



ŠVIETIMO
IR MOKSLO
MINISTERIJA



UGDYMO
PLĖTOTĖS
CENTRAS

*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“. 3 dalis

Turinys

Praktinis darbas Nr. 10 Mobilios aplikacijos kūrimas naudojant Java programavimo kalbą. Teksto rašymas ..	3
Praktinis darbas Nr. 11 Mobilios aplikacijos kūrimas naudojant Java programavimo kalbą. Mygtuko įkėlimas	4
Praktinis darbas Nr. 12 Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą - 1	6
Savarankiškas darbas Nr 3 Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą	11
Praktinis darbas Nr. 13 Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą - 2.....	13
Praktinis darbas Nr. 14 Aktyvaus lango gyvavimo ciklo metodai.....	15
Praktinis darbas Nr. 15 Pranešimas ekrane (<i>Toast</i>)	17
Praktinis darbas Nr. 16 Vartotojo šriftų pritaikymas	19
Praktinis darbas Nr. 17 Dviejų langų mobili aplikacija, langų aktyvavimo metodai	21
Papildoma užduotis	22
Kitas lango aktyvavimo metodas	22
Praktinis darbas Nr. 18 Dviejų langų mobili aplikacija, teksto persiuntimo metodai	23
Praktinis darbas Nr. 19 Skambinimas ir skambučio priėmimas	26
Praktinis darbas Nr. 20 Skambinimo funkcijos tobulinimas.....	29

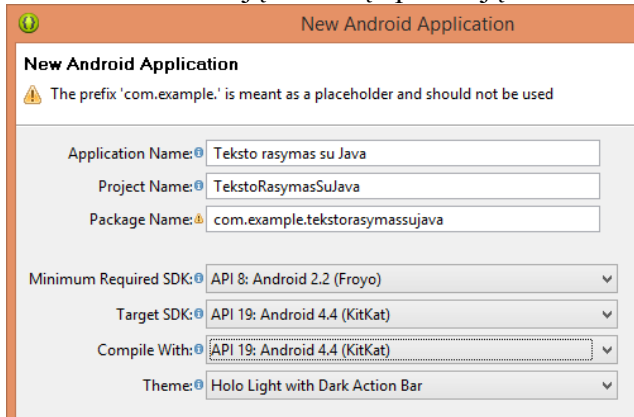
Praktinis darbas Nr. 10

Mobilios aplikacijos kūrimas naudojant Java programavimo kalbą. Teksto rašymas

Darbo tikslas – mobiloje aplikacijoje rašyti tekstą naudojant Java programavimo kalbą.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:



2. Ištrinkite teksto rašymo eilutes iš **activity_main.xml**:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context=".${relativePackage}.${activityClass}"
>
</RelativeLayout>
```

3. Ištrinkite maketo vaizdavimą ekrane **MainActivity.java** faile.

4. Įrašykite tris teksto pateikimo į ekraną eilutes **MainActivity.java** faile:

```
package com.example.tekstorasymassujava;

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

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

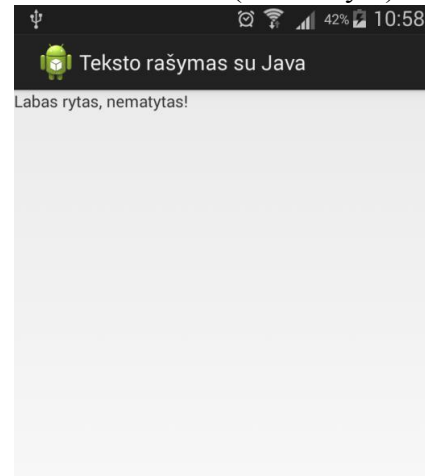
        TextView RodykTeksta = new TextView(this); // kuriame naują eilutę
        RodykTeksta.setText("Labas rytas, nematytas!"); // nurodom tekstą kurį eilutė rašys
        setContentView(RodykTeksta); // nurodom kad parodytų tekstą ekrane

        // ši eilutė naudoja setContentView() metoda, kuris nupiešia main.xml failą jūsų ekrane
        // setContentView(R.layout.activity_main);
    }
}
```

Vaizdas kompiuteryje (užrašo nėra):



Vaizdas telefone (užrašas yra):



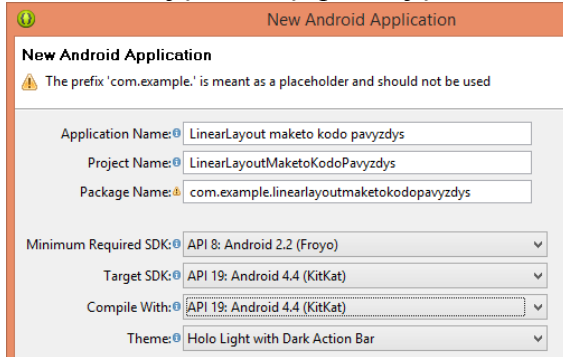
Praktinis darbas Nr. 11

Mobilios aplikacijos kūrimas naudojant Java programavimo kalbą. Mygtuko įkėlimas

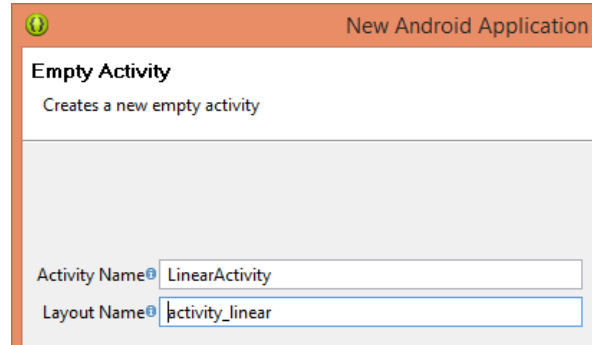
Darbo tikslas – mobiloje aplikacijoje įkelti mygtuką naudojant Java programavimo kalbą.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:



-->



2. Java faile aprašykite tris naujai kuriamus objektus ir juos apibūdinkite:

```
package com.example.linearlayoutmaketokodopavyzdys;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

public class MainActivity extends Activity {

    LinearLayout ll;
    TextView t;
    Button b;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ll=new LinearLayout(this);
        t=new TextView(this);
        b=new Button(this);

        setContentView(R.layout.activity_linear);
    }
}
```

3. Aprašykite visus jų veiksmus:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    ll=new LinearLayout(this);
    t=new TextView(this);
    b=new Button(this);

    LayoutParams dimensions=new LayoutParams(LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT);
    ll.setLayoutParams(dimensions);

    LayoutParams viewDimensions=new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
    t.setLayoutParams(viewDimensions);
    b.setLayoutParams(viewDimensions);

    ll.setOrientation(LinearLayout.VERTICAL);
    t.setText("Labas pasauli!");
    b.setText("mygtukas");
    ll.addView(t);
    ll.addView(b);

    setContentView(ll);
}
```

4. **Xml** failą palikite tuščią, be jokių grafinių objektų:

```
<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=".MainActivity" >

</RelativeLayout>
```

5. Testuokite programą, ir pamatykite jums jau pažįstamus objektus:



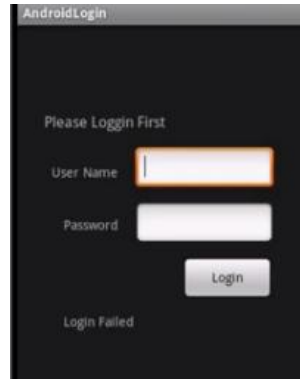
Praktinis darbas Nr. 12

Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą - 1

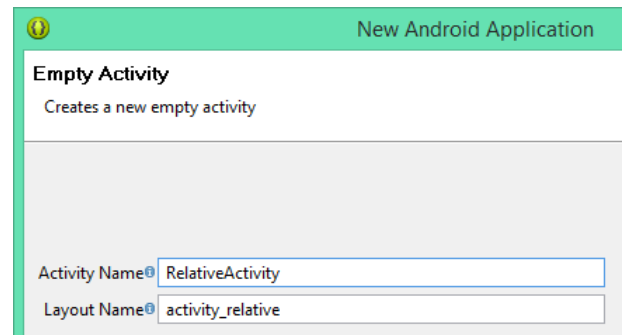
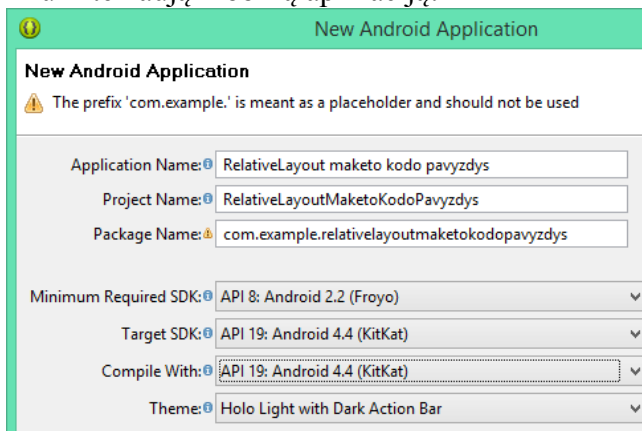
Darbo tikslas – susipažinti su *Java* programavimo kalbos metodais ir funkcijomis, *Java* failo struktūra. Sukurti mobilią aplikaciją su tekstu ir teksto įvedimo lauku, kurioje objektai aprašoma naudojant *Java* metodus.

