Android ile Uygulama Geliştirme



Sürüm: 1.0

volkangezer@gmail.com

http://volkangezer.co.cc

İçindekiler

Android4
Android Yerli Uygulamaları5
Android SDK Özellikleri6
Dalvik6
Kurulum7
Sistem Gereksinimleri7
Gerekli Dosyalar7
Kurulum7
Android SDK neler içerir?18
Başlarken
İlk Açılış
Eclipse
İlk Uygulama22
Geliştirme
Giriş31
Proje Yönetimi32
Android Projeleri32
AndroidManifest.xml
Android'e Isınma
Uygulama Hayat Döngüsü52
Aktiviteler
Ek: Bazı Kısayollar54

Android

Cep telefonlarının GPS, ivmeölçer, wifi ve dokunmatik ekran özelliklerini bir arada bulundurması sebebiyle yaratıcı cep telefonu uygulamaları oluşturmak geliştiricilerin ilgisini çekti. Yine bu telefonlar, kullanıcılar tarafından daha çok ilgi görmeye başladı. Varolan cep telefonu işletim sistemlerinin üçüncü parti yazılımları kısıtlamasıyla da, Android'in açık kaynaklı ve %100 özelleştirilebilir olması birçok geliştiricinin ilgisini çekti.

Açık kaynak çalışma ortamının yanında, güçlü SDK (Software Development Kit – Yazılım Geliştirme Seti) kütüphaneleri sunmasıyla birlikte, kendisini daha önce cep telefonu üzerinde çalışmamış geliştiricilere açtı. Bununla birlikte programlamada uzman veya yeni başlayan olsanız da yaratıcı cep telefonu uygulamaları yazma fırsatı size sunuluyor.

Daha önce Symbian tarzı işletim sistemleri ile de uygulama geliştirmek ve cihazın donanımına erişmem mümkündü ancak bu çok karmaşık C/C++ kodu ile kullanımı zor olan API'ların kullanımını gerektiriyordu. Kodların her telefonun donanımına göre değiştirilmesi gerekiyordu.

Uygulama geliştirmede en büyük gelişim Java'ya sahip MIDlet'ler oldu. Java sanal makinesinde (Java VM) çalışan MIDlet'ler donanımı soyutlayarak, birçok donanımda çalışabilen uygulamalar oluşturulmasına izin verdi.

Günümüzdeki Windows Mobile ve Apple iOS, daha basit ve zengin geliştirme ortamı sunuyor. Ancak Android'den farklı olarak, kendi uygulamalarını üçüncü parti uygulamalardan önce tuttuğu için uygulamalar ve telefon verileri arasındaki iletişimi kısıtlıyor. Android'de ise tüm donanımların erişimlerine API'lar ile izin veriliyor. Telefonla gelen ve sonradan üretilen uygulamalar aynı seviyede ve aynı çalışma zamanında bulunuyor. Bu nedenle herhangi bir uygulamayı değiştirmek, yani tamamen özelleştirmek mümkün. Android Linux 2.6 çekirdeği üzerinde çalışmaktadır. Google'ın desteklemesinden dolayı gelen sorularda Gphone üretip üretmeyeceklerine yanıtı bu adreste bulunmaktadır: <u>http://googleblog.blogspot.com/2007/11/wheres-my-</u> <u>gphone.html</u>. Bu adrese göre Android, cep telefonu yaratıcılığını önleyen engellere takılmadan bir telefonda uygulamaların çalıştırılabildiği ilk açık platformdur.

Açık kaynaklı bir platform olması, sizin de açık kaynaklı ve ücretsiz yazılımlar sunmanızı gerektirmiyor.

Not: Android uygulamaları Java ile geliştirilmesine rağmen, Android telefonlarında Java ME (Mobile Edition) sınıfları ve uygulamaları çalışmayacaktır.

Android Yerli Uygulamaları

Android aşağıdaki yerli uygulamalarla gelir ancak telefona göre değişiklik gösterebilir ve sürümüne göre uygulama isimleri değişebilir veya yeni uygulamalar eklenebilir:

- Gmail destekli e-posta istemcisi, ancak diğer postaları da destekler.
- SMS yöneticisi
- Takvim ve kişi listesi yöneticisi (Google çevrim içi hizmetleri ile tümleşik çalışır)
- Google Haritalar uygulaması. Sokak görünümü, uydu görünümü ve rehber özellikli.
- WebKit (Opera ve Chrome gibi tarayıcılarda da kullanılan, sayfa derlemesini sağlayan motor) tabanlı İnternet Tarayıcısı
- Anında Mesajlaşma istemcisi
- Müzik oynatıcı ve resim gösterici
- Android uygulamalarını indirebileceğiniz Market (Artık Google Play)

Tüm yerli uygulamalar da Android SDK ile yazılmıştır ve Dalvik'te (Sonraki bölümlerde anlatılacak) çalışır. Buna ek olarak bu uygulamaların bilgilerine üçüncü parti yazılımlar erişebilir.

Bu varsayılan uygulamalara telefonu satın aldığınız sağlayıcı ve/veya donanım üreticileri değişiklik yapabilir.

Android SDK Özellikleri

Aşağıdaki özellikler tam bir liste değildir ancak SDK'nın verdiği olanaklar için iyi bir başlangıç olabilir:

- Wi-Fi erişimi
- Konum tabanlı servislere erişim (GPS)
- Tam ortam donanım kontrolü (Kamera, mikrofon).
- İvmeölçer ve pusula API'ı
- IPC (Inter-process communication Çoklu işlem haberleşmesi) mesaj geçişi
- Paylaşımlı veri depolaması
- Tümleşik internet tarayıcısı
- Google Talk ile P2P desteği
- 2D grafik kütüphanesi ve OpenGL ES kullanılarak 3D grafik desteği
- Ses/görüntü oynatmak veya kaydetmek için kütüphaneler
- Yerli uygulamaların değiştirilmesini sağlayan çalışma çerçevesi
- Uygulamaların hizmet olarak çalışabilmesi

Dalvik

Android uygulamalarının Java ile yazıldığını daha önceden söylemiştim. Ancak Android uygulamaları, Java uygulamalarından farklı olarak Java VM'de değil Dalvik VM'de çalışıyor. Bu şekilde Android cihazı üzerinde birden fazla uygulama daha etkili çalışıyor.

Her uygulamanın kendine ait ayrı bir Dalvik örneği bulunuyor. Çalışan tüm uygulamalar bellek ve işlem yönetimini, kaynakları ihtiyaca göre yöneten Android çalışma zamanına bırakıyor. Dalvik VM, Dalvik Executable (Uygulama) dosyalarını çalıştırır. Bu dosyalar, en düşük bellek kalıntısı bırakmak için iyileştirilmiş dosyalardır. .dex dosyaları, SDK tarafından gelen araçlarla oluşturulur.

Android, bu şekilde bellek yönetimini yaparken çalışmayan veya kullanılmayan uygulamaları otomatik olarak durdurur.

Kurulum

Sistem Gereksinimleri

Gerekli Dosyalar

Kurulum için indirilmesi gerekli dosyalar ve adresleri şunlardır:

- Android SDK developer.android.com/sdk/index.html
- Eclipse eclipse.org
- JDK java.oracle.com

Kurulum

Kurulum için aşağıdaki resimlerdeki kırmızı çerçeveli alanları izleyin:



Resim 1 – Android SDK İndirme Sayfası



Resim 2 – Eclipse İndirme Sayfası



Resim 3 – Eclipse Sürüm Seçimi



...or pick a mirror site below.

