

JAVA İLE ANDROID PROGRAMLAMA

ÇAĞLAR ARTAR

İÇİNDEKİLER

BÖLÜM 1: ANDROID İŞLETİM SİSTEMİ	1
Android Nedir?	2
Neden Android?	3
Android'in Gelişim Süreci	4
Android'in Kullanım Alanları	5
Android'in Geleceği?	6
Android Uygulama Geliştirerek Para Kazanmak Mümkün Mü?	7
Mobil İşletim Sistemleri	8
iOS	9
Windows Phone	9
Blackberry OS	9
Symbian OS	9
Web OS	10
Maemo OS	10
MeeGo OS	10
Bada OS	10
Tizen OS	11
Limo OS	11
Sailfish OS	11
Neler Öğrendik?	11
BÖLÜM 2: GEREKLİ ARAÇLARIN KURULUMU VE AYARLARI	13
Giriş	14
Windows Ortamı için Java JDK Kurulumu	15
Mac OS X Ortamı için Java JDK Kurulumu	19
Linux için Java JDK Kurulumu	20
Windows Ortamı için Android Studio ve SDK Kurulumu	20
Mac OS X Ortamı için Android Studio ve SDK Kurulumu	23

Linux Ortamı için Android Studio ve SDK Kurulumu	26
Yeni Proje Oluşturmak	27
Android Studio Kullanıcı Arayüzü	31
Araç Çubuğu (Toolbar)	32
Proje Hiyerarşi Paneli (Project Panel)	34
Tasarım Paneli (Design Panel)	35
Android SDK Manager	36
AVD (Android Virtual Device) Manager	38
Intel HAXM (Hardware Accelerated Execution Manager)	44
Genymotion - Hızlı Android Sanal Aygıtı	47
Neler Öğrendik?	51
BÖLÜM 3: ANDROID İÇİN JAVA	53
Giriş	54
Java Nedir ve Nasıl Çalışır?	55
Eclipse Editörü	56
Eclipse Kullanıcı Ara Yüzü	58
Yeni Java Projesi Oluşturmak	59
Yazım Kuralları	61
Yorum Satırları	62
Sınıf ve Nesne Kavramları	62
Sınıf Tanımlama	64
Nesne Tanımlama	65
Değişkenler Ve Değişken Tipleri	68
Referans Veri Tipleri	70
Değişkenleri Tanımlamak	70
Operatörler	71
Aritmetik Operatörler	72
İlişkisel Operatörler	72
Atama Operatörleri	73

Metodlar	74
Metod Tanımlama	74
Main Metodu	79
Overloading Metodlar	79
Constructor Yapıcı Metodlar	81
Erişim Belirteçleri	83
Public	83
Private	84
Protected	84
Default Ön Tanımlı (Friendly)	84
Static Anahtar Kelimesi	86
Koşul Yapıları	89
if-then-else Koşul Yapısı	89
Soru İşareti Operatörü	93
switch case Koşul Yapısı	94
Döngüler	97
while Döngüsü	98
do while Döngüsü	100
for Döngüsü	102
Diziler	104
Dizilere Değer Atamak	106
İstisnalar	108
Çoklu catch Yapısı	113
Paketler	116
Paketleri Oluşturmak ve Kullanmak	117
Giriş Çıkış ve Dosya İşlemleri	120
Dosya İşlemleri	120
Stream	123
Byte Stream	125

Character Stream	126
Buffered Stream	128
final Anahtar Kelimesi	129
this Anahtar Kelimesi	133
Neler Öğrendik?	135

BÖLÜM 4: NESNE YÖNELİMLİ PROGRAMLAMA 137

Giriş	138
Kalıtım (Inheritance)	138
Kalıtım İşlemleri	139
super Anahtar Kelimesi	143
Geçersiz Kılmak (Override)	146
Çok Biçimlilik (Polymorphism)	149
Soyut (Abstract)	152
Kapsülleme (Encapsulation)	156
Ara Yüzler (Interfaces)	160
İş Parçacıkları (Thread)	163
Neler Öğrendik?	168

BÖLÜM 5: XML (EXTENSIBLE MARKUP LANGUAGE) 169

Giriş	170
XML Nedir?	170
XML Kullanım Alanları?	171
Yeni XML Dosyası Oluşturmak	171
Deklarasyon	172
Etiketler	174
Özellikler	176
Ölçü Birimleri	178
strings.xml	180
dimens.xml	181

colors.xml	182
Yorum Satırları	184
Varlık Referansları	185
Karakter Verileri (CDATA)	186
Neler Öğrendik ?	186

