

Projekat iz Razvoja mobilnih aplikacija

Spirala 3

Prije implementacije zadataka implementirajte sljedeće:

- Klasa **Knjiga** treba da sadrži sljedeće atribute: **id** tipa *string*, **naziv** tipa *string*, **autori** tipa *ArrayList<Autor>*, **opis** tipa *string*, **datumObjavljivanja** tipa *string*, **slika** tipa *URL* i **brojStrinica** tipa *int* (klasa treba da ima konstruktor koji prima parametre u istom redoslijedu kako su atributi nabrojani, te gettere i settere za sve atribute npr. `getNaziv` i `setNaziv`)
- Klasa **Autor** treba da sadrži sljedeće atribute: **imeiPrezime** tipa *string* i **knjige** tipa *ArrayList<string>* (klasa treba da ima konstruktor koji prima imeiPrezime autora te id knjige koji će se dodati u niz knjige, potrebno je implementirati gettere i settere kao i metodu **dodajKnjigu(string id)** koja dodaje dati id u listu knjige (ako već ne postoji)
- Prethodne klase obavezno trebaju imati navedene atribute, ali niste ograničeni na samo ove atribute. Tj. možete imati i dodatne atribute ako vam je lakše da realizujete zadatke, ali ne smijete mijenjati konstruktore koji su navedeni ranije (možete dodati nove konstruktore).

Zadatak a. Implementirajte klasu **DohvatiKnjige** koja je nasljeđena iz `AsyncTask` klase. Kao parametre za izvršavanje zadatka ova klasa prima nazive knjiga. Zadatak je da prođete kroz sve nazive knjiga i dohvatite listu knjiga koje imaju dati naziv. Koristite Google Books API i sljedeću metodu web servisa <https://www.googleapis.com/books/v1/volumes?q=intitle:Naziv>. Za svaki prosljeđeni naziv trebate pozvati web servis, a na kraju formirajte listu svih knjiga koje ste skupili sa ovog web servisa. Svaki put postavite da servis vrati najviše 5 rezultata (koristeći parametar `maxResults`). Konstruktor **DohvatiKnjige** prima jedan parametar, a to je implementacija interfejsa **IDohvatiKnjigeDone** koji ima samo jednu metodu **onDohvatiDone(ArrayList<Knjige>)**. Nakon što skupite sve knjige pozovite metodu `onDohvatiDone`. Ovaj interfejs neka bude deklarisan unutar klase **DohvatiKnjige** (kao interfejs `OnMuzicarSearchDone` iz vježbe 4) i koristite ga za komunikaciju sa fragmentom koji će pozivati **DohvatiKnjige**.

Zadatak b. Implementirajte klasu **DohvatiNajnovije** koja je nasljeđena iz `AsyncTask` klase. Kao parametre za izvršavanje zadatka ova klasa prima ime i prezime autora. Zadatak je da koristeći isti web servis kao u zadatku 1 pronađete zadnjih 5 najnovijih knjiga koje je dati autor napisao. Konstruktor **DohvatiNajnovije** prima jedan parametar, a to je implementacija interfejsa **IDohvatiNajnovijeDone** koji ima samo jednu metodu **onNajnovijeDone(ArrayList<Knjige>)**. Nakon što dobijete odgovor od web servisa pozovite metodu `onNajnovijeDone` sa knjigama koje je web servis vratio. Ovaj interfejs neka bude deklarisan unutar klase **DohvatiNajnovije** i koristite ga za komunikaciju sa fragmentom koji će pozivati **DohvatiNajnovije**.

Zadatak c. Implementirajte `IntentService` **KnjigePoznanika**. Ovaj servis u intent extra podacima prima sljedeće: **idKorisnika** tipa *string* i **receiver** tipa *ResultReceiver*. Kada servis primi intent poziva Google Books API web servis <https://developers.google.com/books/docs/v1/using#WorkingBookshelves>. Kao rezultat **KnjigePoznanika** receiver-u treba vratiti listu svih knjiga, sa svih bookshelves koji su javni za korisnika sa datim id-em (`userId` je isto što i `idKorisnika`). Unutar klase definišite tri statusa: `STATUS_START` sa vrijednosti 0, `STATUS_FINISH` sa vrijednosti 1 i `STATUS_ERROR` sa vrijednosti 2. Prije pozivanja web servisa receiveru pošaljite `STATUS_START` sa `Bundle.EMPTY`, nakon uspješno obrađenog odgovora od web servisa receiveru pošaljite

STATUS_FINISH sa Bundle koji ima podatak **listaKnjiga** koja je parcelableArrayList sa knjigama koje ste dobili pozivajući web servis, a ako je došlo do greške pošaljite STATUS_ERROR sa Bundle.EMPTY.

Zadatak U početnoj aktivnosti (KategorijeAkt) dodajte dugme sa id-em **dDodajOnline** sa tekstom *dodaj knjigu online*. Klik na ovo dugme otvara fragment **FragmentOnline**. FragmentOnline sadrži spinner sa kategorijama koji ima id **sKategorije**, edit text sa id-em **tekstUpit**, spinner **sRezultat**, dugme sa id-em **dRun** i tekstom *Pretraga*, dugme sa id-em **dAdd** i tekstom *dodaj knjigu* i dugme sa id-em **dPovratak** koje vraća nazad na početni izgled aplikacije. Klik na dugme dRun radi sljedeće:

- a.1) [1 bod] Ako je u tekstUpit upisana jedna riječ pozovite DohvatiKnjige sa datom riječi
- a.2) [2 boda] Ako je u tekstUpit upisano više stringova koji su odvojeni sa ; pozovite DohvatiKnjige gdje svaki string predstavlja jedan naziv knjige
- b) [3 boda] Ako je u tekstUpit upisano autor:Neko Ime. Pozovite DohvatiNajnovije gdje je parametar string koji slijedi nakon 'autor:'
- c) [4 boda] Ako je u tekstUpit upisano korisnik:nekiUserId. Pozovite KnjigePoznanika gdje je parametar string koji slijedi nakon 'korisnik:'

Nakon što se završi pretraga u spinner sRezultat upišite nazive knjiga koje je web servis vratio. Klik na dugme dAdd dodaje knjigu koja je odabrana u spinneru sRezultat u kategoriju koja je odabrana u spinneru sKategorije. Sliku za ove knjige zasada hardkodirajte. Knjige koje web servis vrati moraju imati sve podatke popunjene (atribut slika popunite sa vrijednosti iz imageLinks.thumbnail).

Napomena: Rok za rad na spirali 3 je **22.05.2018 do 23:59**. Spirala se treba nalaziti na istom repozitoriju kao i prethodna spirala. Novu spiralu postavite na master branch, a prethodnu spiralu postavite na branch spirala2.