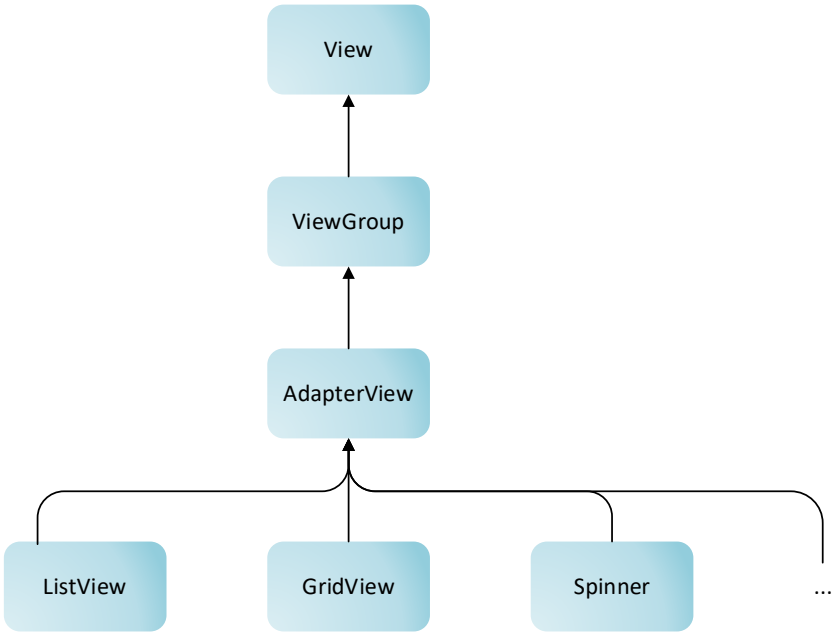
Rukovanje kolekcijama

AdapterView komponenta

Prikaz kolekcije podataka u Androidu omogućen je komponentom *AdapterView* (i istoimenom klasom) i komponentama koje je nasleđuju. Komponente koje nasleđuju AdapterView komponentu su:

- *ListView* - komponenta za prikaz kolekcije gde su podaci postavljeni jedni ispod drugih
- *GridView* - komponenta za prikaz kolekcije kao mreže podataka
- *Spinner* - komponenta za prikaz kolekcije kao padajuće liste.

Hijerarhija *AdapterView* komponenti je prikazana na slici - Slika 1.



Slika 1 - Hijerarhija AdapterVIew komponenti

Za svaki element komponente *AdapterView* može da se generiše poseban izgled. Tako, na primer, možemo da definišemo da se jedan red liste sastoji iz dva *TextView* elementa ili jednog *TextView* i jednog *ImageView* elementa i slčno.

AdapterView komponente ne poznaju detalje objekata koje sadrže, one određuju gde će i kako unutar Aktivnosti biti prikazani podaci. Za dobavljanje podataka zaslužan je *Adapter* koji će biti detaljnije opisan u narednim poglavljima.

Pregled AdapterView komponenata

ListView

ListView je komponenta koja omogućava prikaz liste objekata. *ListView* je komponenta koja određuje gde će unutar Aktivnosti biti prikazani podaci. Za dobavljanje podataka zaslužan je *Adapter*.

Primer definisanja *ListView* komponente unutar pogleda Aktivnosti:

```
<ListView

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:id="@+id/lista"/>
```

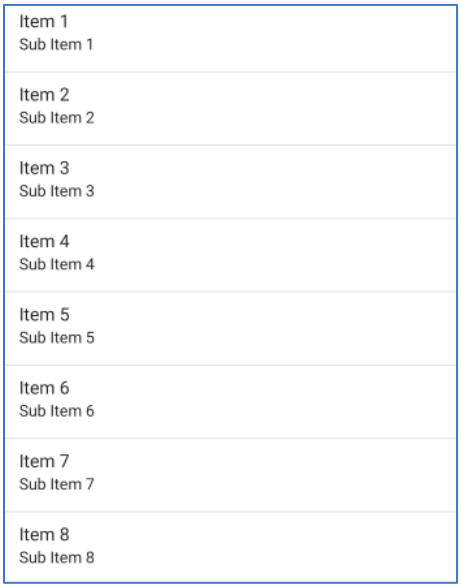
Komponenta *ListView* je u Javi implementirana kao istoimena klasa. Ova klasa omogućava postavljenje osluškivača događaja kao što su: klik na listu, klik na jedan element liste, dugi klik na element liste i drugi. Primer postavljanja osluškivača događaja - klik na jedan element liste dat je u nastavku.

```
ListView list = findViewById(R.id.lista);

list.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    @Override

    public void onItemClick(AdapterView<?> p, View v, int pos, long id) {

        // handle click on item

    }

});
```