Darbo eiga:

Suprojektuosite paprastą prisijungimo formą:



1. Kurkite naują mobilią aplikaciją:



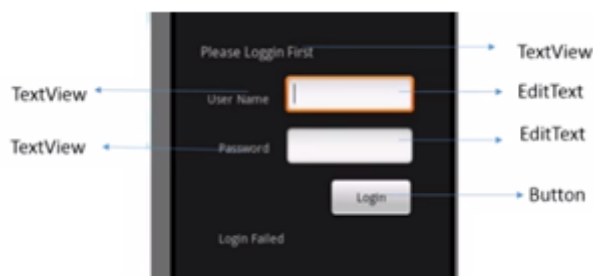
2. **activity_relative.xml** faile palikite tik pačio maketo aprašymą:

```
<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}" >

</RelativeLayout>
```

3. Visą kitą vaizdą, matomą ekrane, suformuosite faile **RelativeLayout.java**. Pirmiausia, **RelativeLayout.java** faile, iš kart po eilutės **public class RelativeLayout extends Activity {**, aprašykite vaizde matomus objektus:

- RelativeLayout `main`;
- EditText `virtotojoVardas`, `virtotojoSlaptazodis`;
- TextView `zinute`, `virtotojas`, `slaptazodis`;
- Button `prisijungti`;



4. Iš kart po metodo onCreate sukurkite procedūrą init(), kurioje aprašysite kiekvieno objekto veiksmus:

```
private void init() {
    // TODO Auto-generated method stub
    main = new RelativeLayout(this);
    vartotojoVardas = new EditText(this);
    vartotojoSlaptazodis = new EditText(this);
    zinute = new TextView(this); //Please login first
    vartotojas = new TextView(this);
    slaptazodis = new TextView(this);
    prisijungti = new Button(this);
}
```

5. Patį kreipinį į procedūrą įkelkite į metodą onCreate:

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

6. Aprašykite

- pagrindinio maketo vaizdą – procedūroje init():

```
main=new RelativeLayout(this);
LayoutParams mainDimensions=new LayoutParams(
    LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT);
main.setLayoutParams(mainDimensions);
```

- ir pirmojo objekto vaizdą – iš kart po procedūros init():

```
private void createMessageTextView() {
    // TODO Auto-generated method stub

    LayoutParams messageDimensions = new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
    messageDimensions.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
    zinute.setText("Please Login First");
    zinute.setLayoutParams(messageDimensions);
}
```

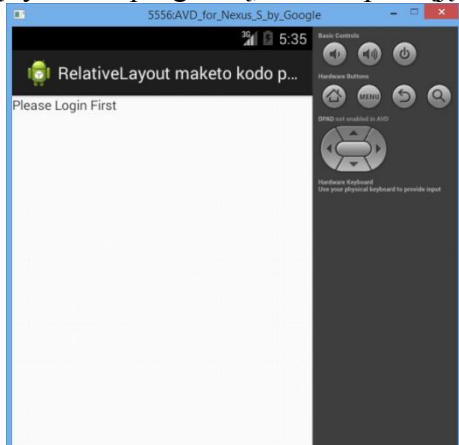
- Aprašykite naują parametą, prie objektų aprašų sąrašo (viršuje):

```
LayoutParams messageDimensions;
```

7. Koreguokite metodą onCreate, kad sukurtą vaizdą parodytų ekrane:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    init();
    createMessageTextView();
    main.addView(zinute, messageDimensions);
    setContentView(main);
}
```

8. Įvykdžius programą, matote pirmąjį objektą - pranešimo eilutę:



9. Toliau po objektu „zinate“ reikia įkelti objektus „vartotojas“ ir „vartotojoVardas“. Norėdami įkelti objektus vieną po kito, jiems turite suteikti identifikatorių:

```
int zinateId=1,...
```

Identifikatorius aprašykite iš karto po objektų aprašo (viršuje):

```
RelativeLayout main;  
EditText vartotojoVardas, vartotojoSlaptazodis;  
TextView zinate, vartotojas, slaptazodis;  
LayoutParams messageDimensions;  
Button prisijungti;  
int zinateId=1, vartotojoVardasId=2, vartotojoSlaptazodisId=3, vartotojasId=4,  
slaptazodisId=5, prisijungtiId=6;
```

Identifikatoriaus naudojimas aprašomas ir prie kiekvieno objekto savybių, viduje procedūros:

```
zinate.setId(zinateId);
```

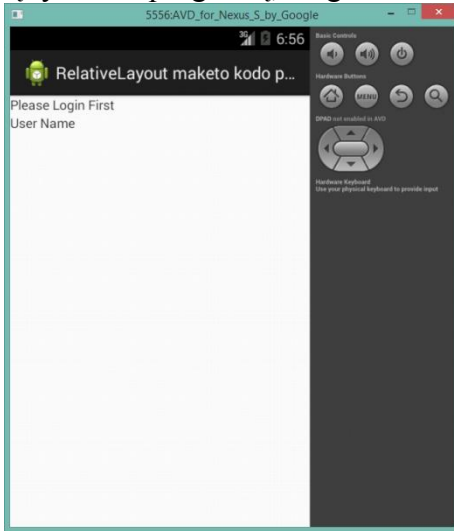
10. Kurkite naują procedūrą antrajam objektui:

```
private void createUserNameTextView() {  
    // TODO Auto-generated method stub  
    userNameDimensions = new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);  
    userNameDimensions.addRule(RelativeLayout.ALIGN_PARENT_LEFT);  
    userNameDimensions.addRule(RelativeLayout.BELOW, zinateId);  
    vartotojas.setText("User Name");  
    vartotojas.setId(vartotojasId);  
}
```

11. Koreguokite metodą onCreate, kad įtrauktų antrojo objekto vaizdą:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    init();  
    createMessageTextView();  
    createUserNameTextView();  
    main.addView(zinate, messageDimensions);  
    main.addView(vartotojas, userNameDimensions);  
    setContentView(main);  
}
```


12. Įvykdžius programą, lange matote jau du objektus:



13. Kurkite naują procedūrą trečiajam objektui (teksto redagavimo laukui):

```
private void vardoivedimolaukas () {  
    // TODO Auto-generated method stub  
    vardoIvedimoLaukoDydis = new LayoutParams (LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);  
    vartotojoVardas.setId (vartotojoVardasId);  
    vardoIvedimoLaukoDydis.addRule (RelativeLayout.BELOW, zinuteId);  
    vardoIvedimoLaukoDydis.addRule (RelativeLayout.RIGHT_OF, vartotojasId);  
}
```

14. Koreguokite metodą onCreate:

```
@Override  
protected void onCreate (Bundle savedInstanceState) {  
    super.onCreate (savedInstanceState);  
    init();  
    createMessageTextView();  
    createUserTextView();  
    vardoivedimolaukas();  
    main.addView (zinute, messageDimensions);  
    main.addView (vartotojas, userNameDimensions);  
    main.addView (vartotojoVardas, vardoIvedimoLaukoDydis);  
    setContentView (main);  
}
```

15. Patikrinkite:

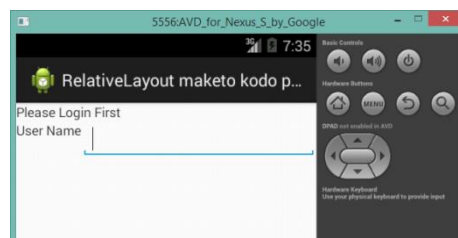


Teksto įvedimo laukas.
Įvedant tekstą – didėja.

16. Procedūroje pridėkite savybę, kuri iškarto išplės teksto lauką iki dešinės pusės:

```
vardoIvedimoLaukoDydis.addRule (RelativeLayout.ALIGN_PARENT_RIGHT);
```

