ANDROID

LEZIONE 2 - 23/12/2015

https://github.com/elbuild/corso-android.git

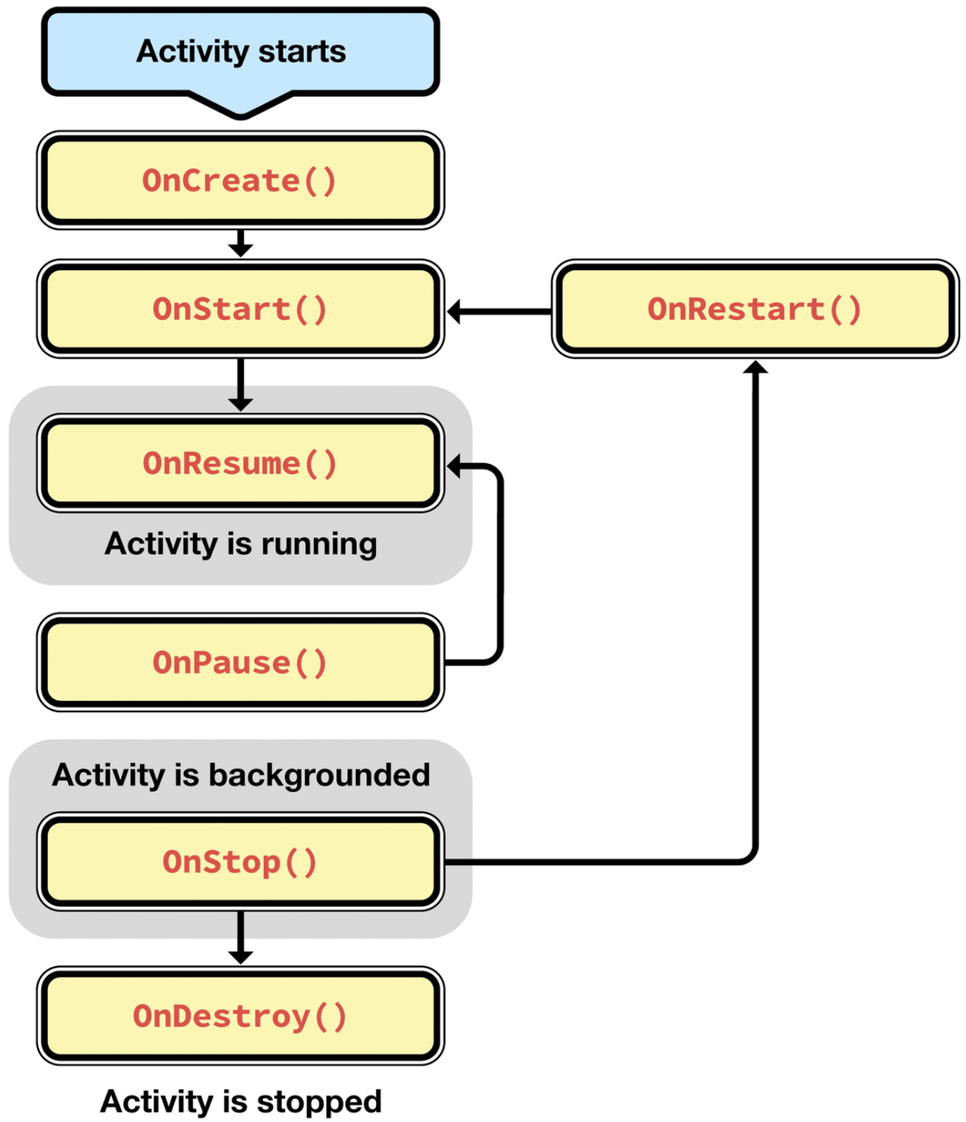
ActivityLifeCycle

https://github.com/elbuild/corso-android.git

APPLICAZIONE PRATICA - ActivityLifeCycle

Lo scopo di questa app è prendere familiarità con il concetto di *life cycle* di una *Activity Android*.

