

Kehittämisosas: Listasovellus – projekti

Harjoitustyön aihe:

Harjoitustyön aiheena oli tarkoitus tehdä listasovellus, jonka avulla käyttäjä voi luoda, tarkastella ja muokata omia listoja ja merkata listojen kohtia valintaruuduilla. Sovellus on tarkoitettu sovellettavaksi erilaisiin kauppa- tai tehtävälistöihin.

Olioluokka

Työssä on käytetty kahta olioluokkaa `UserList` ja `ListOfLists`.

Luokassa `UserList` on neljä kenttää:

- Merkkijono listan nimelle
- Totuusarvolista listan kohtien merkkaamiselle
- Merkkijonolista listan kohdille
- Kokonaisluku listan pituudelle

Nimelle, itse listalle ja merkattujen kohtien listaamiselle löytyy `get`- ja `set`-metodit, kun taas pituudelle on jätetty pelkkä `get`-metodi. Näiden lisäksi on kaksi parametrillistä alustajaa, ensimmäinen, joka luo tyhjän listan annetun nimen perusteella ja toinen, joka ottaa parametreina jo valmiin listan. Jälkimmäistä alustajaa käyttäessä pitää huomioida, että nimi tulee antaa erikseen.

Luokassa `ListOfLists` on kaksi kenttää:

- Lista käyttäjälistoille, eli `UserList`-olioille, eräänlainen ”päälista”
- Viittaus tiedostoon mihin listat tallennetaan `File`-luokan oliona

Luokalla on parametrillinen alustaja joka saa parametreinaan avattavan tiedoston ja hakee sieltä tallennetut erillisellä metodilla. Tämän lisäksi luokka sisältää metodit listojen tallentamiselle binääritiedostoon, päälistan päivittämiseksi sekä listan lisäämiselle, poistamiselle ja löytämiselle päälistasta.

Ohjelman rakenne

Ohjelma koostuu yhdeksästä luokasta: Olioluokista `UserList` ja `ListOfLists`, `Fragment`-luokista `MainListFragment`, `MainListEntryFragment`, `UserListFragment` ja `UserListEntryFragment`, `RecyclerView`-adapteriluokista `MainListAdapter` ja `UserListAdapter` sekä pääohjelmasta `MainActivity`.

Olioluokka `UserList` määrittää käyttäjän listan informaation ja sisältää metodit käyttälistojen muokkaamiselle.

`ListOfLists` taas on eräänlainen päälista, jonka tarkoitus on säilöä kaikki käyttäjän luomat listat ja olla vastuussa näiden tallentamisesta ja uudelleen lukemisesta, kun ohjelma taas avataan. Olioluokat ollaan kuvattu jo tarkemmin ylempänä ja `JavaDoc`issa.

Ohjelma sisältää kaksi `Fragment`tiä asetettuna samaan `NavHostFragment`tiin, joka vaihtaakin näiden `Fragment`tien välillä riippuen siitä halutaanko olla päävalikossa (`MainListFragment`) vai listaa lukemassa (`UserListFragment`).

MainListFrament toimii ohjelman päävalikkona, minkä käyttäjä ensimmäiseksi näkee. Fragmentti itsessään sisältää kaksi graafista komponenttia: RecyclerView rvMainList ja painike btnNewList. Luokassa on metodit onCreateView(), joka ajetaan kun näkymä luodaan, updateList() ja getInstance(). Metodi onCreateView() määrittää käytettävän RecyclerView-olion ja asettaa sille oikean adapterin ja tämän käyttämän datan. Metodi updateList() ollaan vain tarkoitettu näkyvän päälistan päivittämiseen ja getInstance()-metodi palauttaa viittauksen fragmentin olioon. Painike btnNewList käyttää MainActivity-luokan newList()-metodia.

