



*Projektas „Pedagogų kvalifikacijos tobulinimo ir perkvalifikavimo sistemos plėtra (III etapas)“, Nr. VP1-2.2-ŠMM-02-V-01-010
Projekto vykdytojas – Ugdymo plėtotės centras*

Stażuotės vieta	UAB DEVBRIDGE LT
Stażuotės vykdymo laikotarpis	2014 m. rugsėjo 1 d. – 2015 m. vasario 28 d.
Stażuotojas	Edita Vinčienė
Stażuotojo vadovas (mentorius)	Viktoras Gurgždys

Užduotys mokiniams tema „Programavimas mobiliems įrenginiams“. 4 dalis

Turinys

Praktinis darbas Nr. 21 Sąlyginis sakiny. Mygtuko aktyvavimo metodai	3
Praktinis darbas Nr. 22 Kelių aktyvių langų mobili aplikacija. Sąlyginio sakinio panaudojimas	5
Praktinis darbas Nr. 23 Pranešimas su paveikslu.....	8
Praktinis darbas Nr. 24 Rėmelio maketas (<i>FrameLayout</i>).....	11
Praktinis darbas Nr. 25 Žymės langelio (<i>CheckBox</i>) veikimas.....	14
Praktinis darbas Nr. 26 Jungiklio (<i>ToggleButton</i>) veikimas	16
Praktinis darbas Nr. 27 Sąrašas su piktogramomis	19

Praktinis darbas Nr. 21

Sąlyginis sakiny. Mygtuko aktyvavimo metodai

Darbo tikslas – mygtukų aktyvavimo metode panaudoti sąlyginį sakinį.

Darbo eiga:

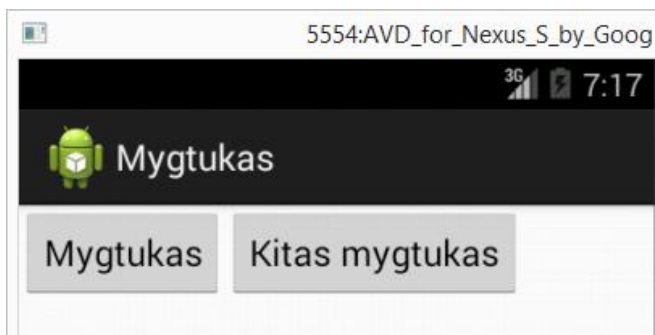
1. Kurkite naują mobilią aplikaciją **Mygtukas**.
2. Įkelkite du mygtukus su jų vardais (ID) ir aktyvavimo veiksmu „mygtukoVeiksmas“:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context="com.example.myapplication" >

    <Button
        android:id="@+id/button1"
        android:onClick="mygtukoVeiksmas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mygtukas" />

    <Button
        android:id="@+id/button2"
        android:onClick="mygtukoVeiksmas"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Kitas mygtukas" />

</LinearLayout>
```



3. Aprašykite metodą **mygtukoVeiksmas**:

```
package com.example.mygtukas;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

public class MygtukasActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mygtukas);
    }

    public void mygtukoVeiksmas(View v)
    {
        Log.d("VIVZ", v.getId()+"Mygtukas paspaustas");
    }

}
```

4. Testuokite programą. Paspaudus vieną ir kitą mygtukus, bus išvedami pranešimai (jei pamiršote kaip juos peržiūrėti, žiūrėkite Praktinį darbą Nr. 14):

```
D 10-15 07:14:10.328 2575 2575 com.example.mygtukas VIVZ 2131034172Mygtukas paspaustas
D 10-15 07:14:15.868 2575 2575 com.example.mygtukas VIVZ 2131034173Mygtukas paspaustas
```

Writable

Smart Insert

23 : 1

134M of 682M

5. **2131034172** ir **2131034173** yra objektų numeriai, todėl galite panaudoti šią `getId()` funkciją, tam kad patikrinti, kuris mygtukas paspaustas:

```
package com.example.mygtukas;

import android.app.Activity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;

public class MygtukasActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_mygtukas);
    }