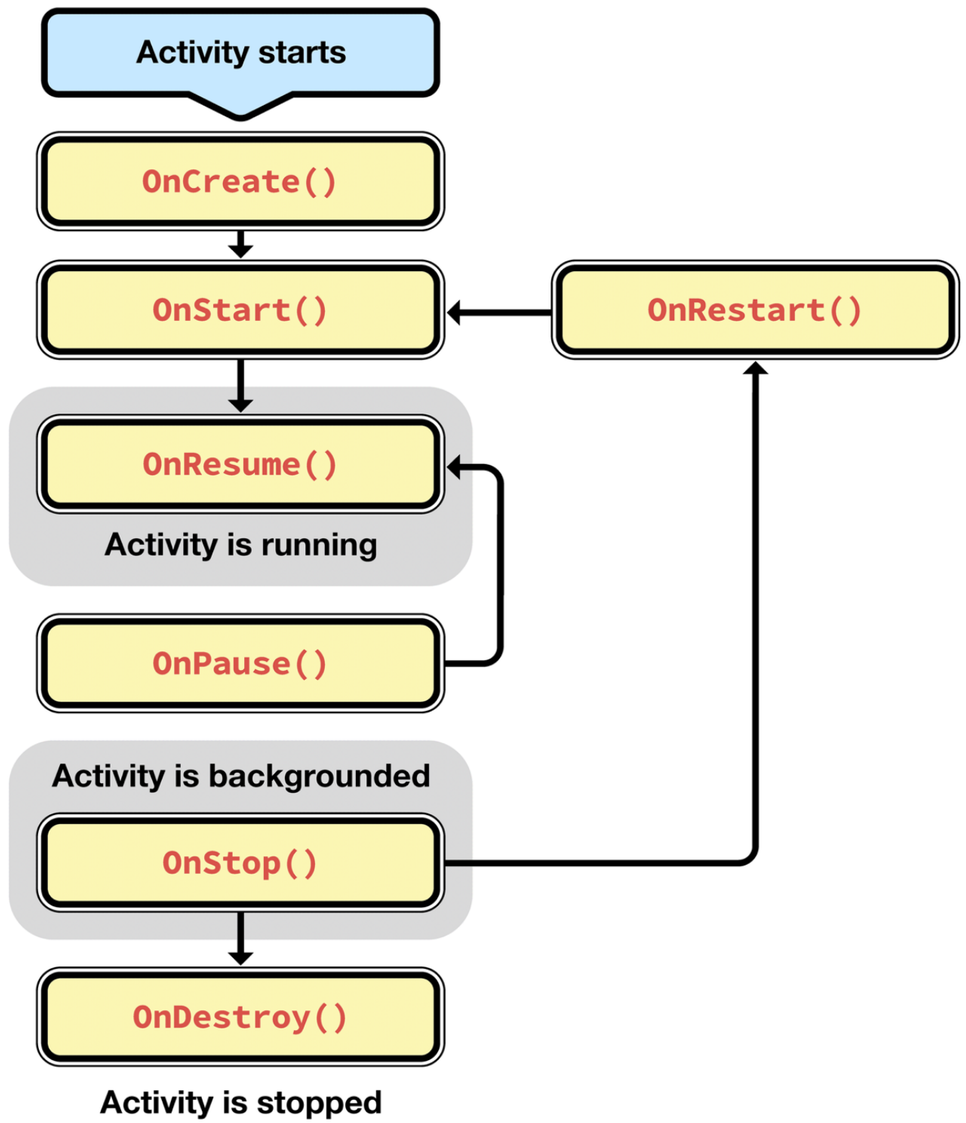
```
public class MainActivity extends Activity {

    private static final String TAG = "LIFECYCLE";

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.d(TAG, "onCreate() event called");
    }

    .... other methods here

}
```



