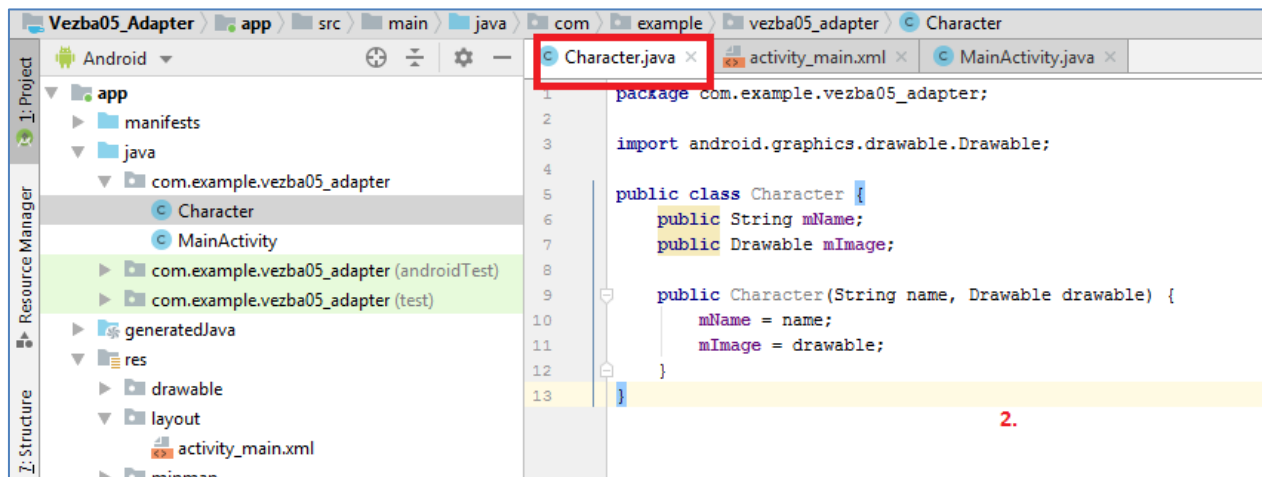


Adapter vežba

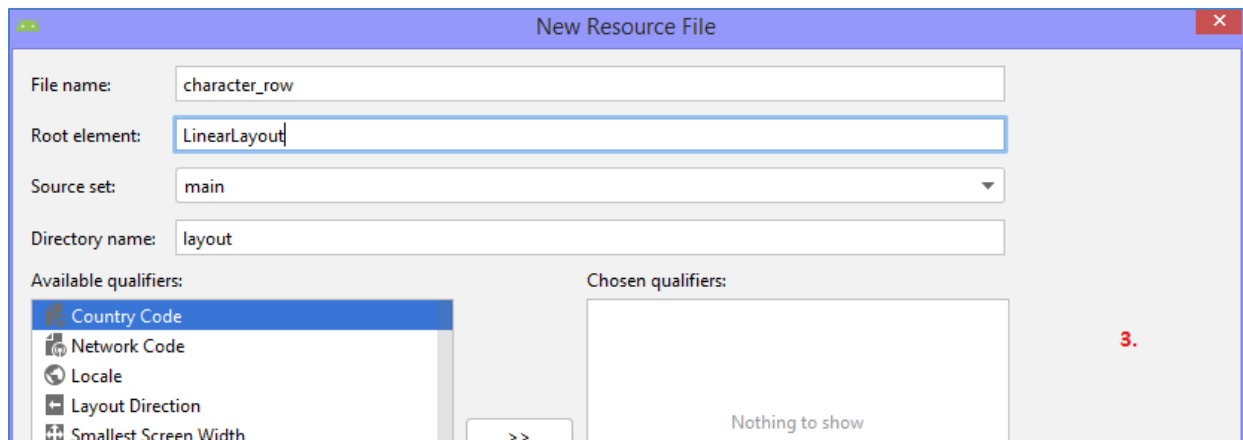
Do sada smo se upoznali sa ListView komponentom u Androidu i načinu rukovanja listom preko Adaptera (Vežba 4). U toj vežbi koristili smo ugrađeni adapter – ArrayAdapter u koji smo smeštali String – ove. U ovoj vežbi, pošto želimo da napravimo listu čiji su elementi kompleksniji, ne možemo da koristimo ugrađeni adapter, već ćemo sami napisati Adapter komponentu.

