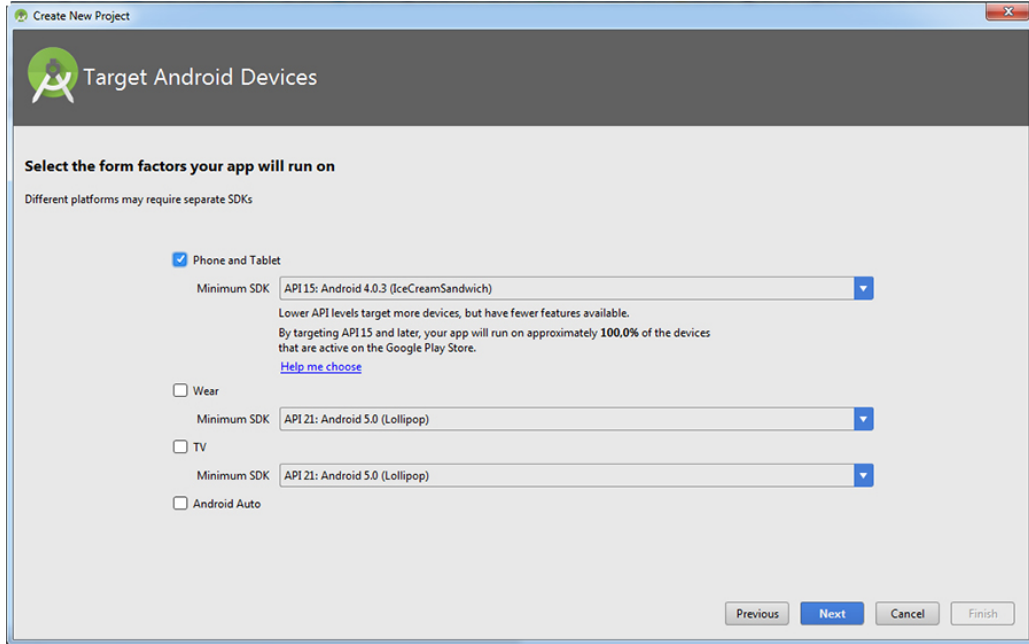


Fotoğraf 2.2: Yeni proje konfigürasyon ekranı

Yukarıdaki ekran, oluşturulacak projenin konfigürasyon ayarlarının yapıldığı ekrandır. Burada verilen isimler önemlidir. Bu nedenle bu alanların neler olduğu açıklanmalıdır.

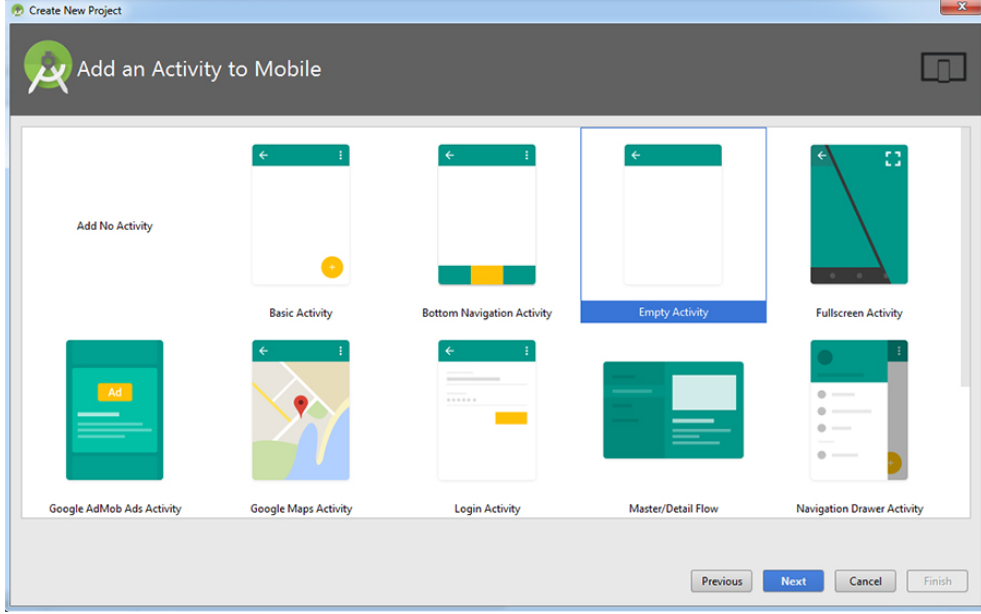
- **Application Name:** Uygulama adımız (Türkçe karakter kullanılmaması önerilir.). Cihazlarda ikonların altında yer alır. Uygulama isminin büyük harfle başlaması önerilir.
- **Company Domain:** Uygulamanın geliştirildiği şirketin domainidir (Tersten yazılması önerilir.).
- **Package Name:** Company domain ve packagename'e göre otomatik oluşur. Değiştirilmek istenirse sağ tarafındaki edit ile değiştirilebilir. Package name Google Play'deki uniqueid gibi düşünülebilir. Aynı package name ile başka bir uygulama yüklenemez.
- **Project Location:** Projenin oluşturulacağı klasördür.

Gerekli alanlar doldurulduktan sonra next'e basılarak ilerlenir.



