

GUI II

Mobilne aplikacije

Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 Adapteri

Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 Adapteri

Stilovi

- Stil je skup svojstava koja specificiraju izgled pogleda
- Neka od tih svojstava su visina (height), širina (width), popuna (padding), margina (margin), font (typeface), boja teksta (font color), veličina teksta (font size), boja pozadine (background color), itd.
- Stilovi se definišu kao poseban resurs (odgovara im XML dokument u res/values/themes direktorijumu) i moguće ih je nasleđivati

main_activity.xml

```
<TextView
2   android:id="@+id/text_view"
   android:text="@string/hello_world"
4   android:layout_width="match_parent"
   android:layout_height="wrap_content"
6   android:typeface="monospace"
   android:textSize="15sp"
8   android:textStyle="italic"
   android:textColor="#00FF00" />
10
```

main__activity.xml

```
<TextView
2   android:id="@+id/text_view"
   android:text="@string/hello_world"
4   android:layout_width="match_parent"
   android:layout_height="wrap_content"
6   style="@style/example_style" />
```

style.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <style name="example_style">
4         <item name="android:typeface">monospace</item>
5         <item name="android:textSize">15sp</item>
6         <item name="android:textStyle">italic</item>
7         <item name="android:textColor">#00FF00</item>
8     </style>
9 </resources>
10
```

style.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <style name="example_style" parent="@android:style/Theme.Material">
4         <item name="android:typeface">monospace</item>
5         <item name="android:textSize">15sp</item>
6         <item name="android:textStyle">italic</item>
7         <item name="android:textColor">#00FF00</item>
8     </style>
9 </resources>
10
```


Boje

- Bojama je moguće dodeljivati imena i definisati ih kao zaseban resurs.
- U datoteci `res/values/colors.xml` se svaka boja zasebno konfiguriše preko osnovnih komponenti (RGB ili ARGB).
- Jednom definisanu boju je moguće koristiti u stilovima, temama ili direktno na pogledima.

colors.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3
4      <color name="red">#FFFF0000</color>    <!-- referenced as @color/red -->
5
6      <color name="green">#FF00FF00</color>    <!-- referenced as @color/green -->
7
8      <color name="yellow">#FFFFFFF0</color>    <!-- referenced as @color/yellow -->
9
10 </resources>
```

Teme

- Tema je stil primenjen na celu aktivnost ili celu aplikaciju umesto na pojedinačan pogled
- Tada se na svaki pogled aktivnosti ili aplikacije primenjuje svako svojstvo teme koje pogled podržava
- Stilovi su lokalni, teme su globalne

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3     <application android:theme="@style/Theme.MyApplication" ... >
4         <!-- ... -->
5     </application>
6 </manifest>
```

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest ... >
3      <application ... >
4          <activity android:name=".ExampleActivity"
5              android:theme="@style/Theme.MyApplication" ... >
6              <intent-filter>
7                  <action android:name="android.intent.action.MAIN" />
8                  <category android:name="android.intent.category.LAUNCHER" />
9              </intent-filter>
10         </activity>
11     </application>
12 </manifest>
```

res/themes/themes.xml

```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2
3     <!-- Base application theme. -->
4     <style name="Theme.MyApplication" parent="Theme.MaterialComponents">
5
6         <item name="colorPrimary">@color/purple_500</item>
7
8         <item name="colorSecondary">@color/teal_200</item>
9
10        <item name="colorSecondaryVariant">@color/teal_700</item>
11
12    </style>
13
14 </resources>
```

Nasleđivanje stilova / tema

- Između stilova se može uspostaviti hijerarhijski odnos navođenjem roditeljskog stila iz kojeg se preuzimaju sve definicije, koje se mogu dopunjavati ili modifikovati.
- Isto važi i za teme.
- Roditeljski stil/tema se navodi u atributu parent ili se povezivanje vrši put prostora imena. Na primer, teme "MyApp.MyTheme.Light" i "MyApp.MyTheme.Dark" nasleđuju temu "MyApp.MyTheme".

Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 Adapteri

Material Design

- Material Design je skup principa za vizuelni dizajn, dizajn pokreta i dizajn interakcija
- Aplikacije dizajnirane po ovim principima pružaju korisnicima konzistentno iskustvo na različitim platformama (mobilnim, web i desktop) i u različitim aplikacijama
- Material Design koristi metafore da bi korisničko iskustvo bilo intuitivno

Material Design

Principi Material Design preporuka mogu se grupisati u tri kategorije:

- opipljive površine (tengable sufaces)
- smeo grafički dizajn (bold graphic design)
- smisleni pokreti (meaningful motion)

Opipljive površine

Senke simuliraju visinu listova papira koja određuje njihov međusobni odnos:

- seam (dva lista papira koji dele zajedničku ivicu se kreću zajedno)
- step (dva lista papira koji se preklapaju se kreću nezavisno)
- floating action button (dugme odvojeno od toolbar-a)

Smeo grafički dizajn

Na listovima se prikazuje:

- tekst (Roboto i Noto)
- fotografije, ilustracije i ikonografija (predefinisane ikone za uobičajene akcije)
- boje (primarna, sekundarna i akcentovana)

Smisleni pokreti

- autentični pokreti (pokreti treba da budu usklađeni sa masom, zapreminom i fleksibilnošću objekta)
- interakcija sa kratkim odzivom (aplikacije reaguju na akcije korisnika i obezbeđuju vizuelnu potvrdu)
- smisleni prelazi (prelazi treba da usmere pažnju korisnika i da budu glatki)

Material Design za Android

Android podržava Material Design tako što pruža:

- nove teme
- nove poglede (npr. RecyclerView, CardView, itd.)
- novi API za senke i animacije

build.gradle (module)