BÖLÜM 6: ANDROID KULLANICI ARAYÜZ TASARIM BİLEŞENLERİ 187

Giriş	188
Kullanıcı Arayüz Elemanları	188
Widgets	189
TextFields	190
Containers	191
Date & Time	191
Expert	191
Layouts	192
Arayüz Elemanları Özellikleri	195
id	195
layout_height	196
layout_width	196
layout_gravity	196
layout_weight	196
gravity	196
layout_margin	196
layout_marginTop	197
layout_marginBottom	197
layout_marginLeft	197
layout_marginRight	197
padding	197
paddingTop	197

paddingBottom	197
paddingLeft	197
paddingRight	197
background	197
enabled	198
Arayüz Tasarım Uygulaması	198
Dikey Ekran için Arayüz Tasarımı	198
Yatay Ekran için Arayüz Tasarımı	201
Sanal Aygıt Üzerinde Projeyi Çalıştırmak	203
Genymotion Üzerinde Projeyi Çalıştırmak	204
Gerçek Mobil Aygıtlar Üzerinde Uygulamayı Çalıştırmak	205
Hata Görüntüleme Pencereleri	206
Neler Öğrendik?	207
BÖLÜM 7: TEMEL ANDROID BİLEŞENLERİ VE KAVRAMLARI	209
Giriş	210
Android Mimarisi	210
Linux Çekirdeği (Linux Kernel)	211
Kütüphaneler (Libraries)	211
Çalışma Zamanı (Android Runtime)	211
Uygulama Çerçevesi (Application Frameworks)	211
Uygulamalar (Applications)	211
Dalvik Sanal Makinesi	211
ART (Android Run Time)	212
Aktiviteler (Activities)	212
Yeni Bir Aktivite Oluşturmak	213
Aktivite (Activity) Sınıfı	213
Aktivite (Activity) Sınıfının Yaşam Döngüsü	213
R.java Sınıfı	217

AndroidManifest.xml Dosyası	219
Loglama	221
Dinleyiciler (Listener)	223
Diyalog Yapıları	227
Alert Dialogue	227
Toasts	233
Bildiri (Notifications)	234
Liste Görünümü (List View)	237
Intent	243
Sayfalar (Aktiviteler) Arası Geçiş İşlemleri	244
Sayfalar (Aktiviler) Arası Veri Taşıma İşlemleri	245
Sayfalardan (Aktivitelerden) Sonuç Döndürme	249
Yayın Alıcıları (Broadcast Receivers)	252
Servisler (Services)	257
AsyncTask	267
Action Bar	275
Fragments	280
Neler Öğrendik?	285
BÖLÜM 8: ANDROID STUDIO İLE GELEN HAZIR UYGULAMA ŞABLONLARI	287
Giriş	288
Uygulama Şablonları	289
Sekmeli Sayfalar (Tabbed Activity)	289
Navigasyon Menüler (Navigation Drawer Menu)	299
Master/Detail Flow (Ana/Detay Akışı)	313
Tam Ekran Sayfalar (Full Screen Activity)	322
Google Haritalar (Google Maps Activity)	326
Google Play Servisleri (Google Play Service Activity)	326
Activity Şablonları	327

Blank Activity	327
Blank Activity With Fragment	327
Android TV Activity	327
Blank Wear Activity	327
Login Activity	327
Settings Activity	327
Diğer Şablonlar	328
Neler Öğrendik?	328

BÖLÜM 9: KONUM SERVİSLERİ VE GOOGLE HARİTALARI 329

Giriş	330
Google Servislerin Genymotion Üzerine Kurulumu	330
Konum Servisleri	331
Google Haritalar Servisi	342
Neler Öğrendik?	355

BÖLÜM 10: VERİ DEPOLAMA İŞLEMLERİ 357

Giriş	358
SharedPreferences	358
Internal Storage	364
External Storage	366
SQLite	369
Veritabanı Oluşturmak	370
Tablo Oluşturmak ve Silmek	371
Kayıt Ekleme (Insert)	373
Kayıtları Listeleme (Select)	375
Kayıt Güncelleme (update)	379
Kayıt Silme (delete)	380
SQLite Uygulaması	381
Neler Öğrendik?	390

