**Fragments**

Un fragment è un controller che si sostituisce ad un activity nell’eseguire determinati task.

Il task principale per un fragment è quello di gestire la UI

* I fragments (introdotti nelle API 11) erano inizialmente disponibili in due implementazioni
  + L'implementazione **nativa** è già compilata all'interno di ogni device; ogni device può dunque contenere una versione leggermente differente dei fragments
  + L'implementazione **support** è una libreria che viene inclusa nella propria applicazione; in questo modo ogni device che esegue l'applicazione usa la stessa implementazione dei fragments.
* Queste due implementazioni sono deprecate da Android 9.0 (API 28), e sono

state sostituite dalla versione inclusa in Jetpack (androidx)

* + A livello di codice non cambia praticamente nulla, cambiano gli import!

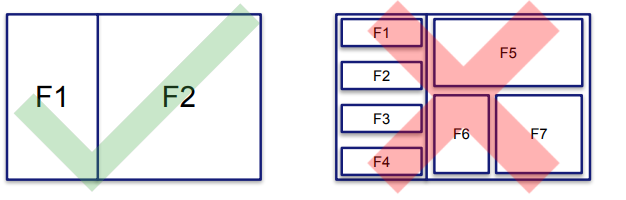
Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Rule of thumb: un'applicazione non deve mai mostrare più di 2-3 fragment allo stesso tempo

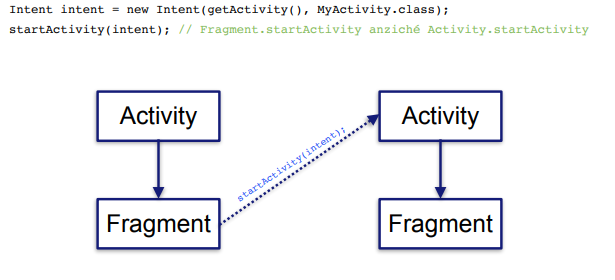


Ci sono però diverse API in Android che si basano sui fragments (ad esempio ViewPager**), se si fa uso di queste API si deve dunque fare uso anche dei fragments.**

**Rule of thumb**: utilizzare sempre i fragments, a meno che si stia sviluppando un'applicazione piccola, che siamo sicuri non verrà ampliata.

**Fragment e intent**

Per lanciare un activity da un fragment si utilizza un intent



Ogni istanza di un fragment può contenere un oggetto di tipo Bundle.

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

**RecyclerView**

È utilizzato per gestire la visualizzazione di una lista.

L’unica operazione che fa il recycler view è quello di riciclare le view e metterle in lista.

Mantiene delle view fisse, ma cambia il contenuto.

**ViewHolder e Adapter**

Il lavoro del viewholder è quello di tenere una view

L’adapter crea viewholders necessare e di fare il bind dei viewholders ai dati.

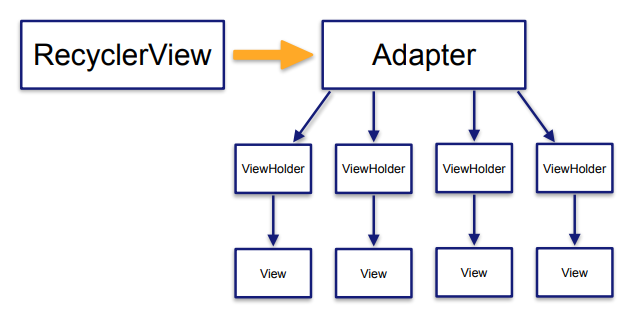


Immagine che contiene testo

Descrizione generata automaticamente

**ViewPager**

Il ViewPager permette all'utente di navigare attraverso delle view facendo lo "swipe" con il dito (a destra o a sinistra)

Il ViewPager si comporta in modo "simile" al RecyclerView, richiede cioè anch'esso un PagerAdapter

Lo sviluppatore deve prendersi carico di implementare due semplici metodi: getCount() e getItem(int)

**Localizzazione GPS**

Fused localtion provider, utilizza gps, internet, reti telefoniche per ottenere la posizione

Blocca automaticamente l’eccesso delle richieste per evitare sovraccarichi della batteria ecc.

Richiede il permesso di accedere al gps nel manifest.

È possibile scegliere tra ACCESS\_COARSE\_LOCATION (la location viene ritornata con una "bassa" precisione) e ACCESS\_FINE\_LOCATION (la location viene ritornata con una precisione più alta).

Le API native sono complesse e cambiano spesso, evitare, usare dipendenza esterna, Smart Location Libray

Nel file build.gradle aggiungere la dipendenza e fare sync

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'com.android.support:appcompat-v7:28.0.0'

implementation 'com.android.support.constraint:constraint-layout:1.1.3'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test:runner:1.0.2'

androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'

implement 'io.nlopez.smartlocation:library:3.3.3'

}

La Smart Location Library è estremamente semplice da usare!!

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Indicare le permission per accedere alla posizione all'interno del file AndroidManifest.xml non basta

A partire da Android 6.0, per alcune permission (quelle marcate come dangerous) è richiesta una conferma a runtime.

Se non viene implementato nessun meccanismo di richiesta delle permission, l'utente deve abilitare manualmente una determinata permission nei settings del device.

La risposta data dall'utente alla richiesta di un permesso viene intercettata nel metodo onRequestPermissionsResult

Immagine che contiene testo, screenshot, interni

Descrizione generata automaticamente