PROJEKT

Testno voden razvoj pri predmetu ita

Aleks sagadin

2021

Kazalo

[1 VZORCI 4](#_Toc74605433)

[1.1 Uporaba vzorcev 4](#_Toc74605434)

[1.2 Vzorec ViewHolder Design Pattern 4](#_Toc74605435)

[1.3 Identity Field 4](#_Toc74605436)

[1.4 MVVM model 4](#_Toc74605437)

[1.5 Singleton 5](#_Toc74605438)

[1.6 Table data gateway 5](#_Toc74605439)

[1.7 Repository 6](#_Toc74605440)

[1.8 Table Module 6](#_Toc74605441)

[1.9 DAO pattern 6](#_Toc74605442)

[1.10 VZOREC LAZY LOAD 7](#_Toc74605443)

[2 FUKCIONALNOST: REGISTRACIJA Z SQLITE 7](#_Toc74605444)

[**2.1** **TEST st 1 za id v MainActivityu** 7](#_Toc74605445)

[2.1.1 Implementacija 7](#_Toc74605446)

[2.1.2 Preoblikovanje 7](#_Toc74605447)

[**2.2** **TEST st 2: za kreiranje baze** 8](#_Toc74605448)

[2.2.1 Implementacija 8](#_Toc74605449)

[2.2.2 Preoblikovanje 9](#_Toc74605450)

[2.2.3 Preoblikovanje 2.0 9](#_Toc74605451)

[**2.3** **TEST st 3 preverjnje če baza že obstaja** 9](#_Toc74605452)

[2.3.1 Implementacija 9](#_Toc74605453)

[2.3.2 Preoblikovanje 10](#_Toc74605454)

[**2.4** **TEST st 4 za insertiranje v DB** 10](#_Toc74605455)

[2.4.1 Implementacija 10](#_Toc74605456)

[2.4.2 Preoblikovanje 10](#_Toc74605457)

[**2.5** **TEST st 5 preverjanje imena v bazi** 11](#_Toc74605458)

[2.5.1 Implementacija 11](#_Toc74605459)

[2.5.2 Preoblikovanje 11](#_Toc74605460)

[**2.6** **TEST st 6** 12](#_Toc74605461)

[2.6.1 Implementacija 12](#_Toc74605462)

[2.6.2 Preoblikovanje 12](#_Toc74605463)

[3 FUNKCIONALNOST: PRIJAVA 13](#_Toc74605464)

[**3.1** **TEST st 7 za preverjanje če je prijava uspešna** 13](#_Toc74605465)

[3.1.1 Implementacija 13](#_Toc74605466)

[3.1.2 Preoblikovanje 13](#_Toc74605467)

[**3.2** **TEST st 8** 13](#_Toc74605468)

[3.2.1 Implementacija 14](#_Toc74605469)

[3.2.2 Preoblikovanje 14](#_Toc74605470)

[4 OSNOVNO OKNO PO PRIJAVI/REGISTRACIJI 14](#_Toc74605471)

[**4.1** **TEST st 9** 14](#_Toc74605472)

[4.1.1 Implementacija 15](#_Toc74605473)

[4.1.2 Preoblikovanje 15](#_Toc74605474)

[**4.2** **TEST st 10 preverjanje če geslo vsebuje številko** 16](#_Toc74605475)

[4.2.1 Implementacija 16](#_Toc74605476)

[4.2.2 Preoblikovanje 16](#_Toc74605477)

[4.3 TEST st 11 17](#_Toc74605478)

[4.3.1 Implementacija 17](#_Toc74605479)

[5 FUNKCIONALNOST : INTRO 18](#_Toc74605480)

[5.1 TEST st 12 18](#_Toc74605481)

[5.1.1 Preoblikovanje 18](#_Toc74605482)

[5.2 TEST st 13 18](#_Toc74605483)

[5.2.1 Preoblikovanje 18](#_Toc74605484)

[5.2.2 Implementacija 18](#_Toc74605485)

[5.3 TEST st 14 19](#_Toc74605486)

[5.3.1 Preoblikovanje 19](#_Toc74605487)

[5.4 Implementacija 19](#_Toc74605488)

[6 FUNKCIONALNOST: PRIKAZ MENI APP 19](#_Toc74605489)

[6.1 TEST st 15 20](#_Toc74605490)

[6.1.1 Preoblikovanje 20](#_Toc74605491)

[6.1.2 Implementacija 20](#_Toc74605492)

[6.2 TEST st 16 20](#_Toc74605493)

[6.2.1 Preoblikovanje 21](#_Toc74605494)

[6.2.2 Implementacija 21](#_Toc74605495)

[6.3 TEST st 17 21](#_Toc74605496)

[6.3.1 Preoblikovanje 21](#_Toc74605497)

[6.3.2 Implementacija 21](#_Toc74605498)

[7 FUNKCIONALNOST: PRIKAZ SEZNAMA 22](#_Toc74605499)

[7.1 TEST st 18 in 19 22](#_Toc74605500)

[7.1.1 Preoblikovanje 22](#_Toc74605501)

[7.1.2 Implementacija 22](#_Toc74605502)

[7.2 TEST 20 in 21 22](#_Toc74605503)

[7.2.1 Preoblikovanje 23](#_Toc74605504)

[7.2.2 Implementacija 23](#_Toc74605505)

[7.3 TEST st 22 23](#_Toc74605506)

[7.3.1 Preoblikovanje 24](#_Toc74605507)

[7.3.2 Implementacija 24](#_Toc74605508)

[7.4 TEST st 23 24](#_Toc74605509)

[7.4.1 Preoblikovanje 24](#_Toc74605510)

[7.5 TEST st 24 24](#_Toc74605511)

[7.5.1 Preoblikovanje 25](#_Toc74605512)

[7.5.2 Implementacija za test 23 in 24 25](#_Toc74605513)

[8 FUNKCIONALNOST: DODAJANJE V SEZNAM 25](#_Toc74605514)

[8.1 TEST st 25,26,27 25](#_Toc74605515)

[8.1.1 Preoblikovanje 26](#_Toc74605516)

[8.1.2 Implementacija 26](#_Toc74605517)

[8.2 TEST st 28 26](#_Toc74605518)

[8.2.1 Preoblikovanje 26](#_Toc74605519)

[8.2.2 Implementacija 27](#_Toc74605520)

[8.3 TEST st 29 27](#_Toc74605521)

[8.3.1 Preoblikovanje 27](#_Toc74605522)

[8.3.2 Implementacija 27](#_Toc74605523)

[8.4 TEST st 30 in 31 27](#_Toc74605524)

[8.4.1 Preoblikovanje 28](#_Toc74605525)

[8.4.2 Implementacija za test 30,31 28](#_Toc74605526)

[8.5 TEST st 32 28](#_Toc74605527)

[8.5.1 Preoblikovanje 29](#_Toc74605528)

[8.5.2 Implementacija 29](#_Toc74605529)

[9 FINAL ODDAJ ZADNJI TESTI 29](#_Toc74605530)

[9.1 TEST 33,34 29](#_Toc74605531)

[9.2 TEST 35 30](#_Toc74605532)

[9.3 TEST 36 30](#_Toc74605533)

[9.4 TEST 37 30](#_Toc74605534)

[9.5 TEST 38 in 39 30](#_Toc74605535)

[10 VIRI 30](#_Toc74605536)

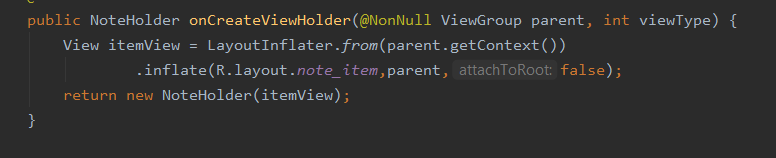
# VZORCI

## Uporaba vzorcev

Iz svoje seminarske sem uporbil MVVM vzorec, ki sem ga tam opisal.

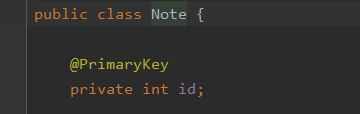
## Vzorec ViewHolder Design Pattern

Vzorec zasnove ViewHolder nam omogoča dostop do vsakega pogleda elementa seznama, ne da bi ga bilo treba iskati. S tem se izognemo večkratnemu klicanku findViewById().



## Identity Field

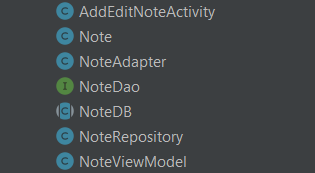
Shrani polje ID baze podatkov v predmet, da ohrani identiteto.