    public void mygtukoVeiksmas(View v)
    {
        if(v.getId()==R.id.button1)
        {
            Log.d("VIVZ", "Paspaustas pirmas mygtukas");
        }
        else if(v.getId()==R.id.button2)
        {
            Log.d("VIVZ", "Paspaustas kitas mygtukas");
        }
    }
}
```

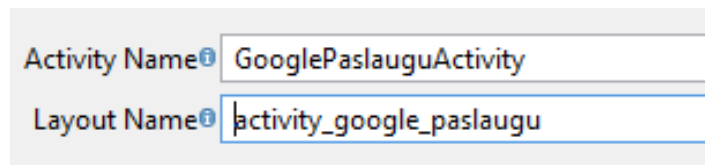
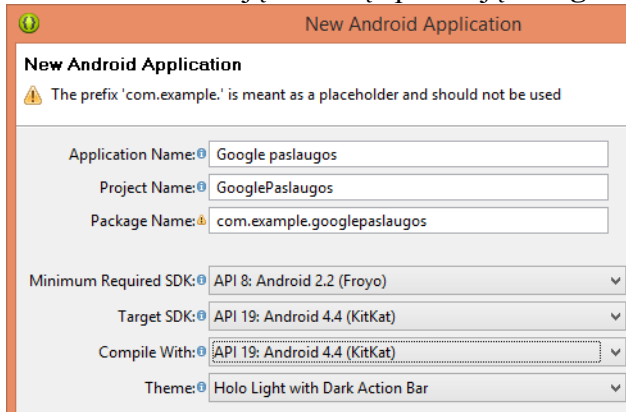
Praktinis darbas Nr. 22

Kelių aktyvių langų mobili aplikacija. Sąlyginio sakinio panaudojimas

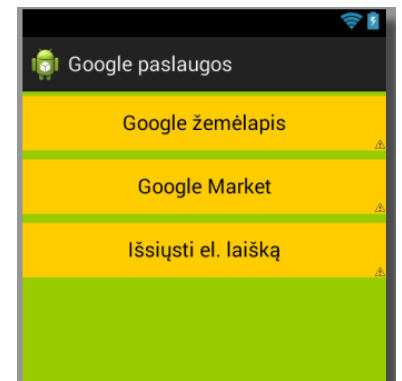
Darbo tikslas – sukurti kelių langų mobilią aplikaciją, lango aktyvavimo pasirinkimui panaudoti sąlyginį sakinį.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją **GooglePaslaugos**:



2. Įkelkite ir vienas paskui kitą išdėliokite tris mygtukus:



3. Vaizdui pagražinti kiekvienam mygtukui uždėkite:

- paraštes viršuje ir apačioje:
`android:layout_marginTop="4dp"`
`android:layout_marginBottom="4dp"`
- spalvą:
`android:background="#FFCC00"`
- spalvą visam langui:
`android:background="#99CC00"`

Visas `activity_google_paslaugu.xml` failas:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#99CC00"
    tools:context="$${relativePackage}.${activityClass}" >

    <Button
        android:id="@+id/zemelapis"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        android:background="#FFCC00"
        android:onClick="veiksmas"
        android:text="Google žemėlapis" />
```

```

<Button
    android:id="@+id/market"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/zemelapis"
    android:layout_alignLeft="@+id/zemelapis"
    android:layout_below="@+id/zemelapis"
    android:layout_marginTop="4dp"
    android:layout_marginBottom="4dp"
    android:background="#FFCC00"
    android:onClick="veiksmas"
    android:text="Google Market" />

<Button
    android:id="@+id/laiskas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/market"
    android:layout_alignLeft="@+id/market"
    android:layout_below="@+id/market"
    android:layout_marginTop="4dp"
    android:layout_marginBottom="4dp"
    android:background="#FFCC00"
    android:onClick="veiksmas"
    android:text="Išsiųsti el. laišką" />

</RelativeLayout>

```

4. Aprašykite metodą veiksmas:

- pirmiausia aprašykite tris IF funkcijas:

```

public void veiksmas(View view) {
    if (view.getId() == R.id.zemelapis) {
    }

    if (view.getId() == R.id.market) {
    }

    if (view.getId() == R.id.laiskas) {
    }
}

```

- Sukurkite **Intent** mygtukui „Google žemėlapis“, kurį paspaudus programa atvers žemėlapio vaizdą su nurodytomis koordinatėmis (**latitude** ir **longitude**):

```

public void veiksmas(View view) {
    Intent intent=null;
    if (view.getId() == R.id.zemelapis) {

        intent=new Intent (android.content.Intent.ACTION_VIEW);
        intent.setData (Uri.parse ("geo:54.893645,23.924391"));
        //** tai yra koordinatės, kurias atvers žemėlapyje **
        startActivity(intent);
    }
    ...
    ...
}

```

- Koreguokite programos kodą įtraukdami pasirinkimo veiksmą **chooser**, kuris parodys, visus akt. langus jūsų telefone, kurie naudoja šią Intent:

```

public void veiksmas(View view) {
    Intent intent=null, chooser=null;
    if (view.getId() == R.id.zemelapis) {

        intent=new Intent (android.content.Intent.ACTION_VIEW);
        intent.setData (Uri.parse ("geo:54.893645,23.924391"));
        //** tai yra koordinatės, kurias atvers žemėlapyje **
        chooser=Intent.createChooser(intent, "Pasirinkite:");
        startActivity(chooser);
    }
    ...
    ...
}

```

5. Testuokite programą. Programos veikimą galima patikrinti tik su realiu prietaisu turinčiu prieigą prie interneto.
6. Dabar aprašykite veiksmus antram aktyvaus lango mygtukui „Google Market“. Veiksmų seka bus panaši:
 - Atsiverskite internete google play paslaugų puslapį adresu:
https://play.google.com/store/apps/details?id=dolphin.developers.com
 - Šitą adresą panaudosite kurdami programos URI. Nukopijuokite jį į URI laukelį, ir pakeiskite taip:
market://details?id=dolphin.developers.com

Visa IF funkcija:

```

if (view.getId() == R.id.market) {

    intent=new Intent (android.content.Intent.ACTION_VIEW);
    intent.setData (Uri.parse ("market://details?id=dolphin.developers.com"));
    /** tai yra puslapio uri adresas */
    chooser=Intent.createChooser(intent, "Ijunk Market");
    startActivity(chooser);
}

```

7. Testuokite programą telefone, turinčiame prieigą prie interneto.
8. Dabar aprašykite veiksmus trečiam aktyvaus lango mygtukui „Išsiųsti el. laišką“:

```

if (view.getId() == R.id.laiskas) {

    intent=new Intent (android.content.Intent.ACTION_SEND);
    intent.setData (Uri.parse ("mailto:"));
    String[] to={"iteditvin@gmail.com", "geramdraugui@gmail.com" };
    intent.putExtra (Intent.EXTRA_EMAIL, to);
    intent.putExtra (Intent.EXTRA_SUBJECT, "Tai siunciama is mano app");
    intent.putExtra (Intent.EXTRA_TEXT, "Sveiki, kaip laikotės? Tai yra mano pirma žinutė. ");
    intent.setType ("message/rfc822");
    chooser=Intent.createChooser(intent, "Pasirinkite:");
    startActivity(chooser);
}

```

Kiekviena eilutė aprašo tokius veiksmus:

- sukuria Intent objektą su ACTION_SEND;
 - apibrėžia URI kaip **mailto:** protokolą;
 - leidžia Intent objektui naudoti minėtą protokolą;
 - apibrėžia siuntėjus;
 - apibrėžia el.pašto temą;
 - apibrėžia laiško turinį;
 - nurodo MIME tipą kaip rfc822 (skirta el. paštui)
 - įjungia pasirinkimo veiksmą.
9. Testuokite programą ir paspaudus mygtuką „Išsiųsti el. laišką“ turėtų pasiūlyti pasirinkimą, pagal jūsų telefone įrašytas programas el. paštui.

