

MVVM APP

MVVM è un pattern molto utilizzato tra le applicazioni per via della sua organizzazione.

1 Creazione Classe Obj

2 Creazione DAO Obj

Nel DAO vanno posti i metodi usati per "usare" il Database. Va annotato come `@Dao`.
È importante usare `LiveData<List>>` per il pattern MVVM. `LiveData` può essere osservato, in modo che quando i dati cambiano, la view viene notificata.

Creazione della classe Database

Questa classe è un **Singleton**.

Creazione della classe Repository

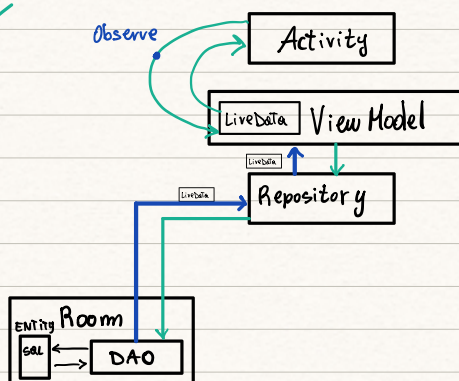
Questa classe ci permette di comunicare con il database; per questo motivo ha delle classi `AsyncTask` che ci permettono di contattare il DB su thread separati. La Repository non è parte del pattern, ma è considerata una **Best Practice** perché fornisce un layer di astrazione al di sopra di **diversi tipi di data source**.

Creazione della ViewModel

Questa classe ha come scopo quello di salvare dei dati per l'interfaccia utente e comunicare con il Model.

Prende i dati della Repository in modo che una Activity possa rappresentarle sullo schermo.

Mettere i nostri dati nella ViewModel invece dell'UI Controller, è utile perché le ViewModel non fanno perdere i dati quando la View viene ricreata. Quando la Activity viene ricreata, semplicemente richiede la stessa istanza della ViewModel che usava prima di essere distrutta.



Reference della ViewModel nell'Activity



Creazione Adapter per
la RecyclerView