UserListFragment on taas Fragmentti, jonka on tarkoitus esittää käyttäjän luoma käyttäjälista. Kun MainListFragment, se pitää sisällään vain kaksi graafista komponenttia: RecyclerView rvUserList ja painike btnNewItem. Luokasta löytyy myös metodit onCreateView() ja updateList(). Metodi onCreateView ajetaan joka kerta kun näyttö itsessään luodaan ja sen tehtävä on vain asettaa RecyclerView-oliolle adapteri ja tämän käyttämä data. Metodi updateList() taas toimii myös samalla tavalla kuin toisessa Fragmentissa, sen on tarkoitus saada parametreinaan merkkijono, minkä se lisää listalle uudeksi kohdaksi. Tämä ei vielä itsessään luo uutta käyttöliittymäkomponenttia, vaan ainoastaan lisää annetun merkkijonon käsiteltävään käyttäjälistaan.

MainListEntryFragment – ja UserListEntryFragment – luokat ovat ainoastaan määrittämässä miltä MainListFragmentin ja UserListFragmentin RecyclerViewien listojen alkiot näyttävät. Tällä hetkellä MainListEntryFragment sisältää yhden ainoan TextView-olion, jonka on tarkoitus esittää listan nimi ja UserListEntryFragment sisältää CheckBox-olion.

MainListAdapter – luokka ollaan tehty määrittämään adapteri MainListFragment – luokan RecyclerView – oliolle. Se luo RecyclerView – listauksen alkiot ylempänä mainitun MainListEntryFragmentin pohjalta ja asettaa jokaiselle näistä alkioista kuuntelijat mainClickListener ja mainListLongClickListener. Kuuntelija mainClickListener on onClickListener, joka tarjoaa metodin onClick, mikä avaa NavControllerin avulla UserListFragment - näkymän, tarjoten sille Bundleen pakattuna sen UserList-olion, mitä painettu kohta RecyclerViewissä edustaa. Kuuntelija mainListLongClickListener taas odottaa pitkää painallusta ja sen onLongClick – metodi avaa AlertDialog-ikkunan, joka varmistaa käyttäjältä haluaako tämä poistaa listan mitä juuri painoi pitkään. Vastauksesta riippuen, metodi poistaa listan ja päivittää listauksen.

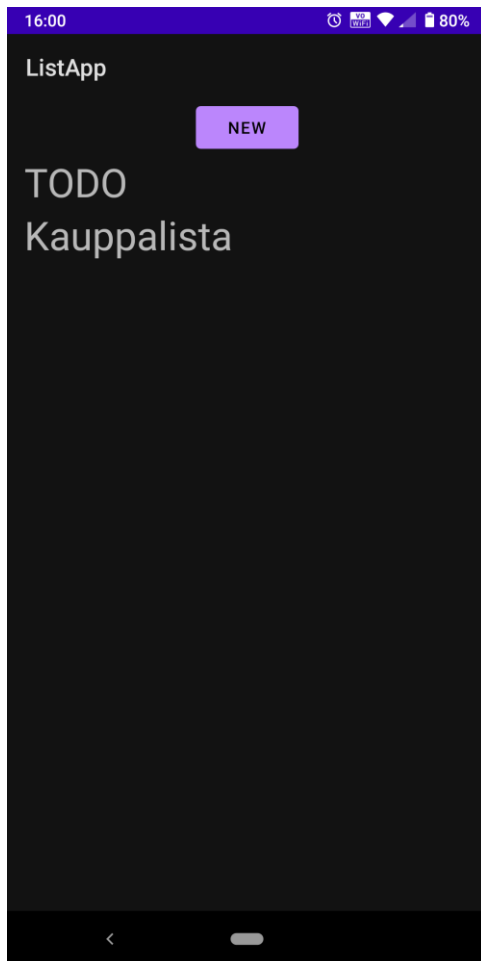
UserListAdapter – luokka taas määrittää adapterin UserListFragment – luokan RecyclerView – oliolle. Samalla tavalla kuin päälistauksin, se luo listan alkiot UserListFragmentin pohjalta ja asettaa alkiolle kuuntelijat, tällä kertaa lambda – lausekkeilla. Alkiot sisältävät Checkbox-oliot, joille annetaan tekstiksi aina yhden käyttäjälistan kohdan merkkijono ja valintaruutu merkataan käyttäjälistan checkedItems – totuusarvomuuuttujalistan mukaan. Kuuntelijoina on onCheckedChangeListener, joka kuuntelee valintaruutujen muutosta ja onLongClickListener, joka toimii samalla tavalla kuin MainListAdapter-luokassakin. Jos valintaruutu merkataan, se lisää tämän merkkauksen muistiin tarkasteltavalle UserList-oliolle ja päivittää olion päälistassa.