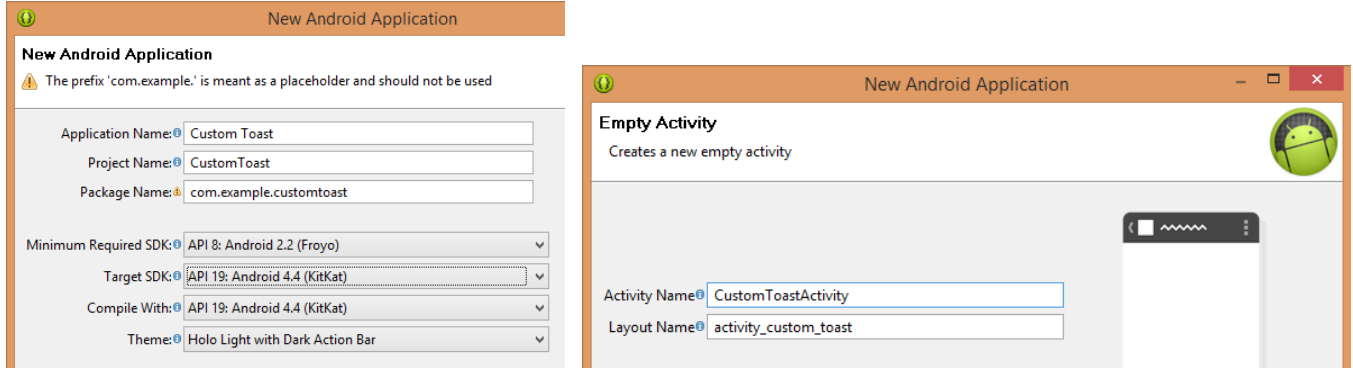
Praktinis darbas Nr. 23

Pranešimas su paveikslu

Darbo tikslas – sukurti mobilią aplikaciją su pranešimu, pranešime panaudoti grafinius objektus.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:

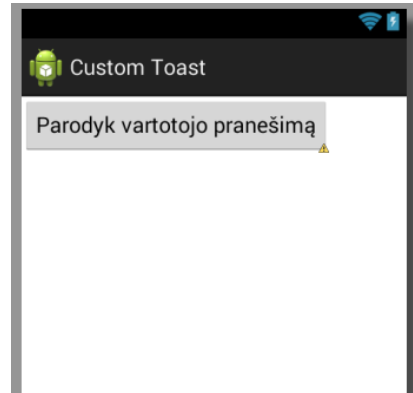


2. Pakeiskite maketo objektų išdėstymą, įkelkite mygtuką:

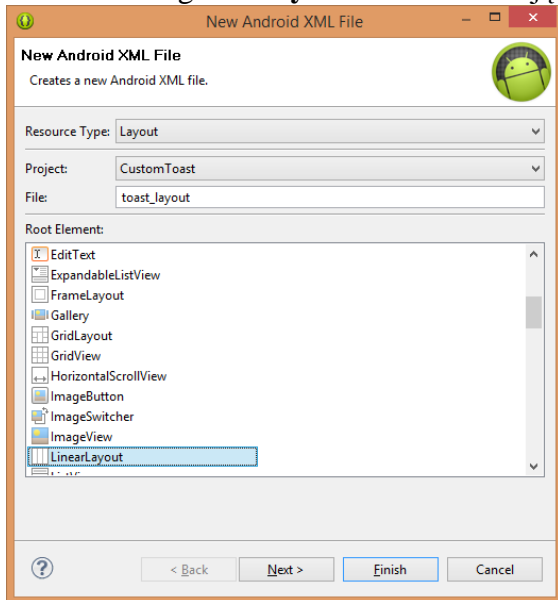
```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".${relativePackage}.${activityClass}" >

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Parodyk vartotojo pranešimą" />

</LinearLayout>
```



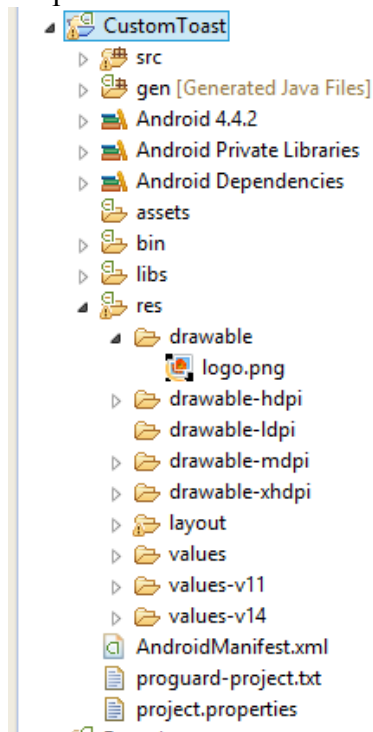
3. Kataloge **res/layout** sukurkite naują maketo failą **toast_layout.xml**:



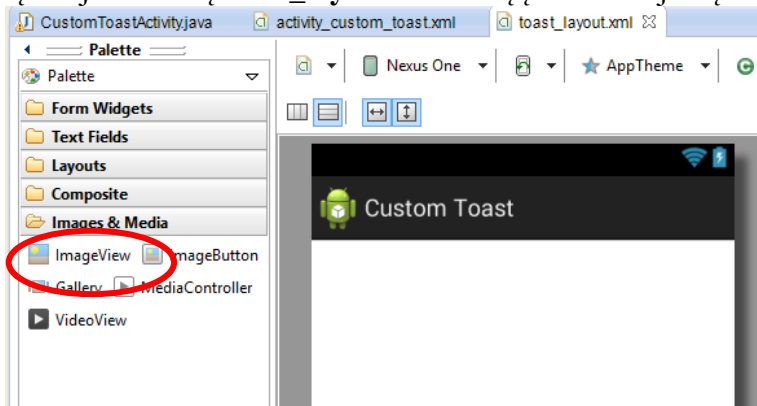
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

</LinearLayout>
```

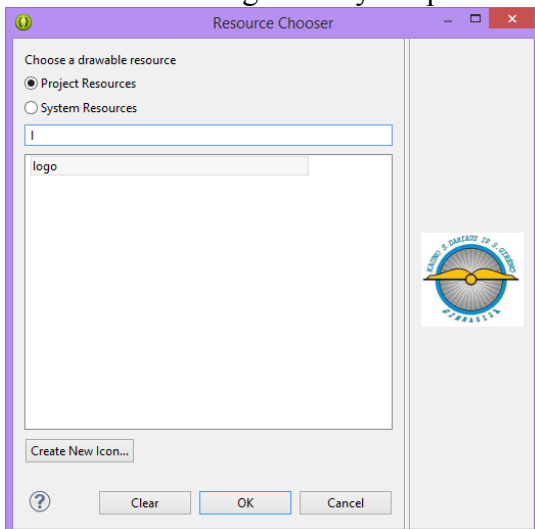

4. Kataloge **res** sukurkite dar viena katalogą **drawable** ir įkelkite į jį paveiksluką **logo.png**, kurį panaudosite savo pranešime:



5. Į naujai sukurtą **toast_layout.xml** failą įkelkite objektą **ImageView (Images&Media)**:



Atsidariusiame lange nurodykite paveiksluko pavadinimą:



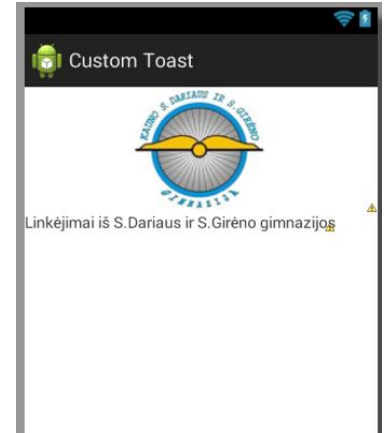
6. Po paveikslu pridėkite **TextView** (Form Widgets):

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:src="@drawable/logo" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Linkėjimai iš S.Dariaus ir S.Girėno gimnazijos" />

</LinearLayout>
```



7. Įrašykite pagrindinio mygtuko veiksmus:

```
android:onClick="pranesimas"
```

8. Identifikuokite pranešimo maketą, pridėdami eilutę:

```
android:id="@+id/root"
```

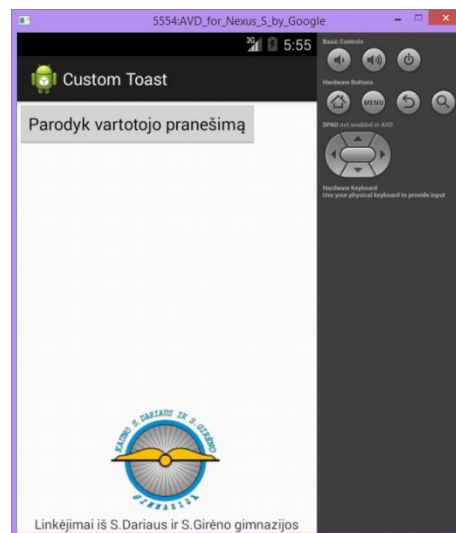
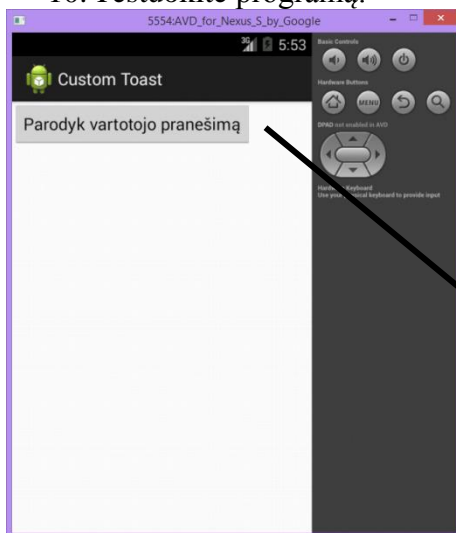
9. Aprašykime patį metodą pagrindiniame java faile:

```
public void pranesimas (View v) {
    Toast toast=new Toast (this);
    toast.setDuration (Toast.LENGTH_LONG);
    toast.setGravity (Gravity.BOTTOM, 0, 0);

    LayoutInflater inflater=getLayoutInflater();
    View apperance=inflater.inflate (R.layout.toast_layout, (ViewGroup) findViewById (R.id.root));
    toast.setView (apperance);
    toast.show ();
}
```

Pastaba: su **LinearLayout** maketu naudojama **ViewGroup**, visi maketai yra **ViewGroup** poklasė.