Fotoğraf 2.3: Projenin konfigürasyon ayarları ekranı

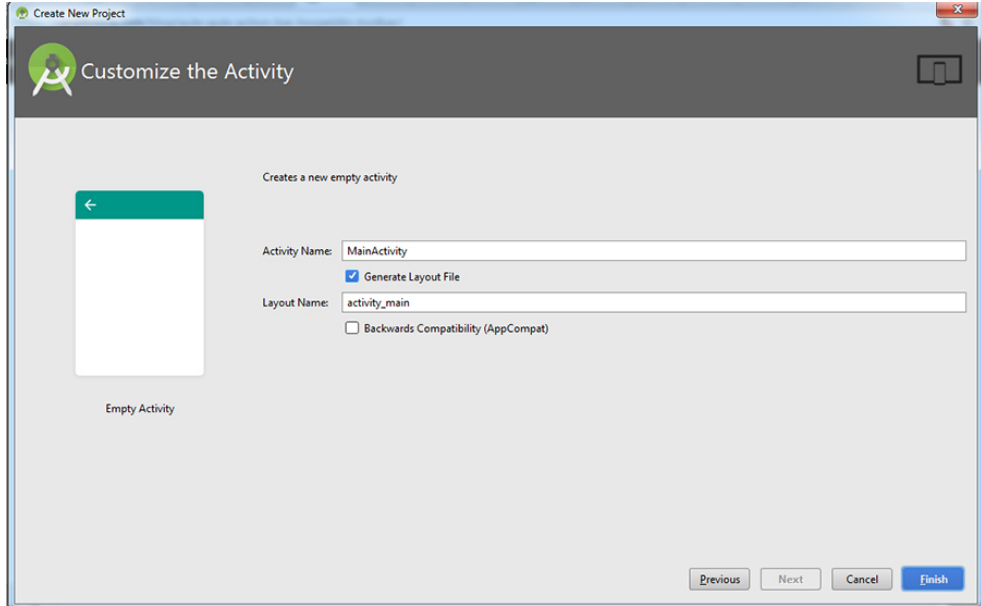
Bu ekranda telefon ve tablet, giyilebilir cihaz (gözlük, saat vb.) veya televizyon gibi farklı platformlardan hangisinde uygulama geliştirileceği seçilir. Platform seçimi yapıldıktan sonra en düşük hangi Android sürümünün destekleneceği seçilir. Seçilen versiyonun alt kısmında android destekli cihazların tahminen ne kadarında çalışacağı gösterilmektedir. Örnek olarak Api 15 üzeri versiyon seçilmiştir. Bu versiyonun cihaza tahminen %100'lük bir destek sunacağından bahsedilmektedir. Seçim yaptıktan sonra **next** ile ilerlenir.



Fotoğraf 2. 4: Activity seçim ekranı

Bu ekran Android template Activity'ler sunmaktadır. Empty Activity seçilerek Next'e tıklanır. İlerleyen zamanlarda proje ihtiyacına göre farklı templateler seçilebilir.

- **Activity** : Kullanıcının gördüğü sayfayı oluşturan java sınıfları (class) dir.

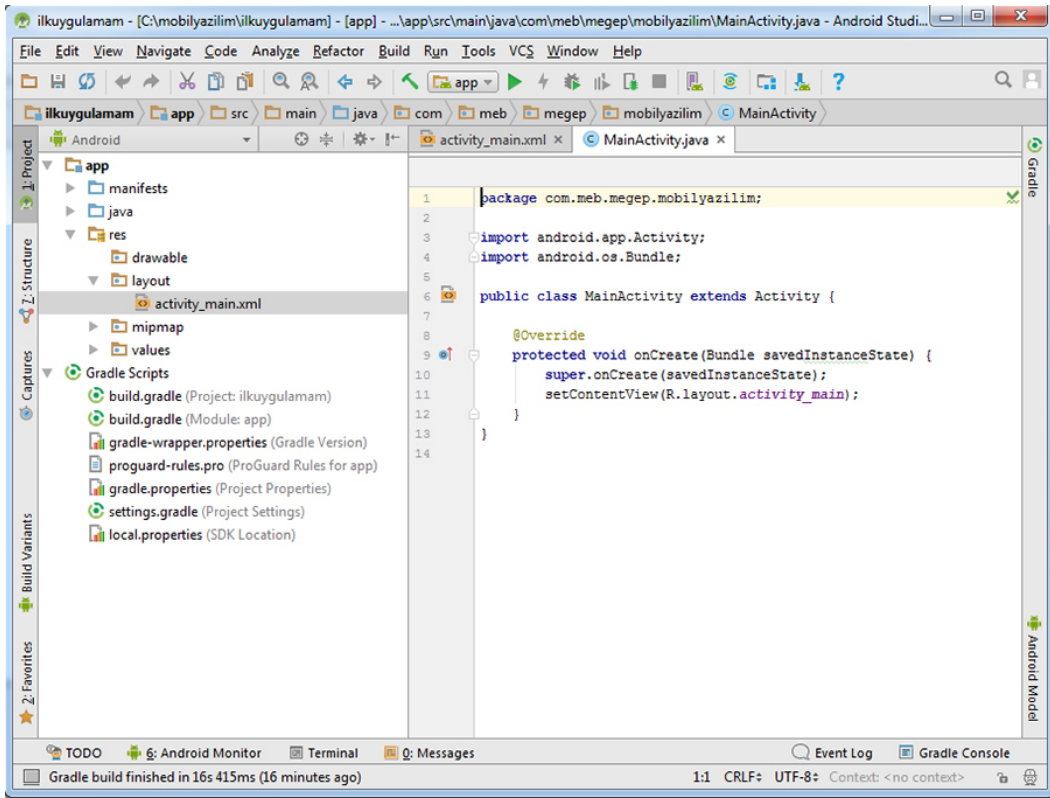


Fotoğraf 2.5: Activity ayarlama ekranı

Yukarıdaki ekranda, açılacak olan ilk Activiy'e ve onun layoutuna isim verilir. Bu isimler değıştirilebilir. Layout, oluşturulan Activiy'nin görsel kısmı olarak düşünülebilir. Bu, istenirse Generate Layout file seçilerek otomatik veya seçilmeyerek uygulama içinden manuel oluşturulabilir.

NOT: Backwards compability (geri gidebilme) seçeneđi seçilirse, uygulama Tabbar destekli gelecektir(App Combat Activity). Böylelikle Tabbar'a back tuşu ve farklı butonlar konulabilir. Seçilmezse Tabbar destekli gelmez. Burada ilk uygulama yazıldığı için Backwards compability özelliđi kaldırılır ve **Finish**'e basılır.

Aşađıda oluşturulan ekran projenin ilk açılış ekranıdır.