## MVVM model

Pri ustvarjanju recycleview-a smo uporabili mvvm arhitekturo.

Aplikacija je razdeljena na 3 dele to so:



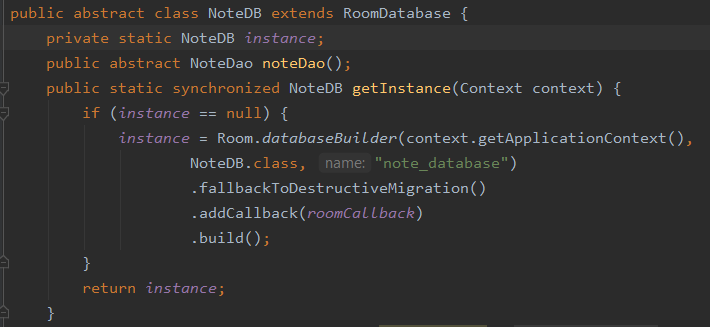
Model: v našem primeru je to Note -> Tu so shranjeni podatki aplikacije. Ne more neposredno govoriti s pogledom.

View : activityi, ki nimajo povezave direktno z modelom ampak imajo do viewModela

ViewModel: v našem primeru je to NoteViewModel -> Deluje kot vez med modelom in pogledom. Odgovorna je za preoblikovanje podatkov iz modela.

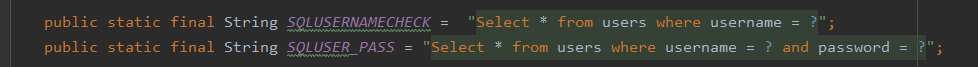
## Singleton

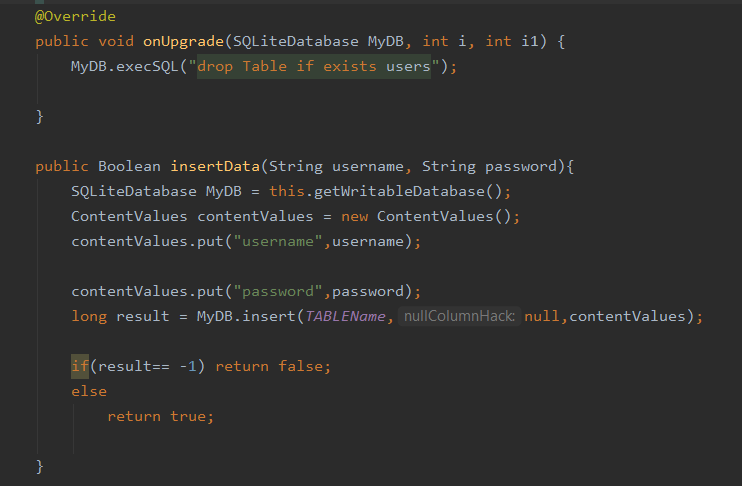
Nemoremo ustvariti vec instanc NoteDB-ja. S tem ne moremo naredit dveh instanc te tabele.



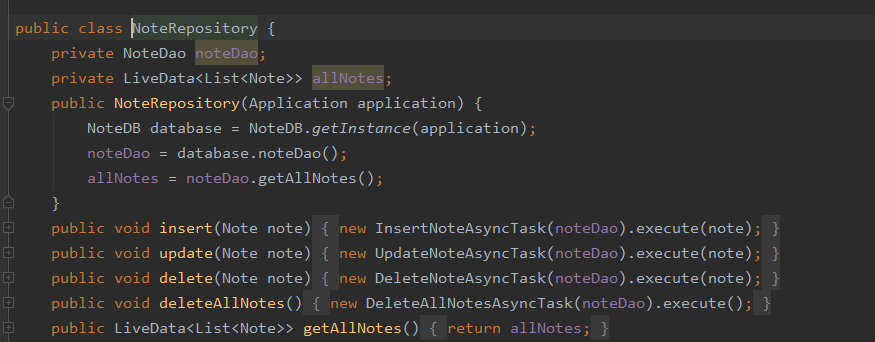
## Table data gateway

Uporabli smo vzorec table gate way spodaj na sliki je primer saj dostopamo do dolocene tabele kjer se isce prav tako pa imamo tudi insert in update.

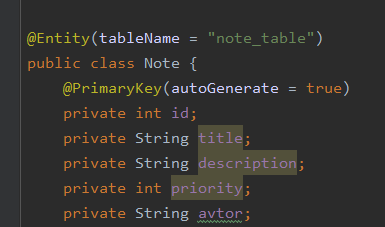




## Repository

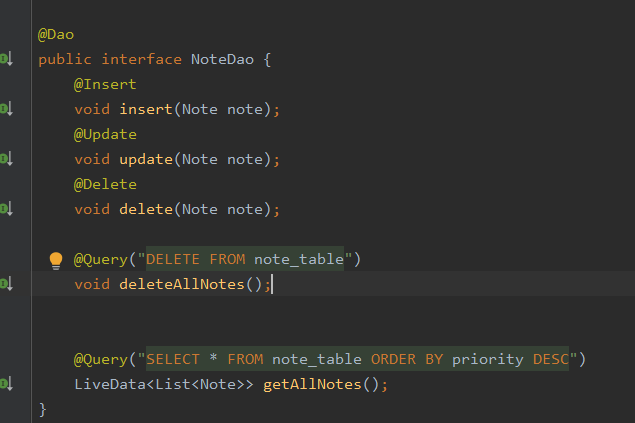


## Table Module



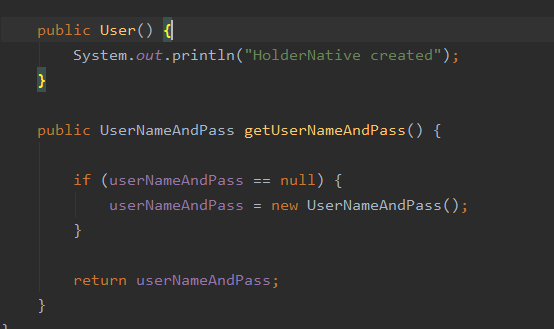
## DAO pattern

Predstavlja vmesnik do podatkovne baze v nasem primeru NoteDB.



## VZOREC LAZY LOAD

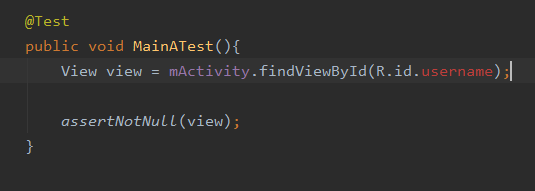
Dodal sem vzorec in ga vkljucil pri loginanju



# FUKCIONALNOST: REGISTRACIJA Z SQLITE

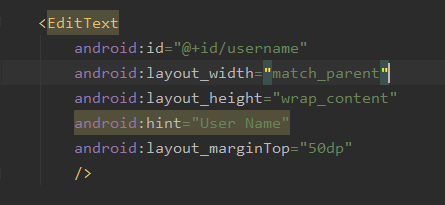
## **TEST st 1 za id v MainActivityu**

1. Naredil sem naredil test za sam id , ki preverja, če je že definirano v samem xml-ju . Ker seveda ni po test padel.



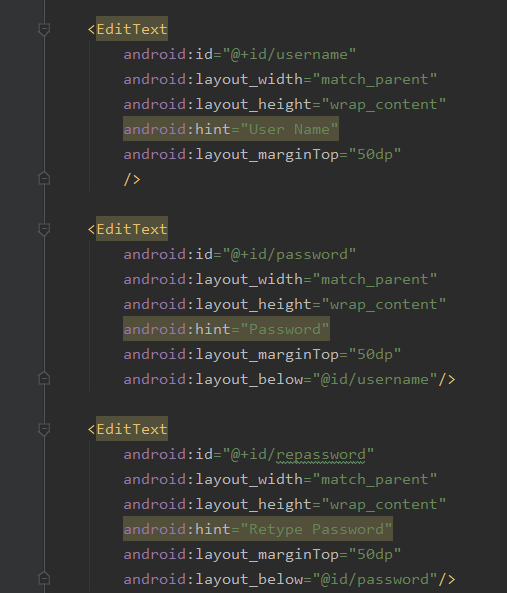
### Implementacija

1. V activity\_main.xml smo dodali dodatne



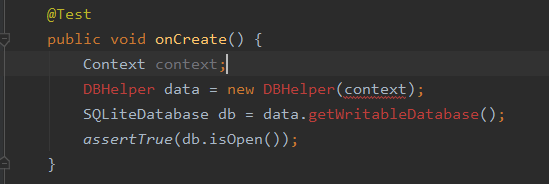
### Preoblikovanje

Dodal sem še za geslo in ponovitev gesla



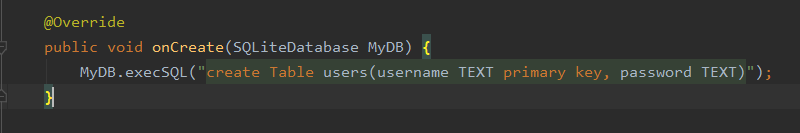
## **TEST st 2: za kreiranje baze**

Test za onCreate- Funkcijo v DBHelp.java. Test bo definitivno padel.

