

Corso Interazione Uomo-Macchina



a.a. 2019-20



Francesco Cutugno
Enrico Leone
Vincenzo Norman Vitale
Marco Grazioso



Outline

- Elementi di un'app
- Android manifest
- Application life-cycle
- Activity



Reference: *Professional Android, Reto Meier*





Elementi di un'app Android

Activity: il livello di presentazione dell'app. La UI è costruita attorno a una o più activities. Usano *Fragments* e *Views* per organizzare e mostrare le informazioni;

Service: sono in esecuzione senza UI e comunicano con gli altri oggetti. Utili per eseguire task asincroni (es. comunicazioni di rete);

Intent: elementi per il *message-passing* tra le applicazioni;

Content Provider: gestiscono la persistenza dei dati dell'app. Sono usati per gestire dati tra le applicazioni (es. *Contatti* e *Galleria*);





Elementi di un'app Android

Broadcast Receiver: riceve intent dal sistema operativo (es. eventi sulle chiamate);

Widget: componenti visive delle applicazioni che possono essere aggiunti sulla home-screen. Sono una variazione dei Broadcast Receiver;

Notification: abilitano l'app all'invio di *alert* all'utente senza togliere il focus all'app in esecuzione.





Android Manifest

File XML che descrive i metadati e la struttura dell'app

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="it.priscalab.hci"
    android:versionCode="1"
    android:versionName="1.0" >

</manifest>
```





Android Manifest (1)

Permessi:

Syntax

```
<uses-permission android:name="string"  
                  [android:maxSdkVersion="integer"] />
```

Esempi

```
<uses-permission android:name="android.permission.READ_CONTACTS"/>  
  
<uses-permission android:name="android.permission.INTERNET" />
```





Android Manifest (2)

SDK:

Syntax

```
<uses-sdk android:minSdkVersion="integer" android:targetSdkVersion="integer"  
[android:maxSdkVersion="integer"] />
```

Platform Version	API Level	VERSION_CODE	Notes
Android 7.1.1 Android 7.1	25	N_MR1	Platform Highlights
Android 7.0	24	N	Platform Highlights
Android 6.0	23	M	Platform Highlights
Android 5.1	22	LOLLIPOP_MR1	Platform Highlights
Android 5.0	21	LOLLIPOP	
Android 4.4W	20	KITKAT_WATCH	KitKat for Wearables Only
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights





Android Manifest (3)

Features:

Syntax

```
<uses-feature android:name="string" android:required=["true" | "false"]  
              android:glEsVersion="integer" />
```

Esempi

```
<uses-feature android:name="android.hardware.nfc" />
```

```
<uses-feature android:name="android.hardware.camera" />
```

Richiede

```
<uses-feature android:name="android.hardware.camera.autofocus" />
```




Android Manifest (4)

Schermi supportati:

Syntax

```
<supports-screens android:resizeable=["true" | "false"]  
                  android:smallScreens=["true" | "false"]  
                  android:normalScreens=["true" | "false"]  
                  android:largeScreens=["true" | "false"]  
                  android:xlargeScreens=["true" | "false"]  
                  android:anyDensity=["true" | "false"]  
  
                  android:requiresSmallestWidthDp="integer"  
                  android:compatibleWidthLimitDp="integer"  
                  android:largestWidthLimitDp="integer"/>
```

NB:

`android:resizeable` è deprecato e non va usato





Android Manifest (5)