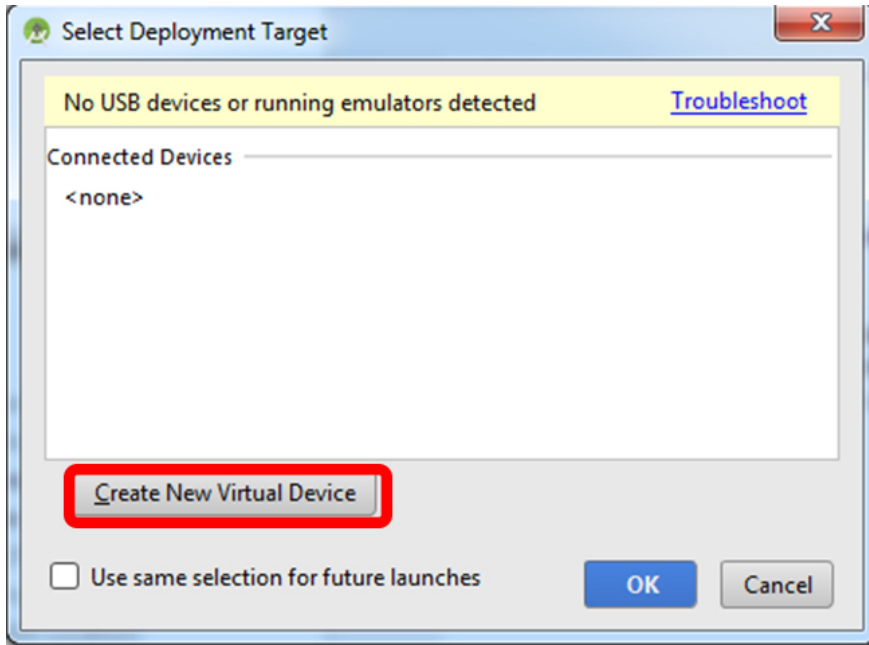
Fotođraf 2.6: Proje açılış ekranı

Oluşturulan template'e göre ekrana boş bir activity ve o activity'nin layoutu gelir. Oluşturulan proje çalıştırılabilir bir Android Uygulaması'dır. Çalıştırıldığında boş bir ekranda Hello World yazısı açılır. Bunu çalıştırabilmek için bir Android cihaza ihtiyaç vardır. Bu gerçek bir Android cihaz da olabilir, emülatör de olabilir. Bu materyalde proje çalıştırma emülatör üzerinden gösterilecektir.

2.2 Emülatör Oluşturma

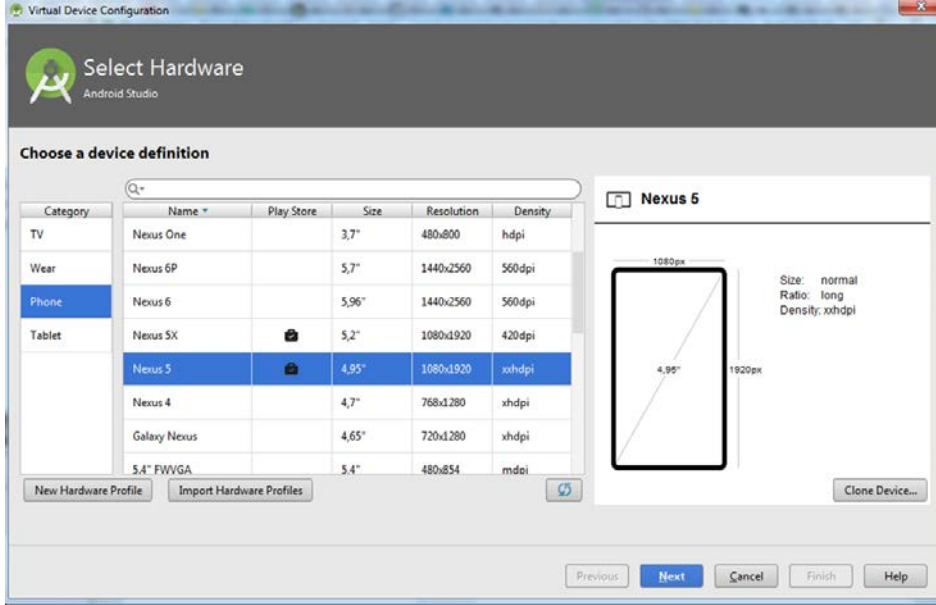
Emülatör, bilgisayar üzerinden çalışan sanal bir android cihaz demektir. Uygulama ilk oluşturulduğunda herhangi bir kod yazılmadan otomatik olarak çalıştırılabilecek bir uygulama hazır bir şekilde açılır. Üst taraftaki **Toolbar**'dan **Run** tuşuna (Yeşil OK) basılır.

Herhangi bir Emülatör tanımlaması yapılmadığı veya gerçek bir Android cihaz bilgisayara bağlanmadığı için fotoğraf 2.7'deki gibi bir ekran açılır. Ekranda görüldüğü gibi uygulamayı çalıştırabilecek bir cihaz yoktur. Bunun için Create New Virtual Device'e basılarak devam edilir.



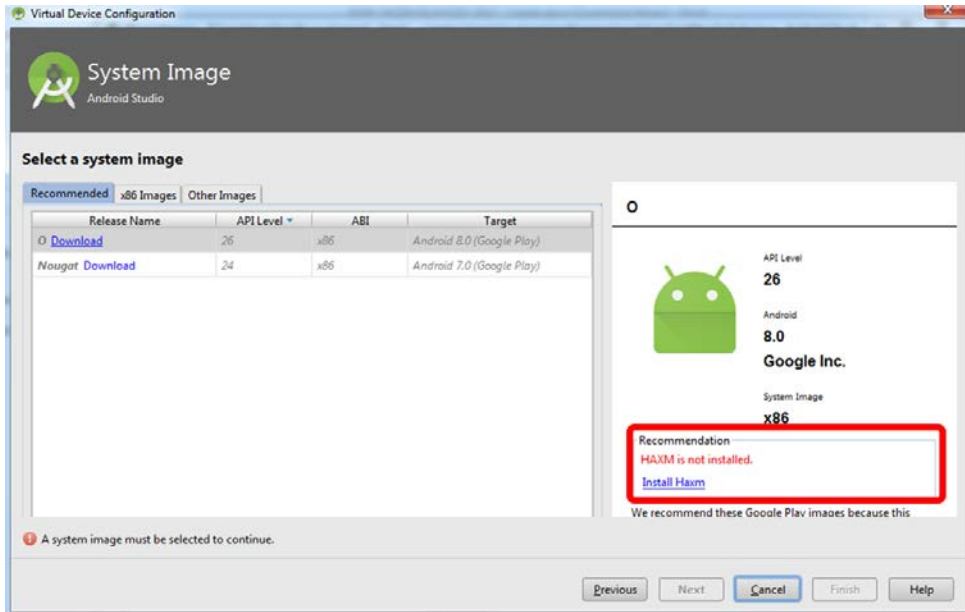
Fotoğraf 2.7: Sanal cihaz oluşturma ekranı

Fotoğraf 2.8'deki ekranda kurulmak istenen cihaz sorulmaktadır. Burada görüldüğü gibi onlarca farklı cihaz oluşturulup uygulama bunların üzerinden denenebilir. Herhangi biri seçilerek Next tuşuna basılır.