1. Kreiramo novi projekat – AdapterExample. Naša aplikacija se sastoji iz liste čiji svaki element sadrži sliku i tekst.
2. Kako bismo predstavili jedan element liste potrebno je da najpre kreiramo novu klasu *Character.java* sa poljima koja odgovaraju komponentama elementa – u našem slučaju slika i ime. Kreiramo dva polja, jedno tipa Drawable (za sliku), drugi tipa String (Slika 1).

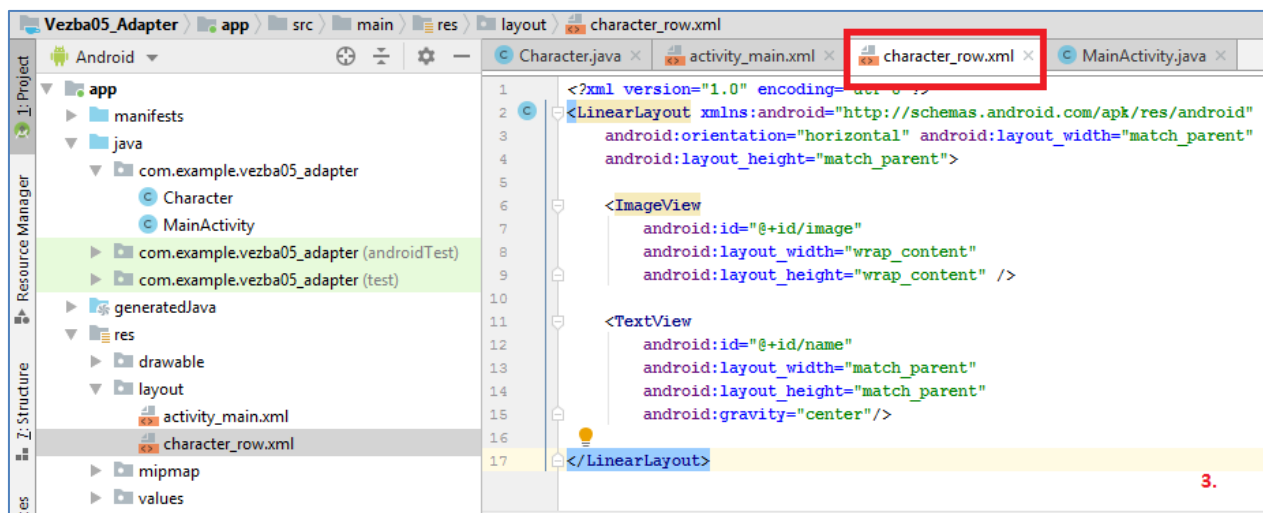


Slika 1 - Character.java klasa

3. Sledeći korak je kreiranje layout-a. Pored *activity_main.xml* koji obuhvata vertikalno orijentisani Linear Layout sa jednom ListView komponentom, pravimo još jedan layout koji odgovara prikazu jednog elementa liste – *character_row.xml* (Slika 2). Ovaj layout treba da sadrži ImageView i TextView, raspoređeni jedno pored drugog (Slika 3).