BÖLÜM 11: ANDROID'DE AĞ İŞLEMLERİ 391

Giriş	392
Volley Kütüphanesi	392
Volley Kütüphanesini Projeye Dahil Etmek	393
Volley Kütüphanesi Kullanımı	395
StringRequest	395
JsonRequest	397
ImageRequest	399
İstekleri İptal Etmek	400
WebView Kullanımı	401
Neler Öğrendik?	405

BÖLÜM 12: ANDROID UYGULAMALARIN YAYINA HAZIR HALE GETİRİLMESİ 407

Giriş	408
Uygulamaların Paketlenmesi ve İmzalanması	408
Uygulamaların Google Play Store Üzerine Yüklenilmesi	411
Neler Öğrendik?	414

3

ANDROID İÇİN JAVA

BU BÖLÜMDE

Giriş	54
Java Nedir ve Nasıl Çalışır?	55
Eclipse Editörü	56
Yazım Kuralları	61
Sınıf ve Nesne Kavramları	62
Değişkenler ve Değişken Tipleri	68
Operatörler	71
Metodlar	74
Erişim Belirteçleri	83
Static Anahtar Kelimesi	86
Koşul Yapıları	89
Döngüler	97
Diziler	104
İstisnalar	108
Paketler	116
Giriş Çıkış ve Dosya İşlemleri	120
final Anahtar Kelimesi	129
this Anahtar Kelimesi	133
Neler Öğrendik?	135

Android uygulamaların yapı taşlarını java dilinin temel kavram ve prensipleri oluşturur. Dolayısıyla bu yola baş koyan bir android severin, java dilinin temel kavram ve prensiplerini çok iyi talim etmesi gerekir.

Bu bölümde, Java yazılım dili nedir? Java yazılım dilinin temel kavram ve prensipleri nelerdir? Bu kavram ve prensiplerin kullanım şekilleri ve bunlara ait gerekli teknik ayrıntılar nelerdir? Bu ve buna benzer birçok sorunun cevabını birlikte arayacağız.

JAVA NEDİR VE NASIL ÇALIŞIR?

Java nesne yönelimli programlama tekniğini benimsemiş, platformlardan bağımsız olarak çalışan çok işlevli bir programlama dilidir. Java diliyle geliştirilen uygulamaların platformdan bağımsız olarak çalışması, bu uygulamaların her işletim sistemi üzerinde çalıştırılabilir olduğu manasına gelmektedir. Java dilinin sahip olduğu bu özellik, kendi kategorisinde bulunan diğer dillerden kendisini ayıran en önemli özelliklerinden bir tanesidir. Java diliyle masaüstü uygulamaları dahil olmak üzere, web uygulamaları, mobil cihazlara yönelik uygulamalar gibi bir çok alanda ve cihazda kullanılabilecek uygulamalar geliştirmek mümkündür. Java diliyle uygulama geliştirmek için kullanılan 3 farklı java sürümü mevcuttur. Bu sürümler geliştiricilere ücretsiz olarak sunulmaktadır. Bunları sırasıyla tanıyalım:

- » **Java JSE (Java Standart Edition):** Sunucu uygulamaları, masaüstü uygulamaları geliştirmek için gerekli olan araç ve kütüphaneleri içerisinde barındıran en temel sürümdür.
- » **Java EE (Java Enterprise Edition):** Geniş ölçekli kurumsal ve web uygulamaları geliştirmek için kullanılan sürümdür.
- » **Java ME (Java Micro Edition):** Taşınabilir mobil telefonlar, tabletler vb. cihazlar ve PDA ve TV setleri gibi aygıtlar için uygulama geliştirmek için kullanılan sürümdür.
- » Java diliyle ilgili bahsedilmesi gereken diğer kavramlar ise, bir önceki bölümde anımsadığımız **JVM**, **JRE** ve **JDK** kavramlarıdır.
- » **JVM (Java Virtual Machine):** Türkçede **java sanal makinesi** olarak adlandırılan bu bileşen, java diliyle yazılmış olan kodların tüm platformlarda çalıştırılmak üzere geliştirilmiş bir bileşendir. Bununla birlikte her platforma uygun ayrı sürümleri de mevcuttur. Dolayısıyla Java, platformlardan bağımsız olarak çalışmasını ve **bir kere yaz her yerde çalışsın** felsefesini **JVM** ile gerçekleştirebilmektedir.
- » **JRE (Java Runtime Environment):** Java programlarını çalıştırmak için gerekli olan yazılımdır. İçerisinde **JVM** ve **Java** kütüphanelerini de barındırır. Dolayısıyla bir bilgisayar üzerinde, java diliyle geliştirilmiş bir uygulamayı çalıştırabilmek için bu yazılımın bilgisayara kurulu olması gerekmektedir.
- » **JDK (Java Development Kit):** JDK 'nın bir önceki bölümde, java uygulamalarını geliştirmek için gerekli olan tüm araçları barındıran bir yazılım geliştirme ortamı olduğundan bahsetmiştik. Burada JDK ile ilgili değinmek istediğim bir diğer husus ise, JDK'nın bünyesinde **JVM** ve **JRE**'yi de barındırdığı gerçeğidir. Dolayısıyla bir önceki bölümde gösterildiği gibi JDK'yı bilgisayarımıza kurduğumuz zaman