Näiden lisäksi ohjelmassa on pääaktiviteetti MainActivity. Se ensimmäisenä onCreate()-metodissaan määrittää tiedoston, minkä pohjalta päälista haetaan. Tiedoston nimi itsessään on vakio jotta sitä voi muuttaa helposti tarvittaessa. Pääaktiviteetilla on yhtenä kenttäänään ListOfLists – olio, joka toimii nimenomaan staattisena päälistana, mikä sisältää kaikki käyttäjän listat. Täten kaikki päälistan muokkaukset voi tapahtua yksinkertaisesti kutsumalla oliota "MainActivity.lists" - viittauksella. Tämän lisäksi pääaktiviteetti määrittää ohjelman painikkeiden käyttämät newList() ja newItem() - metodit.

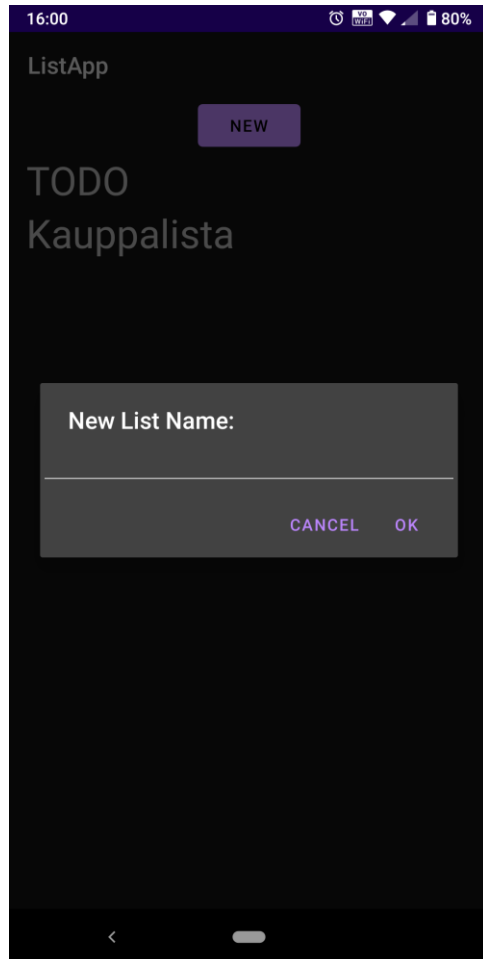
Metodi newList() avaa uuden AlertDialogin, kysyy käyttäjältä nimeä uudelle listalle ja luo syötteen pohjalta uuden listan, lisäten sen päälistaan välittömästi. Metodi newItem() avaa myös AlertDialogin, joka kysyy uudelle listan kohdalle merkkijonoa, minkä pohjalta se lisää sillä hetkellä tarkastelussa olevalle UserList – oliolle uuden kohdan.

Kaikki metodit ja kentät löytyvät JavaDocista.