17. Patikrinkite:



Visas *.java failas:

```
package com.example.relativelayoutmaketokodopavyzdys;

import android.app.Activity;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.RelativeLayout.LayoutParams;
import android.widget.TextView;

public class RelativeActivity extends Activity {

    RelativeLayout main;
    EditText vartotojoVardas, vartotojoSlaptazodis;
    TextView zinute, vartotojas, slaptazodis;
    LayoutParams messageDimensions, userNameDimensions, vardoIvedimoLaukoDydis;
    Button prisijungti;
    int zinuteId=1, vartotojoVardasId=2, vartotojoSlaptazodisId=3, vartotojasId=4, slaptazodisId=5,
        prisijungtiId=6;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        init();
        createMessageTextView();
        createUserNameTextView();
        vardoIvedimoLaukas();
        main.addView(zinute, messageDimensions);
        main.addView(vartotojas, userNameDimensions);
        main.addView(vartotojoVardas, vardoIvedimoLaukoDydis);
        setContentView(main);
    }
    private void init() {
        // TODO Auto-generated method stub
        main=new RelativeLayout(this);
        LayoutParams mainDimensions=new LayoutParams( LayoutParams.MATCH_PARENT, LayoutParams.MATCH_PARENT);
        main.setLayoutParams(mainDimensions);

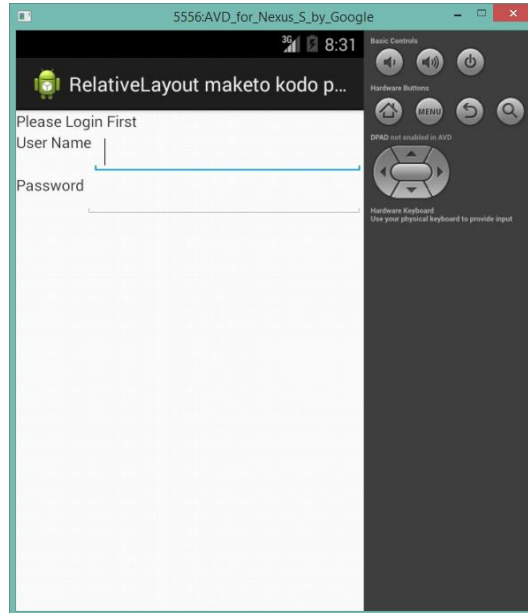
        vartotojoVardas = new EditText(this);
        vartotojoSlaptazodis = new EditText(this);
        zinute = new TextView(this); //Please login first
        vartotojas = new TextView(this); //User Name
        slaptazodis = new TextView(this); //Password
        prisijungti = new Button(this); //Prisijungti
    }
    private void createMessageTextView() {
        // TODO Auto-generated method stub
        messageDimensions = new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
        messageDimensions.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
        zinute.setText("Please Login First");
        zinute.setId(zinuteId);
        zinute.setLayoutParams(messageDimensions);
    }
    private void createUserNameTextView() {
        // TODO Auto-generated method stub
        userNameDimensions = new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
        userNameDimensions.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
        userNameDimensions.addRule(RelativeLayout.BELOW, zinuteId);
        vartotojas.setText("User Name");
        vartotojas.setId(vartotojasId);
    }
    private void vardoIvedimoLaukas() {
        // TODO Auto-generated method stub
        vardoIvedimoLaukoDydis = new LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);
        vartotojoVardas.setId(vartotojoVardasId);
        vardoIvedimoLaukoDydis.addRule(RelativeLayout.BELOW, zinuteId);
        vardoIvedimoLaukoDydis.addRule(RelativeLayout.RIGHT_OF, vartotojasId);
        vardoIvedimoLaukoDydis.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    }
}
```

Savarankiškas darbas Nr 3

Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą

Darbo tikslas – sukurti mobilią aplikaciją, kurioje vartotojo sąsaja aprašoma naudojant *Java* metodus.

Užduotis: pratęskite darbą – į jau sukurtą langą įkelkite teksto žinutę „Password“ ir antrą teksto įvedimo lauką:



Atsakymas:

- Dvi papildomos procedūros:

```
private void createPasswordTextView() {
    // TODO Auto-generated method stub
    passwordDimensions = new LayoutParams (LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT);
    passwordDimensions.addRule (RelativeLayout.ALIGN_PARENT_LEFT);
    passwordDimensions.addRule (RelativeLayout.BELOW, vartotojoVardasId);
    slaptazodis.setText ("Password");
    slaptazodis.setId (slaptazodisId);
}

private void slaptazodzioivedimolaukas() {
    // TODO Auto-generated method stub
    slaptazodzioIvedimoLaukoDydis = new LayoutParams (LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT);
    vartotojoSlaptazodis.setId (vartotojoSlaptazodisId);
    slaptazodzioIvedimoLaukoDydis.addRule (RelativeLayout.BELOW, vartotojoVardasId);
    slaptazodzioIvedimoLaukoDydis.addRule (RelativeLayout.RIGHT_OF, slaptazodisId);
    slaptazodzioIvedimoLaukoDydis.addRule (RelativeLayout.ALIGN_PARENT_RIGHT);
}
}
```

- dviejų naujų kintamųjų aprašymas:

LayoutParams messageDimensions, userNameDimensions, vardoIvedimoLaukoDydis, passwordDimensions, slaptazodzioIvedimoLaukoDydis;

- papildomos eilutės onCreate metode:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate (savedInstanceState);
    init ();
    createMessageTextView ();
    createUserNametextView ();
    vardoivedimolaukas ();

    createPasswordTextView ();
    slaptazodzioivedimolaukas ();

    main.addView (zinate, messageDimensions);
    main.addView (vartotojas, userNameDimensions);
    main.addView (vartotojoVardas, vardoIvedimoLaukoDydis);

    main.addView (slaptazodis, passwordDimensions);
    main.addView (vartotojoSlaptazodis, slaptazodzioIvedimoLaukoDydis);

    setContentView (main);
}
}
```

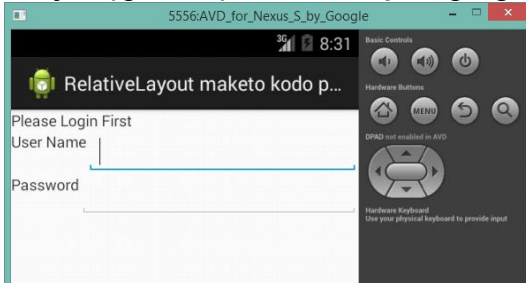
Praktinis darbas Nr. 13

Vartotojo sąsajos kūrimas naudojant Java programavimo kalbą - 2

Darbo tikslas – susipažinti su *Java* programavimo kalbos metodais ir funkcijomis, *Java* failo struktūra. Sukurti mobilią aplikaciją su tekstu, teksto įvedimo lauku ir mygtuku, nustatyti objektams jų savybes.

Darbo eiga:

Praėjusią pamoką naudodami **java** programavimo kalbos kodus sukūrėte maketo pagrindus:



Šią pamoką įkelsite siuntimo mygtuką ir sutvarkysite maketo vaizdą (stilių).

1. Mygtuko įkėlimas

a) Atlikite tuos pačius veiksmus, kaip ir kitiems objektams. Galite pabandyti tai atlikti savarankiškai.

- Mygtuko procedūra:

```
private void mygtukas () {  
    // TODO Auto-generated method stub  
    mygtukoSavybes = new LayoutParams (LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT);  
    prisijungti.setId (prisijungtiId);  
    mygtukoSavybes.addRule (RelativeLayout.BELOW, vartotojoSlaptazodisId);  
    mygtukoSavybes.addRule (RelativeLayout.ALIGN_PARENT_RIGHT);  
    prisijungti.setText ("Login");  
}
```

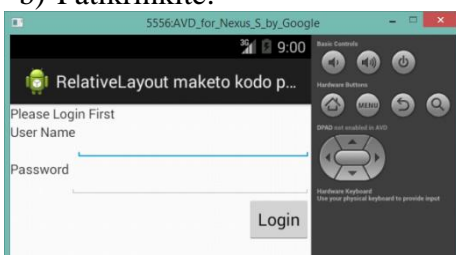
- Kintamojo aprašymas:

LayoutParams `messageDimensions`, `userNameDimensions`, `vardoIvedimoLaukoDydis`, `passwordDimensions`, `slaptazodzioIvedimoLaukoDydis`, `mygtukoSavybes`;

- Vaizdo sugeneravimas:

```
@Override  
protected void onCreate (Bundle savedInstanceState) {  
    super.onCreate (savedInstanceState);  
    init ();  
    createMessageTextView ();  
    createUserNametextView ();  
    vardoivedimolaukas ();  
    createPasswordTextView ();  
    slaptazodzioivedimolaukas ();  
    mygtukas ();  
    main.addView (zinute, messageDimensions);  
    main.addView (vartotojas, userNameDimensions);  
    main.addView (vartotojoVardas, vardoIvedimoLaukoDydis);  
    main.addView (slaptazodis, passwordDimensions);  
    main.addView (vartotojoSlaptazodis, slaptazodzioIvedimoLaukoDydis);  
    main.addView (prisijungti, mygtukoSavybes);  
    setContentView (main);  
}
```

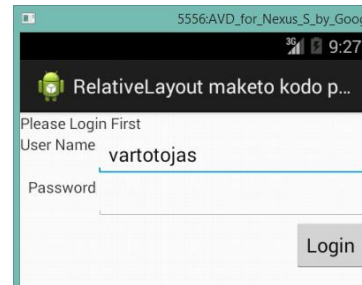
b) Patikrinkite:



2. Vaizdo redagavimas

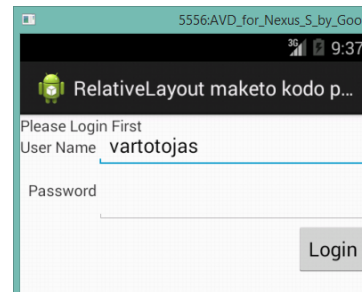
- Sulyginkite teksto užrašus „User Name“ ir „Password“ dešinėje pusėje, kad abu teksto laukai prasidėtų vienoje vertikaloje linijoje:

```
private void createPasswordTextView() {
    // TODO Auto-generated method stub
    passwordDimensions = new LayoutParams(LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT);
    passwordDimensions.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
    passwordDimensions.addRule(RelativeLayout.BELOW, vartotojoVardasId);
    passwordDimensions.addRule(RelativeLayout.ALIGN_RIGHT, vartotojasId);
    slaptazodis.setGravity(Gravity.RIGHT);
    slaptazodis.setText("Password");
    slaptazodis.setId(slaptazodisId);
}
```



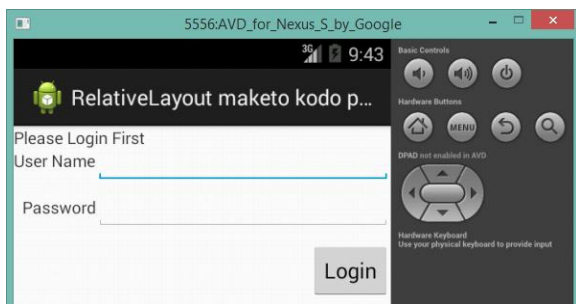
- Sulyginkite užrašą „User Name“ ir įvedimo lauką horizontaliai per vidurį:

```
private void vardoivedimolaukas () {
    // TODO Auto-generated method stub
    vardoIvedimoLaukoDydis = new LayoutParams(LayoutParams.WRAP_CONTENT,
        LayoutParams.WRAP_CONTENT);
    vartotojoVardas.setId(vartotojoVardasId);
    vardoIvedimoLaukoDydis.addRule(RelativeLayout.BELOW, zinuteId);
    vardoIvedimoLaukoDydis.addRule(RelativeLayout.RIGHT_OF, vartotojasId);
    vardoIvedimoLaukoDydis.addRule(RelativeLayout.ALIGN_BASELINE, vartotojasId);
    vardoIvedimoLaukoDydis.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
}
```



- Tą patį padarykite su slaptažodžio įvedimo lauku:

```
slaptazodzioIvedimoLaukoDydis.addRule(RelativeLayout.ALIGN_BASELINE, slaptazodisId);
```



- Padidinkite tarpus tarp objekto dalių:

- apsirašykite kintamąjį:

```
int tarpelis=10;
```

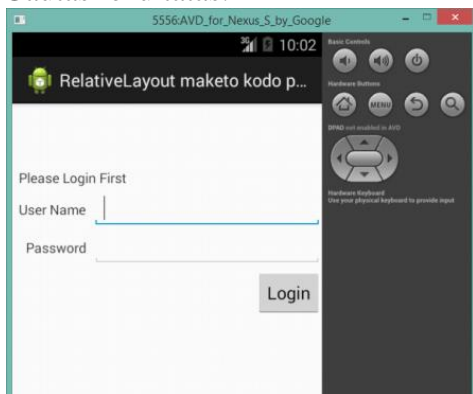
- pritaikykite kintamąjį objektams:

```
zinute.setPadding(tarpelis, 100, tarpelis, tarpelis);
```

```
vartotojas.setPadding(tarpelis, tarpelis, tarpelis, tarpelis);
```

```
slaptazodis.setPadding(tarpelis, tarpelis, tarpelis, tarpelis);
```

Gautas rezultatas:



Praktinis darbas Nr. 14

Aktyvaus lango gyvavimo ciklo metodai

Darbo tikslas – susipažinti su aktyvaus lango veikimo principais, mobilijoje aplikacijoje stebėti aktyvaus lango gyvavimo ciklus.

Įvadas:

Aktyvaus lango gyvavimo ciklo metodai:

- onCreate()
- onStart()
- onResume()
- onPause()
- onStop()
- onRestart()
- onDestroy()

Darbo eiga:

1. Norėdami pamatyti ar programoje veikia ciklas, galime įdėti pranešimo veiksmą. Čia pateiktas **MainActivity.java** failas su ciklais ir jų veikimo patikrinimo eilute:

```
package com.example.vivzactivitylifecycle;

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

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d("VIVZ", "onCreate was called");
    }

    @Override
    protected void onResume() {
        // TODO Auto-generated method stub
        super.onResume();
        Log.d("VIVZ", "onResume was called");
    }

    @Override
    protected void onStart() {
        // TODO Auto-generated method stub
        super.onStart();
        Log.d("VIVZ", "onStart was called");
    }

    @Override
    protected void onPause() {
        // TODO Auto-generated method stub
        super.onPause();
        Log.d("VIVZ", "onPause was called");
    }

    @Override
    protected void onStop() {
        // TODO Auto-generated method stub
        super.onStop();
        Log.d("VIVZ", "onStop was called");
    }

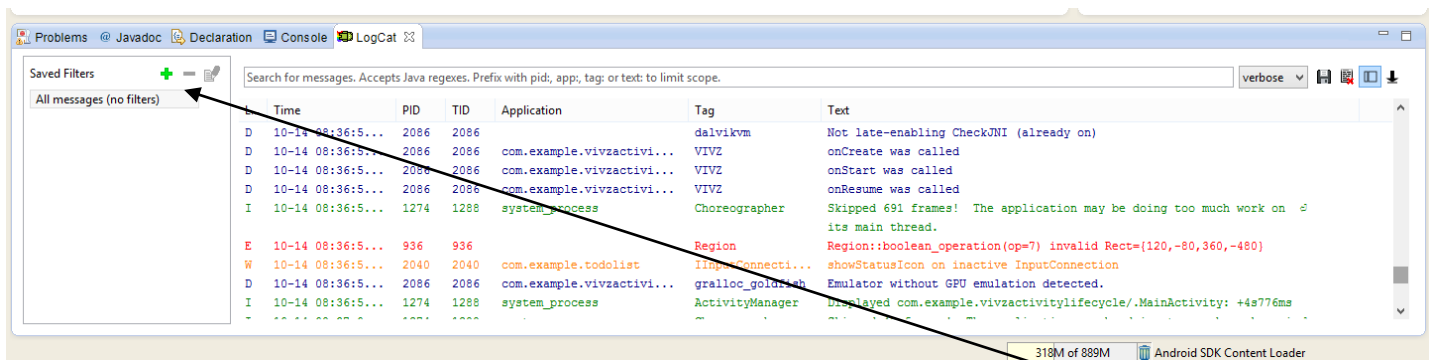
    @Override
    protected void onRestart() {
        // TODO Auto-generated method stub
        super.onRestart();
        Log.d("VIVZ", "onRestart was called");
    }

    @Override
    protected void onDestroy() {
        // TODO Auto-generated method stub
        super.onDestroy();
        Log.d("VIVZ", "onDestroy was called");
    }
}
```

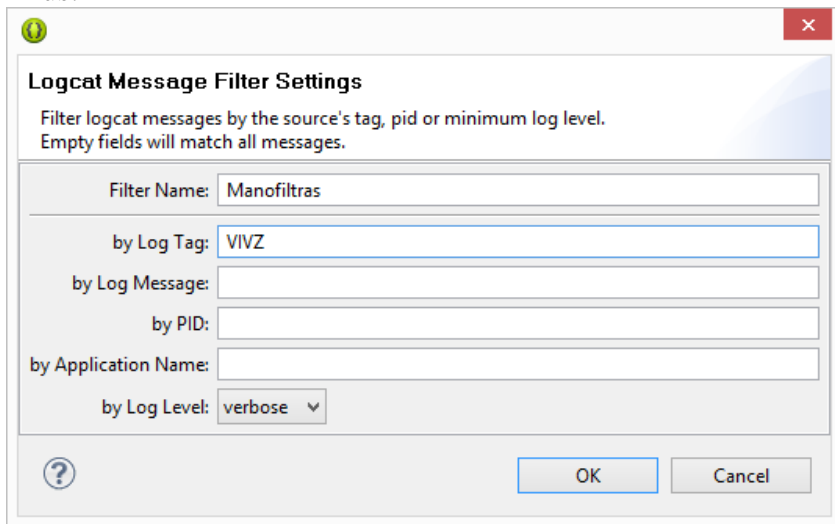
Eilutė