SINIF TANIMLAMA

Sınıf kavramı hakkında bilgi edindikten sonra sınıf tanımlama işleminin nasıl gerçekleştirildiğini görelim. Sınıflar aşağıda gösterilen sözdizimine göre tanımlanırlar.

Java Kodu

```
class SinifAdi
{
    // Sınıf kod bloğu
}
```

1. `class SinifAdi` : Sınıf tanımlaması öncelikle `class` anahtar sözcüğü ile başlatılır. Sonrasında ise sınıfın ismi belirtilir.
2. `{}` : Sınıfın yapısı `{}` süslü parantezler içerisinde kurulur. Bu kısma ayrıca sınıf kod bloğu adı da verilmektedir. Sınıfla ilgili tüm özellikler ve metodlar bu parantezler içerisinde tanımlanır. Aşağıda verilen sözdizimine göre oluşturulmuş bir sınıf örneği yer almaktadır.

Java Kodu

```
1 class MesajYaz
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Android uygulama geliştirme dersleri");
6     }
7 }
```

1. Örnekte gösterildiği üzere `MesajYaz` isimli bir sınıf yapısı tanımlanmış ve sınıfın kod bloğu içerisinde yer alan 3. satır üzerinde `main` isimli bir metod oluşturulmuştur. Bu metodun önemine binaen ayrıca üzerinde durmamız gerekiyor.
2. Java'da programların çalıştırılması için kullanılan `main` metodu aynı zamanda programların ana metodudur. Programlar ana metod olmadan çalıştırılmazlar. Java kodu çalıştırma işlemine sokulduğu zaman ilk olarak sınıf içerisinde tanımlanan `main` metoduna bakılır ve bu metod içerisinde yer alan kodlar çalıştırılır. `main` metodu projemizde yer alan sınıflar arasında herhangi birinin içerisinde bir kez tanımlanır. `main` metodu verilen örnek üzerinde gösterildiği gibi tanımlanmaktadır. Metod kavramını sonraki bölümlerde daha ayrıntılı ele alacağız.

```
21         Kitaplar kitap = new Kitaplar();
22         kitap.kitapAdi = "Android Programlama";
23         kitap.kitapYazari = "Anonim";
24         kitap.kitapKategorisi = "Mühendislik Kitabı";
25         MuhendislikKitaplari muhKitap = new MuhendislikKitaplari();
26         muhKitap.kitapTuru = "Bilgisayar" ;
27         // Oluşturulan kitabın bilgilerini yaz.
28         System.out.println("KITAP BİLGİLERİ");
29         System.out.println("Kitap Adı : " + kitap.kitapAdi);
30         System.out.println("Kitap Yazarı : " + kitap.kitapYazari);
31         System.out.println("Kitap Kategorisi : "+kitap.kitapKategorisi);
32         System.out.println("Kitap Turu : " + muhKitap.kitapTuru);
33
34     }
35
36 }
```

1. Verilen örnekte görüldüğü üzere **Kitaplar**, **MuhendisKitaplari** ve **CocukKitaplari** isimleriyle 3 tane sınıf yapısı bulunmaktadır. Amacımız bu sınıfların özelliklerini kullanarak ortaya bir kitap çıkarmak olacaktır.
2. Kitaplar sınıfını bu 3 sınıf arasında ana sınıf olarak kabul edebiliriz. Bu sınıf içerisinde her türlü kitabın ortak özelliği olarak kabul edilebilir olan özellikler tanımlıdır. **MuhendislikKitaplari** ve **CocukKitaplari** sınıfları da kendi kitap kategorilerine ait olan özellikleri içerisinde barındırmaktadır.
3. Bu sınıflara ait özellikler kendi kod blokları içerisinde yer alan değişken tanımlamalarıyla belirlenmektedir. Sınıfların içerisinde String ifadesiyle başlayan tanımlamaların hepsi birer değişkendir.

