# I.1 - SharedPreferences

## Scopo delle SharedPreferences

Le SharedPreferences sono un modo semplice per memorizzare dei dati su disco che non richiedano particolari strutture o coordinazione, fornendo al contempo dei metodi semplici di lettura e scrittura. Sono una lista di elementi chiave-valore, i cui valori possono essere solo delle primitive o delle stringhe.

Sono solitamente utilizzate per memorizzare dati di setup dell’app che devono permanere tra una sessione e l’altra, o semplici dati che non richiedono un database.

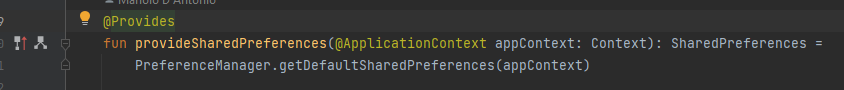
## PreferenceManager

Per recuperare una istanza di SharedPreferences, utilizzeremo PreferenceManager.

Tramite questa classe è possibile creare, recuperare e cancellare istanze di SharedPreferences.

Tra queste, è presente un’istanza di default, che andiamo ad usare

Aggiungi a DataModule:



* PreferenceManager richiede sempre un context per creare o recuperare istanze.
* Hilt ci fornisce due context, Application e Activity, tramite annotations.

MovieListViewModel:

A screen shot of a computer program

Description automatically generated with low confidence

## Salvare e Recuperare valori

Puoi utilizzare il builder di SharedPreferences per salvare dati su disco, in modo che i preferiti permangano tra le sessioni.

A screen shot of a computer code

Description automatically generated with low confidence

* .edit() comincia la modifica, .apply() la applica
* È necessario utilizzare i metodi in base al tipo di dato che si vuole salvare (Boolean, stringa..)
* I metodi richiedono una stringa (key) e un valore (value)
* Creano un record, o lo aggiornano se già presente.

Recuperare i valori è simile:

A screen shot of a computer

Description automatically generated with low confidence

* C’è un metodo per ogni primitiva (getBoolean, getInteger…)
* I metodi richiedono la chiave e un default in caso il valore non venisse trovato.

## Unit Test

Nei test le preferences non vanno testate, ma ci sono molte interazioni con le nostre classi.

Negli Instrumented test non c’è problema, dato che abbiamo a disposizione il contesto reale dell’app, e Hilt effettua l’iniezione dove necessario.

Negli UnitTest invece, creiamo le preferences in questo modo:

A screen shot of a computer

Description automatically generated with medium confidence

* ApplicationProvider.getApplicationContext() fornisce il contesto reale del nostro test. Questa funzione è **utilizzabile solo dentro @Before**

Nei test è importante ricordare di non testare la funzionalità di SharedPreferences, ma quella della nostra funzione.

A screen shot of a computer code

Description automatically generated with low confidence

Sebbene indirettamente, abbiamo testato il corretto comportamento di changeMovieFavourite.

## Esercizio

L’app deve memorizzare i preferiti, che devono permanere tra sessioni.

Salva i film preferiti dell’utente, al momento del click della stella

Quando carichi getMovies, cerca se nelle preferences il film è un preferito, e aggiorna il dataset prima di pubblicarlo con livedata.

* Modificare un elemento di una lista in livedata NON genera una notifica da livedata. In questo caso, si modifica la lista, e poi si re-imposta la lista come contenuto del livedata, con setValue o postValue.