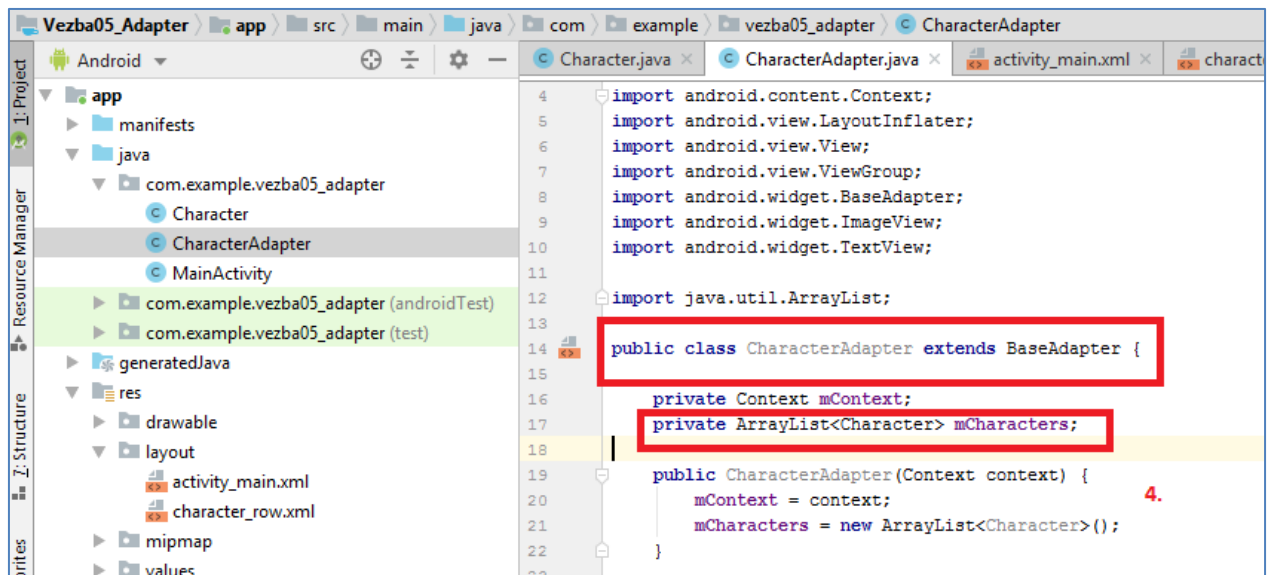


Slika 2 - Dodati novu layout datoteku



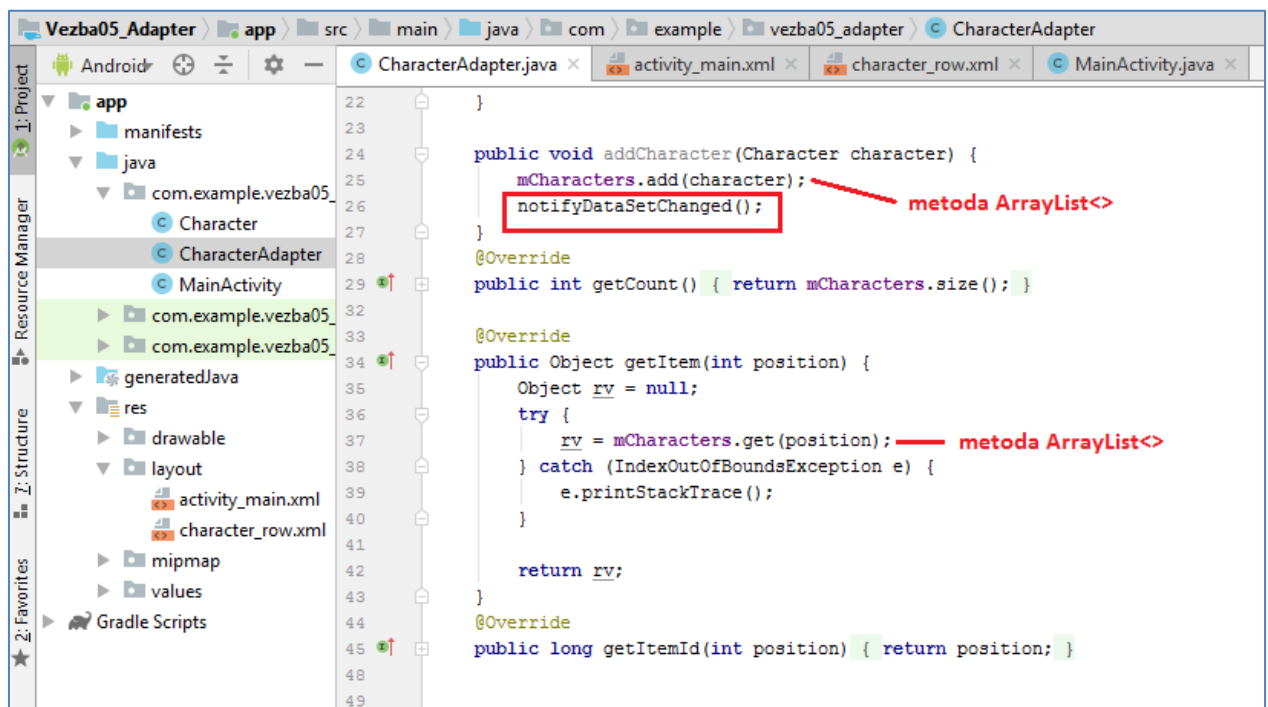
Slika 3 - character_row.xml datoteka

4. Sledeći korak je pravljenje Adaptera. Kreiramo novu Java klasu *CharacterAdapter.java*. Ova klasa treba da nasleđuje *BaseAdapter* i samim tim moramo implementirati *getView* metodu. Pored ove, imamo još par dodatnih metoda koje nam olakšavaju rukovanje listom (Slika 4). Koristimo tip **ArrayList** u koji smeštamo elemente tipa **Character** (instance klase koju smo napravili u drugom koraku).