10. Testuokite programą:

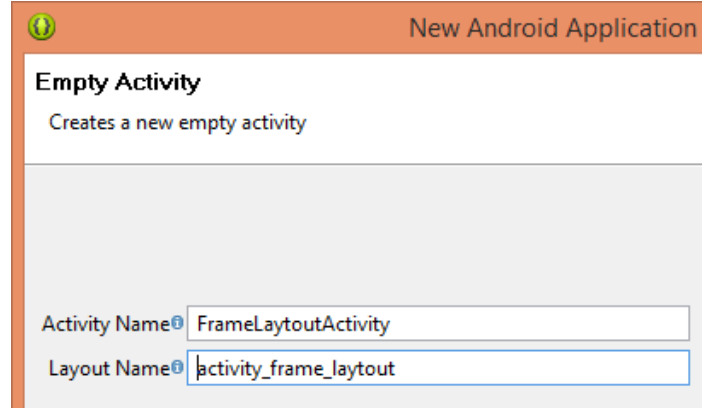
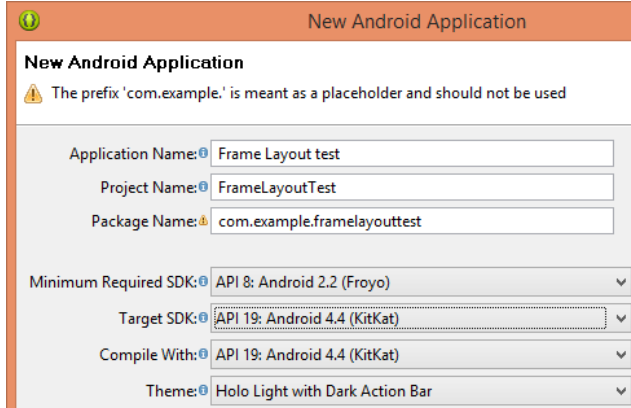


Praktinis darbas Nr. 24 Rėmelio maketas (*FrameLayout*)

Darbo tikslas – kurti vartotojo sąsają naudojant rėmelio maketą, *Java* programoje naudoti sąlyginius sakinius.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:

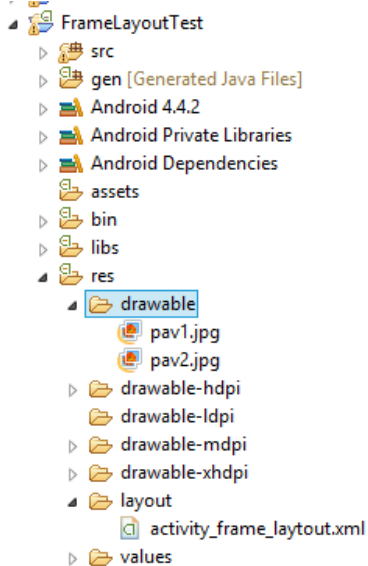


2. `activity_frame_layout.xml` faile ištrinkite tekstinį lauką, o maketą pakeiskite į **FrameLayout**:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="$.{relativePackage}.${activityClass}" >

</FrameLayout>
```

3. Kataloge `res` sukurkite katalogą `drawable` ir įkelkite į jį du nedidelius paveikslukus iš interneto (480x640):



4. Į maketą įkelkite pirmą paveiksluką:

```
<ImageView
    android:id="@+id/imageView1"
    android:layout_gravity="left|top"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/pav1" />
```

5. Į maketą įkelkite antrą paveiksluką:

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".${relativePackage}.${activityClass}" >

  <ImageView
    android:id="@+id/imageView1"
    android:layout_gravity="left|top"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/pav1" />

  <ImageView
    android:id="@+id/imageView2"
    android:layout_gravity="left|top"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/pav2" />

</FrameLayout>
```



6. Pridėkite paveiksluko savybę **visibility="gone"**, tuomet antro paveiksluko nesimatys:

```
<ImageView
  android:id="@+id/imageView2"
  android:layout_gravity="left|top"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:visibility="gone"
  android:src="@drawable/pav2" />
```

7. **FrameLayoutActivity.java** faile aprašykite veiksmus, kad paspaudus ant vieno paveiksluko, atsi-vertų kitas ir t.t.

- aprašykite kintamuosius:

```
public class FrameLayoutActivity extends Activity {

    ImageView p1, p2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_frame_layout);
    }
}
```

- aprašykite veiksmus:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_frame_layout);
    p1 = (ImageView) findViewById(R.id.imageView1);
    p2 = (ImageView) findViewById(R.id.imageView2);
    p1.setOnClickListener(this);
    p2.setOnClickListener(this);
}
```

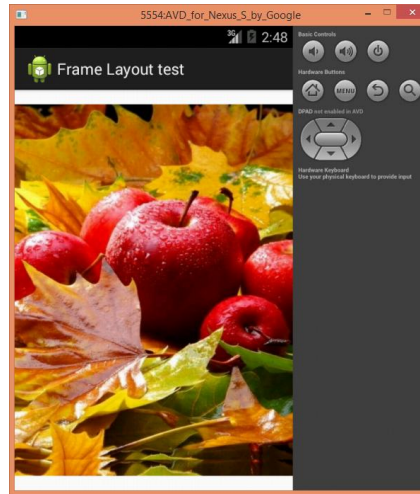
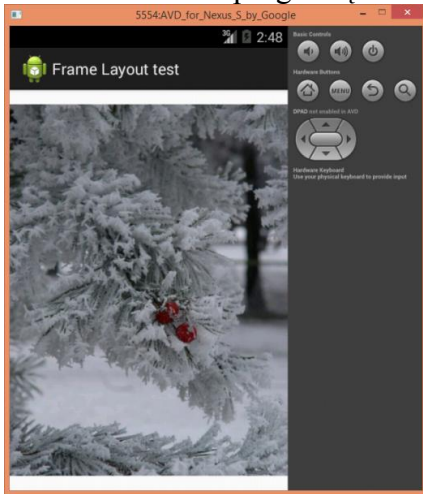
-