```
Log.d("VIVZ", "onDestroy was called");
```

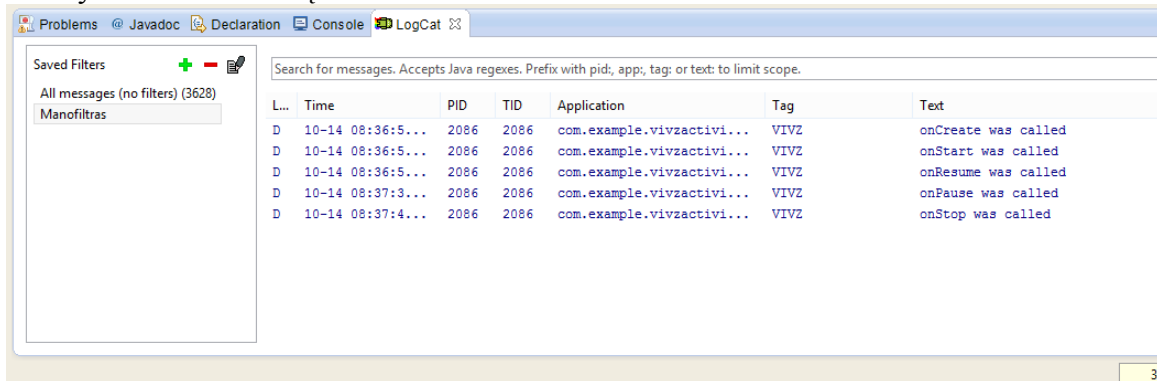
išves pranešimą su žyme VIVZ LogCat pranešimų lange.



2. Norėdami matyti tik pranešimus su VIVZ žyme, susikurkite naują filtrą. Spauskite +, surašykite pavadinimus:



ir matysite tik savo filtrą:



Tai labai naudinga, programos testavimui. Jei programa veikia ne taip, kaip jūs norėtumėt, galima įdėti tokį tikrinimą ir pažiūrėti, ar pasiekama ta vieta, kurią jūs užprogramavote. Galima uždėti skirtingus lygius, kad lengviau galima būtų rasti.

Log.I(String tag, String message)

Log.E(String tag, String message)

Log.W(String tag, String message)

Log.V(String tag, String message)

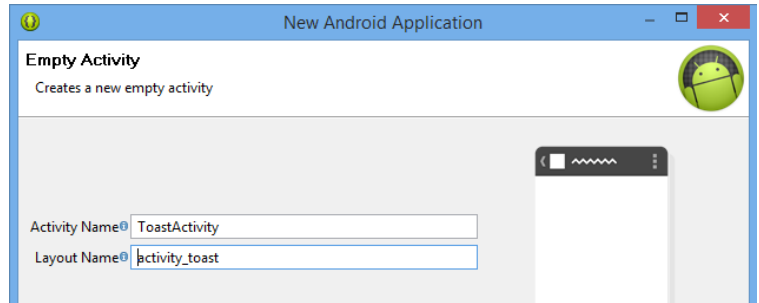
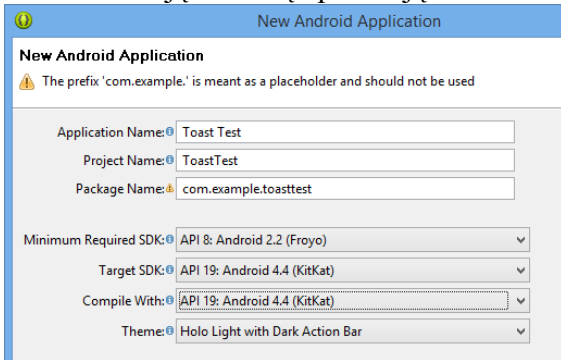
Praktinis darbas Nr. 15

Pranešimas ekrane (Toast)

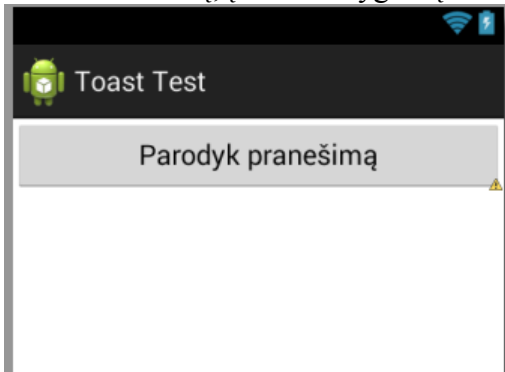
Darbo tikslas – įterpti į mobilią aplikaciją pranešimo išvedimo metodus.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:



2. Ištrinkite tekstą, įkelkite mygtuką ir nurodykite kreipinį paspaudimo veiksmui **onClick**.



```
<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="com.example.toasttest.ToastActivity" >

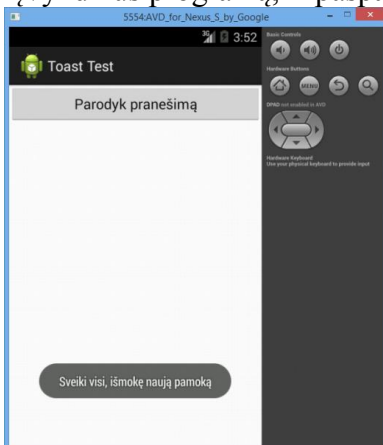
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:onClick="pranesimas"
        android:text="Parodyk pranešimą" />

</RelativeLayout>
```

3. Kurkite **java** metodą mygtuko paspaudimo veiksmui:

```
public void pranesimas(View v) {
    Toast toast=Toast.makeText(this, "Sveiki visi, išmokę naują pamoką",Toast.LENGTH_LONG );
    toast.show();
}
```

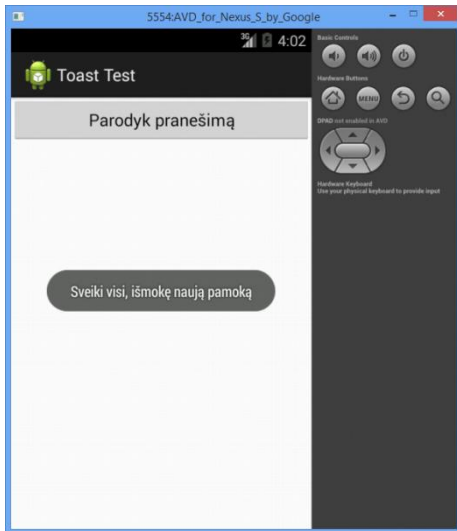
4. Įvykdžius programą, ir paspaudus mygtuką išmetamas pranešimas:



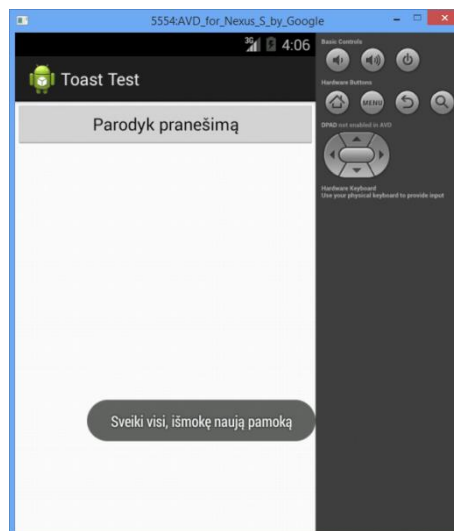
5. Nurodykite, kad pranešimas būtų centre:

```
public void pranesimas(View v) {  
    Toast toast=Toast.makeText(this, "Sveiki visi, išmokę naują pamoką",Toast.LENGTH_LONG );  
    toast.setGravity(Gravity.CENTER, 0, 0);  
    toast.show();  
}
```

6. Patikrinkite:



7. Lygiavimo eilutėje įrašyti „0, 0“ nurodo x ir y poziciją ekrane nuo centro:



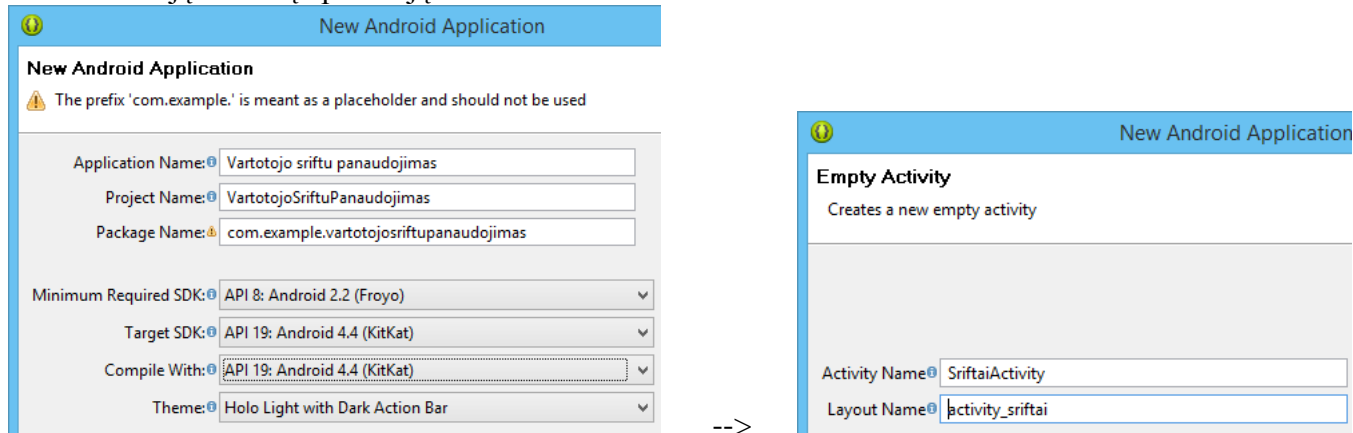
Praktinis darbas Nr. 16

Vartotojo šriftų pritaikymas

Darbo tikslas – surasti internete ir pritaikyti savo mobilią aplikacijai įvairius vartotojo šriftus.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:



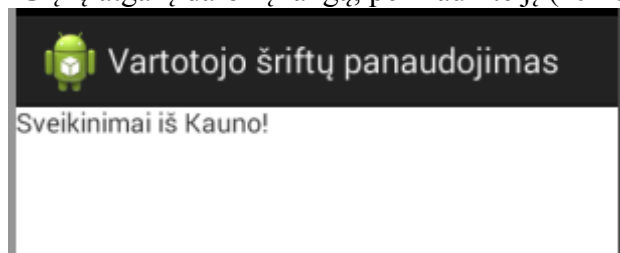
2. Pakeiskite lange matomą tekstą „Hello World!“ kitu, lietuvišku „Sveikinimai iš Kauno!“, taip pat ekrano meniu juostos užrašę įrašykite lietuviškas raides. Visa tai atliksite redaguodami **res/values/strings.xml** failą:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Vartotojo šriftų panaudojimas</string>
    <string name="hello_world">Sveikinimai iš Kauno!</string>

</resources>
```