Slika 4 - CharacterAdapter.java

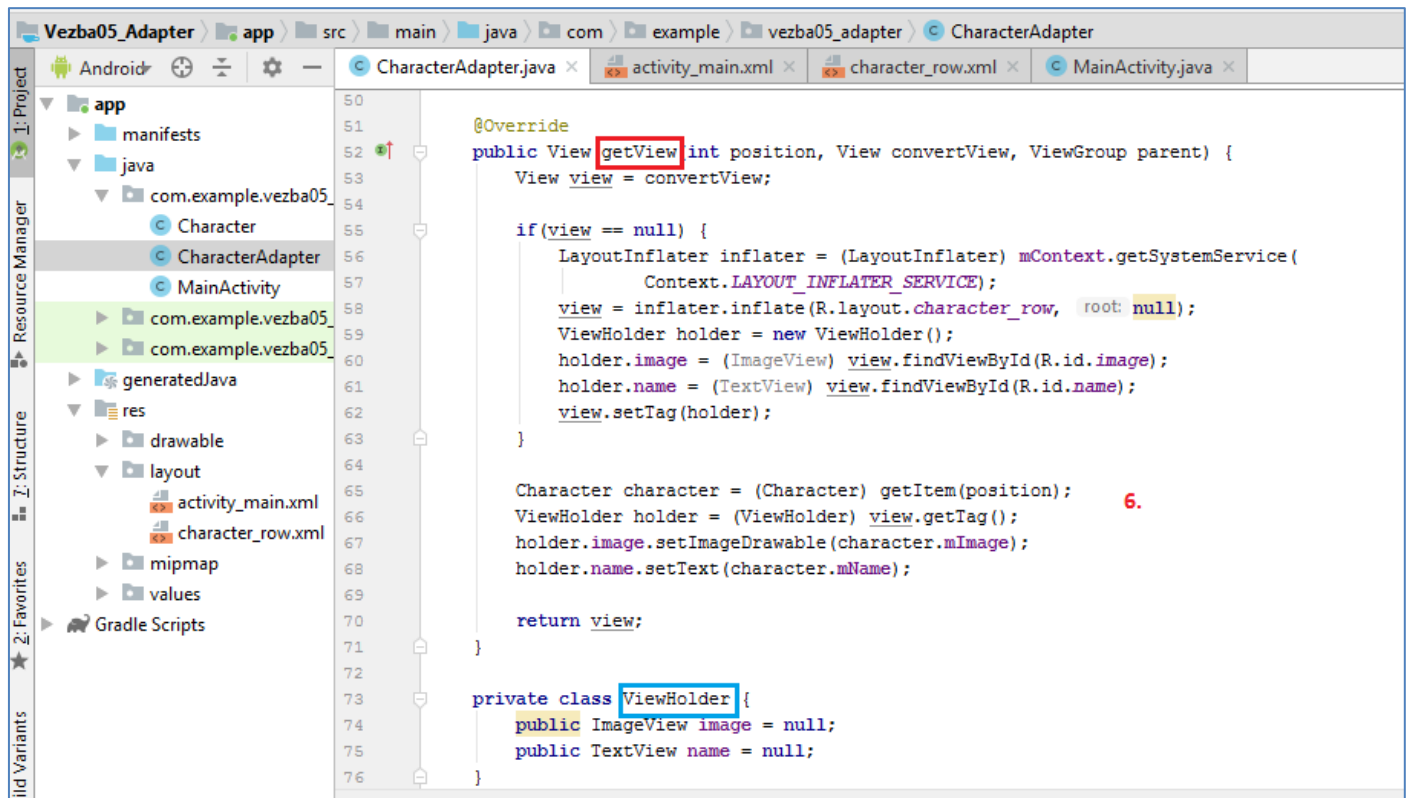
- Implementiramo metode za rukovanje listom: *addCharacter()* - za dodavanje elemenata u listu, *getCount()* – vraća veličinu liste, *getItem()* –vraća objekat sa prosledene pozicije, *getItemId()* – vraća indeks elementa sa pozicije (Slika 5). **Pitanje:** pogledati šta radi metoda *notifyDataSetChanged()*.



