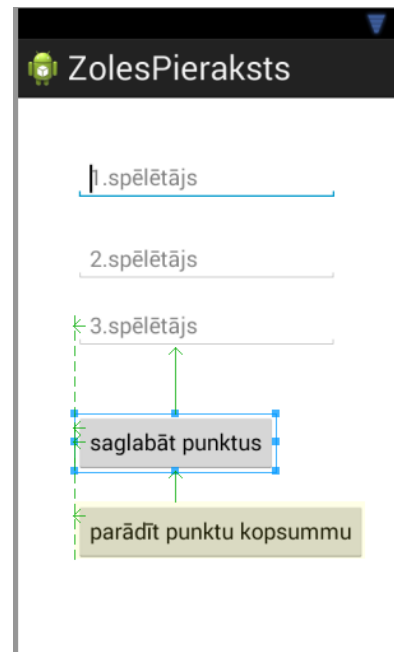


DARBS AR TEKSTA DATNĒM 21.nodarbība

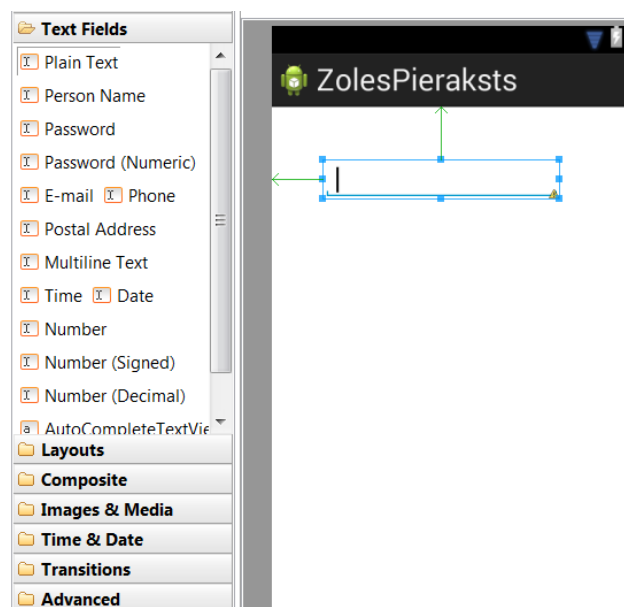
ANDROID PROJEKTS «ZolesPieraksts»

Veidosim programmu, kas pēc pogas **saglabāt punktus** nospiešanas saglabās datnē *zole.txt* trīs zoles spēlētāju iegūtos punktus šajā spēles partijā, bet pēc pogas **parādīt punktu summu** nospiešanas visus datnē esošos katra spēlētāja punktus sasummēs un parādīs ekrānā.



ANDROID PROJEKTS «Zoles pieraksts» - SĀKUMS

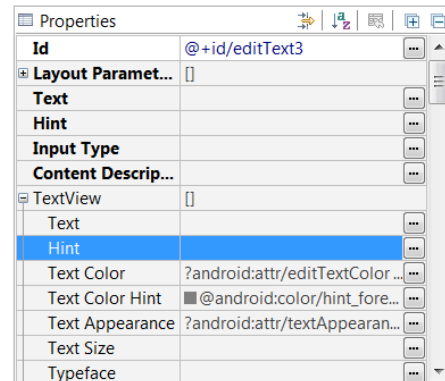
- Izveidojam jaunu Android Application Project **ZolesPieraksts**
- Atveram **activity_main.xml** datni grafiskajā skatā un pievienojam 3 **PlainText** elementus



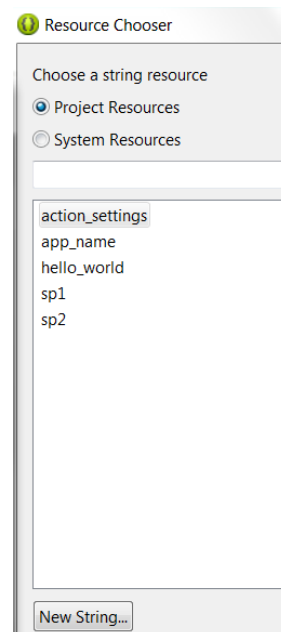
ANDROID PROJEKTS «ZolesPieraksts» - ŠPIKERI