Resim 4 - Yansı site seçimi (Size en yakın olan otomatik seçilecektir)

DRACLE		(Sign In	/Register for /	Account Help)	United States	s 💌 Communiti	es▼ lama.	1. v . 1
roducts and Services	Solutions	Downloads	Store	Support	Training	Partners	About	
r <mark>acle</mark> Technology Network	< > Java							



Resim 5 - Java Web Sayfası

anced & Suite	<u>Latest Release</u>	<u>Next Release (Early Acce</u>	ess) Embedded Use	Previous Releases
ed				
	Java	JavaFx	⊗ NetBeans	Java EE
	Download ±	Download ±	Download ±	Download ±
	Java Platform (JDK) 7u3	JavaFX 2.0.3	JDK 7u3 + NetBeans	JDK 7u3 + Java EE
ie	Here are the Java S	e downloads in deta	ail:	



.ooking for the JavaFX 2.0 SDK? The JavaFX SDK 2.0 is now included in JDK 7u2 for Windows. For the JavaFX 2.0 Developer preview on Mac, go <u>here</u> .						
Java SE Development Kit 7u3						
You must accept the <u>Oracl</u> software. O Accep	le Binary Code Licens of License Agreemer	se Agreement for Java SE to down t ① Decline License Agreemen	load this t			
Product / File Description	File Size	Download				
Linux x86 (32-bit)	63.65 MB	👱 jdk-7u3-linux-i586.rpm				
Linux x86 (32-bit)	78.66 MB	🛓 jdk-7u3-linux-i586.tar.gz				
Linux x64 (64-bit)	64.53 MB	👱 jdk-7u3-linux-x64.rpm				
Linux x64 (64-bit)	77.3 MB	👱 jdk-7u3-linux-x64.tar.qz				
Solaris x86 (32-bit)	135.96 MB	🛓 jdk-7u3-solaris-i586.tar.Z				
Solaris x86 (32-bit)	81.4 MB	🛓 jdk-7u3-solaris-i586.tar.gz				
Solaris SPARC (32-bit)	138.92 MB	🛓 jdk-7u3-solaris-sparc.tar.Z				
Solaris SPARC (32-bit)	86.07 MB	🛓 jdk-7u3-solaris-sparc.tar.gz				
Solaris SPARC (64-bit)	16.14 MB	🛓 jdk-7u3-solaris-sparcv9.tar.Z				
Solaris SPARC (64-bit)	12.31 MB	🛓 jdk-7u3-solaris-sparcv9.tar.gz				
Solaris x64 (64-bit)	14.46 MB	🛓 jdk-7u3-solaris-x64.tar.Z				
Solaris x64 (64-bit)	9.25 MB	🛓 jdk-7u3-solaris-x64.tar.gz				
Windows x86 (32-bit)	84.12 MB	idk-7u3-windows-i586.exe				
Windows x64 (64-bit)	87.41 MB	idk-7u3-windows-x64.exe				

Resim 7 - Lisans Sözleşmesi ve İşletim Sistemi Seçimi

Resim 1 – Resim 7 arasındaki işlemleri yaptıysanız şu dosyaları indirmiş olmalısınız (Sürüm numaraları ve 32/64 bit sürümleri farklı olabilir):



Resim 8 – İndirilen Dosyalar

Kurulum sırası olarak şunu tercih edin:

1. JDK Kurulumu:





2. Android SDK Kurulumu



Resim 10 – Android SDK Kurulum Ekranı

Eğer JDK'yı ilk olarak kurmazsanız, aşağıdaki adımdan sonra ilerleyemezsiniz.

Android SDK Tools Setup	
Java SE Development Kit Detect whether Java SE Development Kit is installed.	C C C C C C C C C C C C C C C C C C C
Android SDK relies on the Java SE Development Kit (JDK). Java SE Development Kit (JDK) version 1.7 has been found. Location: C:\WINDOWS\system32\java.exe	
Nullsoft Install System v08-Mar-2012.cvs ————————————————————————————————————	Cancel

Resim 11 – JDK Bulunma Ekranı

Kurulum bittikten sonra *Start SDK Manager* tıklayıp sistem resimlerini indirmeyi sağlayacak uygulamayı açın.



Resim 12 - Android SDK Kurulum Sonu

Otomatik olarak en son sürüm seçili gelecektir. Ancak tercihinize göre cihazların %97'sini destekleyen 2.1 sürümünü de indirmeniz önerilir. Hangi sürümleri indirirseniz aşağıda anlatılacak sanal cihaz oluşturulmasında o sürüm Android sistemleri oluşturabilirsiniz. Oluşturduğunuz uygulamaları mümkün olduğunca düşük sürümde tutmak daha fazla kullanıcı tarafından kullanılmasını sağlayacaktır.

🖡 Android SDK Manager					×
Packages Tools					
SDK Path: C:\Program Files\Android\android-sdk					
Packages					
. Name	ADT	Desi	Cabu		2
	API	Nev.	Julius		
1005 Andraid SDK Teals		17	Testallad		
Android SOK Tools		17	Afat installed		
Android Stat Matronin-tools		11	 Noc inscalled 		
Commentation for forded 50%	15	2	Alat installed		
Contraction for Android Stat	15	2	Alat installed		
Succession for SDV	15	3	 Noc inscalled Noc inscalled 		
✓ () Samples for SUM	15	2 1	 Not installed 		
ARTY EADI V/a System image	15	1	 Not installed 		
Courses for Android SDK	15		 Not installed 		
P Sources for Android Sol	15	1	 AUX #ISTORED 		
Android 3.2 (API 14)					
Android 3.1 (API 13)					
Android 3.1 (APT 12)					
Android 3.3 (APT11)					
Android 2.3.3 (APT 10)					
H Android 2.1 (API 7)					
Android 2.1 (AP17)					
Android 1.5 (AP1 4)					-
E Estrar					
Andraid Support		7	Aint installed		
- Gaade Addah Ade SDK		4	Aint installed		
Google Analytics SDK		2	Aint installed		
Google Play APX Expansion Library		2	Aint installed		-
			- not and allo		
Show: Updates/New Installed Obsolete Select New	w or <u>Updates</u>			Install 11 packages	
Sort by: API level C Repository Deselect #	All		(Delete packages	
Done loading packages.					

Resim 13 – Android SDK Yöneticisi Paket Seçim Ekranı

3. Eclipse SDK kurulumu, indirdiğiniz sıkıştırılmış dosyaya sağ tıklayıp Extract All

(Tümünü Ayıkla) seçerek yapılır.



Kurulum tamamlandıktan sonraki Eclipse'i ilk başlattığınızda Resim 14 ile karşılaşacaksınız. Resimdeki kutuyu işaretlerseniz bir daha sorulmayacaktır.

🖨 Workspa	ice Launcher	\mathbf{X}
Select a w	orkspace	
Eclipse SDK s Choose a wo	stores your projects in a folder called a workspace. rkspace folder to use for this session.	
Workspace:	C:\Documents and Settings\VGezer\workspace	Browse
Use this a	s the default and do not ask again	
		OK Cancel

Resim 14 – Eclipse Çalışma Dizini Seçimi

Ardından Eclipse ana ekranı görülecektir:



Resim 15 - Eclipse ana ekranı

Geliştirmeye başlamadan önce Eclipse için hazırlanmış *ADT (Android Developer Tools – Android Geliştirme Araçları)* eklentisini yüklemeniz gerekir. Bunun için aşağıdaki adımları izleyin:



Resim 16 - ADT Kurulumu

2	or select a	site		1	Add
Í	or poloce a	Find more sof	tware by working with the "Av	vailable Software S	<u>Sites"</u> preferences
_					
			Versio	n	
i	🖨 🖊 🖨	Repository	2		
	<u>N</u> ame:	Android DT Eklentisi	2	L <u>o</u> cal	
	Location:	https://dl-ssl.google.com	n/android/eclipse/	<u>A</u> rchive	
				·	
	?		3 ок	Cancel	
4					
at	est version:	s of available software	Hide items that are a	lready installed	

Resim 17 – ADT İndirme Dizini Ekleme

Ardından karşınıza gelecek seçeneklerden *Developer Tools (Geliştirme Araçları)* işaretleyin ve sonraki adıma geçin.

Name	Version
💻 🔽 🎹 Developer Tools	
🔽 ጭ Android DDMS	17.0.0.v201203161636-291853
🔽 ጭ Android Development Tools	17.0.0.v201203161636-291853
🔽 ጭ Android Hierarchy Viewer	17.0.0.v201203161636-291853
🔽 ጭ Android Traceview	17.0.0.v201203161636-291853

Resim 18 – İndirilecek Geliştirme Araçları

İndirme işleminden sonra Eclipse'in yeniden başlatılması istenecektir. Yeniden başlattıktan sonra **İlk Açılış** bölümünden devam edin.

Android SDK neler içerir?

- Android API'ları. Android yığınına erişim için gerekli API'lar.
- Developer Tools (Geliştirme Araçları). Android kodunuzu derlemek ve test etmek için gerekli araçlar.
- Android Emulator (Android Öykünücü). Android uygulamalarınızın gerçek cihazda nasıl çalışacağını görmenizi sağlayan sanal Android cihazı oluşturmanızı sağlar.
- Belgelendirme. Android geliştirme için gerekli tüm resmi belgelendirmeyi içerir. Bu belge, en son sürümün çevrim dışı kopyasıdır.
- Örnek kodlar. En basit uygulamadan karmaşık uygulamalara kadar, Android'i nasıl kullanılabileceğinizi görmenizi sağlayan örnek uygulamalar içerir.