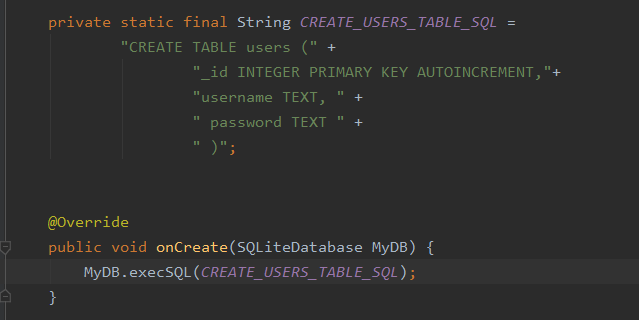


### Implementacija

Implementacija funkcije onCreate v DBhelper razredu.

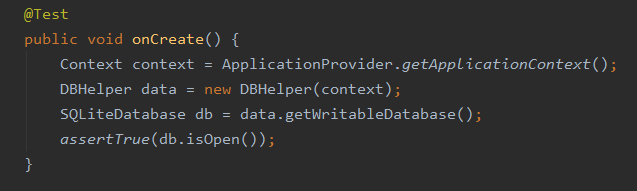


### Preoblikovanje

Preoblikoval sem tako da sem ustvaril spremenljivko za SQL stavek in nato to klical v sami funkciji.

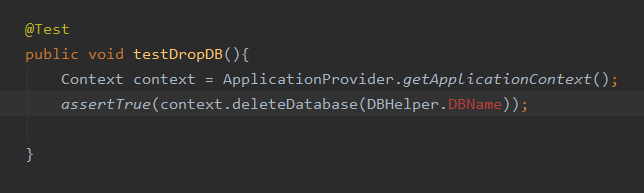
### Preoblikovanje 2.0

Sam test je bilo treba še popraviti ker je treba bazo zaprt in context nastavi vrednost.



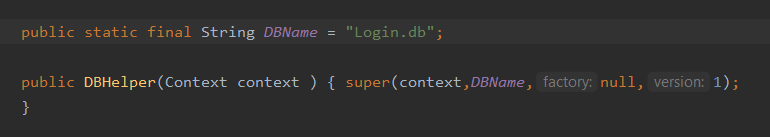
## **TEST st 3 preverjnje če baza že obstaja**

Test pade ker še sama baza ni bila izdelana.



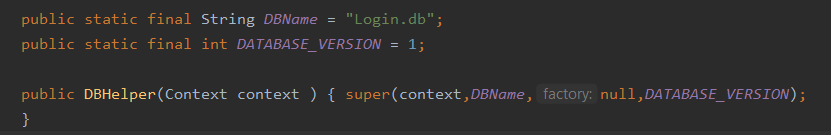
### Implementacija

Implementacija v DBhelperi same metode in spremenljivke.



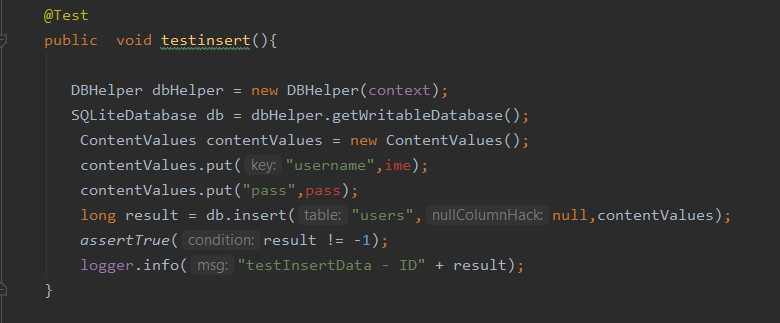
### Preoblikovanje

Dodal novo spremenljivko za verzijo in s tem naredil berljivejšo



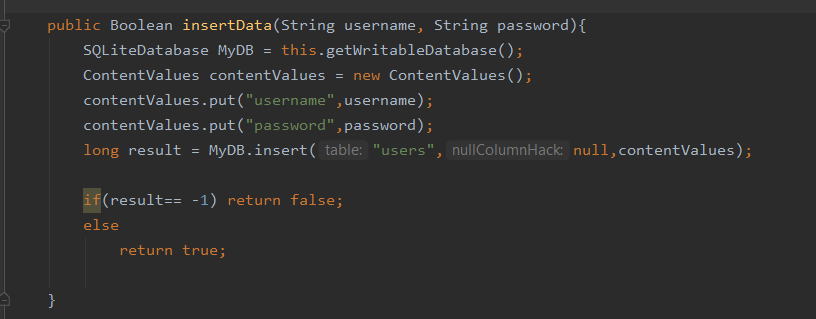
## **TEST st 4 za insertiranje v DB**

Test za insertiranje v podatkovno bazo. Test bo padel ker še nismo kreirali samih vrednosti, ter za to ker še nimamo ustvarjenih atributov(username,password) za bazo.



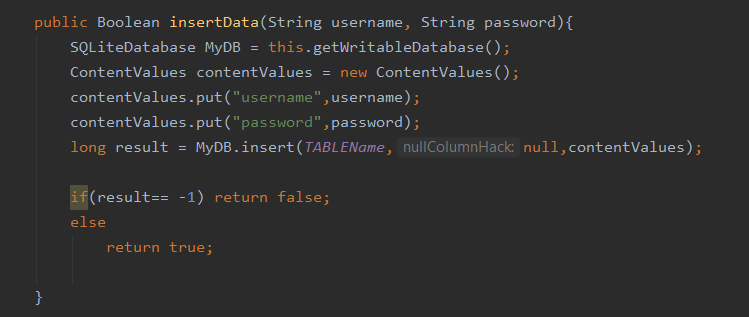
### Implementacija

Implementacija metode v DBhelper.java.



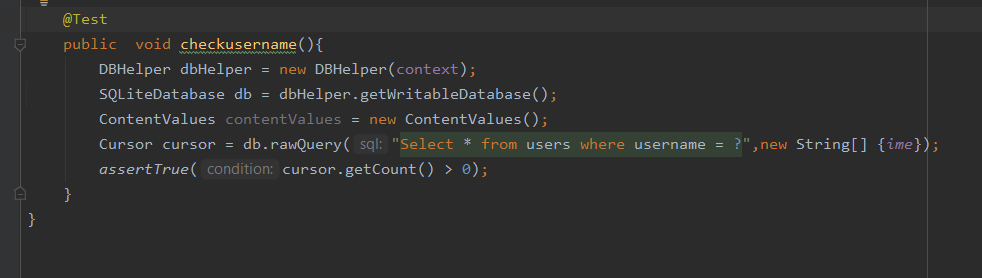
### Preoblikovanje

Za ime tabele »users« sem ustvaril spremenljivko in jo rezerviral na vrhu.



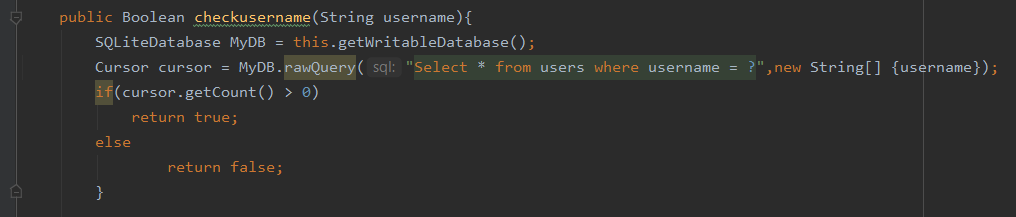
## **TEST st 5 preverjanje imena v bazi**

Test bi moral vrnit true če je ime že v bazi. Test bo pa padel zato ker še nimamo vnosa v bazo.