Katram no trim teksta elementiem izveidojam «špikeri» **hint** - tas ir teksts, kas pasaka, ko vajadzētu ievadīt:

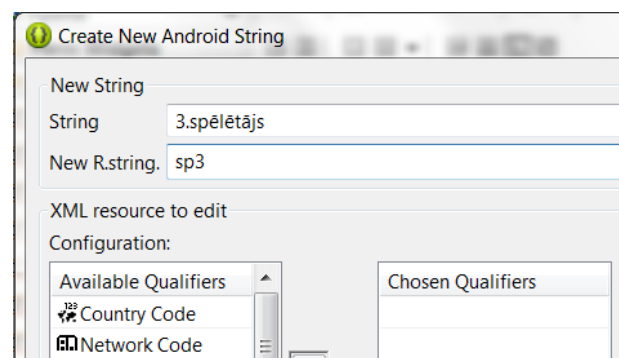
- *Properties* logā **Hint ...**



- *Resource Chooser* logā **New String..**



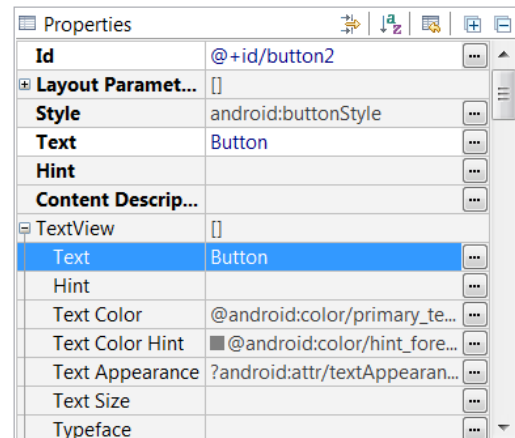
- *Create New Android String* logā pie **String** - špikera teksts, pie **New.R.String** - teksta identifikators
- Katram teksta elementam savs teksts un tā identifikators



ANDROID PROJEKTS «ZolesPieraksts» - POGAS

Izveidojam divas *button* tipa pogas un nomainām to tekstu:

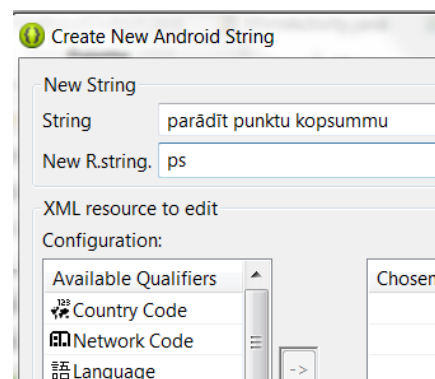
- *Properties* logā **Text**-...



- *Resource Chooser* logā **New String...**



- *Create New Android String* logā pie *String* – pogas teksts, pie *New.R.String* – pogas teksta identifikators
- Katrai pogai cits teksts un teksta identifikators!



«ZolesPieraksts» - PUNKTU SAGLABĀŠANA

- Grafiskajā skatā aktivizējam pogu, kas paredzēta punktu saglabāšanai
- *Properties* logā pie īpašības *OnClick* ierakstām *sagLabatPunktus*

Properties	
Next Focus For...	...
Clickable	<input checked="" type="checkbox"/> true
Long Clickable	<input type="checkbox"/>
Duplicate Paren...	<input type="checkbox"/>
Min Height	48dip
Min Width	64dip
Content Descrip...	...
On Click	saglabatPunktus
Over Scroll Mode	...
Alpha	...
Text Alignment	...
Important For A...	...
Label For	...
Deprecated	(Deprecated Properties)

- pārejam uz skatu *MainActivity.java* un izveidojam jaunu metodi *saglabatPunktus*

```
public class MainActivity extends Activity {
    public void saglabatPunktus(View v){
```

```
}
```

```
@Override
```

```
public class MainActivity extends Activity {
    public void saglabatPunktus(View v){
```

```
    try{
```

```
    }catch(Exception e){
```

```
    }
```

```
@Override
```