3. Grįžę atgal į darbinį langą, perkraukite jį (**refresh**), kad užrašai atsinaujintų:



4. Kadangi pagrindinis objektas su kuriuo dirbsite yra tekstinis laukas, todėl priskirkite jam identifikatorių.

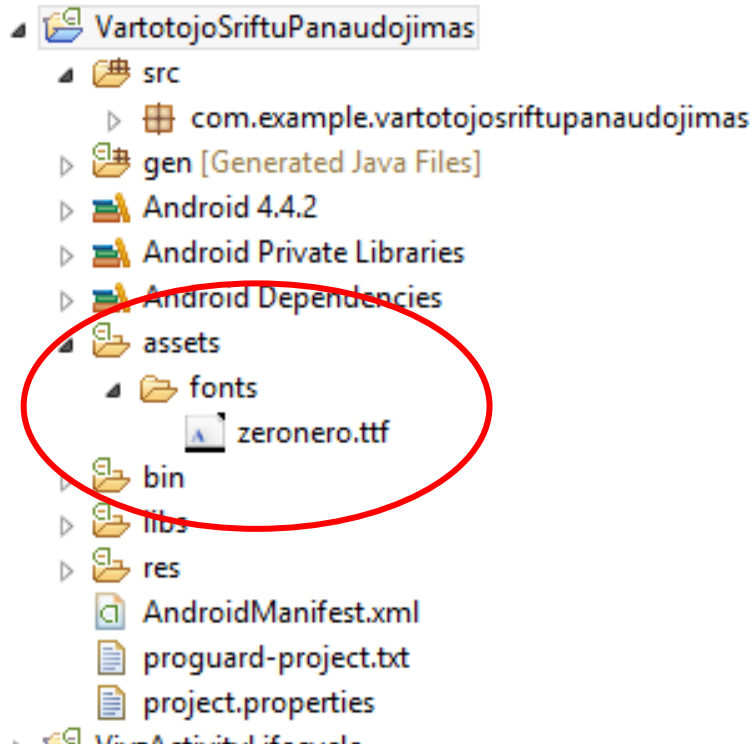
```
<TextView
    android:id="@+id/mano_tekstas"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/hello_world" />
```

5. Aprašykite objektą ***.java** faile:

```
public class SriftaiActivity extends Activity {

    TextView t;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sriftai);
        t=(TextView) findViewById(R.id.mano_tekstas);
    }
}
```

6. Kataloge **assets** sukurkite naują katalogą – **fonts**, į kurį įkelkite jums patikusį šriftą (pvz. - zeronero.ttf)
Šriftą galite atsisiųsti iš interneto adresu <http://www.1001freefonts.com>



7. Aprašykite šriftų panaudojimą *.java faile:

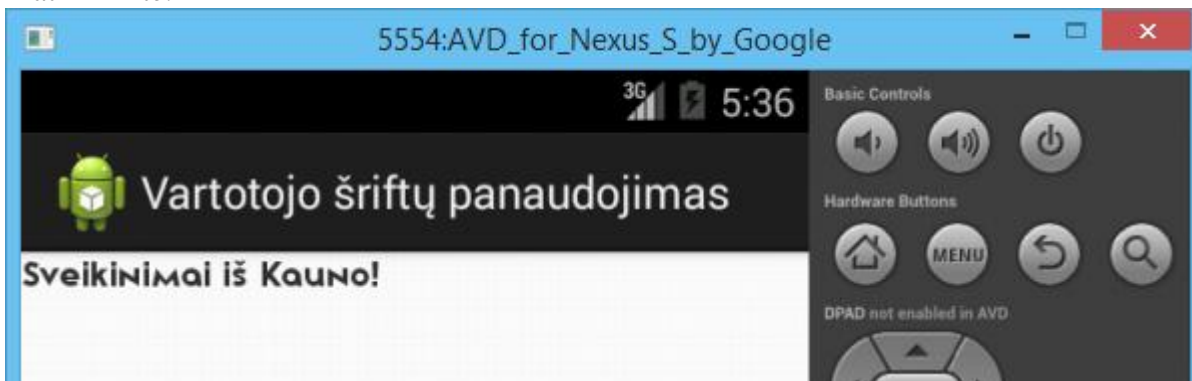
```
package com.example.vartotojosriftupanaudojimas;

import android.app.Activity;
import android.graphics.Typeface;
import android.os.Bundle;
import android.widget.TextView;

public class SriftaiActivity extends Activity {

    TextView t;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sriftai);
        t=(TextView) findViewById(R.id.mano_tekstas);
        Typeface manoSriftai=Typeface.createFromAsset(getAssets(), "fonts/zeronero.ttf");
        t.setTypeface(manoSriftai);
    }
}
```

8. Patikrinkite:



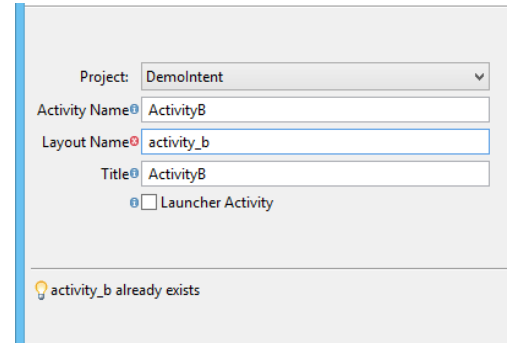
Praktinis darbas Nr. 17

Dviejų langų mobili aplikacija, langų aktyvavimo metodai

Darbo tikslas – sukurti dviejų langų mobilią aplikaciją, panaudoti langų aktyvavimo metodus.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją **DemoIntent**.
2. Į šią mobilią aplikaciją pridėkite dar vieną aktyvų langą **ActivityB**:
 - a) Meniu pasirinkite **File – New – Other... – Android Activity – Next**;
 - b) **Empty Activity – Next**;
 - c) Užpildykite laukus;
 - d) **Finish**.



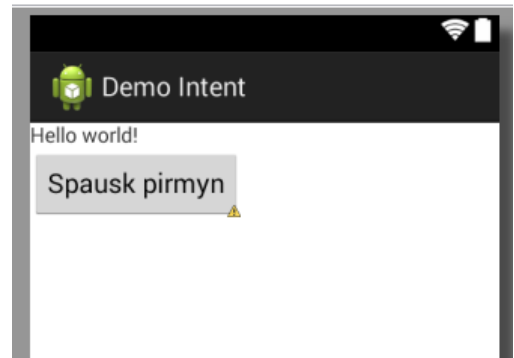
3. Pagrindiniame projekto lange, po tekstu, įkelkite mygtuką.

```
<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="com.example.demointent.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:onClick="vykdyk"
        android:text="Spausk čia" />

</RelativeLayout>
```



4. Pagrindiniame *.java faile įrašykite mygtuko objektą ir jo aktyvavimo metodą **vykdyk**, kuriame aprašyta dviejų langų sujungimo procedūra – **Intent**.

```
package com.example.demointent;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class IntentActivity extends Activity {

    Button b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intent);
        b=(Button) findViewById(R.id.button1);
    }
    public void vykdyk(View v) {
        Intent i=new Intent(this, ActivityB.class);
        startActivity(i);
    }
}
```

5. Testuokite programą. Paspaudus mygtuką atveriamas kitas aktyvus langas.

Papildoma užduotis

Antrame lange įkelkite mygtuką grįžimui atgal į pirmą langą ir sukurkite jo Intent.



Kitas lango aktyvavimo metodas

Irašome paketo pavadinimą:

```
package com.example.demointent;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class IntentActivity extends Activity {

    Button b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intent);
        b=(Button) findViewById(R.id.button1);
    }
    public void vykdyk(View v) {
        Intent i=new Intent();
        i.setClassName("com.example.demointent", "com.example.demointent.ActivityB");
        startActivity(i);
    }
}
```

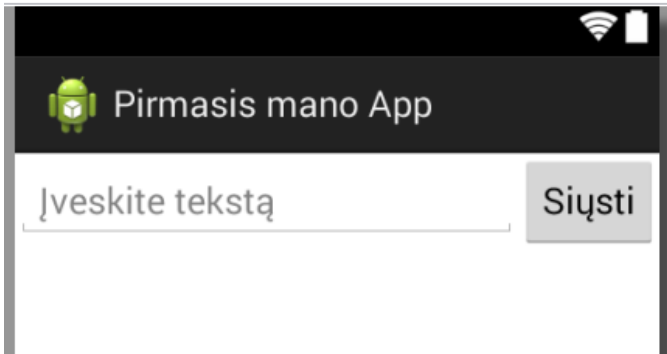
Praktinis darbas Nr. 18

Dviejų langų mobili aplikacija, teksto persiuntimo metodai

Darbo tikslas – sukurti dviejų langų mobilią aplikaciją, panaudoti teksto persiuntimo iš vieno lango į kitą metodus.