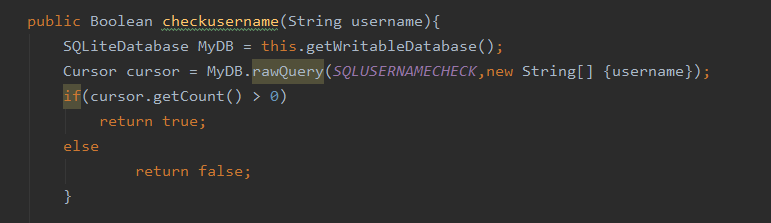
### Implementacija

Implementiral sem metodo, katera bo preverjala, če je že v bazi.



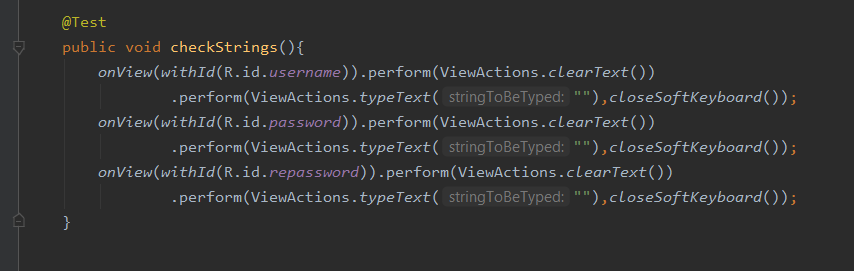
### Preoblikovanje

Za celotni SQL stavek sem ustvaril globalno spremenljivko.



## **TEST st 6**

Test, ki bo preveri če so stringi prazni v input pri registraciji



### Implementacija



### Preoblikovanje

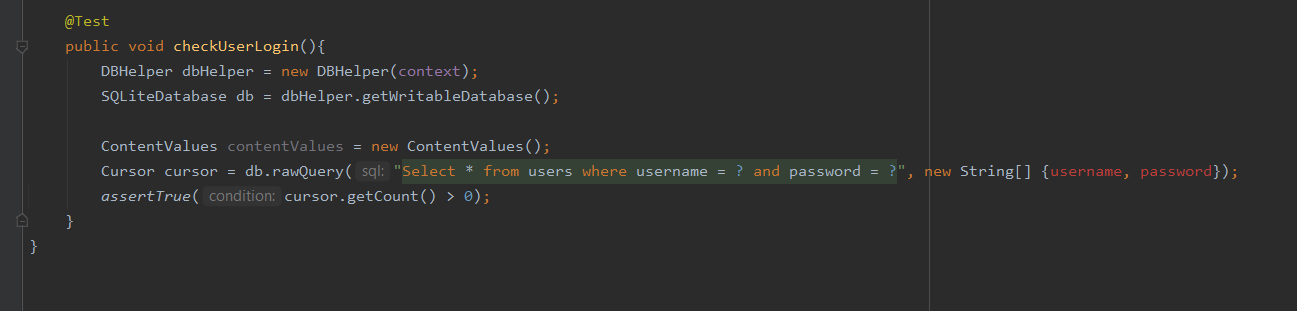
Dodal sem še dodaten izgled vsem elementom.



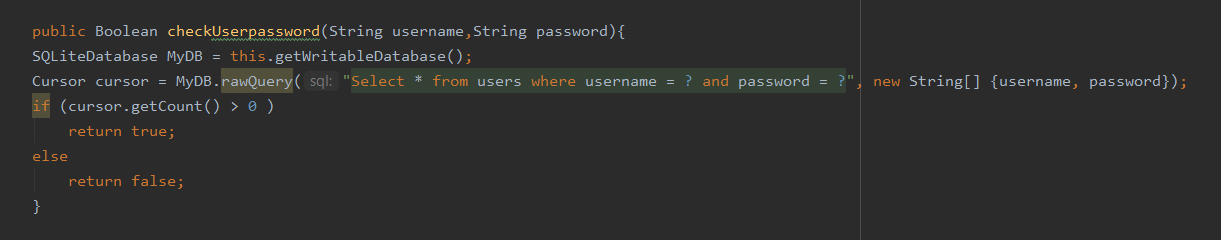
# FUNKCIONALNOST: PRIJAVA

## **TEST st 7 za preverjanje če je prijava uspešna**

Pri tem testu se morata ujemati geslo in ime uporabnika ob prijavi. Test bo padel ker še »ime« in »geslo nista deklarirana oz. podana.

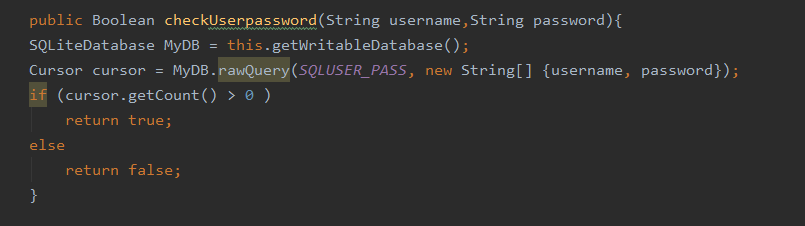


### Implementacija



### Preoblikovanje

Preoblikoval sem ponovno tako da sem spremenil SQL v novo statično spremenljivko in jo tukaj le vstavil(SQLUSER\_PASS)

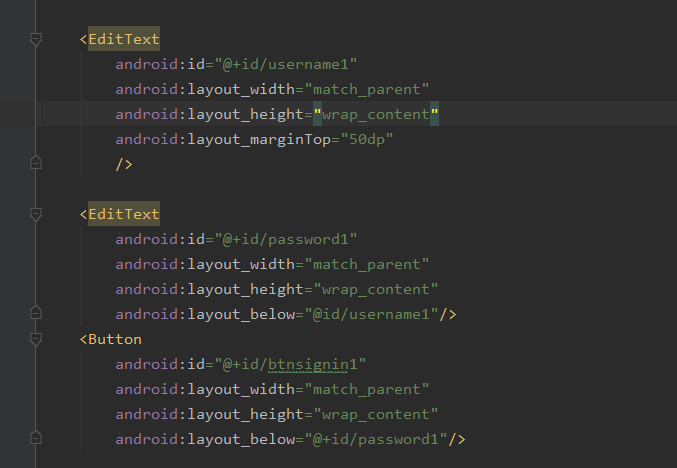


## **TEST st 8**

Preveri če sta vnosa prazna za ime in geslo.



### Implementacija



### Preoblikovanje

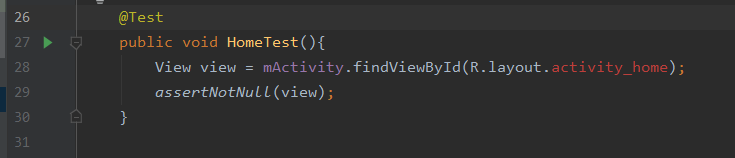
Za obliko sem preoblikoval kodo da je boljsi design.



# OSNOVNO OKNO PO PRIJAVI/REGISTRACIJI

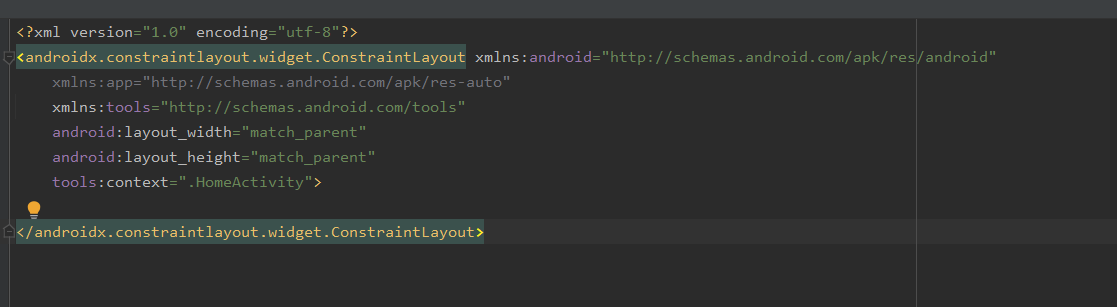
## **TEST st 9**

Izvedba test za pregled layouta. Test bo padel ker še .xml datoteka ni bila ustvarjena.