Slika 5 - CharacterAdapter.java

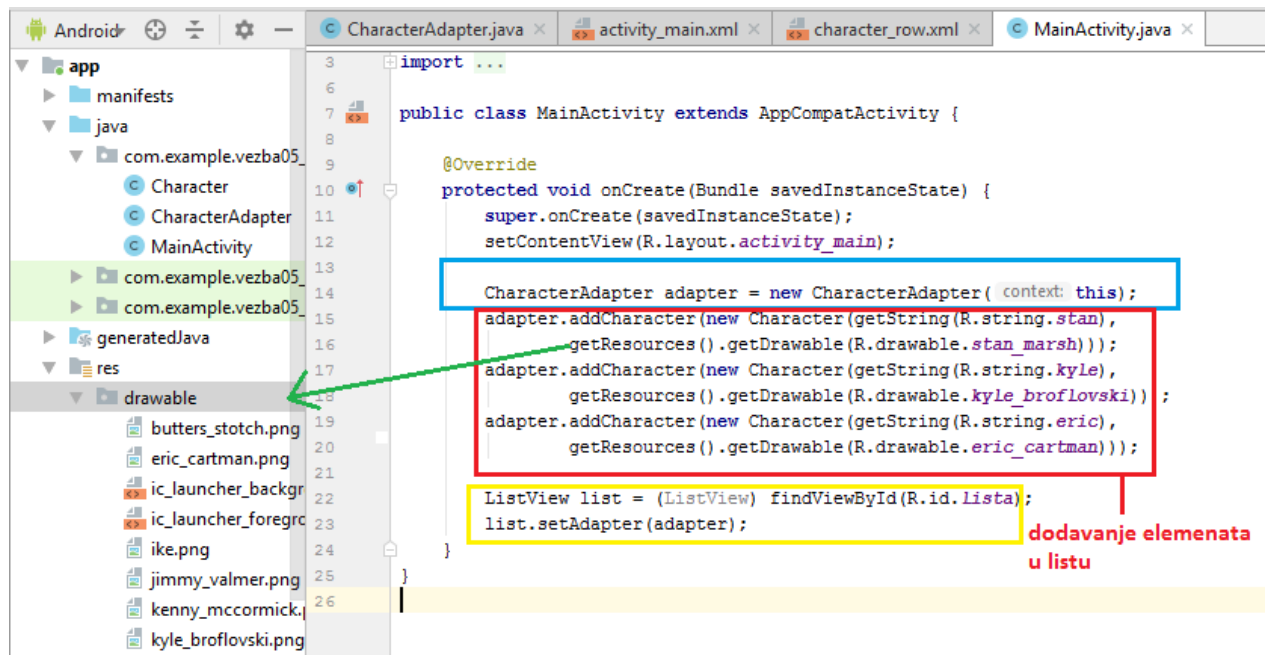
- Sledeći korak je implementacija metode *getView()*. U ovoj metodi regulišemo prikaz elemenata liste, upotrebom ViewHolder komponente. Inicijalno, koristimo klasu *LayoutInflater* i metodu

inflate() kako bismo od layout komponente za prikaz jednog elementa izgenerisali View objekat sa kojim lista rukuje (Slika 6).



Slika 6 - getView() metoda adaptera

7. Sada možemo u okviru MainActivity – ja da instanciramo CharacterAdapter i rukujemo listom. Kako bismo dodavali elemente u listu, pozivamo metodu adaptera *addCharacter*. Drawable resurse u .png formatu smeštati u res -> drawable direktorijum. Na kraju, kako bismo promene videli u listi, nad instancom ListView-a pozivamo metodu *setAdapter* i prosledjujemo instancu našeg adaptera (Slika 7).
8. Prevesti i pokrenuti aplikaciju i testirati sa barem 10 elemenata u listi.



Slika 7 - MainActivity

- 9.
- 10.
- 11.