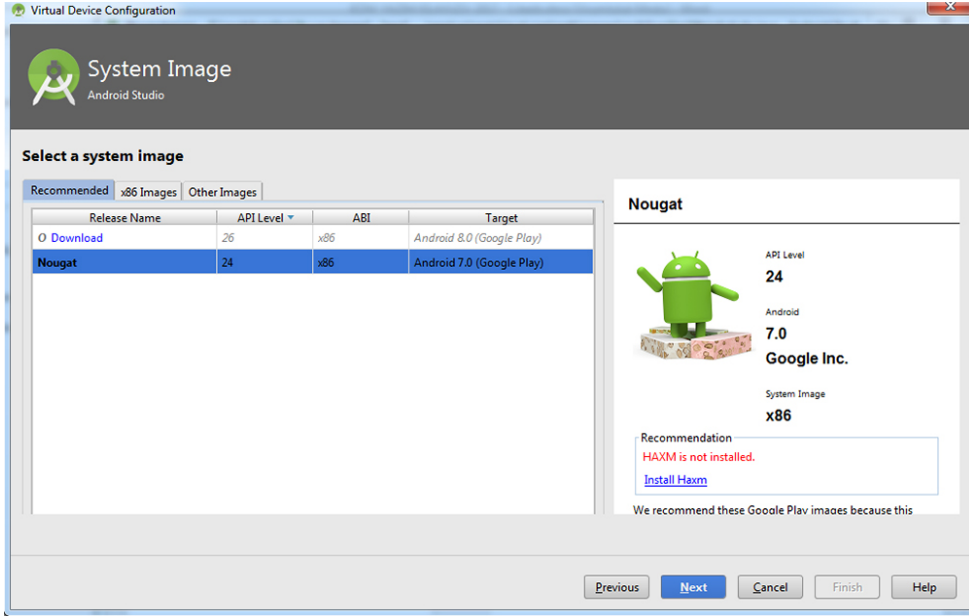
Fotoğraf 2.8: Sanal cihaz seçim ekranı

Eğer bilgisayarda donanım hızlandırıcı özellik varsa ekrana Fotoğraf 2.9'daki gibi bir görüntü gelir. Bu ekranın sağ tarafında öneri (Recommendation) altında install HAXM (Intel Hardware Accelerated Execution Manager) seçeneği yer alır. Bu kısaca şöyle açıklanır: Emülatörlerin daha hızlı çalışması için önerilen bir kurulumdur. Dolayısıyla öncelikle Install Haxm'a basılarak başlanır. Eğer bu seçenek çıkmadıysa bu adım atlanır.



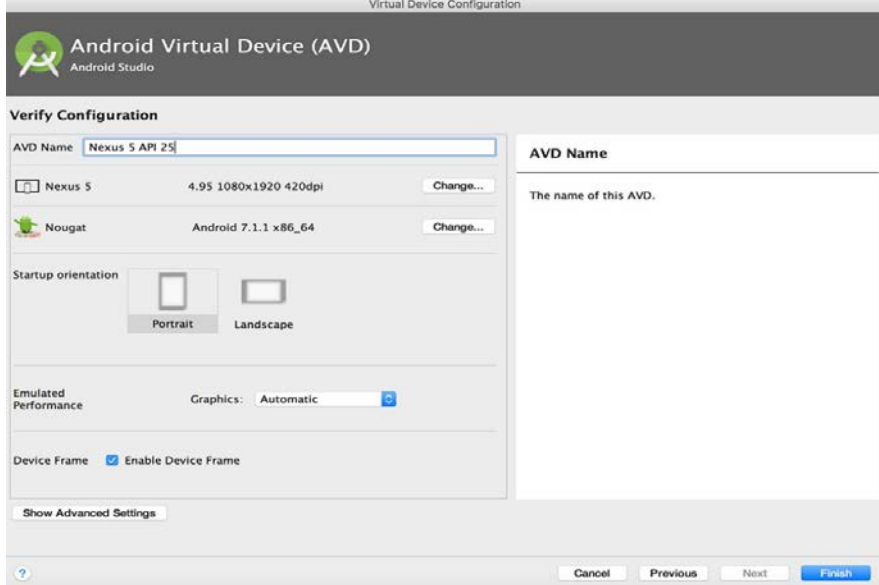
Fotoğraf 2.9: Sanal cihaz işletim sistemi sürümü seçim ekranı

HAXM kurulumu bitince Finish'e basılarak fotoğraf 2.9'daki ekrana geri gelinir. Burada ilk sayfada seçilen cihazın hangi sürümde olması isteniyorsa o seçilir. Android 5'ten 7'ye kadar seçenekler sunulmaktadır (Bilgisayardaki sisteme uygun olan seçilmelidir. Yani bilgisayar 64 bitlik ise 64 olan seçilmelidir.). Seçilen sürümün yanındaki Download linkine tıklanarak o sürüm için gerekli dosyaların indirme işlemleri otomatik olarak başlar.



Fotoğraf 2.10: Android İşletim Sistemi sürümü dosyalarının indirilmiş hâli

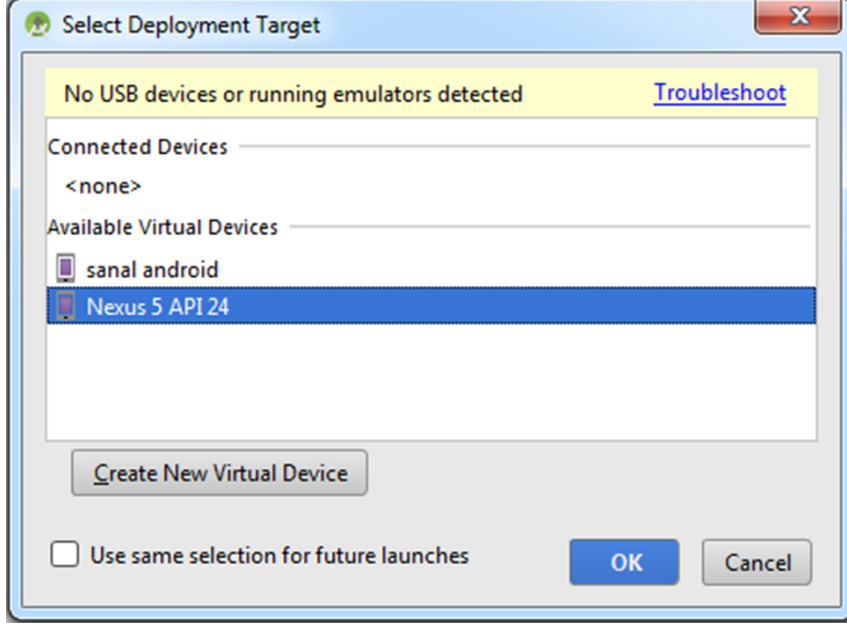
Fotoğraf 2.10'daki bu ekranda indirilen sürüm seçilir, Next'e basılarak devam edilir.