Başlarken

Eğer daha önce cep telefonu veya mikro denetleyiciler üzerinde geliştirme yapmadıysanız, uygulamalarınızı geliştirirken kısıtlı donanım, hız ve hafızayı göz önünde bulunarak geliştirmeler yapmalısınız.

İlk Açılış

Kurulum tamamlandıktan sonraki ilk yeniden başlatmada Resim 19 ile karşılaşacaksınız.

🖨 Welcome to Android Development	
Welcome to Android Development Configure SDK	0
To develop for Android, you need an Android SDK, and at least one version of the Android APIs to compile may also want additional versions of Android to test with.	e against. You
O Install new SDK	
Install the latest available version of Android APIs (supports all the latest features)	
Install Android 2.1, a version which is supported by ~97% phones and tablets	
(You can add additional platforms using the SDK Manager.)	
Target Location: \\cern.ch\dfs\Users\v\vgezer\android-sdks	Browse
⊙ Use existing SDKs	
Existing Location: C:\Program Files\Android\android-sdk	Browse
Rext > Finish	Cancel

Resim 19 – Android Developer Tools (Geliştirme Araçları) yapılandırma ekranı

Eğer Android SDK Manager (SDK Yöneticisi) ile indirme işlemini tamamladıysanız, Use existing SDKs (Mevcut SDK'ları kullan) seçeneğini seçin ve İleri tıklayın. Eğer herhangi bir indirme işlemi yapmadıysanız Install new SDK (Yeni SDK kur) seçip altındaki iki seçeneği de işaretleyin. Bu şekilde en son API'a ve telefon ile tabletlerin %97'sini destekleyen 2.1 sürümünü indirmiş olacaksınız. Not: Bu ekran ile karşılaşmazsanız Eclipse sürümünüz otomatik olarak yapılandırılmış olabilir. Kontrol edebilmek için Window -> Preferences tıklayıp Android sayfasına gelin ve yüklü dizini doğrulayın.

Bu adımları tamamladığınızda, Eclipse uygulamanız Android geliştirmeye hazır duruma gelecektir (Resim 20).

Java - Eclipse SDK	Search Duringto Dury Mindaus Hala					
	l≗ J ₄ ° G° ॐ ▼ O ▼ O ▼ ∞ ∉	© • 😕 🔗 • 🛓		⇒ - ≥		
			Quick Acce	ess	🖹 🕌 Java 🚳	DDMS
I Package Explorer 💥 🖓 🗖 🗖					Outline 🔀	
- -				An	outline is not availa	ble.
	R Problems 💥 @ Javadoc 🔞 Declar	ation			\bigtriangledown	
	0 items					
	Description	Resource	Path	Location	Туре	

Resim 20 – Varsayılan Eclipse Açılış Ekranı

Eclipse

Eclipse, birçok kısmı Java ile yazılmış, genişletilebilir eklenti sistemine sahip bir yazılım geliştirme ortamıdır. Eklentiler sayesinde birçok programlama dilinde kodlama yapılmasına olanak sağlar.

Ücretsiz, açık kaynak kodlu olması ve kolay kullanılabilir arayüzü ve Eclipse için yazılmış Android Geliştirme Araçları içermesi sebebiyle, Android'e yeni başlayanlar için önerilmektedir.

Kurulum sırasında yüklediğiniz ADT Eklentisi ile Eclipse'inize Android dilini geliştirilebilme yetisini verdiniz. Resim 21, ADT Eklentisi kurulmuş Eclipse ana ekranını göstermekte. Bu görüntüde 1 numaralı alan menüyü, 2 numaralı alan araç çubuğunu, 3 numaralı alanlar görünümleri ifade etmektedir. Seçili görünümün diğer görünümlerden farklı renge sahip olduğuna dikkat edin. 4 numaralı alan da yerleşimler (perspective) arasında hızlı geçiş yapmanızı sağlar. İleride bunu kullanmanız faydalı olabilir.

Araç çubuğu menülerde bulunup, sıkça kullanmanız gereken komutlar için simgeler içerir. Sağ tıklayıp *Customize Perspective (Yerleşimi Özelleştir)* tıklayarak yeni komutlar ekleyebilir veya kaldırabilirsiniz.



Resim 21 – Eclipse Ekranı

Editor (düzenleyici) ise Eclipse içerisinde düzenlenebilecek dosyalar açıldığında gösterilir

(Resim 22). Eğer tanımlı bir dosya türü değilse varsayılan harici düzenleyici açılır.



Resim 22 – Eclipse Düzenleyici

Yukarıdaki resimde dosya adının yanındaki yıldız (*) işareti, dosyada kaydedilmemiş değişiklikler olduğunu gösterir. Eğer kapatmadıysanız yazının solundaki renkler de diskte kayıtlı olan sürümle şimdiki sürüm arasındaki farklılıkları gösterir.

Eclipse yerleşimi (perspective) tamamen özelleştirilebilir. Yeni görünümler ekleyebilir, herhangi bir görünüm veya editörü tutarak istediğiniz yere sürükleyebilir, kullanmayacağınız öğeleri gizleyebilir veya yer açmak için simge durumuna küçültebilirsiniz. Dilerseniz *Window (Pencere)* menüsünden yerleşimi kaydedebilirsiniz.

Eğer satır numaralarını görmek istiyorsanız, bir düzenleyiciyi açtıktan sonra, yazının solundaki alana sağ tıklayın ve *Show Line Numbers (Satır Numaralarını Göster)* seçin.

İlk Uygulama

Her programlama dilinde olduğu gibi Android üzerinde de ilk uygulamamız bir "Hello World" (Merhaba Dünya) uygulaması olacak. Bunun için aşağıdaki adımları izleyin:

File (Dosya) – New (Yeni) – Project (Proje) komutunu seçtiğinizde karşınıza yeni proje oluşturma penceresi gelecektir (Resim 23). Bu pencereden *Android – Android Application Project (Android Uygulama Projesi)* seçin ve *İleri* tıklayın.

Resim 24'te görülen *Project Name (Proje ismi)* olarak Türkçe ve özel karakterler içermeyen (tercihen ilk harfi büyük) bir isim girin. Şu an yapacağımız proje için *IlkProje* girin.

Not: Java'da proje isimlerinin ilk harfleri genellikle büyük harfle başlar ve her farklı kelimenin de ilk harfi büyük olur.

Uygulamanızın asıl hedeflediği Android sürümünü seçin. Seçiminizi yaparken Android API'larının geri uyumlu olduğunu (diğer bir deyişle, uygulamalarınızın ileri sürüm uyumlu olduğunu) göz önüne alın. Örneğin 2.3 seçerek derlediğiniz bir uygulama, özel bir durum olmadığı sürece 4.0 sürümünde de çalışacak; ancak 4.0 için derlediğiniz bir uygulama bu API içerisindeki özel fonksiyonları içeriyorsa 2.3 sürümünde çalışmayacaktır. Bu nedenle uygulamanız için mümkün olduğunda düşük API'larda çalışmanız önerilir.

Minimum SDK, uygulamayı çalıştıracak en düşük API seviyesini belirtir. Bir önceki adımda seçilen asıl derleme sürümü ne ise aynı olması olası sorunları önleyecektir.

Burada *Application Name* kullanıcılar tarafından görülecek uygulama adını belirtir. Karakter türü kısıtlaması yoktur. Buradaki *Package Name (Paket Adı)* ise uygulamanızın namespace'idir. Bu alan yegane olmalıdır. Tüm kaynak kodlarınızın bulunacağı paket ismini belirtir. Tüm uygulamalar bir paket ismine sahip olduğundan, anlamlı bir yapı ile uygulamalarınızın geliştirilmesi önerilir. "com.sirketadi.uygulamaadi" bu yapıya güzel bir örnektir. Noktalarla ayrılmış en az iki kelime girmeniz gerekir.

Create custom launcher icon (Özel başlatıcı simgesi oluştur) tıklarsanız bir sonraki adıma geçtiğinizde uygulama simgenizi oluşturmanızı sağlayacak bir sayfa ekrana gelecektir.