```
<application android:allowBackup="true" android:icon="@drawable/ic_launcher"  
             android:label="@string/app_name" android:theme="@style/Theme.Sample" >
```

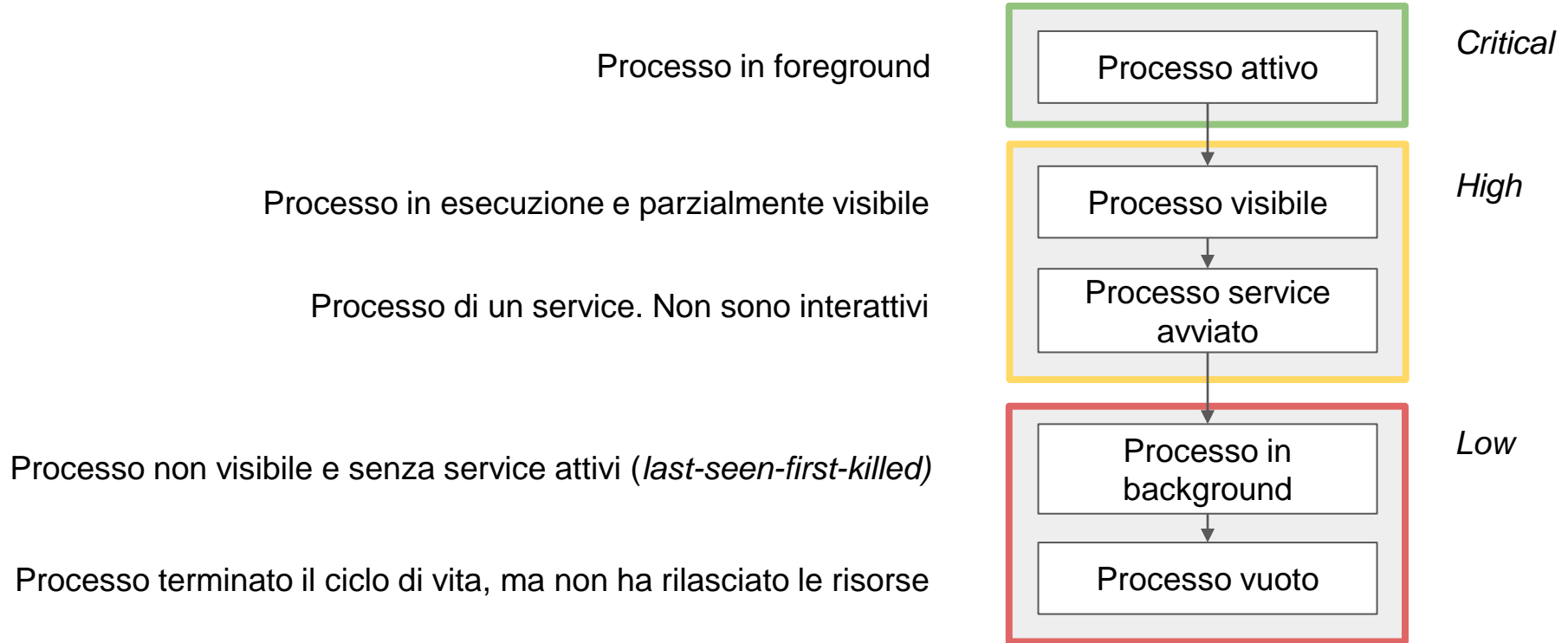
```
<activity>  
  <intent-filter>  
    <action />  
    <category />  
    <data />  
  </intent-filter>  
  <meta-data />  
</activity>
```

```
<service>  
  <intent-filter> . . . </intent-filter>  
  <meta-data />  
</service>  
<receiver>  
  <intent-filter> . . . </intent-filter>  
  <meta-data />  
</receiver>  
<provider>  
  <grant-uri-permission />  
  <meta-data />  
  <path-permission />  
</provider>
```

```
</application>
```



Application priority stack





Application

```
public class MyApplication extends Application {  
    <application android:icon="@drawable/icon" android:name=".MyApplication">  
        private static MyApplication singleton;  
        [...]  
        public static MyApplication getInstance() {  
    </application>        return singleton;  
    }  
  
    @Override  
    public final void onCreate() {  
        super.onCreate();  
        singleton = this;  
    }  
  
    @Override  
    public final void onLowMemory() {...}  
    @Override  
    public final void onTrimMemory(int level) {...}  
    @Override  
    public final void onConfigurationChanged(Configuration newConfig) {...}  
}
```

Specifica la classe dell'app





Activity

```
public class MyActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

Metodo obbligatorio

onRestart()

onStart()

onResume()

onPause()