Fotoğraf 2.11: Emülatör ekran ve açılış ayar ekranı

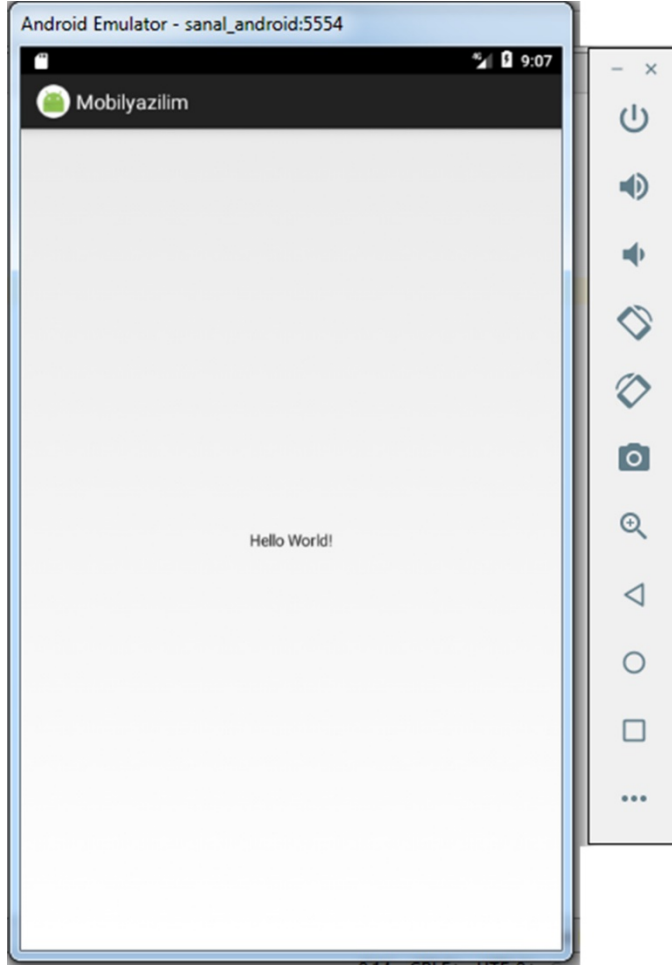
Fotoğraf 2.11'deki ekranda ise oluşturulan emülatöre isim verilir ve başlangıçta açılırken dikey modda ya da yatay modda başlaması istenilen screenorientation seçilir, Finish'e tıklanır. Böylelikle emülatör ayarları tamamlanır.

Finish'e tıklanır. İşlemler bitirildiğinde daha önce fotoğraf 2. 7'de boş gelen emülatör ekranında artık oluşturulan emülatör görülebilir.



Fotoğraf 2.12: Emülatör Seçim Ekranı

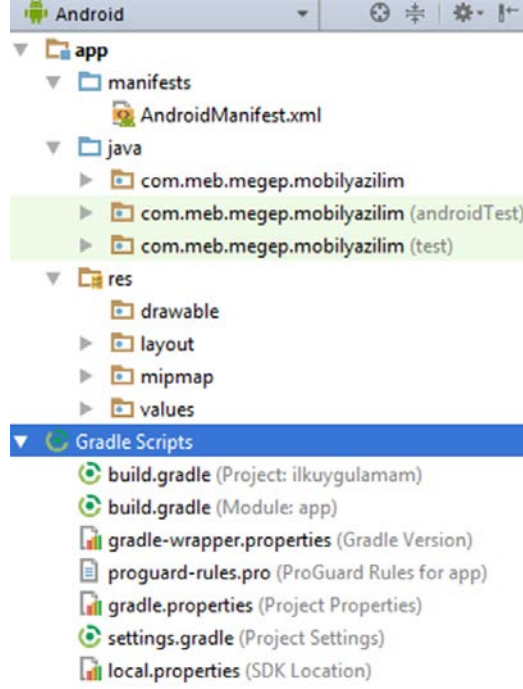
Yukarıdaki ekrana göre uygulama hangi emülatör ekranında çalıştırılmak istenirse o emülatör seçilip ok tuşuna basılır. Fotoğraf 2.13'te ilk android uygulama olan “**Hello World**” uygulaması görülmektedir.



Fotoğraf 2.13: Emulator üzerinde çalışılan programın ekranı

2.3 Proje Yapısı

Android bir projenin temel bileşenlerini tanımak ilerisi için çok önemlidir. Fotoğraf 2.14'te bir projenin temel yapısı yer almaktadır. Bir proje içinde çeşitli klasörler vardır.



Fotoğraf 2.14: Bir Android Programı'nın klasör yapısı

2.3.1. Manifest

Bu klasör bir projenin en temel bileşeni olan Android Manifest.xml dosyasını saklar. Bu dosyada projenin tüm özellikleri mevcuttur. Bu dosyanın her projede bulunması gerekir.

- **Not:** Android Manifest.xml dosyasının adı kesinlikle değiştirilmez. Bu isim sistem tarafından otomatik ve varsayılan olarak verilen bir isimdir.

2.3.2. Java

Bu klasör içinde projede kullanılan java uzantılı class yani sınıflar vardır. Bir önceki derste oluşturulan projede **MainActivity** isimli activity sınıf java uzantılıdır.

2.3.3. Res

Projede kullanılan resim, video, string, menü, layout yani kullanıcı tasarım öğelerini içinde barındıran klasördür. Bu klasör projenin kaynak (resource) klasörüdür. Bu klasör içinde aşağıda açıklanan klasörlerde vardır. Her bir klasör farklı bir kaynağı içerir.