NOT

Değişkenleri kısaca verilerin tutulduğu ve saklandığı yapılar olarak tanımlayabiliriz. Sonraki derslerde bu kavram hakkında daha ayrıntılı bilgi edineceğiz.

4. 21. satırda görüldüğü üzere **Kitaplar** sınıfından **kitap** isminde bir nesne türetilmektedir. Sonrasında **22**, **23** ve **24**. satırlar üzerinde türetilen nesnenin öğelerine, oluşturulacak olan kitabın özelliklerine ait değerler atanmaktadır.
5. 25. satırda bu kez **MuhendislikKitaplari** isimli sınıftan **muhKitap** isimli bir nesne türetilmiş sonrasında bu satırın altında yer alan 26. satır üzerinde, bu nesneye ait **kitapTuru** isimli nesne ögesine erişilmekte ve değer ataması yapılmaktadır.
6. Son olarak **26** ve **33**. satırlar arasında nesne öğelerine atanan değerler ekrana çıktı olarak verilmektedir. Programın ekran çıktısı aşağıda gösterildiği gibi olacaktır.

8

ANDROID STUDIO İLE GELEN HAZIR UYGULAMA ŞABLONLARI

BU BÖLÜMDE

Giriş	288
Uygulama Şablonları	289
Activity Şablonları	327
Diğer Şablonlar	328
Neler Öğrendik?	328

Android Studio, Android uygulamalarının gelişim sürecini hızlandırmak ve kolay bir hale getirmek için geliştiricilere hazır şablonlar sunmaktadır.

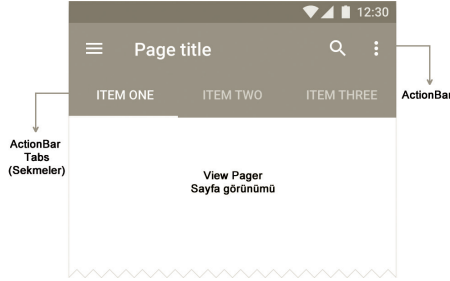
Bu bölümde, Android Studio içerisinde yer alan hazır uygulama şablonlarını ve bu şablonlar içerisinde yer alan uygulama modüllerini ele alacağız.

UYGULAMA ŞABLONLARI

Android Cihazları üzerinde çalıştırılabilinen ve test edilebilen temel android uygulama modüllerinin tümüne verilen isimdir. Bu şablon içerisinde yer alan uygulama modülleri yeni bir projenin activity türü olarak kullanılabilir olduğu gibi, mevcut projenin bir parçası olarak kullanılabilirler.

SEKMELİ SAYFALAR (TABBED ACTIVITY)

Türkçede sekmeli sayfalar olarak adlandırılan **Tabbed Activity**, adından da anlaşılacağı üzere bir activity ekranı üzerinde sekmeli sayfalar oluşturmamızı sağlayan bir uygulama modülüdür. Her sekmenin altında, farklı içerikler görüntüleyen sayfalar yer alır. Böylece bir activity ekranı üzerinde birçok içeriğin görüntülenmesi sağlanmış olunur. Tabbed yapısı aşağıda verilen görsel üzerinde gösterildiği üzere 3 kısımdan oluşmaktadır.