New Project		New Android App
Select a wizard Create an Android Application Project		New Android Application Image: Constraint of the state o
Wizards: type filter text		Application Name:0 IlkProje Project Name:0 IlkProje Package Name:0 com.vgezer.ilkproje Build SDK:0 Android 21 (AP17) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required SDK:0 AP17: Android 21 (Eclair) Iminimum Required S
(?) < Back Next > Einish	Cancel	< Back Next > Finish Cancel

Resim 23 – Yeni proje oluşturma penceresi Resim 24 – Yeni Android Projesi

Sonraki adım aktivite oluşturulup oluşturmak istemediğinizi belirten sayfadır (Resim 25). Bir Activity (Aktivite), çalışan ve işlemler yapan bir sınıf olarak tanımlanabilir. Eğer bir işlem yapılacaksa, Create Activity (Aktivite Oluştur) seçeneği işaretlenerek sonraki adıma geçildiğinde Activity sınıfının alt sınıfı olarak, Activity Name alanına yazılan aktivite oluşturulur (Buraya IlkProjeActivity girin). Activity Layout (Aktivite Yerleşimi), bu aktivite için oluşturulmasını istediğiniz yerleşimi belirtir (Bu alana main girin). Yerleşimlerle ilgili ayrıntılı bilgi ileride verilecektir.

Aktivite ekleme işlemi tercihe bağlıdır ancak neredeyse tüm uygulamalarda kullanılır. Aktivite ekleme işlemini daha sonra da yapabilirsiniz.

Yine aynı sayfada Navigation Type (Gösterim Türü) seçilebilir ancak bu seçtiğiniz SDK sürümüne göre desteklenmeyebilir.

New Android App				[
New Blank Activity Creates a new blank	activity, with optiona	al inner navigation.			0
Activity Name® Layout Name® Navigation Type® Hierarchical Parent® Title®	IlkProjeActivity activity_ilk_proje None IlkProjeActivity		•	(:
?		< <u>B</u> ack	Next >	Einish	Cancel

Resim 25 – Aktivite Bilgisi Ekranı

Son tıklanıldıktan sonra uygulama oluşturulur ve oluşturulan dosyalar *Package Explorer* (*Paket Gezgini*) penceresinde gruplandırılmış şekilde gösterilir (Resim 26).

Not: Son düğmesi etkin değilse, ilgili dosyalar yüklenmemiş olabilir. İleri tıklayarak eksik dosyaların yüklenmesi için *Install (Yükle)* tıklayın.



Resim 26 – Paket Gezgini penceresi

Uygulama dosyalarını veya kodlarını incelemeden önce çalıştırarak nasıl görüldüğüne bakmak için en basit yol sanal bir Android cihazı oluşturmaktır. Bunun için Eclipse içerisinde *Window (Pencere) –> AVD Manager (AVD (Android Sanal Aygıt) Yöneticisi)* veya Başlat menüsünden *Android SDK Tools –> AVD Manager* uygulamasını açın.

Açılan pencereden New tıklayın (Resim 27).

🚦 Android Virt	ual Device Manager			
List of existing Andr	roid Virtual Devices located at			
AVD Name	Target Name	Platform	API Level CPU/A	BI New
	No AVD available			Edit
				Delete
				Descio
				Details
				Start
		destant desta		Refresh
A valid Android Virtual Device. M A reparable Android Virtual Device.				
	uar bevice chac railed to load. Click I	Decails to see the error.		

Resim 27 – AVD Yöneticisi

Yeni sanal aygıt oluşturma ekranı açıldığında (Resim 28) bir isim girin ve içerisine kurulacak Android sürümünü seçin. Bir SD kart boyutu belirleyerek, oluşturulacak sahte harici hafızayı belirleyin.

Snapshot özelliğini etkinleştirirseniz, cihazın ilk açılıştan sonraki açılışları daha hızlı olacaktır.

Bir *Skin (Görünüm)* olarak sanal cihazın ekran boyutunu belirleyebilirsiniz. Android sistemi bilgisayarınızda olduğundan daha yavaş çalıştığından düşük bir boyut seçmeniz (HVGA gibi) önerilir.

Hardware (Donanım) alanından cihazın sahip olduğu donanımları belirleyebilirsiniz. *New* düğmesi ile yeni donanım varsayılanlarını değiştirebilirsiniz (Donanımların varsayılan değerleri için: <u>http://developer.android.com/guide/developing/devices/managing-</u> <u>avds.html#hardwareopts</u>. Herhangi bir ekleme yapmazsanız bu değerler temel alınacaktır). *Create AVD (AVD Oluştur)* tıklayarak *Console (Konsol)* penceresinde (Henüz değinilmedi, ancak Eclipse ana penceresinin en altındaki sekmede görebilirsiniz) yapılan ayarları görebilirsiniz.

Create	new Android Virtual Devi	ce (AVD)	×
Name:	Android-Gbread		
Target:	Android 2.3.3 - API Level 10		*
CPU/ABI:	ARM (armeabi)		~
SD Card:			
	⊙ Size: 50	Mi	В 💙
	O File:	Br	owse
Snapshot:			
Skin:			
		^ ^ L	
Hardware:	Property	Value	New
	Abstracted LCD density	160	Delete
	Device ram size	256	Delece
Override	the existing AVD with the same	name	
		Create AVD	Cancel

Resim 28 – Yeni AVD oluşturma ekranı

Son işlem olarak *Run* menüsünden *Run (Çalıştır)* tıklayın ve *Run As (Farklı çalıştır)* penceresinden *Android Application (Android Uygulaması)* seçeneğini seçip *Tamam* tıklayın (Resim 29).



Resim 29 – Farklı çalıştır penceresi

Bu işlemi her proje için sadece bir kez yapmanız gerekecektir. İlk çalıştırma işleminden sonra bu yapılandırılma kaydedilecektir. Uygulama derlenip telefona yüklenecek ve çalıştırılacaktır. Eğer uygulama görünmüyorsa tuş kilidini açmanız gerekebilir. Sonuç Resim 30 gibi olacaktır:



Resim 30 – İlk Uygulama Görüntüsü

Geri tuşunu kullanarak uygulamadan çıkabilir ve istediğiniz an tekrar uygulamayı *Uygulama Çekmecesinden* çalıştırabilirsiniz.

Sanal cihazı ve uygulamayı, geliştirme yaptığınız süre boyunca açık bırakabilirsiniz. Eğer aynı uygulama yüklenecekse, uygulamanız otomatik olarak kapatılıp tekrar açılır. Bu şekilde sonuçlar daha hızlı görülebilir.

Aygıtı kapatmak için doğrudan sağ üst köşedeki kapatma düğmesini kullanabilirsiniz. Eğer Snapshot seçeneğini etkinleştirmişseniz sonraki açılışta kaldığı yerden devam edecektir.

Bu programdaki kodları doğrudan incelemenin faydalı olmayacağı düşüncesiyle, sonraki bölümleri okumanızı öneriyorum.

Geliştirme

Giriş

Android geliştirme, kurulum aşamasında yüklediğiniz SDK (Software Development Kit – Yazılım Geliştirme Seti) araçları ile yapılmakta. En hızlı şekilde gerekli araçların çalıştırılması için Eclipse ve Eclipse için ADT kullanıldığını söylemiştim.

Ancak herhangi bir metin düzenleyici ile de tüm işlemleri gerçekleştirebilirsiniz. Bu durumda, komut satırını kullanarak gerekli komutları çalıştırmanız gerekecektir.

Android Geliştiricileri (developer.android.com) sitesinden alınan uygulama geliştirme aşamaları şu şekildedir:

• Kurulum

Bu aşamada geliştirme ortamını kuruyor ve Android geliştirme araçlarını yüklüyorsunuz. Yine Sanal Android Cihazı (AVD) kurulumu da bu aşamada yapılıyor.

• Geliştirme

Bu aşamada uygulamanızı oluşturuyor ve geliştiriyorsunuz.

• Hata Giderme ve Test Etme

Bu aşamada uygulamanızı bir .apk dosyası olarak derliyor, bu şekilde sanal veya gerçek bir Android cihazına yükleyerek çalıştırıyorsunuz.

Daha sonra bir hata giderme aracı ile hataları kontrol ediyor ve günlük kontrolü yapıyorsunuz.

• Dağıtım

Bu son aşamada uygulamanızı dağıtım için yapılandırıp, derliyor ve test ediyorsunuz. Bu işlemden sonra kullanıcılara dağıtım veya satım işlemi yapıyorsunuz.

Proje Yönetimi

Proje, kod ve kaynak dosyalarınızı içeren bir sandık olarak düşünülebilir. SDK aracı, projenizin başarılı bir şekilde yönetimi ve derlenmesi için belirli bir yapıda olmasını gerektirir. Bu nedenle Eclipse ve ADT ile veya komut satırı komutları ile oluşturmanız önerilir. Bu dokümanda sadece Eclipse ile proje yönetimi anlatılacaktır.

Android için üç farklı proje türü bulunuyor:

Android Projects (Android Projeleri): Projenizin kaynak kodunu ve gerekli dosyaları içerip .apk dosyasına dönüştürülen ana proje türüdür. Projeler cihaza yüklenerek test edilir.