- **Drawable:** Resim nesneleri burada tutulur. Bu resimler farklı ekran çözünürlüklerine göre aşağıdaki gibi 4 farklı klasöre ayrılabilir. Fotoğraf 2.15'te görüldüğü üzere aynı resim farklı klasörlerde yer alabilir. Bunun sebebi her Android cihazının aynı çözünürlükte olmamasıdır. Eğer çok yüksek çözünürlüklü bir resim, düşük çözünürlüklü bir cihazda gösterilmek istenirse, resim ögesi geç yüklenir sistemin yavaşlamasına neden olur. Bundan dolayı aynı resmin farklı çözünürlükte örnekleri oluşturulur. Bu işlemden sonra uygulama hangi cihazda çalışırsa otomatik olarak o cihaza uygun çözünürlükte resim gösterilir.

```
MyProject/  
  res/  
    drawable-xhdpi/  
      awesomeimage.png  
    drawable-hdpi/  
      awesomeimage.png  
    drawable-mdpi/  
      awesomeimage.png  
    drawable-ldpi/  
      awesomeimage.png
```

Fotoğraf 2.15: Çözünürlüklere göre resim düzenleme ekranı

- **Layout:** Kullanıcı arayüzü için hazırlanan view veya ViewGroup nesneleri burada tutulur.
- **Menu:** Uygulamanın .xml uzantılı tanımlanan menü dosyaları burada tutulur.
- **Mipmap:** Bu klasör AndroidStudio'da icon öğelerini tutmak için kullanılır. Drawable klasöründe olduğu gibi 4 farklı klasörden oluşur.
- **Values:** Ölçü, string ve stil gibi dosyalar burada tutulur. dimens.xml, strings.xml ve styles.xml dosyalarından oluşur. dimens, ölçüleri, strings, uygulamada kullanacak metinleri ve styles ise uygulama için stilleri saklar.

2.3.4. GradleScripts

GradleScripts projenin derlenmesi ve APK'sinin oluşmasını sağlayan yapılar bu bölümdedir. Yeni bir proje açıldığında bu bölümün altındaki kısımlar otomatik olarak gelir.

2.4 Android Kütüphane Yapısı

Programlamadaki anlamıyla kütüphane, birçok sınıfı, fonksiyonu bir arada bulunduran, özel bir amaca yönelik yazılmış kodlar topluluğudur. Paket, kitaplık veya modül olarak da adlandırılır. Kütüphane kullanılarak bir programlama diliyle yapılabilecekler artırılıp işlemler kolaylaştırılabilir.

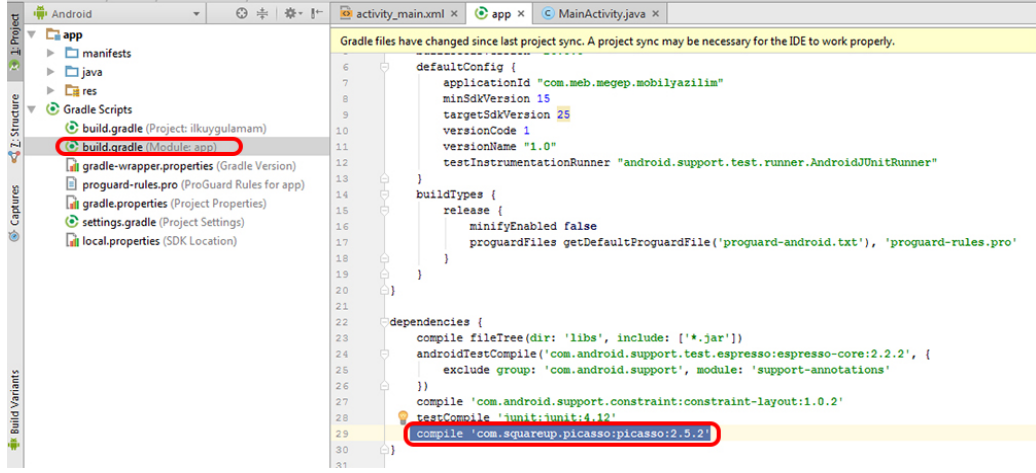
Android kütüphanesi yapısal olarak android uygulamasına benzer. Bir uygulama için gerekli olan androidmanifest dosyası, kaynak dosyalar ve kaynak kod vb. tüm bileşenleri içerir. Ancak bir cihazda çalışacak android uygulaması APK gibi derlenmek yerine, androidarhive dosyası (AAR) içinde android kütüphanesi olarak derlenir.

Android projelerinde kullanılabilecek kütüphane dosya tipi olarak Java Archive dosya (JAR) kütüphaneleri de vardır. Fakat AAR kütüphanesi ile JAR kütüphanesini birbirinden ayıran ana fark AAR kütüphanesi layout, drawables gibi görsel kaynakları içerirken JAR kütüphanesinde bunlar bulunmaz. Bu konu bir örnekle şu şekilde açıklanabilir: JAR ve AAR kütüphaneleri ile yazılmış benzer giriş (login) kütüphaneleri oluşturulur. JAR kütüphanesinde giriş ekranında textbox ve butonlar gibi görseller bulunmaz bunlar yeniden oluşturulur. AAR kütüphanesinde ise bunlar vardır projelerde doğrudan bu görseller eklenip kullanılabilir.

2.5. Android Projesine Kütüphane Ekleme

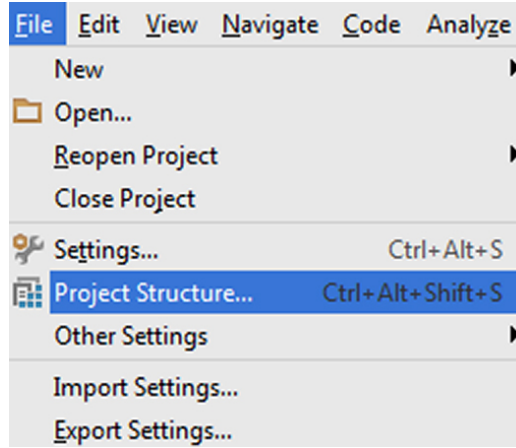
Android bir projeye kütüphane eklemenin iki yöntemi vardır: Gradle inşa sisteminin kullandığı **build.gradle** dosyasına doğrudan kütüphanenin eşsiz bir kimliği (artifactID) yazılarak yapılabilir veya projenin **Dependency** yani bağımlılık ekleme sihirbazı kullanılarak bu yazma işlemini IDE'nin yapması sağlanabilir. İki yöntemde de çok yaygın resim kütüphanesi olan Picasso kütüphanesinin kurulumu anlatılmıştır.

- **1.Yöntem:** Bu yöntemde eklenmek istenen kütüphanenin eşsiz bir kimliği (artifactID), Gradle inşa sisteminin kullandığı build.gradle dosyasına el ile yazılarak yapılabilir.