```
dependencies {  
2    // ...  
  
4    implementation 'com.google.android.material:material:1.8.0'  
  
6    // ..  
    }  
8
```

AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest ... >
3     <application android:theme="@style/Theme.MaterialComponents" ... >
4         <!-- ... -->
5     </application>
6 </manifest>
```


Material Design teme

Konstanta	Opis
MaterialComponents	tamna verzija
MaterialComponents.Light	svetla verzija
MaterialComponents.Light.DarkActionBar	svetla verzija sa toolbar-om

Table 1: Izabrane Material Design teme.

Reference

- Material Design, <https://material.io/>
- Material Design for Android,
<https://developer.android.com/guide/topics/ui/look-and-feel>

Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 Adapteri

Adapteri

- Adapteri povezuju poglede (naslednice `AdapterView` pogleda) i izvore podataka
- Postoje predefinisani adapteri koji povezuju različite poglede (`ListView`, `GridView`, `Spinner`, itd.) i različite izvore podataka (nizove, kolekcije, kursore, itd.)
- Moguće je napraviti adaptere koji povezuju proizvoljan pogled i proizvoljni izvor podataka

ArrayAdapter

- Povezuje TextView pogled (ili pogled koji sadrži TextView pogled) i niz ili kolekciju
- Automatski se poziva `toString()` metoda svakog objekta u nizu ili kolekciji i njena povrana vrednost se prikazuje u pogledu

activity_main.xml

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=". MainActivity">
10
    <ListView
12      android:id="@+id/listView"
        android:layout_width="match_parent"
14      android:layout_height="wrap_content"
        android:layout_weight="1" />
16
18 </LinearLayout>

```

MainActivity.java

```
1 public class MainActivity extends AppCompatActivity {  
2  
3     @Override  
4     protected void onCreate(Bundle savedInstanceState) {  
5         super.onCreate(savedInstanceState);  
6         setContentView(R.layout.activity_main);  
7  
8         String[] days = {"Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"};  
9         ArrayAdapter arrayAdapter = new ArrayAdapter<this, R.layout.list_item, days>;  
10        ListView listView = findViewById(R.id.listView);  
11        listView.setAdapter(arrayAdapter);  
12    }  
13 }  
14
```

list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
2 <TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textView"
4     android:layout_width="wrap_content"
    android:layout_height="wrap_content"
6     android:layout_weight="1"
    android:text="TextView">
8 </TextView>
```


CursorAdapter

- CursorAdapter je adapter koji koristi kursor kao izvor podataka
- Kursor sadrži rezultat upita nad bazom podataka (više o kurzorima na jednom od narednih časova)

Custom Adapters

- Moguće je definisati adapter koji koristi proizvoljan izvor podataka
- Potrebno je definisati klasu koja nasleđuje `Adapter` ili `BaseAdapter` i redefinisati njene metode

ExampleAdapter.java

```
1 public class ExampleAdapter extends BaseAdapter {  
2  
3     Activity activity;  
4  
5     public ExampleAdapter(Activity activity) {  
6         this.activity = activity;  
7     }  
8  
9     @Override  
10    public int getCount() {  
11        // return item count  
12    }  
13  
14    @Override  
15    public Object getItem(int position) {  
16        // return item at position  
17    }  
18  
19    @Override  
20    public long getItemId(int position) {  
21        // return item ID at position  
22    }  
23  
24    @Override  
25    public View getView(int position, View convertView, ViewGroup parent)  
26    {  
27        // return view at position  
28    }  
29 }
```