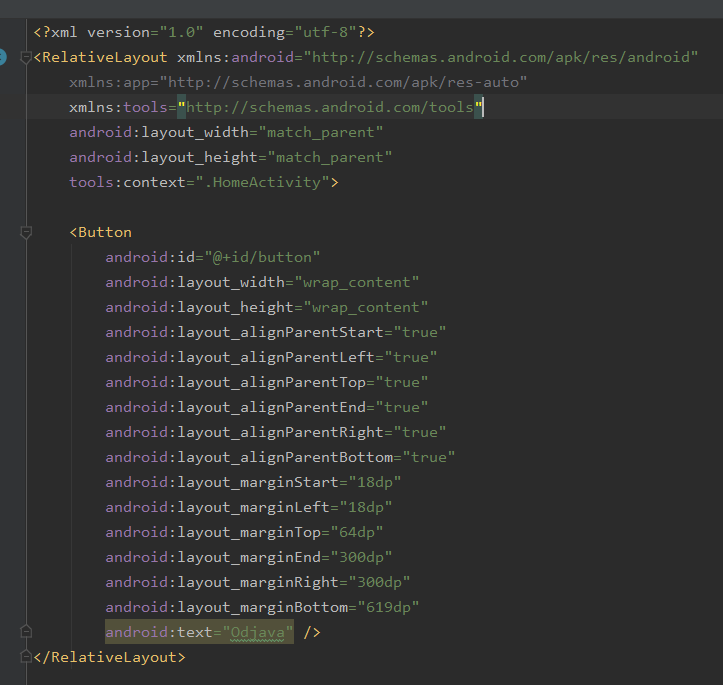
### Implementacija

Minimalna koda ki se generira ob ustvarjanju xml datoteke.



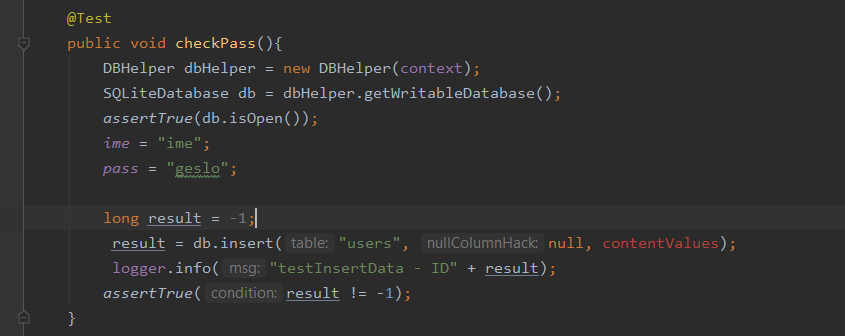
### Preoblikovanje

Kodo sem preoblikoval tako, da sem zamenjal za RelativeLayout in dodal gumb za odjavo, ki te bo locirala nazaj na prijavno okno, ki pa še trenutno ne dela.



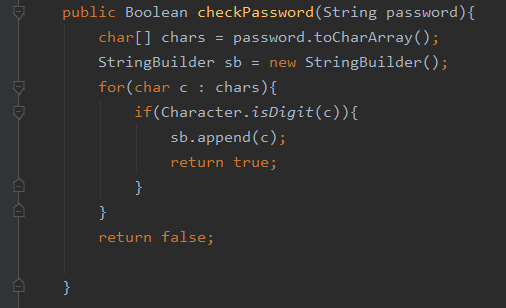
## **TEST st 10 preverjanje če geslo vsebuje številko**

Test bo padel ker še ni ustvarjeni contentValuesov, prav tako še nikjer ne preverja če je stevilka ali ne v geslu.



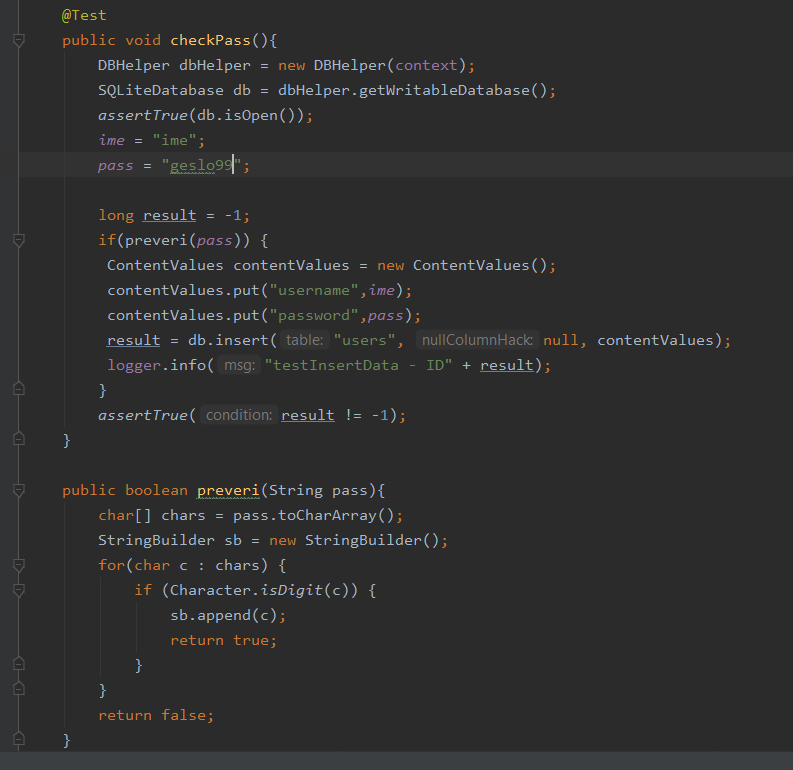
### Implementacija

Implementacija same metode



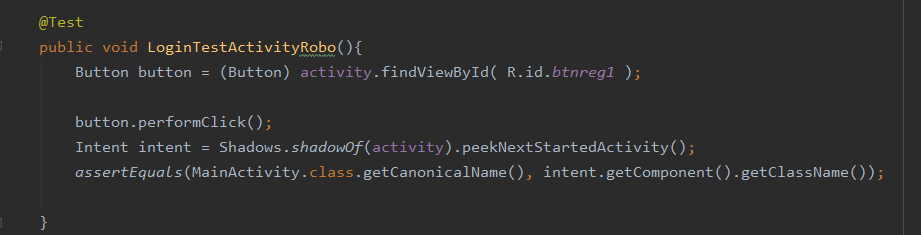
### Preoblikovanje

Z zgornjo funkcijo sem preveril pred insertom. Preoblikoval sem sam test da je preglednejši in delujoč s pomočjo funkcije:

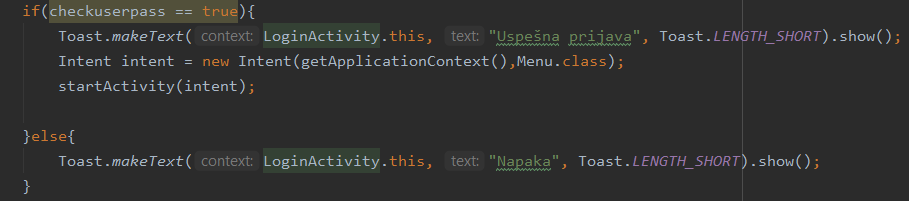


## TEST st 11

Test preverja delovanje Intent s strani Login.

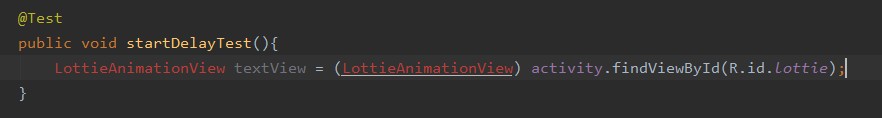


### Implementacija

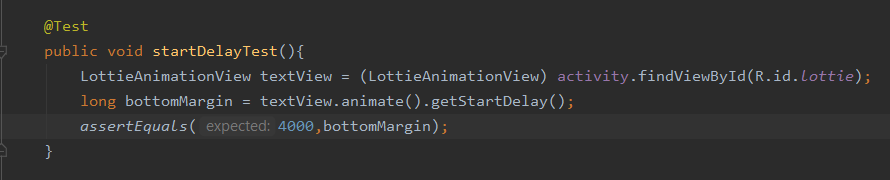


# FUNKCIONALNOST : INTRO

## TEST st 12

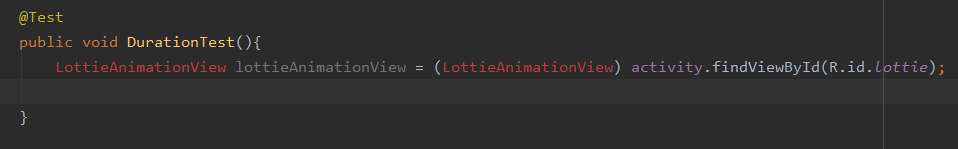
Test preverja če je StartDelay na intro activityu nastavljen na doloceno vrednost. 