```
public class FrameLayoutActivity extends Activity implements View.OnClickListener {
```

- jei šioje vietoje rodytų klaidą (nes naudojate metodą, kuris dar nesukurtas), paspaudę pele ant žodžio **implements**, pasirinkite **add implements methods**:

```
@Override
public void onClick(View v) {
    // TODO Auto-generated method stub
    if (v.getId()==R.id.imageView1)
    {
        p1.setVisibility(View.GONE);
        p2.setVisibility(View.VISIBLE);
    }
    else
    {
        p1.setVisibility(View.VISIBLE);
        p2.setVisibility(View.GONE);
    }
}
```

8. Testuokite programą:



Visas **FrameLayoutActivity.java** failas:

```
package com.example.frameLayoutouttest;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;

public class FrameLayoutActivity extends Activity implements View.OnClickListener {

    ImageView p1, p2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_frame_layout);
        p1 =(ImageView) findViewById(R.id.imageView1);
        p2 =(ImageView) findViewById(R.id.imageView2);
        p1.setOnClickListener(this);
        p2.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (v.getId()==R.id.imageView1)
        {
            p1.setVisibility(View.GONE);
            p2.setVisibility(View.VISIBLE);
        }
        else
        {
            p1.setVisibility(View.VISIBLE);
            p2.setVisibility(View.GONE);
        }
    }
}
```

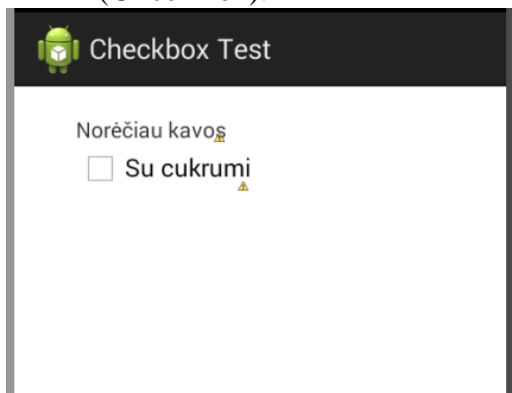
Praktinis darbas Nr. 25

Žymės langelio (*CheckBox*) veikimas

Darbo tikslas – įterpti į mobilią aplikaciją žymės langelį, naudoti jo formatavimo gaires ir jų atributus. Objekto aktyvavimui naudoti sąlyginį sakinį.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją.
2. Aplikacijos lange patalpinkite du objektus – teksto lauką (**TextView**) ir pasirinkimo langelį (**CheckBox**):



```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="30dp"
    android:text="Norėčiau kavos" />

<CheckBox
    android:id="@+id/checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/textView"
    android:layout_below="@+id/textView"
    android:text="Su cukrumi"
    android:checked="false"/>
```

Pastaba: parametras **checked** gali turėti dvi reikšmes: **true** (pažymėtas langelis) arba **false** (nepažymėtas langelis).

3. Redaguokite *.java failą, kuriame surašykite visus reikalingus veiksmus:

- atsidarykite failą redagavimui:

```
package com.example.checkboxtest;

import android.app.Activity;
import android.os.Bundle;

public class ChechBoxActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_check_box);
    }
}
```

- aprašykite kintamąjį **c**:

```
public class CheckboxActivity extends Activity {

    CheckBox c;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_checkbox);
    }
}
```

- ir jo veiksmus:

```
public class CheckboxActivity extends Activity implements View.OnClickListener{

    CheckBox c;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_checkbox);
        c= (CheckBox) findViewById(R.id.checkbox);
        c.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
    }
}
```

- automatiškai sugeneruotas metodas onClick tikrins ar checkbox jungiklis yra išjungtas ar įjungtas:

```
@Override
public void onClick(View v) {
    CheckBox t= (CheckBox) v;
    if(t.isChecked())
    {