ExampleAdapter.java

```
@Override
2 public View getView(int position, View view, ViewGroup parent) {

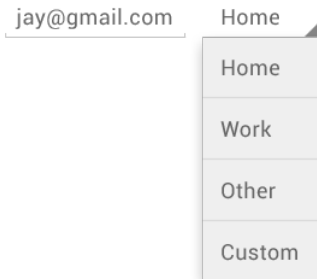
4     if (view == null) {
        view = activity.getLayoutInflater().inflate(R.layout.example_adapter, null);
6     }

8     TextView tvName = (TextView) view.findViewById(R.id.tv_name);
    tvName.setText(...);
10    TextView tvDescription = (TextView) view.findViewById(R.id.tv_description);
    tvDescription.setText(...);
12    ImageView ivIcon = (ImageView) view.findViewById(R.id.iv_icon);
    ivIcon.setImageResource(...);
14
    return view;
16 }
```

example_adapter.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent" >
6
7     <ImageView android:id="@+id/iv_icon" />
8
9     <TextView android:id="@+id/tv_name" />
10
11    <TextView android:id="@+id/tv_description" />
12
13 </LinearLayout>
14
```

Spinner



- Spinner pogled prikazuje stavke u meniju (korisnik može da izabere jednu stavku iz menija)
- Stavke se preuzimaju iz adaptera koji je pridružen pogledu

Figure 1: Spinner.

spinner.xml

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <Spinner
8     android:id="@+id/spinner"
9     android:layout_width="match_parent"
10    android:layout_height="wrap_content" />
11
12 </LinearLayout>
```

SpinnerActivity.java

```

1 public class SpinnerActivity extends Activity {
2
3     @Override
4     public void onCreate(Bundle state) {
5         // ...
6         List<String> list = populate();
7         ArrayAdapter adapter =
8             new ArrayAdapter<String>(this, android.R.layout.spinner_item, list);
9         Spinner spinner = (Spinner) findViewById(R.id.spinner);
10        spinner.setAdapter(adapter);
11
12        spinner.setOnItemSelectedListener(
13            new AdapterView.OnItemSelectedListener() {
14                public void onItemSelected(AdapterView<?> parent, View view,
15                    int position, long id) {
16
17                    Intent intent =
18                        new Intent(SpinnerActivity.this, SecondActivity.class);
19                    intent.putExtra("position", position);
20                    intent.putExtra("id", id);
21                    startActivity(intent);
22
23                }
24                public void onNothingSelected(AdapterView<?> adapterView) {
25
26                }
27            });
28    }

```


ListView



Figure 2: ListView pogled.

- ListView pogled prikazuje listu stavki (koja može da se skroluje)
- Stavke se preuzimaju iz adaptera koji je pridružen pogledu

list_view.xml

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <ListView
8     android:id="@+id/list_view"
9     android:layout_width="wrap_content"
10    android:layout_height="wrap_content" />
11
12 </LinearLayout>
```

ListViewActivity.java

```
1 public class ListViewActivity extends Activity {  
2  
3     @Override  
4     protected void onCreate(Bundle state) {  
5  
6         // ...  
7         List<String> list = populate();  
8         ArrayAdapter adapter = new ArrayAdapter(  
9             this, android.R.layout.simple_list_item_1, list);  
10  
11         ListView listView = (ListView) findViewById(R.id.list_view);  
12         listView.setAdapter(adapter);  
13     }  
14 }
```

GridView

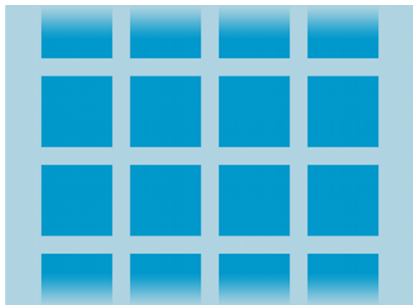


Figure 3: GridView pogled.

- GridView pogled prikazuje tabelu stavki (koja može da se skroluje)
- Stavke se preuzimaju iz adaptera koji je pridružen pogledu

grid_view.xml

```
1 <LinearLayout
2   xmlns:android="http://schemas.android.com/apk/res/android"
3   android:orientation="vertical"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent">
6
7   <GridView
8     android:id="@+id/grid_view"
9     android:numColumns="auto_fit"
10    android:gravity="center"
11    android:columnWidth="50dp"
12    android:layout_width="match_parent"
13    android:layout_height="match_parent" />
14
15 </LinearLayout>
16
```

gridview_item.xml

```
<CheckedTextView
2   xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@android:id/checked_text"
4   android:layout_width="wrap_content"
   android:layout_height="match_parent"
6   android:layout_weight="0.9"
   android:gravity="center_vertical" />
8
```

GridViewActivity.java

```
1 public class GridViewActivity extends Activity {  
2  
3     @Override  
4     public void onCreate(Bundle state) {  
5         // ...  
6         List<String> list = populate() ;  
7         ArrayAdapter adapter = new ArrayAdapter(  
8             this, R.layout.gridview_item, list);  
9  
10        GridView gridView = (GridView) findViewById(R.id.grid_view);  
11        gridView.setAdapter(adapter);  
12    }  
13 }  
14
```