### Preoblikovanje

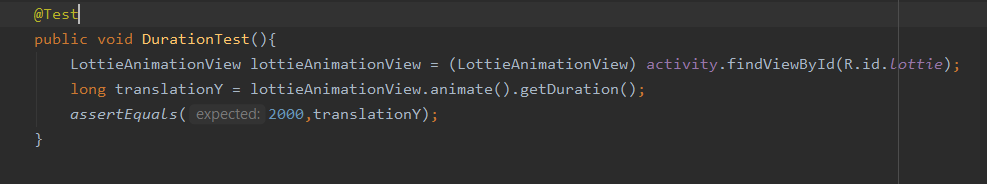


## TEST st 13

Test preverja če je getDuration na intro activityu nastavljen na doloceno vrednost.

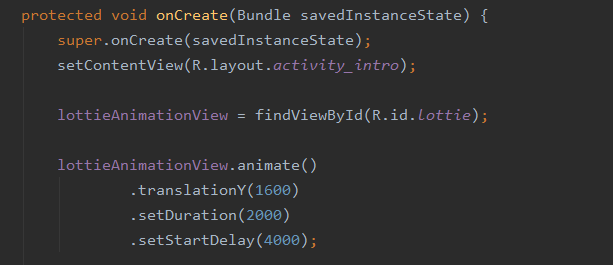


### Preoblikovanje



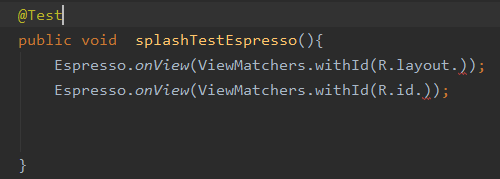
### Implementacija

V tej implementacij je prikaz za test 5.1 in 5.2.

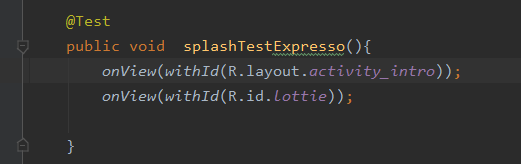


## TEST st 14

Test preverja če imamo layout in id dejansko deklariran.test pade ker še ne vemo kaksen id/layout bomo imeli.

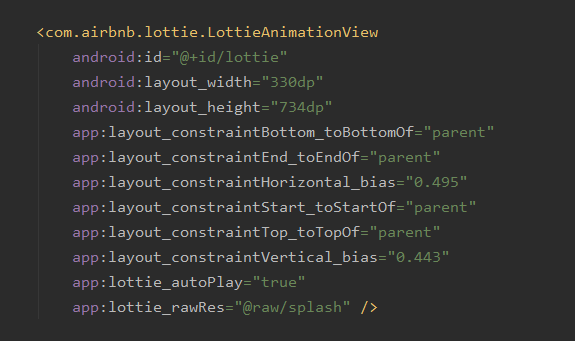


### Preoblikovanje



## Implementacija

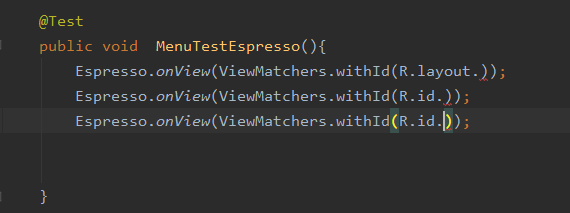
Implementacija samega activity\_intro.xml in v njem uporaba lottie



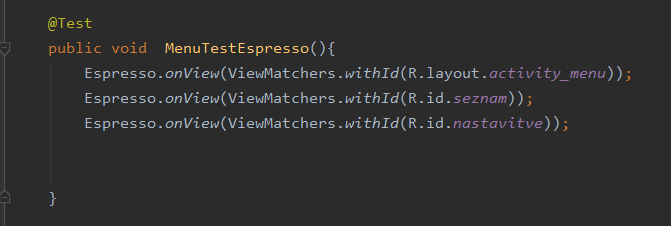
# FUNKCIONALNOST: PRIKAZ MENI APP

Prikaz glavnega menija v apliakciji

## TEST st 15



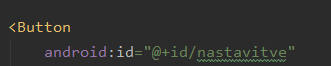
### Preoblikovanje



### Implementacija

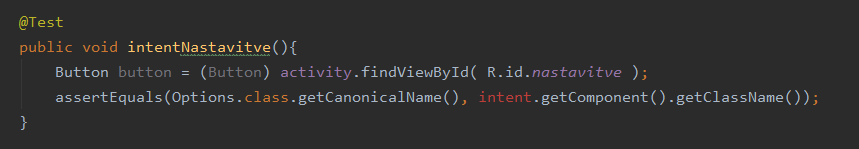
Implementacija v .xml datoteki kot layout je celotna datoteka poimenovana activity\_menu.xml.



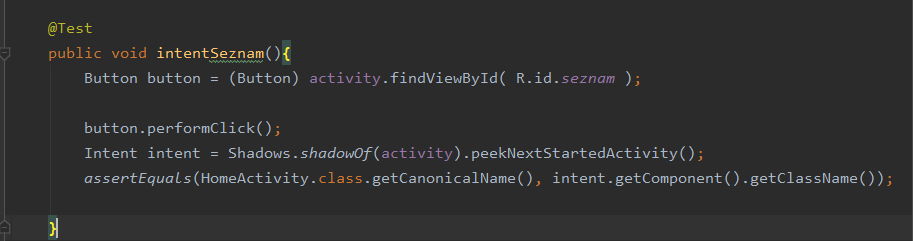


## TEST st 16

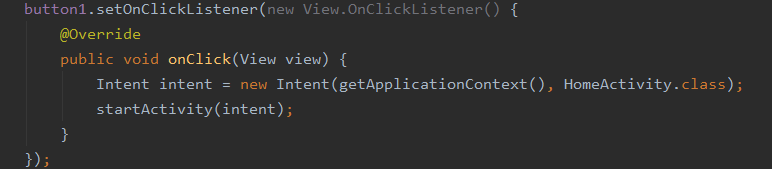
Test preverja če deluje prenos v nov activity HomeActivity.java. Test ne bo delal ker še intent ni definiran.



### Preoblikovanje

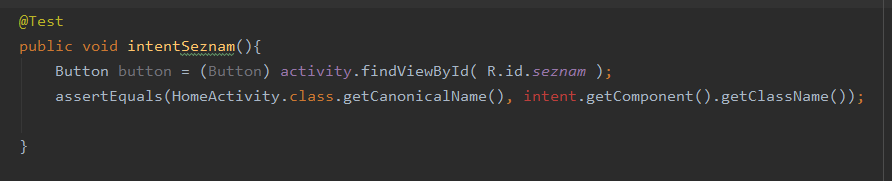


### Implementacija

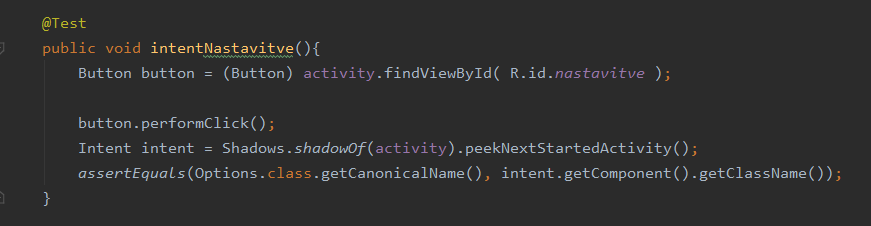


## TEST st 17

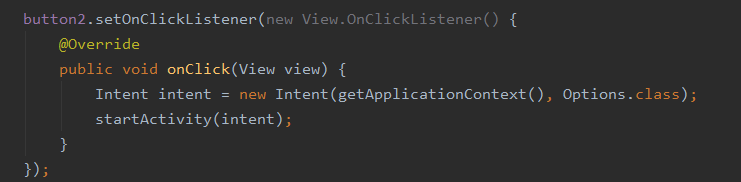
Test preverja če deluje prenos v nov activity Options.java. Test ne bo delal ker še intent ni definiran.