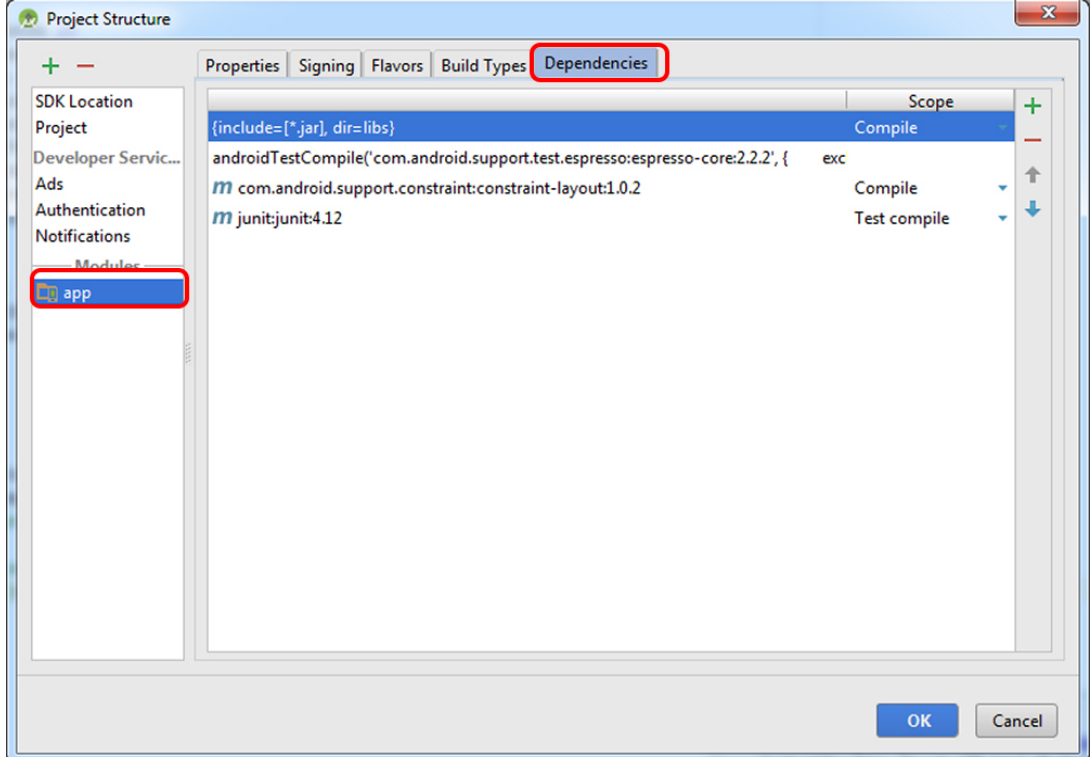
Fotoğraf 2.16: Kodla projeye kütüphane ekleme

- **2.Yöntem:** Bu yöntemde editörün menü yönergeleri kullanılır. Sırası ile **File > Project Structure** yolu takip edilir.



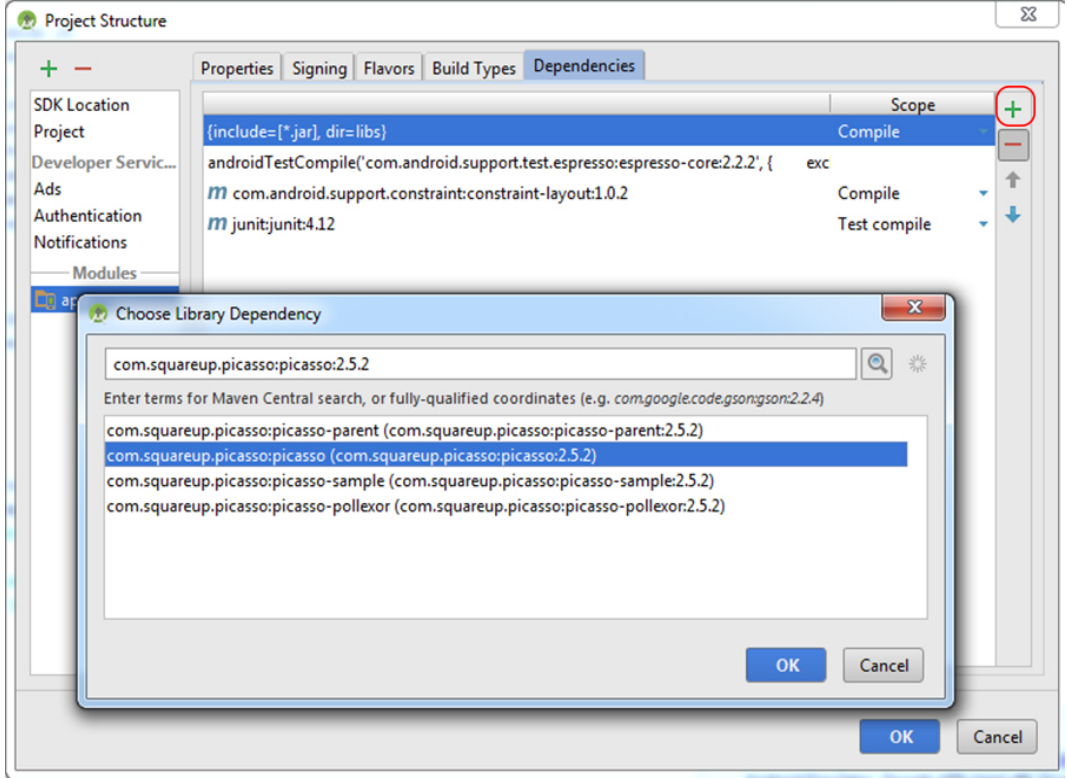
Fotoğraf 2.17: Sihirbaz ile projeye kütüphane ekleme

Açılan pencereden projedeki modül ismine (örnekte "Geleceği Yazanlar") ve ardından da **Dependencies** sekmesi tıklanır.



Fotoğraf 2.18: Sihirbaz ile projeye kütüphane ekleme ekranı

Hemen sağ üstte görülen + düğmesine basılıp Library Dependency ögesi seçilir. Çıkan pencerede varsayılan olarak en çok kullanılan kütüphanelerin listesi yer alır. Picasso Android SDK'yı eklemek için üstteki kutucuğa "picasso" yazılıp arama düğmesine basılır.



Fotoğraf 2.19: Eklenecek kütüphanenin aratılması ekranı

OK düğmesine basıldıktan sonra Picasso Android SDK'sı projenin build. gradle dosyasına eklenecektir. Bir diğer deyişle proje, Gradle sayesinde Picasso Android SDK'sını indirecek ve kullanıma hazır hâle getirecektir.