Klasa *ListView* ima metodu *setEmptyView* kojom se definiše izgled liste kada u njoj nema podataka. Ova metoda kao parametar prima objekat *View* elementa.

```
<!--layout.xml-->

<FrameLayout

     android:layout_width="match_parent"

     android:layout_height="match_parent">

     <ListView

         android:id="@+id/lista"

         android:layout_width="match_parent"

         android:layout_height="match_parent"/>

      <TextView

         android:id="@+id/emptyView"

         android:layout_width="wrap_content"

         android:layout_height="wrap_content"

         android:layout_gravity="center"

         android:text="@string/this_list_is_empty" />

</FrameLayout>


// Activity.java

ListView list = findViewById(R.id.lista);

TextView emptyView = findViewById(R.id.emptyView):

list.setEmptyView(emptyView);
```

Izgled *ListView* komponente prikazan je na slici - Slika 2.



Slika 2 - ListView

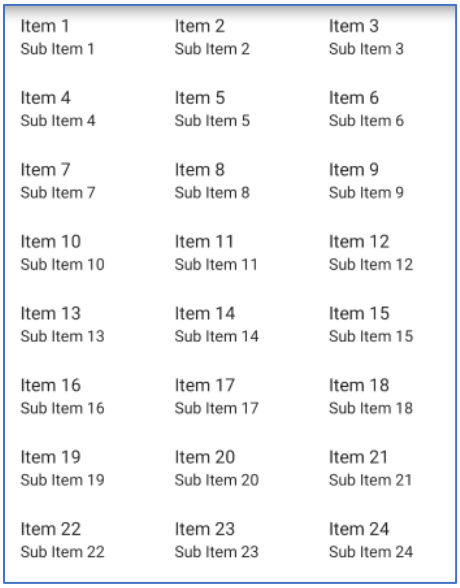
GridView

GridView je komponenta koja označava mesto unutar Aktivnosti na kome će se nalaziti podaci prikazani u mreži. Za dobavljanje podataka zaslužan je Adapter. Primer definisanja *GridView* komponente:

```
    <GridView

            android:layout_width="match_parent"

            android:layout_height="match_parent"

            android:numColumns="3"

            android:id="@+id/grid" />
```

GridView komponenta je u Javi opisana klasom *GridView*. Kao i *ListView*, i *GridView* klasa omogućava postavljanje osluškivača događaja.

Izgled *GridView* komponente sa 3 kolone prikazan je na slici - Slika 3.



Slika 3 - GridView

Spinner

Spinner je komponenta koja označava mesto unutar Aktivnosti na kome će se nalaziti padajuća lista podataka. Primer definisanja *Spinner* komponente.

```
    <Spinner

            android:layout_width="match_parent"

            android:layout_height="match_parent"

            android:id="@+id/spinner"/>
```

Spinner komponenta je u Javi opisana klasom *Spinner* koja omogućava postavljanje osluškivača događaja.

Izgled *Spinner* komponente prikazan je na slici - Slika 4.

Slika 4 - Spinner

Adapter

Adapter klasa predstavlja spregu između podataka i *AdapterView* komponente. Njegov zadatak je da pruži model podataka *AdapterView* komponenti i da konvertuje podatke u polja *AdapterView* komponente.

Slika 5 - Komunikacija izvora podataka

Dva standardna Adaptera u Androidu su:

- *ArrayAdapter* - radi sa podacima koji se nalaze unutar liste
- *CursorAdapter* - radi sa podacima koji su smešteni u bazu podataka.

Pored definisanja AdapterView komponente u XML rasporedima (ListView, GridView, Spinner) potrebno je definisati i izgled jednog elementa unutar kolekcije. U ovu svrhu, moguće je koristiti predefinisane rasporede (*simple_list_item1, simple_list_item2*…) ili kreirati nove rasporede po želji korisnika. Prilikom kreiranja Adaptera navodi se XML raspored jednog elementa kolekcije u koji se smeštaju podaci kao i tip podataka koje Adapter sadrži.

```
ArrayAdapter<T> adapter = new ArrayAdapter<T>(Context c, int resource)
```

Primer kreiranja Adaptera koji objekte tipa String smešta u predefinisani element kolekcije koji se sastoji od jednog TextView elementa:

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

                android.R.layout.simple_list_item_1);
```

Rukovanje elementima kolekcije se obavlja nad objektom Adaptera. Neke od metoda za rukovanje kolekcijama koje pruža Adapter klasa su:

- `getCount()` - dobavljanje broja elemenata u kolekciji
- `getItem(int position)` - dobavljanje elementa na određenoj poziciji
- `isEmpty()` - provera da li je kolekcija prazna
- `getView(int position, View v, ViewGroup parent)` - dobavljanje View elementa za prikaz podataka na određenoj poziciji
- `notifyDataSetChanged()` - obaveštava da su se podaci promenili i da treba osvežiti izgled kolekcije.

U zavisnosti od tipa Adaptera omogućene su i dodatne funkcije za rukovanje podacima. Tako, na primer, *ArrayAdapter* sadrži osnovne funkcije za rukovanje nizovima kao što su:

- `add(T object)` - dodavanje elementa u niz
- `remove(T object)` - uklanjanje elementa iz niza
- `clear()` - uklanjanje svih elemenata iz niza
- `insert(T object, int index)` - dodavanje elementa na određenu poziciju u nizu…

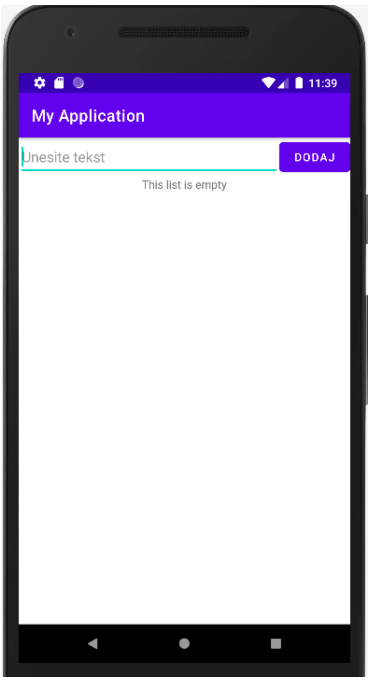
Primer kreiranja Adaptera koji objekte tipa String smešta u predefinisani element kolekcije *simple_list_item1,* povezivanja Adaptera sa ListView komponentom i manipulisanja podacima u listi:

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,

                android.R.layout.simple_list_item_1);

adapter.add("Item1"); // dodavanje elemenata

adapter.add("Item2");

adapter.remove("Item2"); // uklanjanje elementa



ListView list = findViewById(R.id.lista);

list.setAdapter(adapter); // povezivanje Adaptera i ListView
```

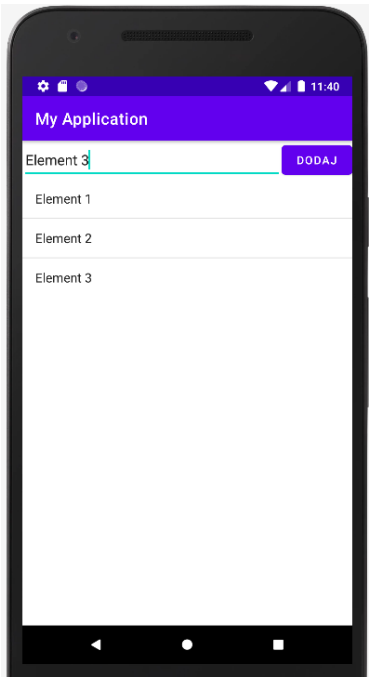
Ukoliko je potrebno smestiti podatke u korisnički definisane objekte, moraju se definisati prilagođeni Adapteri.

Zadatak za vežbu

- Kreirati XML raspored Aktivnosti (Slika 6) koji se sastoji od:
  - EditText polja
  - dugmeta i
  - ListView elementa
- Kreirati ArrayAdapter koji dobavlja elemente tipa String za ListView
- Definisati da jedan element liste ima predefinisani XML raspored *simple_list_item1*
- Ukoliko je lista prazna prikazuje se poruka: *Lista je prazna* (u TextView elementu)
- Klikom na dugme u listu se dodaje element sa tekstom koji je korisnik uneo u EditText polje (zaštititi se od nevalidnog unosa)
- Klikom na jedan element liste isti se briše iz liste
- Dugim klikom na element liste se u *Toast* poruci ispisuje vrednost elementa.



Slika 6 - Izgled Aktivnosti kada je lista prazna



Slika 7 - izgled Aktivnosti nakon dodavanja elemenata