### Preoblikovanje



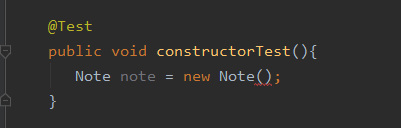
### Implementacija



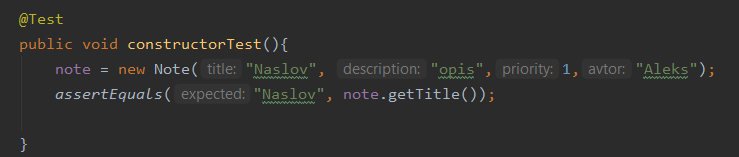
# FUNKCIONALNOST: PRIKAZ SEZNAMA

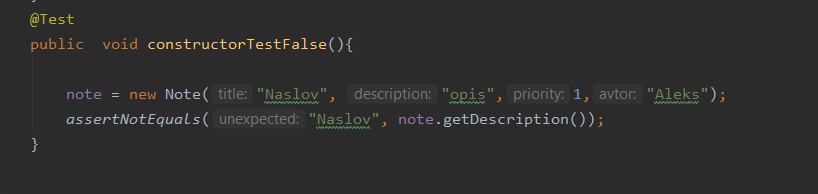
## TEST st 18 in 19

Preverimo če deluje konstruktor za seznam note ki bo uporabljen v recycleview.

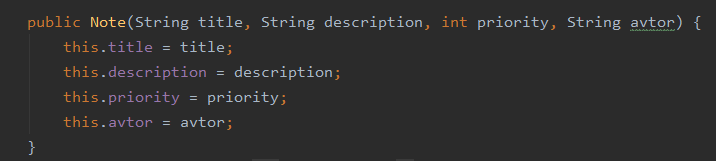


### Preoblikovanje





### Implementacija

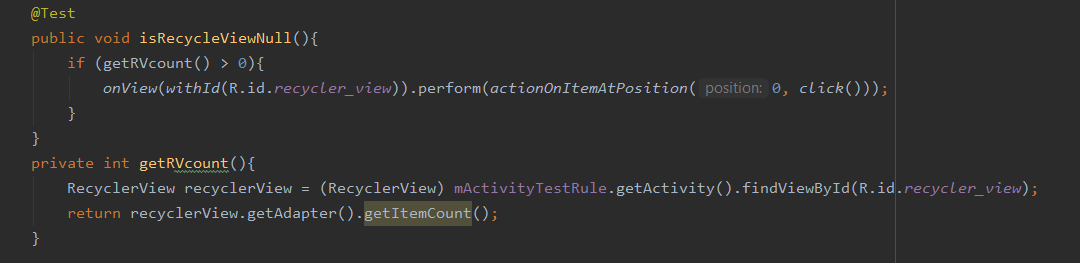


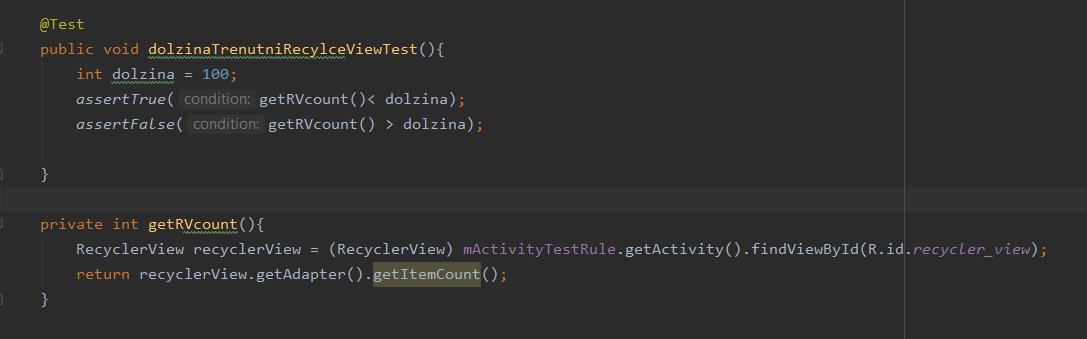
## TEST 20 in 21

Test preverjata če je RecycleView= Null ali kolikšna je dolžina recycleViewa .Test pade ker še ni implementirane funkcije.



### Preoblikovanje



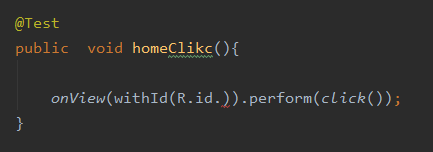


### Implementacija

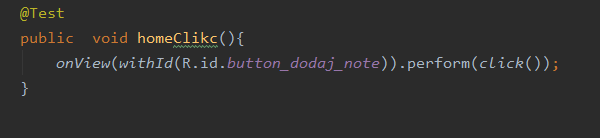
Tu smo implementirali celotni razred NoteAdapter.java, kjer se dodaja v list.

## TEST st 22

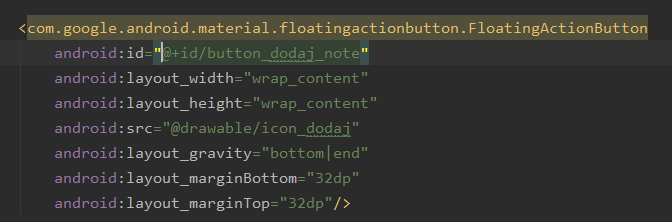
Test preverja če je možnost klika na button\_dodaj\_note, kjer pa bo padel saj še nimamo napisanega .xml



### Preoblikovanje

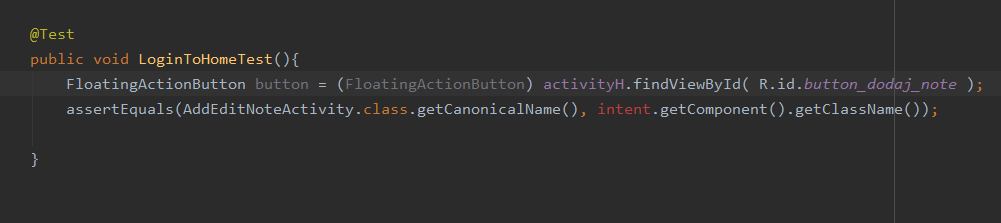


### Implementacija

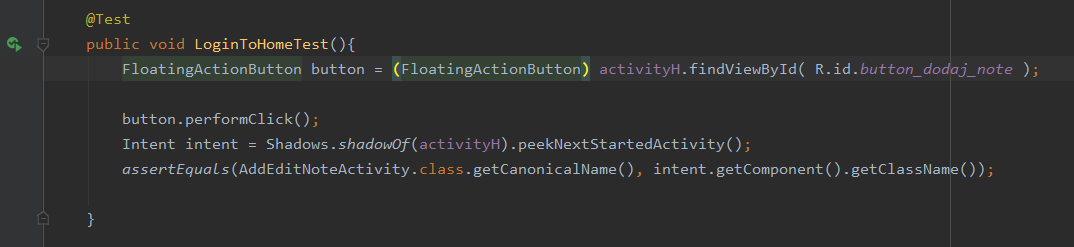


## TEST st 23

Test preverja če nas floatingbutton prenese na nov activity.Test bo padel ker še intent ustvarjen.

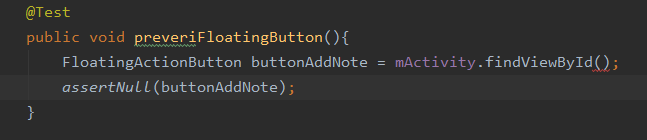


### Preoblikovanje

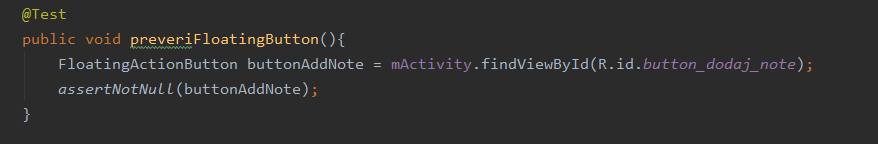


## TEST st 24

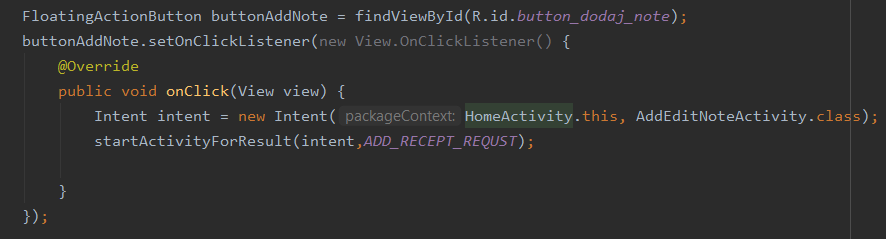
Test preverja če floatingbutton ni null pred zacetkom.