Test Projects (Test Projeleri): Bu projeler uygulamanızı test etmeniz için gereken kodları içerir ve cihazda çalışan uygulamalara yüklenirler.

Library Projects (Kütüphane Projeleri): Bu projeler cihaza yüklenemeyen, ancak kodunuzun tekrar kullanılmasını istediğinizde kullanılan projelerdir. Bu dosyalar uygulama derlenirken .apk dosyasına gömülürler.

Android Projeleri

Bu projeler, cihaza yüklenip çalıştırılabilen .apk dosyası oluşturulan projelerdir. Paket Gezgininde aşağıdaki klasörler otomatik veya gerektiğinde geliştirici tarafından oluşturulur:

Src: içerisinde oluşturduğunuz paket isimlerini içeren alt klasörleri barındıran ve tüm aktivite ile sınıf dosyalarınızı içeren klasördür.

Bin: Derlemeden sonra .apk ve diğer çıktı dosyalarını bulabileceğiniz klasör.

Gen: ADT tarafından oluşturulan Java dosyalarını içerir.

Android x.x.x: Android'in sahip olduğu hazır fonksiyon, metot ve sınıfları içerir.

Android Dependencies: Android'in proje derlerken gerekli olan dosyalarını içerir.

Res: Uygulama kaynaklarını içeren klasördür. Bu kaynaklar resim, ses, metinler vb. olabilir.

Anim: Animasyonları içerir.

Color: Renkleri tanımlayan XML dosyalarını içerir.

Drawable-Xdpi: Resim dosyalarını veya çizilebilir şekillerin XML dosyalarını içerir. Seçilen API'ya bağlı olarak "X", h (high), m (medium) ve l (low) olarak klasörler halinde bulunur ve resimleriniz çözünürlüğe göre sınıflandırılmalıdır.

Layout: Uygulamanızın görünümünü belirleyen XML dosyalarını içerir.

Menu: Uygulama menülerini içeren XML dosyalarını içerir.

Raw: MP3, OGG gibi ses dosyalarını veya diğer dosyaları koyabileceğiniz klasördür.

Values: Uygulamanın kullandığı metinler, diğer dil çevirileri gibi XML dosyaları konulabilen klasördür. Diğer klasörlerdeki XML dosyalarından farklı olarak buraya konulan XML dosyalarındaki elementler, isimlerine göre çağrılır.

AndroidManifest.XML: Uygulama, içerisindeki sınıflar ve diğer bileşenlerin nasıl davranacağını belirleyen kontrol dosyasıdır. Aktiviteler, cihazda kullanılması istenen özellikler ve izinler, uygulamanın çalışabilmesi için gereken API sürümü vb. gibi birçok nitelik bu dosyada belirtilir. Ayrıca dağıtım işlemi yapılırken gerekli olan uygulamanın imzalanması da buradan yapılabilir (Ayrıntılı bilgi için bkz. **AndroidManifest.xml**).

Project.properties: Bu dosya projenin özelliklerini içerir. Paket Gezgininde, projenin adının yer aldığı üst klasörde sağ tıklayarak *Properties (Özellikler)* seçerseniz, görsel olarak düzenleyebilirsiniz.

Bir Android projesi bileşenleri nelerdir?

Altı farklı bileşen bulunmaktadır ancak uygulamanızın bunların tamamını bulundurması gerekmez.

- Aktiviteler (Activity). Uygulamanızın görünen her ekranı Activity sınıfının bir uzantısı olacaktır. View'ları ve Widget'ları (Araçlar) kullanarak kullanıcıya grafiksel arayüz sunar.
- Hizmetler (Services). Uygulamanızın görünmez kısmıdır. Bu bileşenler arkaplanda çalışır ve uygulamanız aktif olmasa bile yapılması gereken işlemleri yapar.
- İçerik Sağlayıcılar (Content Providers). Uygulama veritabanını yönetmede ve paylaşmada kullanılır. Diğer uygulamaların sizin içeriğinizi kullanmanıza izin verebilir veya diğer uygulamaların izin verdiği içeriklere erişebilirsiniz.
- Amaçlar (Intents). Mesaj iletme çalışma çerçevesidir. Amaçları kullanarak Aktivite veya Hizmetlere yapılmasını amaçladığınız mesajları iletebilirsiniz. Sistem otomatik olarak hangi uygulamayı çalıştırması gerektiğini belirleyecektir.
- Yayın Alıcıları (Broadcast Receivers). Yayın alıcısı oluşturup kaydederek, uygulamanız belirtilen filtreleme için sistemi dinleyebilir. Eğer dinlenilen bir Amaç varsa uygulamanız çalıştırılır.
- Bildirimler (Notifications). Genel olarak kullanıcının odağını bozmadan dikkatini çekmek için kullanılır. Örneğin bir mesaj geldiğinde cihazın ışıklarını yakabilir, titremesini sağlayabilir, diyalog gösterebilir veya ses çaldırabilirsiniz.

AndroidManifest.xml

Her Android projesi kök dizininde AndroidManifest.xml dosyasını içerir. Bu dosya her bileşenin düğüm noktasıdır ve Amaç Filtreleri ile İzinleri kullanarak uygulamanızın cihazla ve/veya diğer uygulamalarla nasıl etkileşime geçeceğini belirtir. Ayrıca uygulamanın nereye kurulacağını, yüklemek için gereken en düşük Android sürümü de buradan belirlenir.

Geliştirme aşaması bittikten sonra da dosyanın kolay bir şekilde imzalanmasını (ileride anlatılacak) buradan yapabilirsiniz.

Bu dosya ayrıca uygulamanın metadata (kullanılacak tema ve simgesi gibi) bilgilerini içerir.

Yaptığınız yukarıdaki örneğin manifest dosyası aşağıdaki gibidir:



Tablo 1 – AndroidManifest.XML Kod Görünümü

Eğer HTML bilginiz varsa etiket (tag) kullanım yapısına uzak değilsinizdir. Eğer daha önce HTML ile çalışmadıysanız kısaca bahsetmekte yarar var. Etiket yapısını iki şekilde kullanabilirsiniz:

<etiket özellik="değer"></etiket> veya <etiket özellik="değer"/> şeklinde.

Paket Gezgininde AnrdoidManifest.xml dosyasına çift tıkladığınızda Resim 31 ile karşılaşırsınız.

🐳 Android I	Manifest	^
▼ Manifest Gene	ral Attributes	
Defines general inf	ormation about the AndroidManifest.xml	
<u>Package</u>	com.vgezer.ilkprojemiz Browse	
Version code	1	
Version name	1.0 Browse	
Shared user id	Browse	
Shared user label	Browse	
Install location	×	
Manifest Extras	① ③ ④ ① C ① ① Az Add Remove Up Down	
To export the appli	cation for distribution, you have the following options:	
Use the Export	<u>Wizard</u> to export and sign an APK	
Export an unsid	ined APK and sign it manually	
▼ Links		~
📰 Manifest 🚺 A	pplication P Permissions I Instrumentation F AndroidManifest.xml	

Resim 31 – Android Manifest Dosyası

Sayfanın altında bulunan ilk 4 sekmeyi kullanarak, herhangi kodlama bilgisine sahip olmadan işlem yapabilir veya AndroidManifest.xml sekmesini seçerek Tablo 1'deki görünüme ulaşabilirsiniz.

Kod görünümünde CTRL+Boşluk tuşuna basarsanız kod tamamlama yardımcısını açabilirsiniz.

Şimdi bu dosyanın içeriğini inceleyelim:

<application> Bir manifest dosyası sadece bir application düğümü içerebilir. Yukarıda saydığımız bileşenler bu düğümün altında bulunurlar. Bu düğüm, uygulama simgesi, adı ve temasını içerir. <activity> Uygulamanın gösterdiği Aktivite bu etiket kullanılarak tanımlanmalıdır. android:name niteliği kullanılarak sınıf adı belirtilir. Nokta ile başlanıp java dosyasının adı yazılarak tanımlanır. Dosyada tanımlanmayıp başlatılmak istenen aktiviteler hata döndürür. Bunun altında yer alan <intent-filter> etiketi, bu aktivitenin hangi koşulda çalıştırılması gerektiğini belirtir.

<action> Amaç filtresinin hangi koşulda uygulanması gerektiğini belirtir. android:name niteliği ile çağırılma ismi belirtilir. Dahili isimler için *android.intent.action.* kullanılabilirken kendi işlemleriniz için farklılığı korumak adına paket adınızla başlaması önerilir. Buradaki *MAIN*, bu aksiyonun bu uygulamanın ilk aktivitesi olduğunu ve *LAUNCHER*, ise bu uygulamanın uygulama listesinde bir simgesi olacağını belirtir.