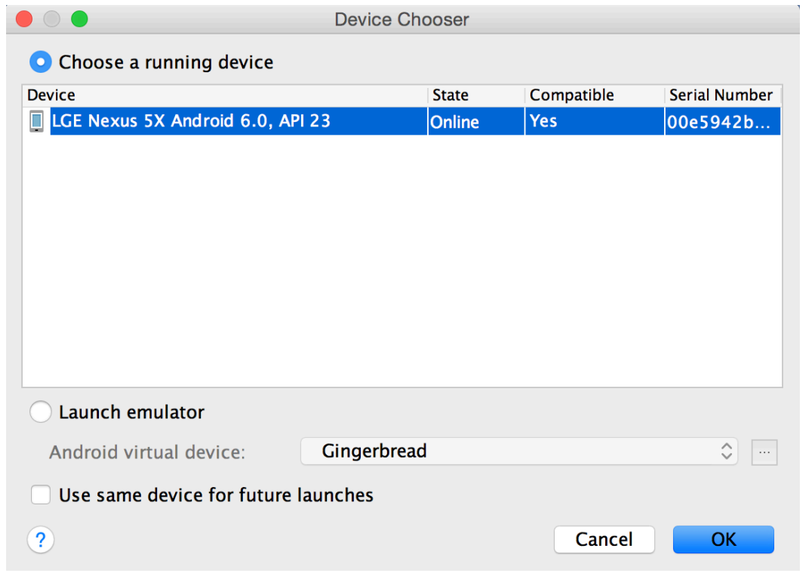
Esaminiamo output con LOGCAT nelle diverse situazioni.

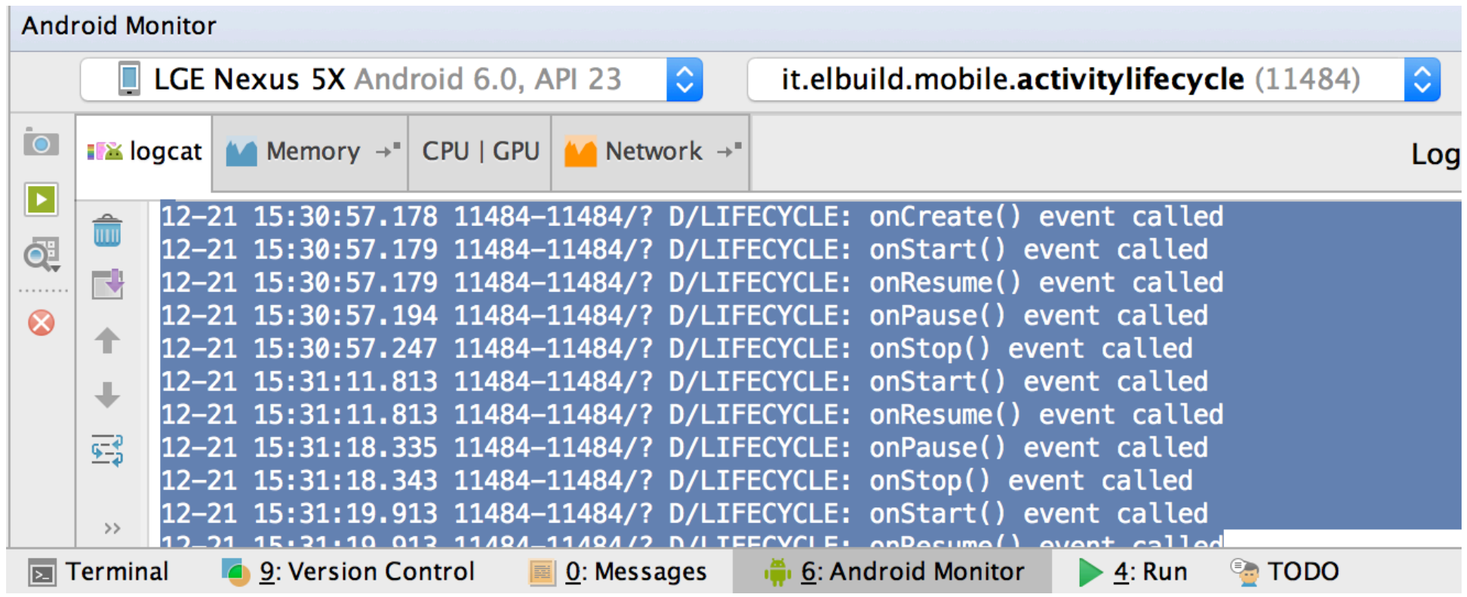
APPLICAZIONE PRATICA - ActivityLifeCycle