Android Studio içerisinde oluşturulan sekmeli sayfalar ön tanımlı olarak içerisinde **ActionBar** bulundurulur, ActionBar'ın hemen altında yer alan kısımda farklı sayfa içeriklerini görüntülemek için kullanılan sekmeler yer alır. Ekranın orta kısmında yer alan **View Pager** yapısı ise, sekmelerin altında yer alan sayfalar arasında geçişi sağlayan ve bu sayfaları bir arada tutan yapıdır. Şimdi sekmeli activity sayfalarının nasıl oluşturulduğunu görelim. Öncelikle **Tabbed** adında yeni bir proje oluşturuyoruz. Projenin activity türünde **Tabbed Activity** olarak seçiyoruz. Activity türünü seçtikten sonra karşımıza oluşturulacak olan Tabbed Activity ile ilgili ön tanımlamaların yapılacağı aşağıda gösterilen bir pencere çıkacaktır. Bu pencere üzerinde Tabbed Activity yapısını oluşturan kaynak ve *xml* dosyalarının isim bilgilerini belirleyeceğiz.

yer alan birinci alan `id` olduğundan, bu alanın indis numarası `0` olacaktır. Dolayısıyla `getString(0)` ifadesi tablonun `id` alanında bulunan veriyi elde edecektir.

KAYIT GÜNCELLEME (UPDATE)

Veritabanına daha öncesinde eklenmiş olan kayıtlar üzerinde değişiklik yapma ihtiyacı duyulabilir. Bu durumumda aşağıda verilen SQL söz dizimi kullanılır.

```
UPDATE TABLO_ADI SET ALAN_ADI = DEGER WHERE ALAN_ADI = DEGER ;
```

SQL sözdizimi `UPDATE` ile başlar ve bu ifadenin hemen ardından güncelleme işlemi yapılacak olan tablo adı yazılır. `SET` ifadesinden sonra güncelleme işlemlerinin yapılacağı alan adları ve bu alanlarda yer alacak yeni değerler yazılır. Güncelleme işlemi için verilecek olan koşul ifadeleri ise yine `WHERE` söz diziminden sonra yazılır. Aşağıda `UPDATE` SQL söz dizimi kullanılarak güncelleme işlemi yapmak için oluşturulmuş bir metod yapısı yer almaktadır.

```
public void ogrenciKaydiniGuncelle(int id) {
    final String TABLO_ADI = "ogrenci" ;
    String ogrenciAdi = "Çağlar";
    String ogrenciSoyadi = "Artar";
    SQLiteDatabase db = this.getWritableDatabase();
    String updateQuery = "UPDATE "+TABLO_ADI+" SET ogrenci_adi = "+ ogrenciAdi+"
, ogrenci_soyadi = " + ogrenciSoyadi+ " WHERE id = " + id + " ";
    db.execSQL(updateQuery);
    db.close();
}
```

Güncelleme işlemi için kullanılan bir diğer yöntemde ise `update` metodundan faydalanılmaktadır. Metod güncelleme işlemi için gerekli olan SQL sorgusunu kendisi oluşturur.

```
public void ogrenciKaydiniGuncelle(int ogrenci_id) {
    public final String TABLO_ADI = "ogrenci" ;
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put("ogrenci_adi", "Çağlar");
    values.put("ogrenci_soyadi", "Artar");
    db.update(TABLO_ADI, values, "id" + " = ?",
        new String[]{String.valueOf(ogrenci_id)});
    db.close();
}
```

Ekranın ortasında yer alan WebView arayüz elemanı içerisinde, adresi verilen web sayfaları görüntülenecektir. Ekranın en üstünde yer alan EditText ise gidilecek olan web adresinin yazılacağı alan olarak kullanılacak. Ekranında yer alan **Git** butonu adresi yazılan sayfaya gitmek, **Geri** butonu bir önceki ziyaret edilen sayfaya gitmek, **İleri** butonu ise geri butonuna basılmasıyla terk edilen bir önceki sayfaya gitmek için kullanılacaktır.

Tasarım dosyasının ardından *MainActivity.java* kaynak dosyamızı da aşağıda gösterilen şekliyle düzenliyoruz.

MainActivity.java

```
1 public class MainActivity extends Activity implements View.OnClickListener {
2     WebView webView;
3     EditText url;
4     Button search;
5     Button back;
6     Button forward;
7     String newUrl;
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13
14        search = (Button) findViewById(R.id.search);
15        back = (Button) findViewById(R.id.back);
16        forward = (Button) findViewById(R.id.forward);
17        url = (EditText) findViewById(R.id.url);
18
19        search.setOnClickListener(this);
20        back.setOnClickListener(this);
21        forward.setOnClickListener(this);
22
23        webView = (WebView) findViewById(R.id.webView);
24        webView.getSettings().setJavaScriptEnabled(true);
25        loadDefaultPage();
26
27    }
28
29    public void loadDefaultPage()
```