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<ul style="list-style-type: none">➤ İş sağlığı ve güvenliği tedbirlerini alınız.➤ Yeni bir proje oluşturunuz.	<ul style="list-style-type: none">➤ Proje Package name isminin eşsiz olmasına dikkat etmelisiniz.➤ Türkçe karakter kullanmamalısınız.
<ul style="list-style-type: none">➤ Uygulamanız için hedef cihaz seçimini yapınız.	<ul style="list-style-type: none">➤ Cihaz seçiminde tablet-phone seçeneğini seçmelisiniz.➤ Hangi android sürümünde çalışacağınıza dikkat etmelisiniz.
<ul style="list-style-type: none">➤ Activity seçiniz.	<ul style="list-style-type: none">➤ Emptyactivity seçerek devam etmelisiniz.➤ Acitivity ve layout isimlendirme yaparken Türkçe karakter kullanmamaya dikkat etmelisiniz.
<ul style="list-style-type: none">➤ Emülatör oluşturunuz.	<ul style="list-style-type: none">➤ Emülatör oluştururken bilgisayarınızın donanımını dikkate alarak emülatör işlemini oluşturmamalısınız.➤ Emülatörün kullanacağı işletim sistemi ile projede kullandığınız işletim sisteminin uyumlu olmasına dikkat etmelisiniz.
<ul style="list-style-type: none">➤ Emülatör açılış ayarlarını yapınız.	<ul style="list-style-type: none">➤ Yazdığınız programın çözünürlüğüne dikkat ederek emülatör açılış ekranı ayarlarını yapmalısınız.
<ul style="list-style-type: none">➤ Emülatörü test ediniz.	<ul style="list-style-type: none">➤ Emülatörü çalıştırdığınızda “Hello World” yazısının çıktığına dikkat etmelisiniz. (bk resim 2.13)
<ul style="list-style-type: none">➤ Projeye kütüphane ekleyiniz.	<ul style="list-style-type: none">➤ Projeye kütüphane ekleme yöntemlerinden birini seçerek kütüphane ekleme işlemini yapmalısınız.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki cümleleri dikkatle okuyarak boş bırakılan yerlere doğru sözcüğü yazınız.

1.'yı android marketteki unique id'i gibi düşünülebilir.
2. Kullanıcının gördüğü sayfayı oluşturan java sınıflarına (class).....denir.
3., bilgisayar üzerinden çalışan sanal bir android cihaz demektir.
4. Projede kullanılan resim, video, string, menü, layout yani kullanıcı tasarım öğelerini içinde barındıran klasördır.
5., ölçü, string ve stil gibi dosyaları tutar.
6.kullanılarak bir programlama diliyle yapılabilecekler artırılabilir ve işlemler kolaylaştırılabilir.

DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme” ye geçiniz.

KONTROL LİSTESİ

Aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. İş sağlığı ve güvenliği tedbirlerini aldınız mı?		
2. Yeni bir proje oluşturduunuz mu?		
3. Uygulama geliştireceğiniz cihaz platformunu seçebildiniz mi?		
4. Uygumanızın tasarımı için gerekli activity seçimini yapabildiniz mi?		
5. Bilgisayarınızın donanımına uygun emülatör kurulum ayarlarını yapabildiniz mi?		
6. Uygulamanızın emülatör ortamında test işlemini yapabildiniz mi?		
7. Android kütüphane yapısını anladınız mı?		
8. Projenize kütüphane ekleme işlemini yapabildiniz mi?		

DEĞERLENDİRME

Değerlendirme sonunda **Hayır** şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız **Evet** ise bir sonraki öğrenme faaliyetine geçiniz.

MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatle okuyarak doğru seçeneği işaretleyiniz.

1. Aşağıdakilerden hangisi mobil işletim sistemlerinden biri değildir?
A) Android
B) Symbian
C) Windows 7
D) iOS
E) BADA
2. iOS için geliştirilen uygulamalarda kullanılan programlama dili aşağıdakilerden hangisidir?
A) Java
B) HTML/CSS
C) C#
D) Objective C
E) C++
3. Java geliştirme paketinin içinde aşağıdakilerden hangisi bulunmaz?
A) Java Çalışma Ortamı (JRE)
B) Java Sanal Makine (JVM)
C) Java API'ler
D) Java Aplet
E) NetBeans
4. Aşağıdakilerden hangisi android bir projenin temel bileşenlerinden biri değildir?
A) Manifest
B) Res
C) Layout
D) Mainvoid
E) GradleScripts
5. Project structure penceresinde projemize kütüphane ekleme menüsü aşağıdakilerden hangisidir?
A) Properties
B) Dependencies
C) Signing
D) Flavors
E) BuildTypes

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise D, yanlış ise Y yazınız.

6. () Android projelere eklenebilecek iki tip kütüphane tipi vardır.
7. () JAR Kütüphanesi Layout, Drawables gibi görsel kaynakları içerir.
8. () GradleScripts projenin derlenmesi ve APK'sının oluşmasını sağlayan yapıdır.
9. () Projede kullanılan görsel dosyalar Manifest klasöründe tutulur.
10. () AndroidManifest.xml dosyasının adı değiştirilmez.
11. () Native dille yazılmış bir program tüm mobil işletim sistemlerinde çalışır.
12. () Her projenin bir activity ve bir layout'a ihtiyacı vardır.
13. () Cross mobil uygulama platformunda HTML, CSS yapısı kullanılmaktadır.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ 1'İN CEVAP ANAHTARI

1	C
2	E
3	A
4	D

ÖĞRENME FAALİYETİ 2'İN CEVAP ANAHTARI

1	Package name
2	Activity
3	Emülatör
4	Res
5	Values

MODÜL DEĞERLENDİRMENİN CEVAP ANAHTARI

1	C
2	D
3	E
4	D
5	B
6	DOĞRU
7	YANLIŞ
8	DOĞRU
9	YANLIŞ
10	DOĞRU
11	YANLIŞ
12	DOĞRU
13	YANLIŞ

KAYNAKÇA

- <https://developer.android.com/studio/projects/android-library.html>
- TUFAN, Mustafa; BAYKARA Muhammet; GÜLER Zafer; AVCI Engin; **Açık Kaynak Mobil İşletim Sistemi** (<http://web.firat.edu.tr/feeb/kitap/C12/89.pdf>)