- Metodē *saglabatPunktus* izveidojam *try-catch* konstrukciju

- Try konstrukcijā ieviešam `PrintWriter` objektu rakstīšanai failā

```
try{
    PrintWriter raksta=new PrintWriter(new FileWriter(getFilesDir()+File.separator+"zole.txt"));
}catch(Exception e){
}
}
```

- Objekts `EditText` vajadzīgs, lai nolasītu ievadīto punktu skaitu no teksta elementiem. To varam izmantot arī kļūdas paziņojumam. Tekstu no teksta elementiem saglabājam datnē un to aizveram.

```
public void saglabatPunktus(View v){
    EditText teksts=(EditText)findViewById(R.id.editText1);
    try{
        PrintWriter raksta=new PrintWriter(new FileWriter(getFilesDir()+File.separator+"zole.txt"));
        raksta.println(teksts.getText().toString());
        teksts=(EditText)findViewById(R.id.editText2);
        raksta.println(teksts.getText().toString());
        teksts=(EditText)findViewById(R.id.editText3);
        raksta.println(teksts.getText().toString());
        raksta.close();
    }catch(Exception e){
        teksts.setText("kaut kas nav labi");
    }
}
```

«ZolesPieraksts» - PUNKTU SUMMAS PARĀDĪŠANA

- Otrai pogai piestiprinām metodi `paradaSummu`
- `EditText` objekts būs vajadzīgs, lai piekļūtu teksta elementiem un ieliktu tajos punktu summu. To izmantosim arī, lai paziņotu par problēmām darbā ar datni.

```

public void paradaSummu(View v){

    EditText teksts=(EditText)findViewById(R.id.editText1);
    String rinda;
    try{
        BufferedReader lasa=new BufferedReader(new FileReader(getFilesDir()+File.separator+"zole2.txt"));

        }catch(Exception e){
            teksts.setText("nelabi lasot");
        }
    }
}

```

«ZolesPieraksts» - PUNKTU SUMMĒŠANA

Jāizdara dažas izmaiņas programmā:

- Metodē *saglabatPunktus* pieliekam *true* pie *FileWriter* parametriem, lai datnē dati papildinātos, nevis katru reizi rakstītos no jauna:

```

PrintWriter raksta=new PrintWriter(new FileWriter(getFilesDir()+File.separator+"zole.txt",true));

```

- Metodē *paradiSummu* ieviešam mainīgos *punkti1*, *punkti2* un *punkti3*, kas uzkrās atbilstošo spēlētāju punktus. `int punkti1=0,punkti2=0,punkti3=0;`
- Nolasīšanai no datnes vajadzīgs cikls. Katrā ciklošanās reizē izvēlamies, kuram spēlētājam pieskaitīt punktus. Kad nolasīti 3. spēlētāja punkti, spēlētāja numuru *i* atkal uzliekam 1.
- Kad visi punkti sasummēti, katra spēlētāja punktus ieliek atbilstošajā teksta elementā:

```

int i=1;
while((rinda=lasa.readLine())!=null){
    switch (i){
        case 1: punkti1=punkti1+Integer.parseInt(rinda);i++;break;
        case 2: punkti2=punkti2+Integer.parseInt(rinda);i++;break;
        case 3: punkti3=punkti3+Integer.parseInt(rinda);i=1;break;
    }
}

```

```

teksts.setText(punkti1+"");
teksts=(EditText)findViewById(R.id.editText2);
teksts.setText(punkti2+"");
teksts=(EditText)findViewById(R.id.editText3);
teksts.setText(punkti3+"");