- **onCreate**: l'activity viene creata. Il programmatore deve assegnare le configurazioni di base e definire quale sarà il layout dell'interfaccia;
- **onStart**: l'activity diventa visibile. È il momento in cui si possono attivare funzionalità e servizi che devono offrire informazioni all'utente
- **onResume**: l'activity diventa la destinataria di tutti gli input dell'utente.
- **onPause** (l'inverso di onResume) notifica la cessata interazione dell'utente con l'activity;
- **onStop** (contraltare di onStart) segna la fine della visibilità dell'activity;
- **onDestroy** (contrapposto a onCreate) segna la distruzione dell'activity.



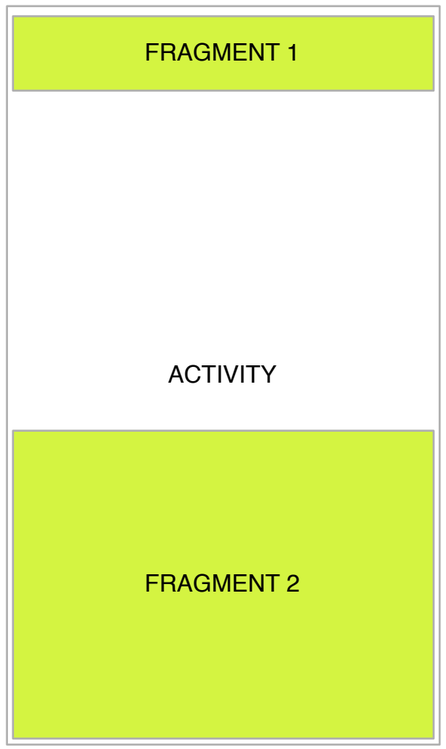
APPLICAZIONE PRATICA - ActivityLifeCycle, running