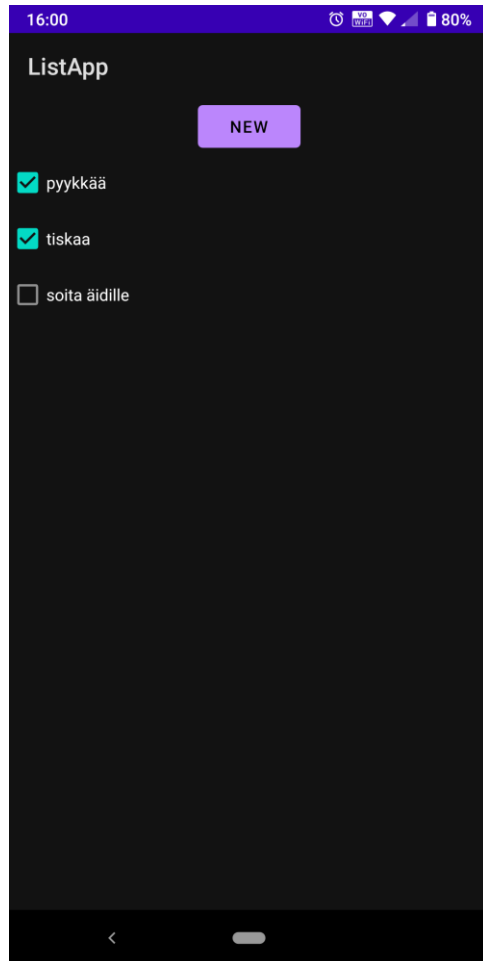
Sovellus toimii tällä hetkellä niin, että kun sen avaa, avautuu ns. “päävalikko” tai päälista. Tämä lista saa alkionsa ohjelman määritellystä binääritiedostosta, mitkä se hakee heti avautuessaan. Ensimmäiseksi ohjelman avautuessa tulisi näkyä listaus tallennetuista käyttäjälistoista, tai jos tiedostoa ei syystä tai toisesta löydy tai saada auki, pohjana luotu vakiolista.



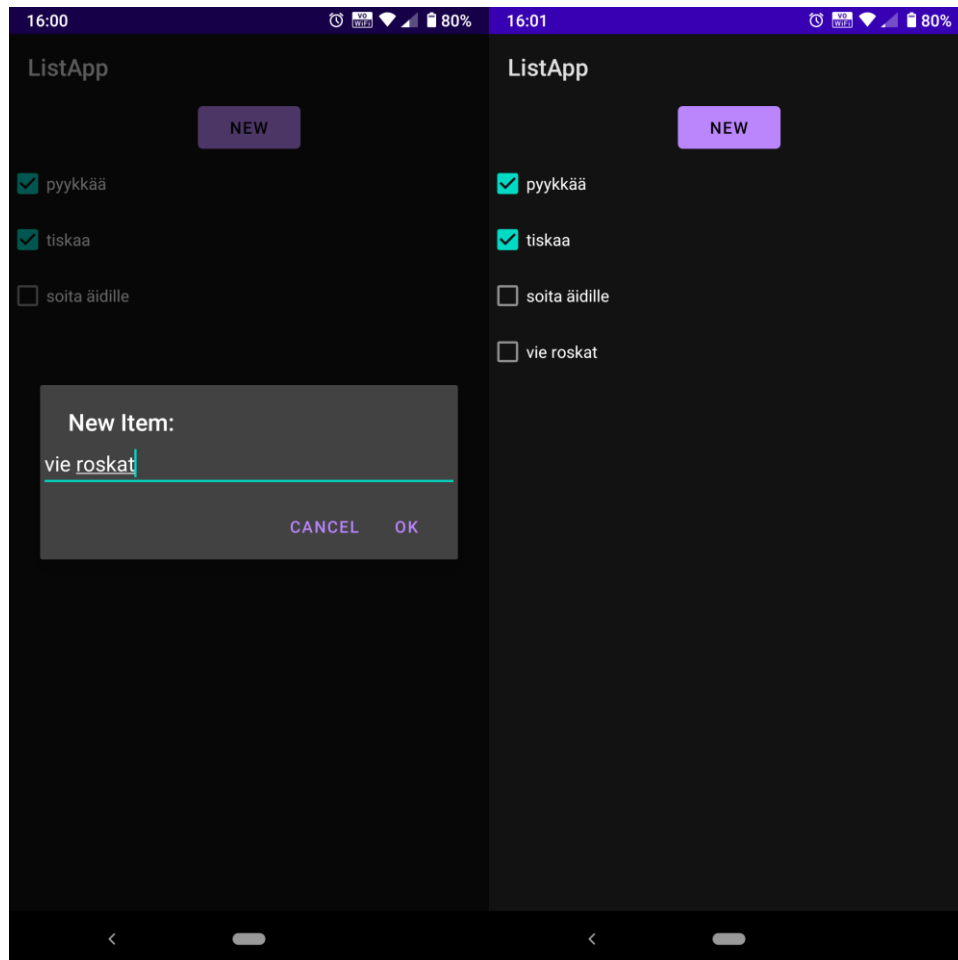
Painamalla painiketta, käyttäjä voi lisätä kokonaan uuden listan.