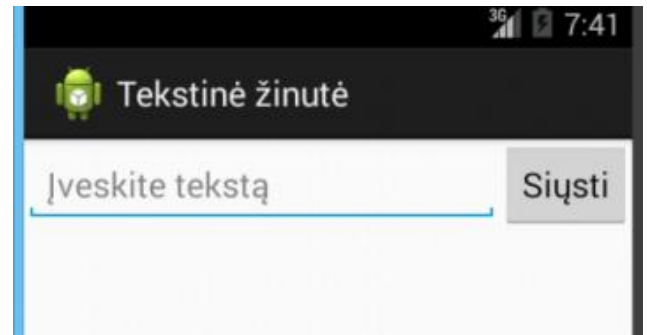
Darbo eiga:

1. Padarykite Praktinio darbo Nr. 1 mobilios aplikacijos „PirmasisManoApp“ kopiją „TekstineZinute“ (arba sukurkite iš naujo):



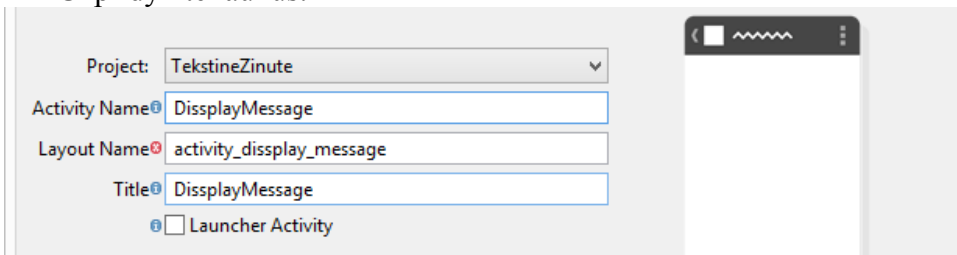
2. Faile **strings.xml** pakeiskite aktyvaus lango pavadinimą:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Tekstinė žinutė</string>
  <string name="edit_message">Įveskite tekstą</string>
  <string name="button_send">Siųsti</string>
  <string name="title_activity_main">MainActivity</string>
  <string name="title_activity_display_message">DisplayMessageActivity</string>
</resources>
```



3. Į šią mobilią aplikaciją pridėkite dar vieną aktyvų langą:

- Meniu pasirinkite **File – New – Other... – Android Activity – Next**;
- **Empty Activity – Next**;
- Užpildykite laukus:



- **Finish.**

4. Sutvarkykite naujo lango maketą **activity_disspaly_message.xml**:

```
<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}" >
</RelativeLayout>
```

5. Pridėkite mygtuko veiksmą `activity_main.xml` faile:

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

    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send"
        android:onClick="sendMessage" />

</LinearLayout>
```

6. Faile `MainActivity.java` įrašykite mygtuko aktyvavimo metodą `sendMessage` ir veiksmus:

```
package com.example.pirmasmanoapp;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends Activity {
    public final static String EXTRA_MESSAGE = "com.example.tekstinezinute.MESSAGE";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    /** Mygtuko paspaudimo metodas atidarys žinutės langą */
    public void sendMessage(View view) {
        Intent intent = new Intent(this, DissplayMessage.class);
        EditText editText = (EditText) findViewById(R.id.edit_message);
        String message = editText.getText().toString();
        intent.putExtra(EXTRA_MESSAGE, message);
        startActivity(intent);
    }
}
```

7. Faile `DissplayMessage.java` įrašykite tekstinės žinutės gavimo iš kito lango veiksmus:

```
package com.example.pirmasmanoapp;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;

public class DissplayMessage extends Activity {

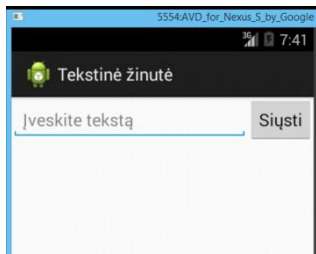
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dissplay_message);

        //Priimama tekstinė žinutė iš kito aktyvaus lango
        Intent intent = getIntent();
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);

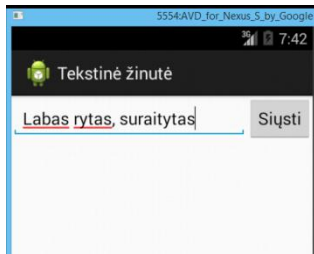
        // Kuriamas tekstinis vaizdas
        TextView textView = new TextView(this);
        textView.setTextSize(20);
        textView.setText(message);

        //Aktyvuojamas tekstinis vaizdas
        setContentView(textView);
    }
}
```

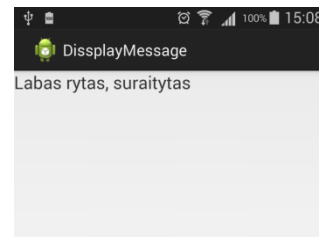

8. Paleidus programą, atsidaro pirmasis langas:



parašius tekstą ir



paspaudus mygtuką, atsidaro antrasis langas:



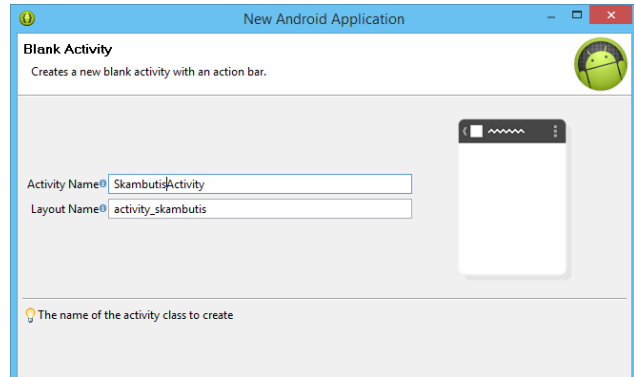
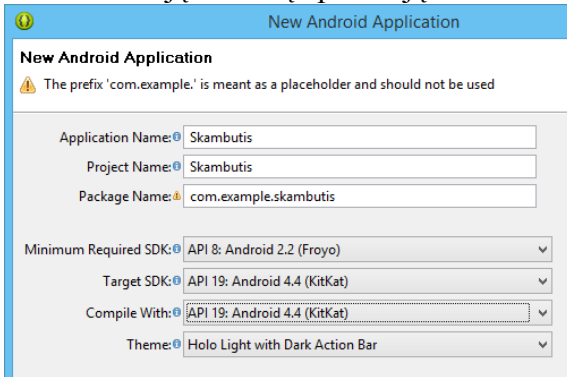
Praktinis darbas Nr. 19

Skambinimas ir skambučio priėmimas

Darbo tikslas – sukurti mobilią aplikaciją imituojančią skambinimą telefonu.

Darbo eiga:

1. Kurkite naują mobilią aplikaciją:



2. **activity_skambutis.xml** faile ištrinkite teksto pateikimo eilutes.

3. Redaguokite **SkambutisActivity.java** failą, įrašydami metodą Intent:

```
package com.example.skambutis;

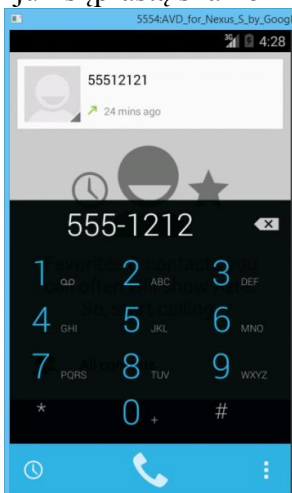
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;

public class SkambutisActivity extends Activity {

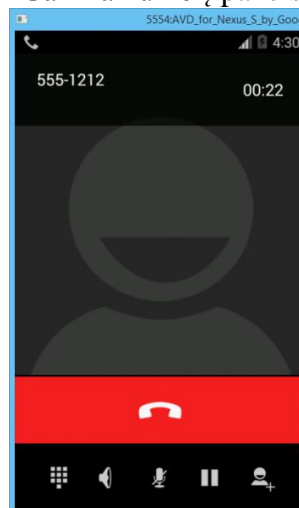
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_skambutis);

        Intent DialIntent = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:5551212"));
        DialIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(DialIntent);
    }
}
```

