

# Esercizi 1

Layout, Risorse e Activities

# 1. Esercizio sul concetto di Risorsa e Manager

Partendo da una basic Activity fare le seguenti operazioni:

1. Creare un qualifier riferito alle impostazioni di Locale per le risorse “string”.
2. Creare una risorsa di tipo string sia nel XML di default che in quello appena creato
3. Da emulatore, verificare che cambiando il Locale (Language) e ricaricando la app, tale valore si mostrerà come campo text di una TextView.
  
4. Ottenere un riferimento ai seguenti Manager:

```
ActivityManager am = (ActivityManager) (istanza di activity).getSystemService(ACTIVITY_SERVICE);  
PackageManager pm = (inst. activity).getPackageManager();  
WindowManager wm = (inst. activity).getWindowManager();  
ConnectivityManager cm = (ConnectivityManager) (inst. activity).getSystemService(CONNECTIVITY_SERVICE);
```

Aiutandovi con la documentazione online o facendovi suggerire da Android Studio, cercate un metodo riferito ad almeno un Manager, in grado di sollevare un’eccezione riguardante la sicurezza/permessi.

## 2. Ciclo di Vita di una App

Partendo da una basic Activity, eseguire i seguenti passaggi relativi al ciclo di vita di una app:

1. Implementare tutti i metodi notevoli visti a lezione. Usando Toast e Log cercate di capire quando vengono chiamati. Date fastidio al vostro dispositivo (e.g. Togliete/Ripristinate la vs. activity dallo stack, capovolgere lo schermo tante volte etc...).
2. La vostra Activity ha un oggetto di tipo Bundle come variabile-attributo privato. **Nient'altro.**
3. In "onCreate" tale Bundle viene sovrascritto con il bundle argomento di onCreate. Viene inizializzato **una volta sola**, con una sola chiave <int> chiamata "Conta". Conta è qui inizializzato ad uno.
4. Il valore di quel Bundle viene incrementato ogni volta che viene chiamato "onStart". Il suo valore viene mostrato sullo schermo impostando il campo "text" di una TextView

Suggerimento: Salvare il valore del Bundle in "onSaveInstanceState"

# 3. Ciclo di Vita di una App e Callback aggiuntive

Partendo da una basic Activity, eseguire i seguenti passaggi relativi al ciclo di vita di una app:

1. Creare un altro file Java che estende Activity/AppCompatActivity. Solo il file Java, **non un'intera activity**. Modificate gli opportuni file .xml per fare in modo di associare il layout che già avete con questa nuova classe Java.
2. Questa nuova activity **non ha alcun attributo**.
3. Si faccia override del metodo “onPostCreate(Bundle b)” e si verifichi quando viene chiamato.
4. Creare la classe FormattaData. Vedi class diagram in figura.
5. Stabilire come creare un'istanza di FormattaData e metterla nel bundle di onSavedInstanceState.
6. L'istanza così salvata serve a scrivere in una TextView il Timestamp Date dell'ultima volta che l'app è passata per onPostCreate.

Suggerimento:

```
DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");  
String dataStringa = dateFormat.format(new Date());
```

Class : FormattaData
-dateFormat : SimpleDateFormat
+FormattaData() : <<constr>> +FormattaData(String) : <<constr>> +getData() : String

## 4. Esercizio con i Layout

1. Prendere confidenza con l'IDE per la generazione di file XML con i diversi layout.
2. Impostare come layout di fondo un LinearLayout con orientamento "vertical"  
`android:orientation="vertical"`.
3. Aggiungere tre Button da codice. Provate prima in `onCreate` e poi in `onStart`.  
Differenze?
4. Cosa fa di preciso "`setContentView`" ?