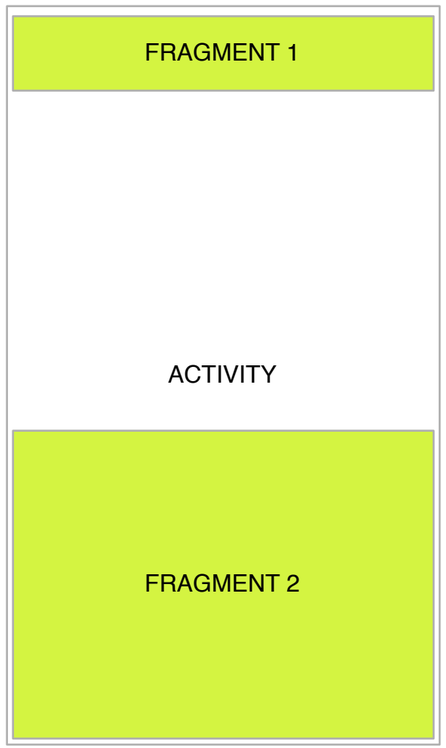
FRAGMENTS

ANDROID FRAGMENTS



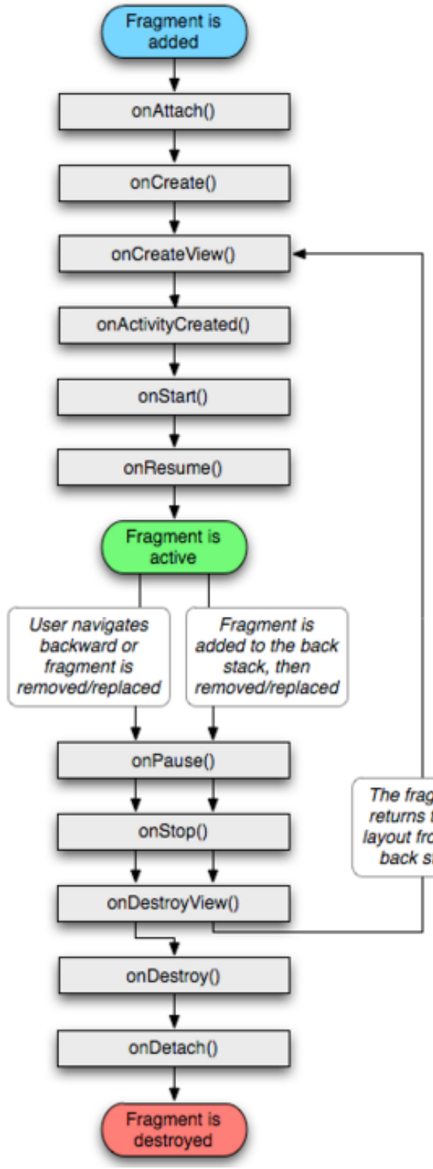
- Un Fragment è un oggetto che ha come scopo principale la gestione di una parte della UI
- Sono stati introdotti nella versione 3.0 HoneyComb nel processo di modernizzazione del sistema di creazione UI culminato con la release di ICS
- La classe FragmentActivity della libreria support ne rende l'uso possibile anche su versioni precedenti alla 3.0
- Un fragment "vive" all'interno di una Activity e dipende in maniera molto stretta dal suo *lifecycle*