### Preoblikovanje



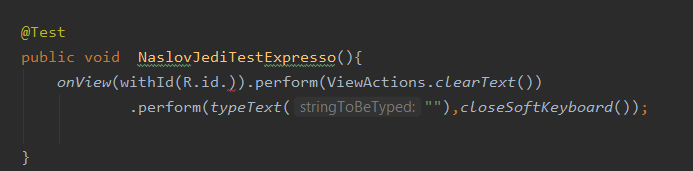
### Implementacija za test 23 in 24



# FUNKCIONALNOST: DODAJANJE V SEZNAM

## TEST st 25,26,27

Testi preverjajo če so stringi prazni pri posameznem input oknu. Testi niso delali ker še id ni bil ustvarjen.



### Preoblikovanje



### Implementacija

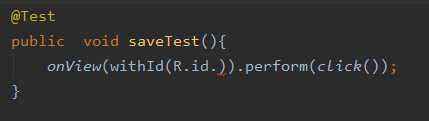




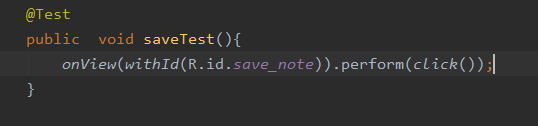


## TEST st 28

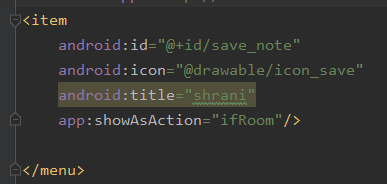
Test preverja če imamo save\_note id in le deluje na njem click saj je button.



### Preoblikovanje

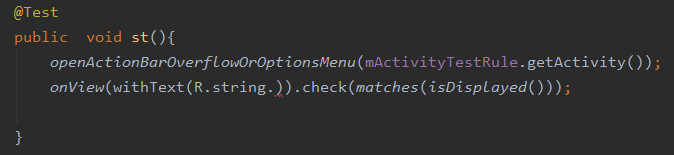


### Implementacija

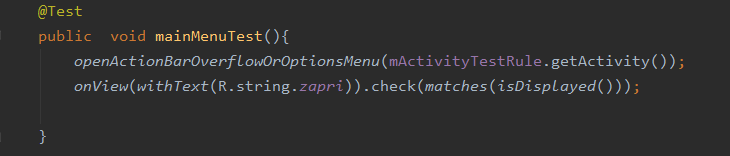


## TEST st 29

Test preverja če se odpre pravi string. Test bo padel ker še nismo imeli pravega stringa.

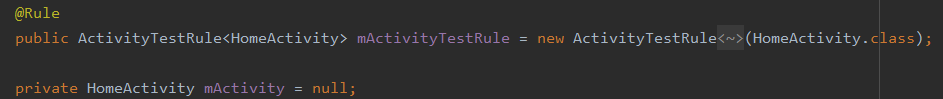


### Preoblikovanje



### Implementacija

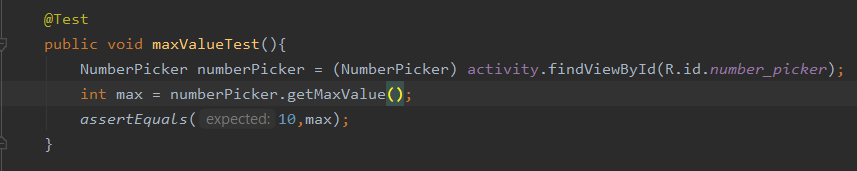
Pri implementaciji je bilo seveda potrebno ustvariti tudi razred v katerem se preverja to.

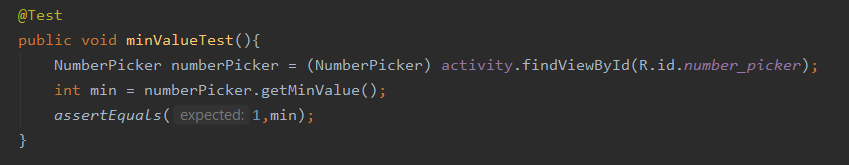




## TEST st 30 in 31

S tem tem testom preverjamo ali je max number pravilno dolocen. Pri drugem pa če je min pravilno nastavljena.

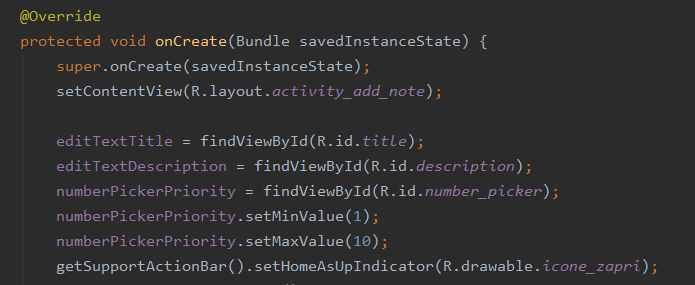




### Preoblikovanje

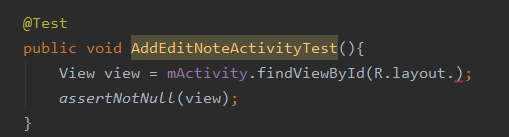


### Implementacija za test 30,31

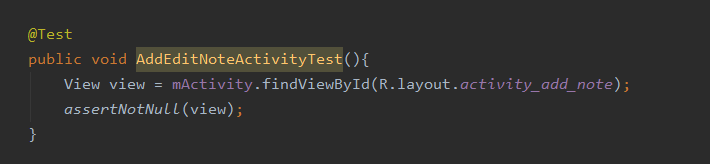


## TEST st 32

Test preverja če layout ni enako null.



### Preoblikovanje



### Implementacija

Ustvarjenje layouta

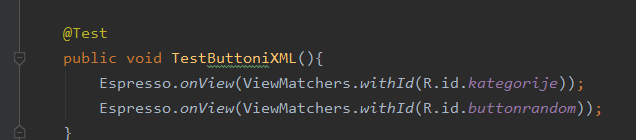


# FINAL ODDAJ ZADNJI TESTI

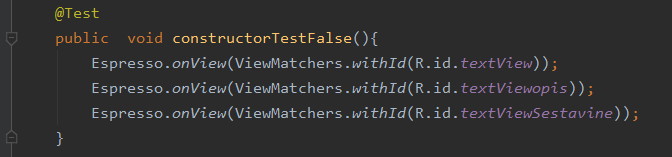
## TEST 33,34



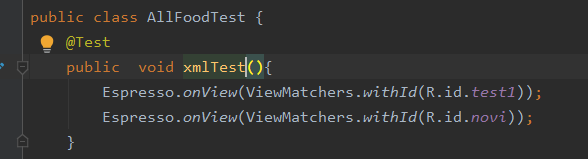
## TEST 35



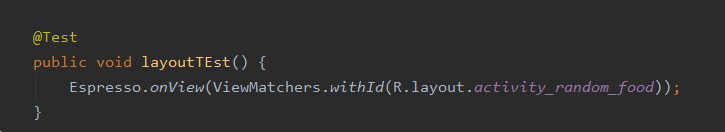
## TEST 36

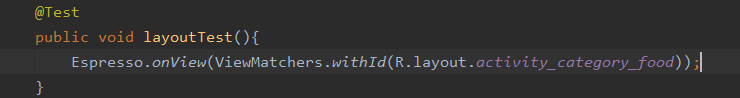


## TEST 37



## TEST 38 in 39





# VIRI

<https://www.javacodegeeks.com/2013/09/android-viewholder-pattern-example.html>

<https://www.geeksforgeeks.org/data-access-object-pattern/>

<https://www.vogella.com/tutorials/Robolectric/article.html>

<http://blogs.quovantis.com/how-to-use-espresso-for-android-ui-testing/>

<https://www.tutorialspoint.com/espresso_testing/espresso_testing_tutorial.pdf>

<https://lottiefiles.com/search?q=food&category=animations&animations-page=1>

<https://www.vogella.com/tutorials/Robolectric/article.html>

<https://square.github.io/retrofit/>

<https://riptutorial.com/design-patterns/example/30584/java-lazy-loading>