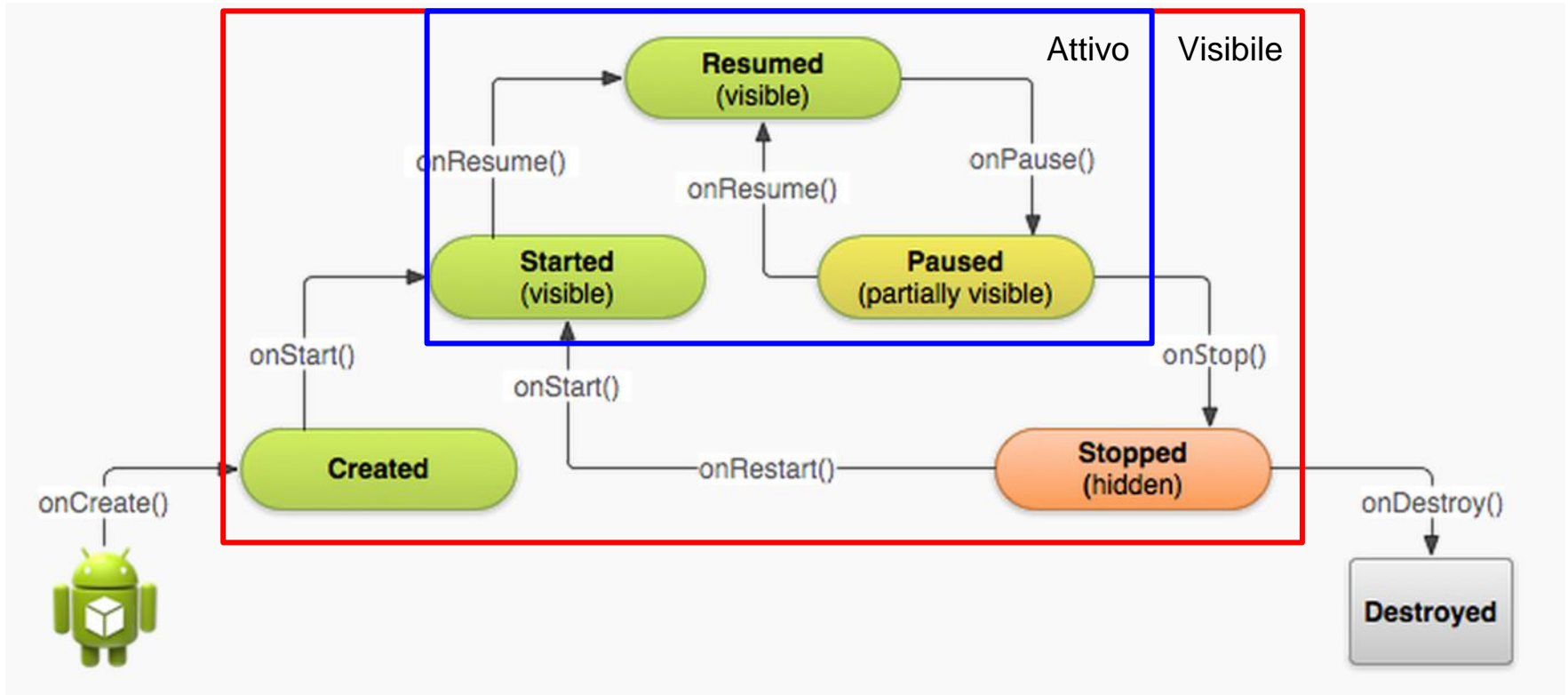
onStop()

onDestroy()





Activity lifecycle

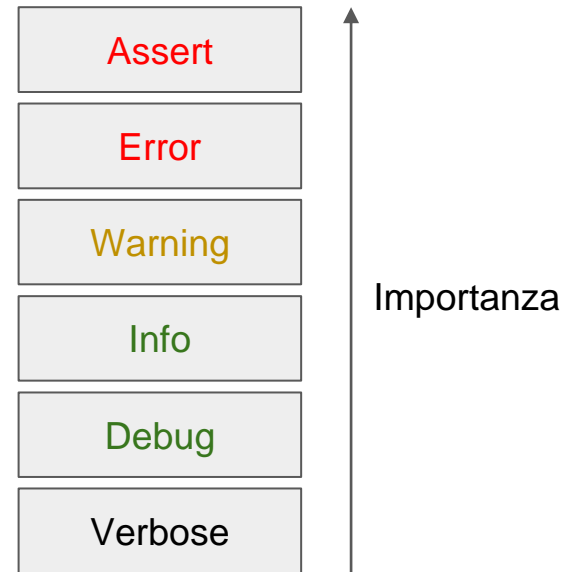




Log

```
private static final String TAG = "MyActivity";
```

Log. wtf (String Tag, String Message);
 e
 w
 i
 d
 v





Resources

XML

`@string/nome_stringa`

`@dimen/nome_dimensione`

JAVA

`getResources().getString(id_stringa)`

`getResources().getDimension(id_dimen)`

`id_stringa = R.string.nome_stringa`

`id_dimen = R.dimen.nome_dimensione`





Esercizio

Creare un'activity e implementare i metodi del ciclo di vita

Per ogni metodo aggiunto, stampare un messaggio di Log





Cosa c'è di sbagliato?