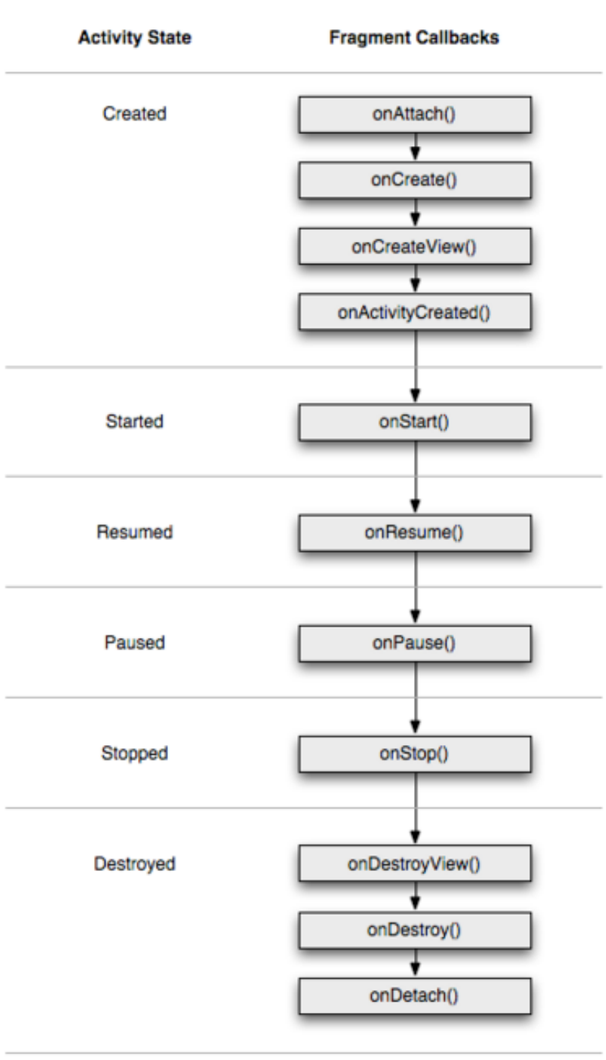
ANDROID FRAGMENTS

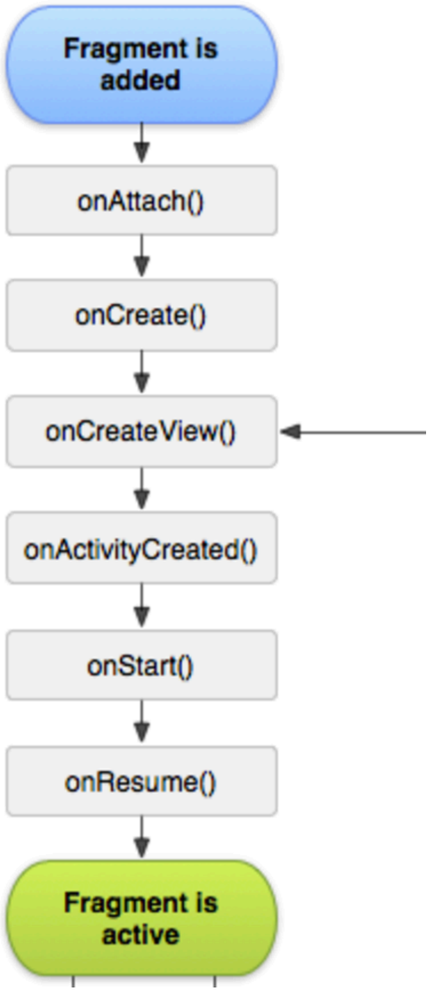


- Un Fragment dal punto di vista di una Activity è semplicemente un componente del layout.
- Una volta che il fragment è *attached* l'Activity padre lo vede come un ViewGroup all'interno della sua gerarchia di layout.
- Il fatto che sia un ViewGroup rende possibile gestirne a runtime il posizionamento in maniera programmatica, come per qualsiasi altro elemento di UI.
- La gestione dei Fragment è possibile attraverso la classe FragmentManager a cui ogni activity può accedere attraverso il metodo *getFragmentManager()*.

ANDROID FRAGMENTS







- **onAttach**: segnala il momento in cui il Fragment scopre l'Activity di appartenenza. L'Activity non è ancora instanziata per cui non si può interagire con i suoi metodi.
- **onCreate**: è la creazione del Fragment in quanto componente;
- **onCreateView**: il programmatore vi lavorerà spesso. È il momento in cui viene creato il layout del Fragment. Solitamente qui si fa uso del *LayoutInflater*;
- **onActivityCreated**: segnala che la creazione dell'Activity è stata completata, vi si può interagire in tutto e per tutto.

ANDROID FRAGMENTS

- Una Activity gestice i Fragment attraverso un FragmentManager, utilizzando delle implementazioni di FragmentTransactions

```
public class MainActivity extends Activity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (savedInstanceState == null)
        {
            getSupportFragmentManager().beginTransaction()
                    .add(R.id.container, new MyFragment()).commit();

        }
    }
}
```

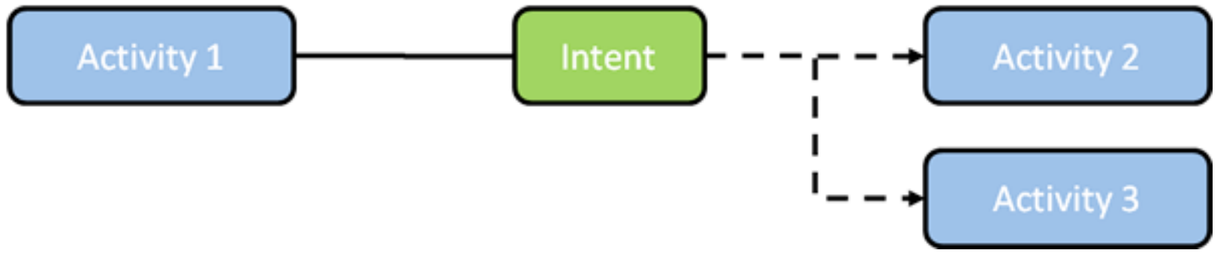
- Le FragmentTransactions più comuni sono le seguenti.

`add`	Aggiunge un Fragment all'Activity
`remove`	Rimuove un fragment precedentemente aggiunto
`replace`	Sostituisce un Fragment con un altro
`hide`	Nasconde un fragment
`show`	Mostra un fragment precedentemente nascosto

FragmentLifeCycle and FragmentTrasactions

https://github.com/elbuild/corso-android.git

INTENT e INTENT FILTER



- Un Intent è un oggetto il cui scopo principale è quello di veicolare un messaggio fra due componenti del sistema
- Esistono Intent espliciti ed impliciti
- Un Intent può avere extra, intesi come oggetti che implementano l'interfaccia Parcelable

FINALMENTE UN PO' DI CODICE

https://github.com/elbuild/corso-android.git

Q&A

https://github.com/elbuild/corso-android.git