    }
    else
    {

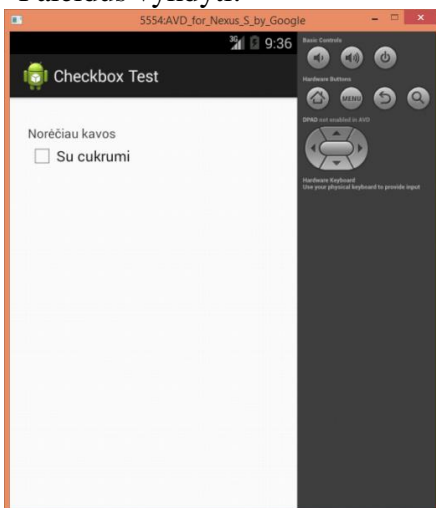
    }
}
}
```

- Priklausomai nuo to, ar langelis bus pažymėtas, ar ne, programa išves į ekraną pranešimą (Toast):

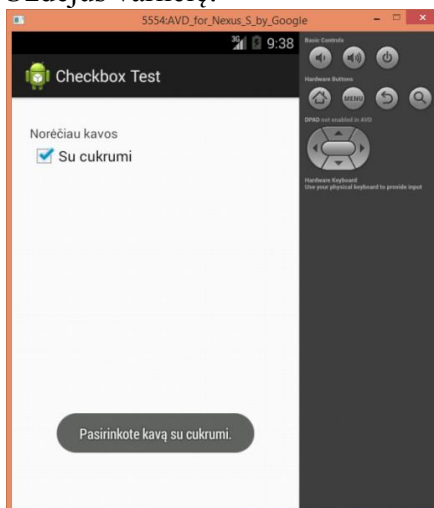
```
if(t.isChecked())
{
    Toast.makeText(this, "Pasirinkote kavą su cukrumi.", Toast.LENGTH_SHORT).show();
}
else
{
    Toast.makeText(this, "Pasirinkote kavą be cukraus.", Toast.LENGTH_SHORT).show();
}
}
```

4. Testuokite programą:

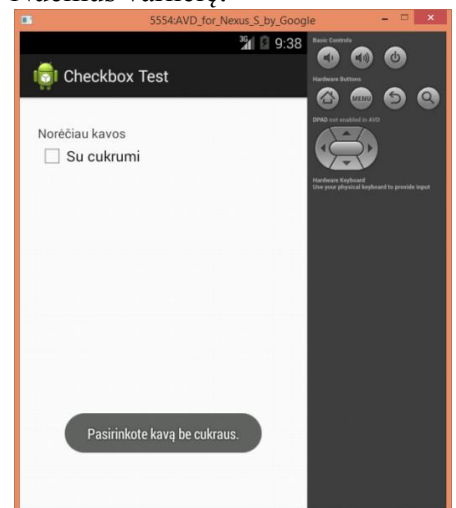
Paleidus vykdyti:



Uždėjus varnelę:



Nuėmus varnelę:



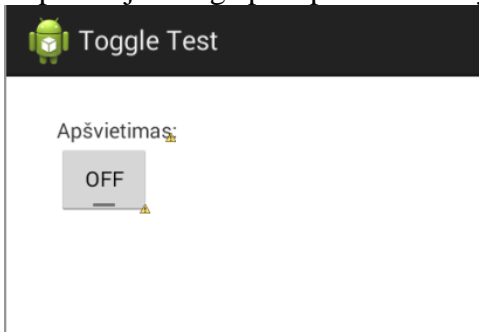
Praktinis darbas Nr. 26

Jungiklio (*ToggleButton*) veikimas

Darbo tikslas – įterpti į mobilią aplikaciją jungiklį, naudoti jo formatavimo gaires ir jų atributus. Objekto aktyvavimui naudoti sąlyginį sakinį.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją.
2. Aplikacijos lange patalpinkite du objektus – teksto lauką (**TextView**) ir jungiklį (**ToggleButton**):



```
<TextView
    android:id="@+id/tekstas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="30dp"
    android:layout_marginTop="30dp"
    android:text="Apšvietimas:" />

<ToggleButton
    android:id="@+id/jungiklis"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/tekstas"
    android:layout_below="@+id/tekstas"
    android:text="ToggleButton" />
```

3. Redaguokite *.java failą, kuriame surašysite visus veiksmus (padarysite taip, kad paspaudus jungiklį, programa pakeistų ekrano foną iš baltos spalvos į tamsią ir atvirkščiai):

- atsidarykite failą redagavimui:

```
package com.example.toggletest;

import android.app.Activity;
import android.os.Bundle;

public class ToggleActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_toggle);
    }
}
```

- aprašykite **ToggleButton** kintamąjį **t** ir jo veiksmus:

```
public class ToggleActivity extends Activity implements CompoundButton.OnCheckedChangeListener {

    ToggleButton t;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_toggle);
        t = (ToggleButton) findViewById(R.id.jungiklis);
    }

    public void onCheckedChanged (CompoundButton compoundbutton, boolean b){

    }
}
```


- Naujai sukurtame metode **onCheckedChanged** bus tikrinama loginio kintamojo **b** reikšmė (**true** arba **false**) – jungiklis įjungtas arba išjungtas – ir priklausomai nuo jungiklio būsenos keičiama fonų spalva. Kadangi veiksme dalyvauja fonas, tai jam turi būti suteikta identifikavimo žymė:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/maketas"
    tools:context="$${relativePackage}.${$activityClass}" >
```

- panaudodami funkciją IF:

```
public void onCheckedChanged (CompoundButton compoundbutton, boolean b){
    if (b)
    {

    }
    else
    {

    }
}
```

- aprašykite mygtuko veiksmus (fonas juodas arba baltas). Tam papildomai įveskite maketo kintamąjį **r**:

```
RelativeLayout r;
ToggleButton t;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_toggle);
    t= (ToggleButton) findViewById(R.id.jungiklis);
    r= (RelativeLayout) findViewById(R.id.maketas);
}
```

-

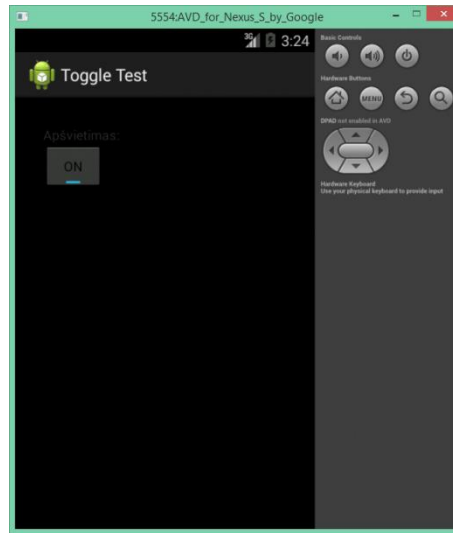
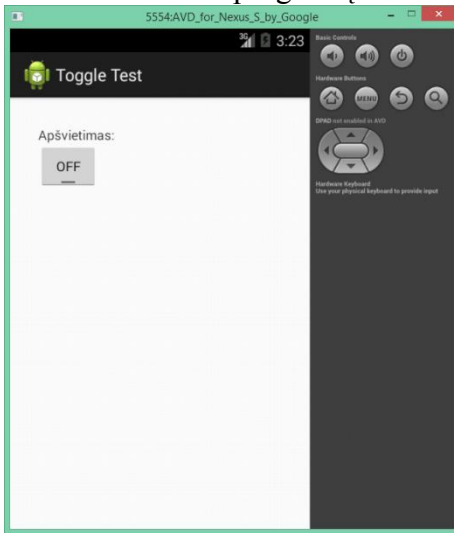
```
public void onCheckedChanged (CompoundButton compoundbutton, boolean b){
    if (b)
    {
        r.setBackgroundColor (Color.BLACK);
    }
    else
    {
        r.setBackgroundColor (Color.WHITE);
    }
}
```