Yukarıdaki XML dosyasında bulunmayan daha bir çok kullanım mevcuttur. Bunlara zamanı geldiğinde değineceğim. Ancak şunu bilmekte yarar var. Bu dosyada belirtilmeyen işlemleri veya gerekli donanım bağlantılarını daha sonra gerçekleştiremezsiniz. Örneğin interneti kullanmanız gerekiyorsa, ilk olarak bu dosyada izinler bölümüne eklemeniz gerekli. Kullanıcılar uygulamayı yüklerken bu izinleri kabul ederek yükleyecekler. Yükleme yapıldıktan sonra bir daha gerekli izinler kullanıcıdan alınmak üzere sorulmayacaktır.

Şimdi çalışma ortamına dönelim.

Android'e Isınma

Proje yönetimini ve kaynakları anlamak için yeni bir proje yapalım.

İlk projeyi tamamladığınız varsayılarak proje oluşturma sihirbazı açıklamaları tekrar verilmeyecektir.

Öncelikle projemizdeki hedefleri belirleyelim:

- Projemiz bir hesap makinesi olacak. Proje adı: Hesap Makinesi. Paket adı: com.vgezer.hesapmakinesi.
- Android 2.3 sürümü ve üstünde çalışacak (Eğer bilgisayarınıza 2.1 kurduysanız bunu da seçebilirsiniz).
- Düğmelere basıldığında ilgili 4 işlemden birini yapıp aynı ekranda gösterecek.
- Türkçe ve İngilizce dil seçenekleri olacak.
- Görünümü yaklaşık şu şekilde olacak:





Bu proje sonunda:

- Proje Gezginini daha fazla öğrenmeyi
- Bir arayüze kontrol yerleştirmeyi
- Uygulamaya resim yüklemeyi

• Çoklu dil desteği kullanmayı

öğreneceksiniz. Yine kodların nasıl çalıştığı burada anlatılmayacak. Bu bölümde önemli olan kodlamadan çok çalışma ortamına alışmanız.

Öncelikle sanal Android cihazınızın açık olduğundan emin olun. Eğer değilse Eclipse içerisinden *Window –> AVD Manager* tıklayın ve aygıtı seçip *Start* tıklayın. Cihaz çalıştıktan sonra **IIII** tıklayarak uygulama çekmecesine girin. Sırasıyla *Settings – Language & Keyboard (Görebilmeniz için ekranı tutup yukarı çekmeniz gerekebilir) – Select Language* tıklayıp Türkçe'yi bularak seçin (Not: Eğer 2.1 sürümünü kullanıyorsanız, uygulama çekmecesinden *Custom Locale* tıklayıp yeni bir dil eklemeli ve tr_TR girmelisiniz.).

Projeyi oluşturduğunuzda, *IlkProje* adlı projenizin de hala Paket Gezgininde olduğunu görebilirsiniz. Üzerine sağ tıklayıp *Delete (Sil)* seçerseniz, çalışma ortamından kaldırabilirsiniz. Projeyi sabit diskinizden <u>tamamen</u> silmek için uyarı penceresindeki işaret kutusunu işaretleyin.

Projenizin res/layout/main.xml dosyasını çift tıklayarak açın ve @string/hello metnini seçip klavyenizden *Delete* tuşuna basarak kaldırın (Resim 33).



Resim 33 – Hesap Makinesi Projesi

Palette (Palet) kısmından Text Fields (Yazı alanı) klasörünü açın ve aşağı kaydırarak Number (Sayı) yazı alanını bulun. Ardından fare ile basılı tutarak telefon ekranına sürükleyin (Resim 34).



Resim 34 – Sayı alanı ekleme

Hemen altına bir tane daha sayı alanı ekleyin.

Form Widgets (Form Yardımcıları) klasöründen bir Button (Düğme) sürükleyin ve sayı alanının altına bırakın.

Düğmeye sağ tıklayıp *Edit Text (Metni Düzenle)* tıklayın. *Button* yazan kısma "+" koyacağız ancak buna izin verilmiyor. Bu nedenle "New String" (Yeni Metin) düğmesine basın ve *String* kısmına "+"; *New R.string.* kısmına *top_is* yazın. Tamam tıklayarak pencereyi kapatın. Otomatik olarak @string/top_is seçilmiş olmalı ve *Resolved Value (Çözülen Değer)* + göstermelidir. Pencereyi Tamam tıklayarak kapatın.

Aslında düğme ve diğer Görünümler için doğrudan

Başka bir düğme ekleyin ve toplama simgesinin yanına koymaya çalışın. Yerleşim planı buna izin vermeyecektir. Yerleşim planı bilgisine projenin devamında değineceğim. Şu an için düğmeleri alt alta ekleyin. Çıkarma, Çarpma ve Bölme için de aynı işlemleri yapın ve isim olarak sırasıyla: *cik_is*, *carp_is*, *bol_is* verin.

Son adımda yerleşim Resim 35 gibi olmalıdır:





Görünümü biraz daha güzelleştirmek için her butona basıp *Toggle Fill Width (Genişliğe Doldur/ma)* tıklayın (Resim 36).



Resim 36 – Genişliğe Doldur

Şimdiye kadar yaptığımız işlemde 4 adet düğmemiz olduğundan sorunla karşılaşmayabiliriz ancak proje büyüdükçe *button1*, *button2* gibi varsayılan isimler kullanılırsa aradığımız nesneyi bulmamız zorlaşır. Bu nedenle düğme isimlerini sırasıyla şu şekilde değiştirin: *top_dug*, *cik_dug*, *carp_dug* ve *bol_dug*. *editText1* ve *editText2* sayı alanlarının adını da *sayi1* ve *sayi2* olarak değiştirin. Dosyayı kaydedin. Klasörlerin altında bulunan *Graphical Layout (Grafiksel Yerleşim)* ve main.xml sekmelerinden düzenleme biçimini seçebilirsiniz.

Main.xml sekmesini tıklayın ve aşağıdaki gibi olduğunu doğrulayın (Eğer düğme ekleme işlemini yapamadıysanız aşağıdaki kodu bu alana yapıştırabilirsiniz):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:layout width="fill parent"
    android:layout height="fill parent"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/sayi1"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:ems="10"
        android:inputType="number" >
        <requestFocus />
    </EditText>
    <EditText
        android:id="@+id/sayi2"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:ems="10"
        android:inputType="number" />
    <Button
        android:id="@+id/top dug"
        android:layout width="match parent"
        android:layout height="wrap content"
        android:text="@string/top is" />
    <Button
        android:id="@+id/cik dug"
        android:layout width="match parent"
        android: layout height="wrap content"
        android:text="@string/cik is" />
    <Button
        android:id="@+id/carp dug"
        android: layout width="match parent"
        android:layout height="wrap content"
        android:text="@string/carp is" />
    <Button
        android:id="@+id/bol dug"
        android:layout width="match parent"
```

```
android:layout_height="wrap_content"
android:text="@string/bol_is" />
```

</LinearLayout>

Tablo 2

src/res/values/strings.xml dosyasını çift tıklayarak açın. Alttaki sekmeden strings.xml

görünümüne geçin. Eğer bir hata yapmadıysanız veya düğme metinlerini yazdıysanız

aşağıdaki metin bulunmalıdır:

Burada kullanmadığınız hello ismindeki satırı silin ve

<string name="sonucyazi">Sonuc:</string> satırını resources etiketleri arasına

ekleyerek dosyayı kaydedin:

Not: Bir ekleme yaptıktan sonra kullanabilmeniz için dosyayı kaydetmeniz gerekmektedir.

src/com.vgezer.hesapmakinesi klasöründeki HesapMakinesiActivity.java dosyasını çift

tıklayarak açın ve aşağıdaki kodu yapıştırın:

```
package com.vgezer.hesapmakinesi;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class HesapMakinesiActivity extends Activity {
    /** Called when the activity is first created. */
```

```
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        final EditText sayi1= (EditText) findViewById(R.id.sayi1);
        final EditText sayi2= (EditText) findViewById(R.id.sayi2);
       Button toplama= (Button) findViewById(R.id.top dug);
       Button cikarma= (Button) findViewById(R.id.cik dug);
       Button carpma= (Button) findViewById(R.id.carp dug);
       Button bolme= (Button) findViewById(R.id.bol dug);
        toplama.setOnClickListener(new View.OnClickListener() {
                  public void onClick(View v) {
                        String sonuc=
String.valueOf((Double.parseDouble(sayi1.getText().toString())+Double.parseDo
uble(sayi2.getText().toString()));
      Toast.makeText(getApplicationContext(),getString(R.string.sonucyazi)+"
"+sonuc, Toast.LENGTH LONG).show();
                  }
            });
        cikarma.setOnClickListener(new View.OnClickListener() {
                  public void onClick(View v) {
                        String sonuc=
String.valueOf((Double.parseDouble(sayi1.getText().toString()) -
Double.parseDouble(sayi2.getText().toString()));
      Toast.makeText(getApplicationContext(),getString(R.string.sonucyazi)+"
"+sonuc, Toast.LENGTH LONG).show();
            });
        carpma.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                  String sonuc=
String.valueOf((Double.parseDouble(sayi1.getText().toString())*Double.parseDo
uble(sayi2.getText().toString()));
      Toast.makeText(getApplicationContext(),getString(R.string.sonucyazi)+"
"+sonuc, Toast. LENGTH LONG). show();
        });
       bolme.setOnClickListener(new View.OnClickListener() {
                  public void onClick(View v) {
                        String sonuc=
String.valueOf((Double.parseDouble(sayi1.getText().toString())/Double.parseDo
uble(sayi2.getText().toString()));
      Toast.makeText(getApplicationContext(),getString(R.string.sonucyazi)+"
"+sonuc, Toast. LENGTH LONG). show();
```

Projeyi çalıştırdığımızda Resim 37Resim 19 görülecektir:



Resim 37 – Hesap makinesi projesi 1

Metin alanlarına tıklayarak iki sayı girip işlemlerden birine dokunduğunuzda (tıkladığınızda) sonuç kısa bir süre *Toast* mesajında (ileride anlatılacak) gösterilecektir. Şimdi bu uygulamaya dil seçeneği ekleyelim.

Android, ülke, dil, telefonun yönü, günün saati, dokunmatik ekrana sahip olup olmaması, ekran boyutu ve diğer bir çok değişkene göre yapılandırma dosyaları oluşturabilir. Bu özelliklerden istediklerimizi kullanarak her özelliğe göre farklı davranışta uygulamalar yapabiliriz. Şimdi projeye sağ tıklayıp *New – File – Other* seçin (Alternatif olarak Ctrl+N tuşlarına basabilirsiniz). *Android XML Values File* seçip ileri tıklayın. Dosya ismi olarak *strings* yazın ve sonraki adıma geçin. Dosyanın zaten var olduğu uyarısı verilecektir ancak farklı bir yapılandırma seçeceğimiz için bu sorun ortadan kalkacaktır. *Choose Configuration Folder (Yapılandırma Klasörü Seç)* sayfasından *Language* tıklayın ve "->" düğmesine basın. Gelen *Language* yazı alanına İngilizce'yi belirten "en" girin. Pencerenin altında yazan klasörün res/values-en olarak değiştiğini göreceksiniz (Resim 38).

Available Qualifiers	^		Chosen Qualifiers	Language
Country Code			言五 en en	en 🗸
Network Code				(2 letter code)
Region				
ESmallest Scree		->		
⊖Screen Width				
🗘 Screen Height		<-		
Size		,		
⊐ ∎¤ Ratio				
Orientation				
🖵 UI Mode				
Night Mode	-			
pi Density				
Touch Screen				
III Text Innut	~			

Resim 38 – Farklı yapılandırma seçim penceresi

Son tıkladığınızda pencere kapatılıp yeni dosya açılacaktır. *Add (Ekle)* tıklayıp "S" elementini seçtikten sonra Tamam tıklayın. İsim alanına *app_name*, değer alanına da *Calculator* yazın. Bir string elementi daha ekleyin ve isim olarak *sonucyazi*, değer olarak da *Result:* yazıp dosyayı kaydedin. Projeyi tekrar çalıştırın ve D düğmesi ile uygulamadan çıkın. Yukarıda belirttiğim adımları tekrar izleyerek telefonun dilini İngilizce yapın. İngilizce ülkesi için herhangi bir ülke seçebilirsiniz. Uygulamayı tekrar çalıştırdığınızda başlık ve sonuç yazılarının değiştiğini göreceksiniz ().



Diğer metinleri eklemediğimize dikkat edin. Eğer farklı bir yapılandırma için ekleme yapmazsak, varsayılan değerler alınacaktır.

Uygulamayı bölümün başlangıcında verilen resme benzetmek üzere devam edelim.

Android'de ekranlar yerleşimler ile gösterilir. Yerleşimleri XML veya programlama ile yapmak mümkündür ancak biz sadece XML ile ilgileneceğiz. Programlama ile geliştirmek isteyenler Android Geliştirici sayfasını ziyaret edebilirler.

Hesap makinesi örneğimize bakarsak, eklediğimiz her metin alanı ve düğme bir Yardımcı'ydı (Widget). Tablo 2'deki koda bakarsanız, buradaki *LinearLayout* alanı Görünüm Grubunu (View Group) temsil eder. XML dosyasındaki yerleşimleri de aşağıdaki gibidir:

> <GörünümGrubu> <Görünüm 1/> <Görünüm 2/>

</GörünümGrubu>

Android içerisindeki bazı görünüm grupları şunlardır:

- FrameLayout
- LinearLayout
- TableLayout
- RelativeLayout

FrameLayout (Çerçeve Yerleşimi): En basit yerleşim şeklidir. Örneğin sadece bir resim kullanacaksanız bu yerleşimi seçmek isteyebilirsiniz. Bu görünümün altına yerleştirilecek her Görünüm, sol üste konulacaktır ve konumları değiştirilemez.

LinearLayout (Doğrusal Yerleşim)*: Orientation (Yönlendirme) etiketinde belirlediğiniz değere göre Görünümler, yatay veya dikey olarak tek bir yönde yerleştirilir.

* Hesap makinesi örneğimizde düğmeleri yan yana yerleştirememe sebebimiz bu yerleşimin seçilmiş olmasından kaynaklanıyordu.

TableLayout (Tablo Yerleşimi): Görünümleri satır ve sütuna yerleştiren yerleşimdir. Örnek kullanımı aşağıdaki gibidir (XML kodlarında yer alan <!-- --> arasındaki metinler yorum satırlarıdır ve kodlamada herhangi bir etkileri yoktur).

<TableLayout> <TableRow> <!-- Ekleyeceğimiz her Görünüm bir sütun oluyor --> </TableRow> <!-- Yeni Satır Oluşturuyoruz --> <TableRow> <!-- Her satırdaki sütun sayısı aynı olmak zorunda değil --> </TableRow> </TableRow>

RelativeLayout (Bağıl Yerleşim): Görünümleri birbirlerinin konumuna göre yerleştirir. Örneğin ilk olarak *düğme1* adında bir düğme yerleştirdiyseniz, sonraki öğelerin konumları düğmenin altından ve solundan X birim uzakta gibi değerler ile tutulacaktır.

Hesap makinesi örneğimize dönersek, Resim 32'te gösterilen yerleşim için iki seçeneğimiz mevcut: Tablo Yerleşimi veya Bağıl Yerleşim. Tablo Yerleşimini seçerek görünümümüzü değiştirelim.

İşleme başlamak için main.xml dosyasını açın ve ekranın boş kısmında sağ tıklayarak *Change Layout (Yerleşimi Değiştir)* seçin. Ardından *TableLayout* seçip Tamam tıklayın. Ekranın yine boş bir kısmında sağ tıklayarak Resim 32 için ihtiyaç duyulan kadar (3 kez) *Add Row (Satır Ekle)* işlemini yapın. Görünümleri tutup sürükleyerek istenilen sonuca ulaşmaya çalışın. Ardından düğmelerin eşit boyutta olması için her satırdaki bir düğmeye tıklayarak, düzenleyicinin araç çubuğunda *Distribute Weights Evenly (Ağırlıkları Eşit Yap) seçin.* Sonuç Resim 39 gibi olmalıdır:

Hesap Makinesi	
+	
х	/



İsterseniz tekrar çalıştırıp test edebilirsiniz. İsterseniz Ctrl+F11 tuşuna basarak telefonu yatay-dikey olarak döndürebilirsiniz.

Projemizin son adımı da hesap makinesine bir arkaplan eklemek olacak. Bunun için Google Görsellerde *arkaplan* aramasını yapın. Resim boyutu olarak genişlik 240, yükseklik 320 yazın. Benim seçtiğim resim şu:



Arama sonucunda renk seçiminden mavi tonunu tıklayarak bu resmi bulmayı kolaylaştırabilirsiniz.

Resmi sadece <u>küçük harflerden</u> oluşan <u>ve özel karakter içermeyen</u> isimle kaydedin (Ben *arkaplan* isminde kaydettim). Eclipse proje gezginine sürükleyerek *res/drawable-mdpi* klasörüne bırakın. Dosyayı link (bağlamayı) değil, kopyalamayı seçin. Bu klasör aslında drawable klasörüdür, ancak yukarıda bahsettiğimiz farklı yapılandırmalar için "yeniyapılandırma" eki gelmektedir. Klasör adı kullanılırken bu ek kullanılmaz.

Şimdilik sadece bir yerleşimimiz olduğundan layout/main.xml dosyasını açın ve XML görünümüne geçin. <TableLayout> arasında *android:id*, *android:layout_width* gibi parametreleri göreceksiniz. android:özellik="değer" yapısını bozmadan ve küçüktür işaretinden önce Enter tuşu ile yeni bir satıra geçin.

android:b yazıp Ctrl+Boşluk tuşu ile kod tamamlayıcıyı açın ve *android:background* (*arkaplan*) çift tıklayarak seçin. İmleç, çift tırnaklar arasına gelecektir.

Paket Gezginine eklediğiniz dosyalara erişmek için @ işaretini ve ardından *res* klasörünün içindeki bir dizinin ismini girmeniz gerekir. Bölü (/) işareti ve ardından dosyanın <u>sadece</u> adını girerek referans çağırma işlemi tamamlanır.

Şimdi tırnakların arasına @d yazın ve ardından kod tamamlayıcı ile tamamlatın. Tekrar kod tamamlayıcıyı açıp eklediğiniz resmin adını seçin. İşlem sonunda *main.xml* dosyasında aşağıdaki düzenleme yapılmış olmalıdır (resmi arkaplan olarak kaydettiğiniz varsayılmaktadır):

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/TableLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_height="fill_parent"
    android:background="@drawable/arkaplan">
```

Tüm dosyaları kaydettikten sonra programı tekrar çalıştırıp sonucu görebilirsiniz.

Bu uygulama ile artık yeni kaynak ekleme ve cihaz yapılandırmalarına göre uygulama ekranları oluşturmayı öğrendiniz. Bir sonraki bölümde Android SDK içerisinde gelen uygulamalara göz atacağız.

Uygulama Hayat Döngüsü

Android içerisindeki uygulamaların, kendilerini Android işletim sistemine bıraktıklarını söylemiştim. Android içerisindeki tüm uygulamalar hiçbir istisna olmadan, aşağıdaki hayat döngüsüne sahiptirler:

- Uygulama oluşturulması (onCreate)
- Uygulama başlatılması (onStart)
- İşlemler yapılması
- Uygulama durdurulması (onStop)
- Uygulama yok edilmesi (onDestroy)

Bu olayların her biri birçok etkiye göre değerlendirilerek (bellek kullanımı, kullanıcı girişi vb.) yönetilir.

Uygulama herhangi bir beklenmedik durum oluşturursa, kapatılması istenecektir. Örneğin yukarıda yaptığımız uygulamada sayı alanlarına değer girmeden herhangi bir düğmeye tıklayın ve cihazın davranışını görün. Bu durum *Force Close (Zorla Kapatma)* durumudur.

Aktiviteler

İlk bölümde bir aktivitenin, kullanıcılara gösterilen ekranlar olduğunu söylemiştim. Temel olarak her oluşturduğunuz yerleşim dosyasını farklı bir aktiviteyle ilişkilendirirsiniz ve yeni aktivitelere Amaçlar ile geçiş yaparsınız. Diğer bir olanak ise farklı yerleşim dosyalarını aynı aktivite içerisinde göstermektir. Bu işlem, yeni bir ekran oluşturmayacak, sadece görünümü değiştirecektir.

Aktiviteler sadece görünür sayfalardan oluşmak zorunda değil. Bir hizmet olarak da çalıştırmanız mümkün.

Bir proje oluşturduğunuzda, eğer işareti kaldırmadıysanız, otomatik bir aktivite oluşturulacaktır.

Yeni bir aktiviteyi yeni *class (sınıf)* dosyası olarak oluşturursunuz ve kullanabilmeniz için *AndroidManifest.xml* dosyasında tanımlamanız gerekir.

Aktiviteler stack (yığın) mekanizmasına (LIFO - Last In First Out) uyar ve her çalıştırdığınız yeni aktivitede, bir önceki aktiviteyi yığına iter. Kullanıcılar geri tuşunu kullanarak bir önceki aktiviteye dönerler.

Aktivitelerin oluşturulduğunda, durdurulduğunda devam ettirildiğinde ve yok edildiğinde çağırdıkları metotlara göre işlem yapılması gerekir. Örneğin bir aktivite durdurulduğunda büyük nesneler olan veritabanı ve ağ bağlantısını kesmeniz gerekir.

Bir aktivite tanımlamak için Activity sınıfından türeyen bir alt sınıf oluşturmanız gerekir. Tanımladıktan sonra kullanmanız gereken en önemli iki metot şunlardır: *onCreate* ve *onPause*.

onCreate, aktiviteniz oluşturulduğunda çağırılan metottur. Aktivitenin en önemli tanımlamalarını burada yapmanız gerekir. Oluşturduğunuz bir yerleşimi burada setContentView() çağırarak gösterirsiniz. onPause, kullanıcı uygulamanın önüne başka bir ekran getirdiğinde (farklı bir uygulama veya aktivite çalıştırıldığında) çağırılan metottur. Saklanmasını istediğiniz değişiklikleri burada kaydetmeniz önerilir. Çünkü kullanıcı bir daha bu ekrana dönmeyebilir.

Tüm oluşturduğunuz aktiviteler, kullanılabilmesi için *AndroidManifest.xml* dosyasında tanımlanmalıdır. Aşağıdaki tabloda görüleceği üzere, eğer oluşturduğunuz dosya adı *AktiviteAdi.java* ise kullanmanız gereken isim *AktiviteAdi* dır.

```
<manifest ... >
<application ... >
<activity android:name=".AktiviteAdi" />
...
</application ... >
...
</manifest >
```

Bir aktiviteden diğer aktiviteye geçmek için *startActivity(Intent)* metodu kullanılır. Intent parametresinin bulunduğu yerde yeni bir Intent türünde nesne oluşturuyoruz.

Çalıştırmak istediğimiz her aktivite kendi oluşturduğumuz aktivite olmak zorunda değil. Örneğin telefonun dahili e-posta aktivitesini de çalıştırabiliriz. Kullanıcı birden fazla uygulamaya sahipse, bir tanesini seçmesi istenecektir.

Aktiviteyi kapatmak için ise *finish()* kullanabilirsiniz. Diğer bir aktiviteyi kapatmak için ise *finishActivity()* kullanın. Ancak bunları kullanmak zorunda değilsiniz, çünkü bellek yönetimi Android içerisinde oldukça başarılı.

Ek: Bazı Kısayollar

Kısayollar her zaman size zaman kazandırır ve uzun menüler arasında arama yapmanızı önler.

Eclipse içerisinde kullanabileceğiniz birçok kısayoldan birkaçını buraya yazıyorum.

Eclipse içerisinde bu kısayolları değiştirebilir veya yeni kısayollar ekleyebilirsiniz. Bunun için Eclipse yardımına bakın.

Eğer daha önce Programlama IDE'leri (Tümleşik Geliştirme Ortamı) ile çalıştıysanız çoğu kısayol size tanıdık gelecektir:

Kısayol	Açıklama			
F5	Paket Gezgini klasörlerini yenile.			
F2	Paket Gezginindeki bir dosyanın adını değiştir			
CTRL+	CTRL ve ilgili tuşlara aynı anda basıldığında geçerli komutlar			
A	Tüm metni seç			
С	Metni kopyala			
V	Kopyalanmış metni yapıştır			
D	Satırı sil			
N	Yeni dosya penceresini aç			
F11	Çalıştır			
S	Dosyayı kaydet			
Shift+S	Tüm değiştirilmiş dosyaları kaydet			
W	Açık düzenleyici penceresini kapat			
F	Bul/Değiştir			
Shift+C	Seçili metni yorum yap/yorumu kaldır			
Boşluk	Kod tamamlayıcıyı aç			
Alt+YukarıOk				
veya	Bulunulan satırı yukarı veya aşağı kopyalayıp yapıştır			
Alt+AşağıOk				

ALT+	ALT ve ilgili tuşlara aynı anda basıldığında geçerli komutlar
Aşağı	Düzenleyici açıkken satırı aşağı kaydır