```

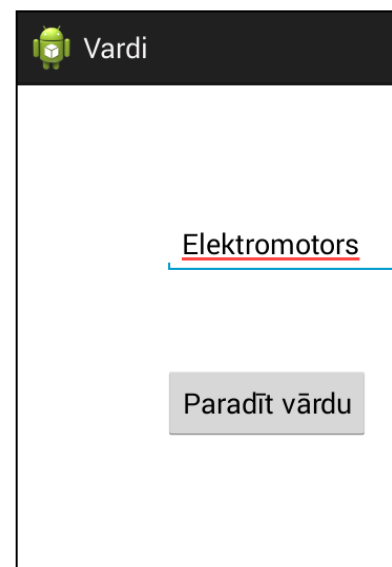
«ZolesPieraksts» - PUNKTU SUMMĒŠANAS KODS

```
public void paradaSummu(View v){  
  
    EditText teksts=(EditText)findViewById(R.id.editText1);  
    String rinda;  
    int punkti1=0,punkti2=0,punkti3=0;  
    try{  
        BufferedReader lasa=new BufferedReader(new FileReader(getFilesDir()+File.separator+"zole.txt"));  
        int i=1;  
        while((rinda=lasa.readLine())!=null){  
            switch (i){  
                case 1: punkti1=punkti1+Integer.parseInt(rinda);i++;break;  
                case 2: punkti2=punkti2+Integer.parseInt(rinda);i++;break;  
                case 3: punkti3=punkti3+Integer.parseInt(rinda);i=1;break;  
            }  
        }  
        teksts.setText(punkti1+"");  
        teksts=(EditText)findViewById(R.id.editText2);  
        teksts.setText(punkti2+"");  
        teksts=(EditText)findViewById(R.id.editText3);  
        teksts.setText(punkti3+"");  
        lasa.close();  
    }catch(Exception e){  
        teksts.setText("nelabi lasot");  
    }  
}
```

ANDROID PROJEKTS «Vardi»

Veidosim jaunu Android programmu «Vardi», kas nospiežot pogu «Parādīt vārdu», no teksta datnes `vardi.txt` uz labu laimi izvēlēsies vienu vārdu un parādīs to uz ekrāna teksta elementā.

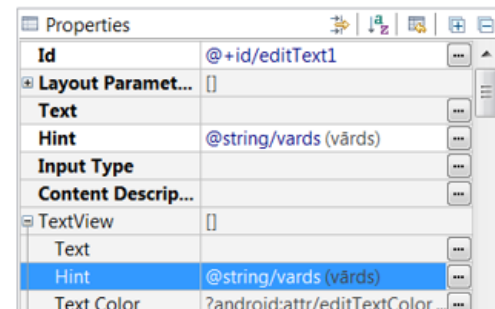
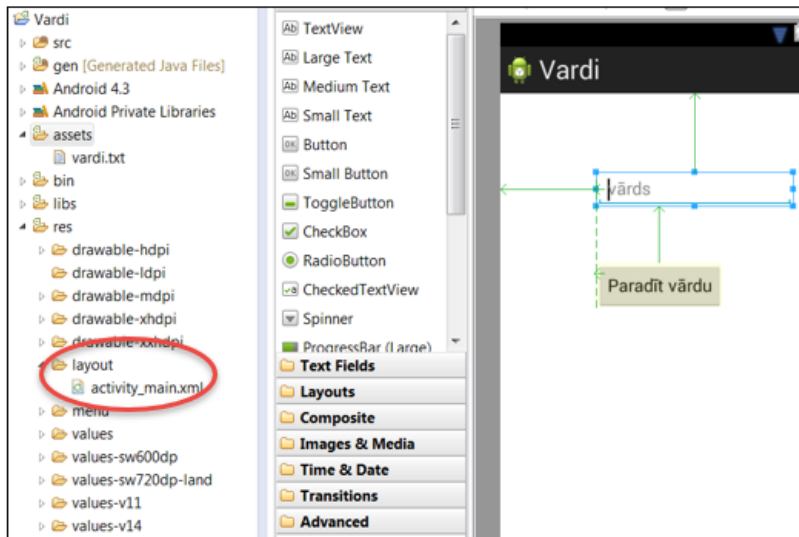
Vārdu saraksta datni `vardi.txt` izveidosim paši un saglabāsim projekta mapē `assets`.