4. Jei testuosite programą telefone, pamatysite jums įprastą skambinimo vaizdą:



5. Paskambinus šiuo numeriu įvyks sujungimas. Galima numerį pakeisti kitu.



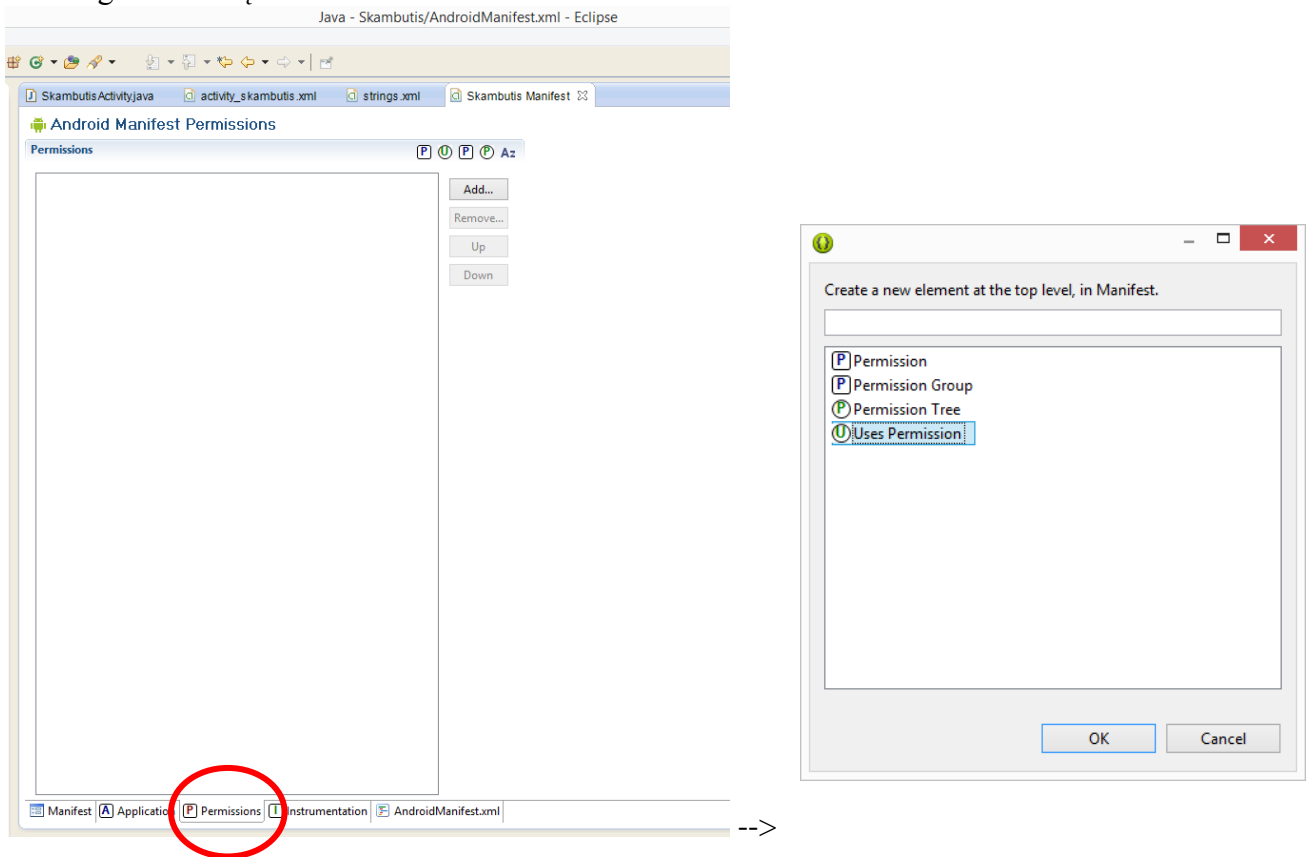
6. Tam, kad iš skambinančiojo taptumėt tu, kuriam skambina, redaguokite parašytas tris eilutes:

```
Intent CallIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:5551212"));
CallIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
startActivity(CallIntent);
```

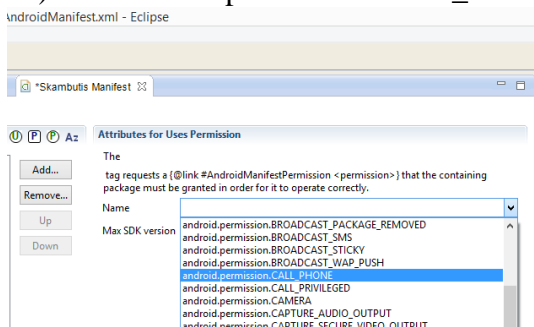
7. Testuokite. Įvykdę programą, turėtumėte gauti vaizdą su klaidų kodais, arba programa visai nepasileis.

8. Reikia padaryti tam tikrus nustatymus, kad būtų leidžiama ši procedūra. Redaguokite aktyvaus lango leidimus:

a) **AndroidManifest.xml** faile pasirinkę **Permissions**, spauskite mygtuką **Add** ir atsidariusiame lange komandą **Uses Permission**



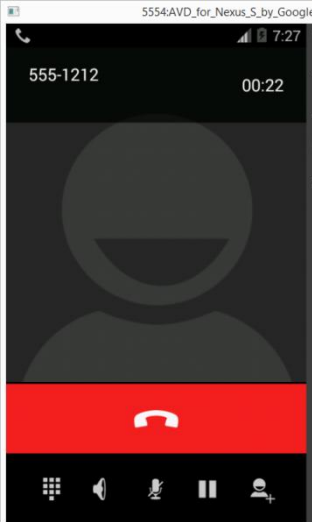
b) Vardo lauke pasirinkite **CALL_PHONE**



c) Manifest faile atsiras viena papildoma eilutė:

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

9. Testuokite programą iš naujo. Ir štai, jums kažkas skambina (testuokite emuliacijoje, nes telefone gali neveikti):



Praktinis darbas Nr. 20

Skambinimo funkcijos tobulinimas

Darbo tikslas – patobulinti mobilią aplikaciją, imituojančią skambinimą telefonu, kad prieš skambinant būtų galima įvesti abonento telefono numerį.

Darbo eiga:

Tęsite Praktinį darbą Nr.19 „Skambutis“. (Jei jo nedarėte, teks susikurti.) Patobulinsite programą, pridėdami mygtuką ir teksto redagavimo langą.

1. Į failą **activity_skambutis.xml** įkelkite mygtuką:

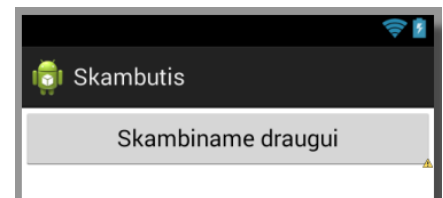
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.skambutis.SkambutisActivity" >

    <Button android:id="@+id/callButton"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Skambiname draugui" />

</RelativeLayout>
```

Šios keturios eilutės nurodo, kad

- bus sukurtas mygtukas vardu **callButton**
- nurodomas mygtuko plotis (fill_parent - užpildyti visa)
- nurodomas mygtuko aukštis (wrap_content)
- nurodomas mygtuko tekstas (Skambiname draugui)



2. Aprašykite mygtuko funkcijas **SkambutisActivity.java** faile:

```
package com.example.skambutis;

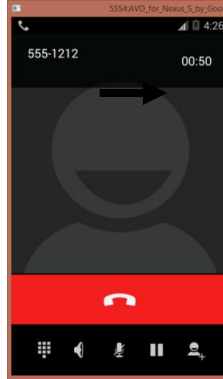
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class SkambutisActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_skambutis);
        final Button callButton = (Button) findViewById(R.id.callButton);
        callButton.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                Intent callIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:5551212"));
                callIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(callIntent);
            }
        });
    }
}
```

Sukūrėte mygtuką **xml** faile, jam suteikėte vardą ID – **callButton**. Funkcijoje taip pat naudojate šį vardą ir mygtuko veiksmus, kaip ir praėjusiame darbe (Nr.19), tam, kad būtų iškvieistas skambinimo numeriui 5551212 langas.

Įvykdę programą gaunate:



3. Kol kas galite paskambinti tik vienu programoje įrašytu numeriu. Pakoreguosite programą taip, kad numerį galima būtų įvesti prieš skambinant. Pakeiskite maketo objektų dėstymo būdą iš reliatyvaus į linijinį (vertikalų). Pridėkite tekstą ir teksto įvedimo langą, kuriame galėsima bus įvesti kitą telefono numerį.

```
<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="com.example.skambutis.SkambutisActivity" >

    <TextView android:id="@+id/tekstolaukas"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Įveskite numerį:"
    />

    <EditText android:id="@+id/phoneNumber"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />

    <Button android:id="@+id/callButton"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Skambinti"
    />

</LinearLayout>
```

A screenshot of the 'Skambutis' app interface. It shows a text input field with the placeholder text 'Įveskite numerį:' and a button labeled 'Skambinti' below it.

4. Redaguokite failą **SkambutisActivity.java**.

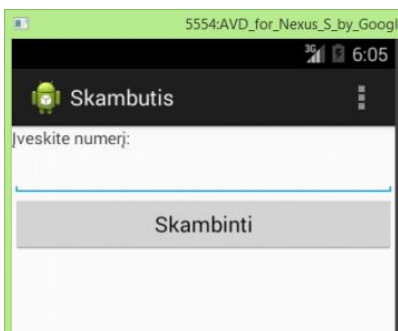
- Aprašykite teksto įvedimo lauko funkciją EditText.

```
final EditText phoneNumber = (EditText) findViewById(R.id.phoneNumber);
```

- Pakeiskite metodą, kuriame buvo tel. nr. 5551212, į teksto įvedimo metodą getText():

```
Intent CallIntent = new Intent(Intent.ACTION_CALL,Uri.parse("tel:" +
phoneNumber.getText()));
```

5. Įvykdžius programą, jau galite skambinti norimu numeriu.



Visas SkambutisActivity.java failas:

```
package com.example.skambutis;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class SkambutisActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_skambutis);
        final EditText phoneNumber = (EditText) findViewById(R.id.phoneNumber);
        final Button callButton = (Button) findViewById(R.id.callButton);
        callButton.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v) {
                Intent CallIntent = new Intent(Intent.ACTION_CALL, Uri.parse("tel:" +
phoneNumber.getText()));
                CallIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK );
                startActivity(CallIntent);
            }
        });
    }
}
```