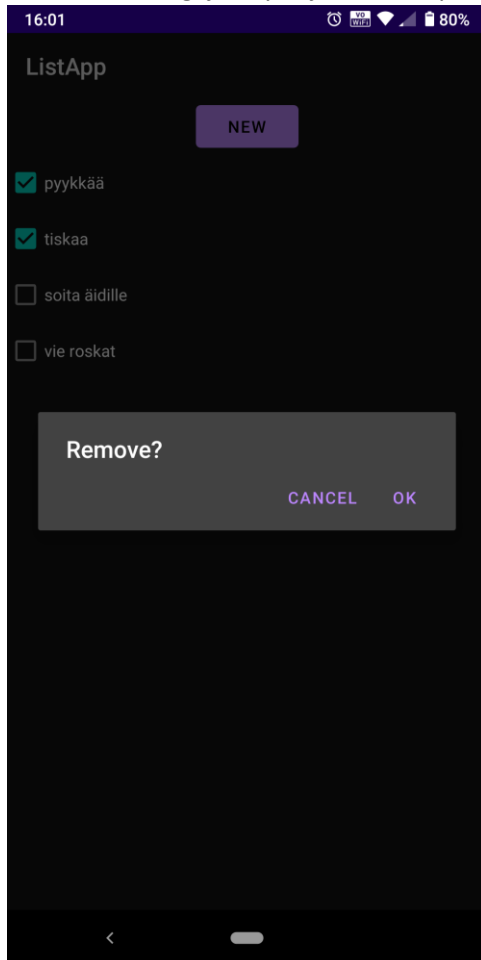
Käyttäjän painaessa jotakin päävalikossa näkyvistä listojen nimistä, ohjelma avaa käyttäjälle näkyviin tätä vastaavan käyttäjälistan. Käyttäjä voi merkata listan kohtia valintabokseilla ja ohjelman olisi tarkoitus tallentaa valinta.



Käyttäjä voi jälleen painiketta painamalla lisätä uuden kohdan listalle.



Jos päävalikossa tai käyttäjälistanäkymässä painaa jotakin listan kohtaa pitkään, avautuu poiston varmistusdialogi ja käyttäjä voi tällä poistaa listojen kohtia tai kokonaisia listoja.



Tunnetut ongelmat

En saanut UserListAdapter – luokassa olevia kuuntelijoita toimimaan muuten kuin lamdba – lausekkeilla. Näiden toiminta on riippuvainen holder.getadapterposition() - metodista, mistä nämä saavat valitun alkion sijainnin. Tälle varmasti olisi jokin yksinkertainen ratkaisu, jolla tämän voisi toteuttaa erillisenä kuuntelijana pitääkseen onBindViewHolder() - metodi siistimpänä, mutta en sitä ainakaan nyt löytänyt. En ole myöskään aivan varma mitä ajoin takaa staattisilla viittauksilla tiettyihin MainListFragment ja UserListFragment – olioihin, mitä voisi myöskin siistiä jatkokehityksessä pois. Eräs ominaisuus mitä en saanut myöskään toimimaan oli käyttäjälisterien valintaruutujen tekstien ylivaihtaminen kun valintaruutu merkataan, mutta se olisi varmaankin ensimmäinen varsinainen ominaisuus mitä lähtisin kehittämään.