ANDROID PROJEKTS «Vardi» - GRAFISKIE ELEMENTI

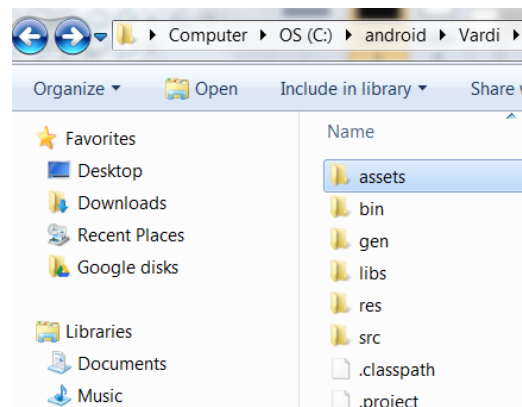
- Izveidojam jaunu Android Application Project `Vardi`.

- Atveram `activity_main.xml` datni grafiskajā skatā.
- Pievienojam vienu `PlainText` elementu, kura īpašībai `Hint` uzliekam vērtību `Vārds`, un vienu button elementu ar tekstu `Parādīt vārdu`.

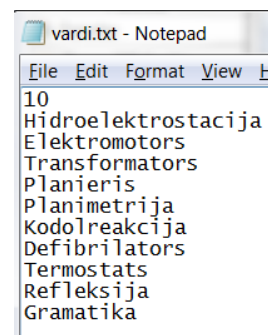


ANDROID PROJEKTS «Vardi» - ASSETS MAPE

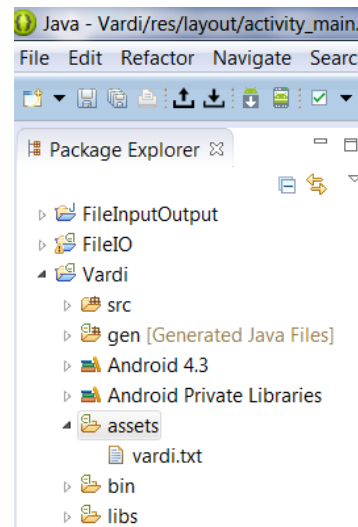
- Atrodam datorā projekta *Vardi* apakšmapi *assets*



- Izveidojam tajā jaunu teksta datni `vardi.txt`
- Datnē `vardi.txt` 1.rindā ierakstām skaitli - cik vārdu būs šajā datnē
- Tālākajās rindās ierakstām atbilstošu skaitu vārdu



- Atgriezieties programmas logā un mapei *assets* izpildām darbību *Refresh* – mapē parādās tur saglabātā datne *vardi.txt*



ANDROID PROJEKTS «Vardi» - METODE *paraditVardu*

- Pogas īpašībai *OnClick* ieliekam vērtību *paraditVardu*
- Pārejām uz skatu *MainActivity.java* un papildinām programmu ar metodi *paraditVardu*

```
public class MainActivity extends Activity {
```

```
    public void paraditVardu(View v){
```

```
    }
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

- Pievienojam *EditText* objektu darbam ar teksta elementu un try-catch konstrukciju dabam ar datni

```
public class MainActivity extends Activity {
    public void paradiVardu(View v){
        EditText vards=(EditText)findViewById(R.id.editText1);
        try{

        }catch(Exception e){
            vards.setText("lasot slikti");
        }
    }
}

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

ANDROID PROJEKTS «Vardi» - VĀRDA IZVĒLE

- Vispirms no datnes *vardi.txt* nolasām pirmo rindu
- Rindu pārveidojam par skaitlisku lielumu, iegūstam rindu skaitu
- No rindu skaita uzģenerējam gadījuma skaitli *n*
- Ciklā lasām no datnes *n* reizes, lai nolasītu *n*-to vārdu
- *N*-to vārdu parāda uz ekrāna

```
public void paradiVardu(View v){
    EditText vards=(EditText)findViewById(R.id.editText1);
    try{
        BufferedReader lasa= new BufferedReader(new InputStreamReader(getAssets().open("vardi.txt")));
        String rinda=lasa.readLine();
        int rinduSkaitis=Integer.parseInt(rinda);
        int n=new Random().nextInt(rinduSkaitis)+1;
        for(int i=0;i<n;i++){
            rinda = lasa.readLine();
        }
        vards.setText(rinda);
        lasa.close();
    }catch(Exception e){
        vards.setText("lasot slikti");
    }
}
```