- kad ši funkcija veiktų:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_toggle);
    t= (ToggleButton) findViewById(R.id.jungiklis);
    r= (RelativeLayout) findViewById(R.id.maketas);

    t.setOnCheckedChangeListener (this);
}
```

4. Testuokite programą:



Visas *.java failas:

```
package com.example.toggletest;

import android.app.Activity;
import android.graphics.Color;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.RelativeLayout;
import android.widget.ToggleButton;

public class ToggleButtonActivity extends Activity implements CompoundButton.OnCheckedChangeListener{

    RelativeLayout r;
    ToggleButton t;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_toggle_button);
        t= (ToggleButton) findViewById(R.id.jungiklis);
        r= (RelativeLayout) findViewById(R.id.maketas);

        t.setOnCheckedChangeListener(this);
    }

    public void onCheckedChanged (CompoundButton compoundbutton, boolean b){
        if (b)
        {
            r.setBackgroundColor (Color.BLACK);
        }
        else
        {
            r.setBackgroundColor (Color.WHITE);
        }
    }
}
```

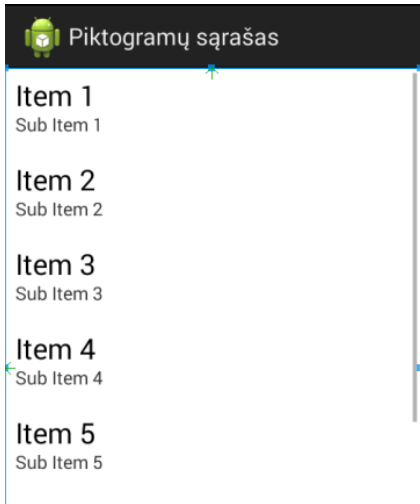
Praktinis darbas Nr. 27

Sąrašas su piktogramomis

Darbo tikslas – informaciją mobilijoje aplikacijoje pateikti sąrašu.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją.
2. Aplikacijos lange patalpinkite sąrašo objektą:



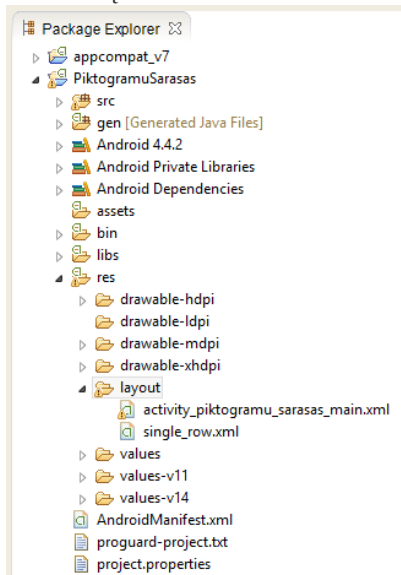
```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="{relativePackage}.{activityClass}" >

<ListView
    android:id="@+id/listView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true" >

</ListView>

</RelativeLayout>
```

3. Maketų aplanke sukurkite naują maketo failą **single_row**, kuriame nurodysite kaip turi atrodyti viena sąrašo eilutė:

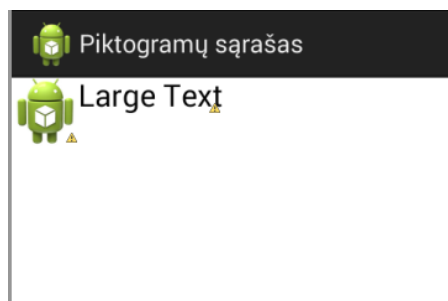


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher"
        android:layout_gravity="center_horizontal|top" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_gravity="center_horizontal|top" />

</LinearLayout>
```



4. Užrašą „Large Text“ galite ištrinti, palikdami tik kabutes, kurios nurodys teksto vietą eilutėje:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:layout_gravity="center_horizontal|top" />
```

5. Turite du *.xml maketo failus. Viename aprašėte kaip atrodys visas sąrašas, kitame – kaip atrodys viena sąrašo eilutė. Dabar *.java faile sujunkite tuos du failus, aprašydami veiksmus:

- aprašykite sąrašą:

```
public class PiktogramuSarasasMainActivity extends Activity {

    String[] sav_dienos={"Pirmadienis", "Antradienis", "Trečiadienis", "Ketvirtadienis",
                        "Penktadienis", "Šeštadienis", "Sekmadienis"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_piktogramu_sarasas_main);
    }
}
```

- aprašykite sąrašo kintamąjį:

```
public class PiktogramuSarasasMainActivity extends Activity {

    String[] sav_dienos={"Pirmadienis", "Antradienis", "Trečiadienis", "Ketvirtadienis",
                        "Penktadienis", "Šeštadienis", "Sekmadienis"};

    ListView l;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_piktogramu_sarasas_main);
        l=(ListView) findViewById(R.id.listView);
    }
}
```

- aprašykite sąrašo išvedimą į ekraną:

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_piktogramu_sarasas_main);
    l=(ListView) findViewById(R.id.listView);
    ArrayAdapter<String> adapter=new ArrayAdapter<String>
        (this, R.layout.single_row, R.id.textView, sav_dienos);
    l.setAdapter(adapter);
}
```

6. Testuokite programą:

