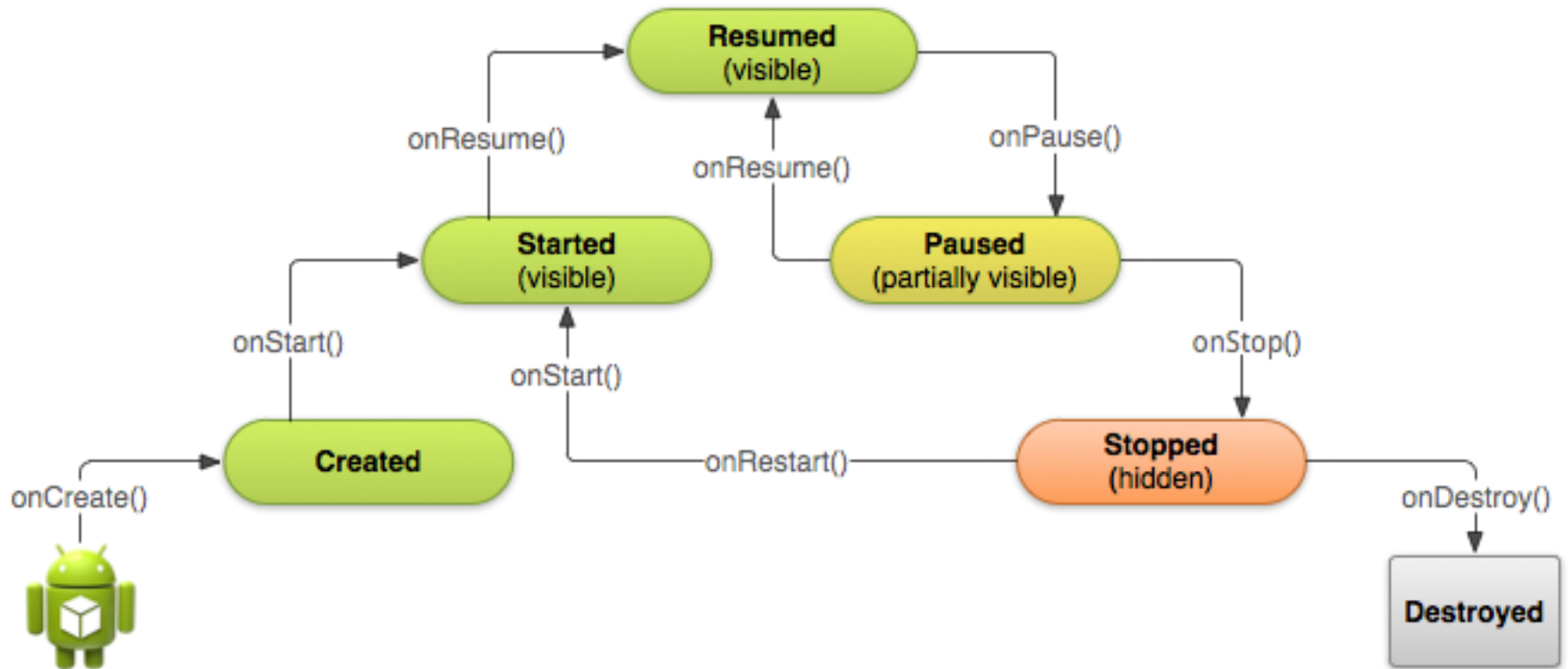


Activity

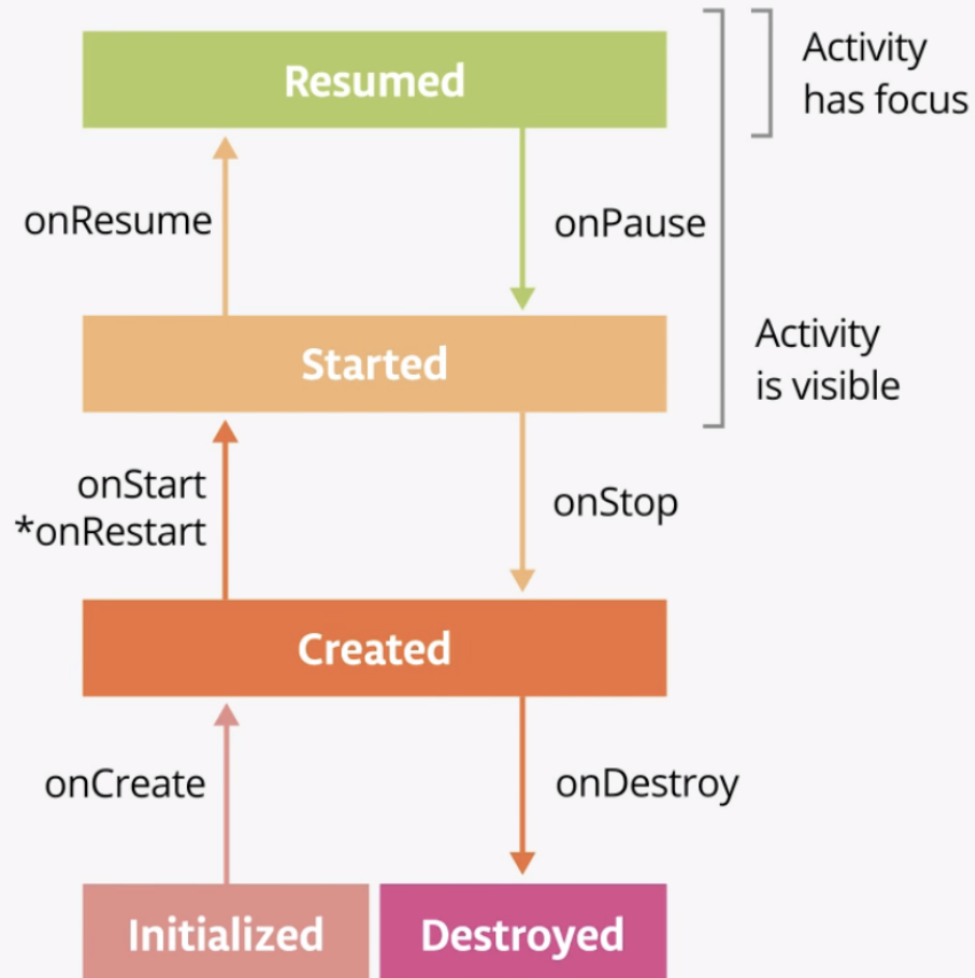
Activity

- **Una Activity**
 - è realizzata con una classe java che estende `android.app.Activity`
 - rappresenta una singola schermata grafica della nostra applicazione
 - viene solitamente associata ad un'azione specifica che l'utente può fare
 - è si attiva se in primo piano
- **Una applicazione contiene solitamente molte Activity**
- **Ogni Activity ha il compito di memorizzare e ripristinare lo stato delle sue variabili di istanza**

Stati di una Activity



The Activity Lifecycle



Intent

Intent

- Cos'è un Intent

- Un messaggio che consente di comunicare tra componenti dell'app (o tra app diverse)
- Contiene informazioni su cosa fare e con quali dati
- Il destinatario esegue un'azione in base al contenuto

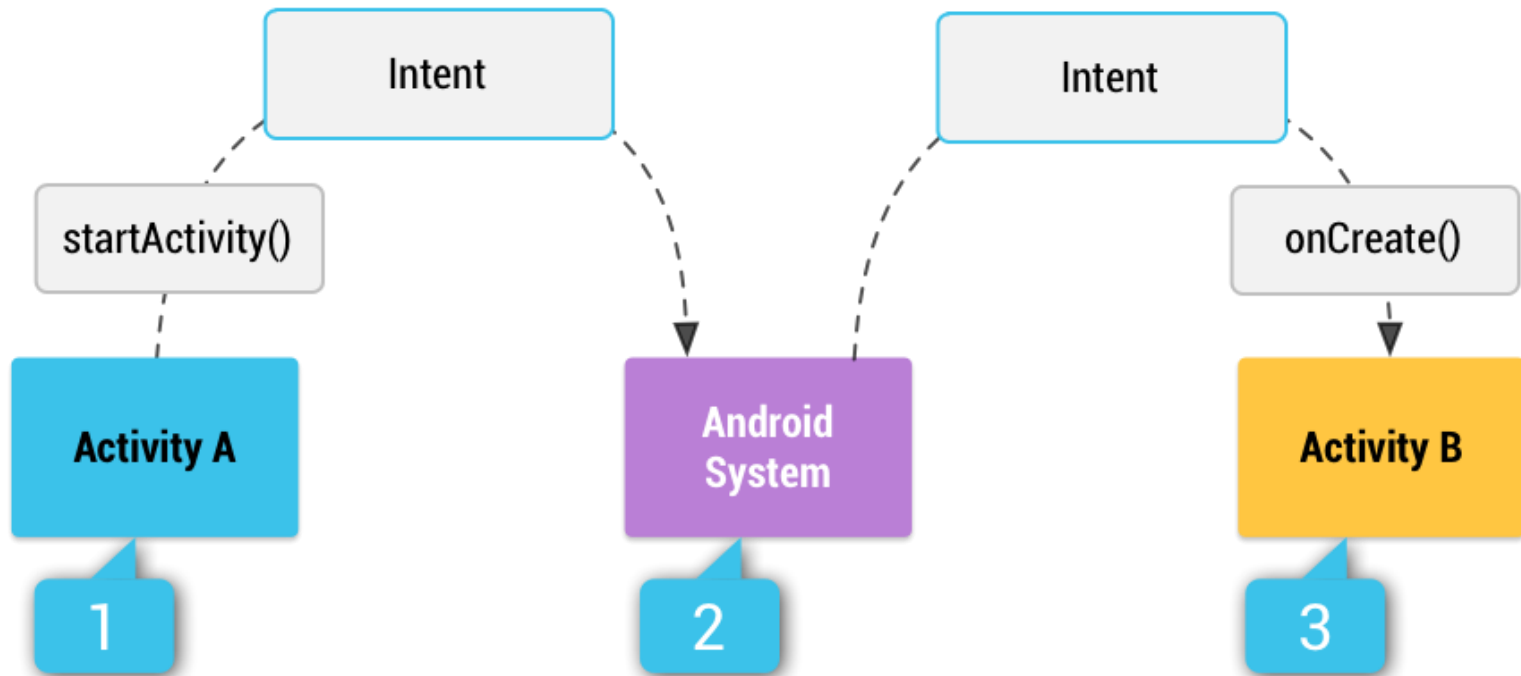
- Usi principali:

- Avviare un'Activity
- Avviare un Service
- Inviare un Broadcast

- Tipi di Intent

- Espliciti
 - Specificano esattamente quale componente avviare
 - Usati per lanciare Activity o Service della stessa app
- Impliciti
 - Descrivono un'azione da compiere

Avviare un'Activity



Intent Esplicito

- Usi quando sai esattamente quale componente vuoi avviare (es. un'altra Activity della tua app):

```
Intent intent = new Intent(this, ColorPickerActivity.class);  
startActivity(intent);
```

- Avvia direttamente ColorPickerActivity definita nella tua app.

Passare dati con gli Intent

- Aggiungere dati all'Intent (mittente)

```
Intent intent = new Intent(this, NoteDetailActivity.class);
intent.putExtra("note_text", "Questa è la mia nota");
startActivity(intent);
```

- Sono coppie chiave/valore di tipi "Base"
 - Integer, String, etc.
- Leggere i dati (destinatario)

```
Intent intent = getIntent();
String noteText = intent.getStringExtra("note_text");
```

Intent Implicito

- Usi quando vuoi eseguire un'azione, ma non sai (o non ti interessa) quale app la gestirà:

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.setType("text/plain");
intent.putExtra(Intent.EXTRA_TEXT, "Testo della mia nota");
startActivity(Intent.createChooser(intent, "Condividi nota con"));
```

- Chiede al sistema di mostrare le app in grado di condividere testo (es. Gmail, WhatsApp, ecc.).

Intent ed Action

- una action è una stringa che identifica il tipo di operazione che sarà eseguita
 - `Intent.ACTION_DIAL`
 - “`android.intent.action.DIAL`”
 - `it.apogeo.android.cap04.senderproject.intent.action.CUSTOM_ACTION`
- Sintassi della stringa
 - `<package>.intent.action.<action name>`
- Metodi
 - `setAction()`

Esempi azioni standard

- Costanti della classe Intent

- ACTION_VIEW
 - content://contacts/people/1
- ACTION_DIAL
 - content://contacts/people/1
- ACTION_VIEW
 - tel:123
- ACTION_DIAL
 - tel:123
- ACTION_EDIT
 - content://contacts/people/1
- ACTION_VIEW
 - content://contacts/people/

- Esempio di esecuzione della activity per inserire un numero

```
Intent intent = new Intent() ;
intent.setAction(Intent.ACTION_DIAL) ;
startActivity(intent) ;
```

Intent, action e dati

- In generale una coppia action/dato specifica l'activity da eseguire
- la forma del dato è una URI (RFC 2396)
 - scheme://host:port/path
- Esempi
 - ACTION_VIEW content://contacts/people/1
 - ACTION_DIAL content://contacts/people/1
 - ACTION_VIEW tel:123
 - ACTION_DIAL tel:123
 - ACTION_EDIT content://contacts/people/1
 - ACTION_VIEW content://contacts/people/

Action custom

Receiver

- Si usa un Intent filter per dichiarare la volontà di gestire l'azione nel receiver

```
<intent-filter>
    <action android:name="cap04.senderproject.intent.action.CUSTOM_ACTION">
    </action>
    <category android:name="android.intent.category.DEFAULT">
    </category>
</intent-filter>
```

Sender

- Si usa un Intent a cui si imposta l'action opportuna

```
Intent intent = new Intent() ;
intent.setAction(CUSTOM_ACTION) ;
startActivity(intent) ;
```

Categorie

- Categoria dell'intent
- Categorie di default
 - **CATEGORY_DEFAULT**, CATEGORY_BROWSABLE, CATEGORY_TAB, CATEGORY_ALTERNATIVE, CATEGORY_SELECTED_ALTERNATIVE, CATEGORY_LAUNCHER, CATEGORY_INFO, CATEGORY_HOME, CATEGORY_PREFERENCE, CATEGORY_TEST ,...
- Si imposta con `addCategory()`

Data

- **MIME Type**

- Identificatore del tipo di dato
- RFC 2046 (http://en.wikipedia.org/wiki/Internet_media_type)
- tipo/sottotipo
 - image/jpeg, image/png, image/*, text/plain, text/html, ...

- **URI**

- `<scheme>://<host>:<port>[<path>|<pathPrefix>|<pathPattern>]`

- **Metodi di Intent**

- `setDataAndType (Uri data, String type)`
- `setType(String)`
- `setData(Uri)`
- `setDataAndTypeAndNormalize(Uri, String)`

Extra

- Coppie chiave-valore
 - Dati richiesti per compiere l'azione voluta
- Metodi
 - `putExtra(String, <Tipo>)`
 - `get<Tipo>Extra`
- Esempi di Extra standard
 - EXTRA_EMAIL, EXTRA_PHONE_NUMBER, EXTRA_REFERRER, EXTRA_SHORTCUT_ICON, EXTRA_SHORTCUT_NAME, EXTRA_SUBJECT, EXTRA_TEMPLATE, EXTRA_TEXT, EXTRA_TITLE, EXTRA_UID

Flags

- Servono a comunicare al sistema operativo come deve gestire il lancio del componente
- Si settano nell'intent
 - `setFlags()` o `addFlag()`

```
intent.addFlag(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
intent.setFlags(intent.getFlags() | FLAG_ACTIVITY_NEW_TASK);
intent.setFlags(Intent.FLAG_ACTIVITY_FORWARD_RESULT |
    Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
```

```
// example...
// value of flags: 1
intent.setFlags(2|4);
// now flags have this value: 110
intent.addFlags(8);
// now flags have this value: 1110
```

Construttori di Intent

- `Intent()`
 - Create an empty intent.
- `Intent(Intent o)`
 - Copy constructor.
- `Intent(String action)`
 - Create an intent with a given action.
- `Intent(String action, Uri uri)`
 - Create an intent with a given action and for a given data url.
- `Intent(Context packageContext, Class<?> cls)`
 - Create an intent for a specific component.
- `Intent(String action, Uri uri, Context packageContext, Class<?> cls)`
 - Create an intent for a specific component with a specified action and data.

Intent Filter

- Dichiarano quali intent impliciti un componente può ricevere
 - Ogni componente ne può dichiarare più di uno
- Un intent filter può essere composto da
 - action – nome action
 - category – nome category
 - data – MIME Type e/o schema URI

```
<activity android:name="ShareActivity">
  <intent-filter>
    <action android:name="android.intent.action.SEND" />
    <category android:name="android.intent.category.DEFAULT" />
    <data android:mimeType="text/plain" />
  </intent-filter>
</activity>
```

Intent Resolution

- Action

- Un Intent filter può dichiarare zero o più action
- **L'action specificata nell'intent deve essere dichiarata nel filtro**

- Category

- Un Intent filter può dichiarare zero o più category
- **Tutte le categorie dell'intent devono essere presenti nel filter**

- Type e URI

- se presenti devono essere dichiarate nel filtro
- sono analizzate separatamente
- possono essere dedotte dai dati nell'intent

Filtri sul contenuto

- Filtri su uri

- `<scheme>://<host>:<port>/<path>`
 - `http`, `mailto`,
- controllato in modo gerarchico
 - prima lo schema poi l'autority e poi il path

- Filtri mimeType

- cioè il tipo di contenuto
 - immagini (`image/jpeg`)
 - video (`audio/mpeg4-generic`)
 - contatti (specifico di Android)
 - etc.

Esempi

```
<intent-filter>
  <action android:name="android.intent.action.VIEW"></action>
  <data android:mimeType="vnd.android.cursor.dir/contact"></data>
  <category android:name="android.intent.category.DEFAULT"></category>
</intent-filter>
```

```
<intent-filter>
  <data android:mimeType="image/*" />
  ...
</intent-filter>
```

```
<intent-filter>
  <data android:scheme="http" android:type="video/*" />
  ...
</intent-filter>
```

Package manager

- La classe che permette gestire le app si chiama Package Manager
 - Permette di scoprire quali app soddisfano un intent

Esempio di Intent filter

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    <category android:name="android.intent.category.HOME" />
    <category
android:name="android.intent.category.DEFAULT"></category>
</intent-filter>
```

Esempio di ricerca delle activity che possono essere eseguite per lanciare una app

```
Intent intent = new Intent(Intent.ACTION_MAIN);
intent.addCategory(Intent.CATEGORY_LAUNCHER);
PackageManager packageManager = getPackageManager();
List<ResolveInfo> infoList =
    packageManager.queryIntentActivities(intent, 0);
```


Activity che restituiscono valori

- Lanciare una activity per far ottenere un valore (URI) di ritorno

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(android.provider.ContactsContract.Contacts.CONTENT_URI);
startActivityForResult(intent, CHOOSE_CONTACT_CODE);
```

- Quando l'activity child “esce” viene chiamato il metodo

```
protected void onActivityResult(int requestCode, int resultCode, Intent data)
```

Esempio

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode,resultCode,data);
    if(requestCode==CHOOSE_CONTACT_CODE) {
        if(resultCode == Activity.RESULT_OK) {
        }else if(resultCode == Activity.RESULT_CANCELED) {
        }else{}
    }else{}
}
```

Generazione del risultato

- Nell'activity child
 - Gestisco l'esito
 - Posso impostare il dato nell'intent che ritorno
- Dati di ritorno
 - URI del dato
 - Extra
 - `public Bundle getExtras ()`
 - `public Intent putExtras (Bundle extras)`

Esito positivo

```
setResult(Activity.RESULT_OK, dataIntent);
finish();
```

Esito negativo

```
setResult(Activity.RESULT_CANCELED, dataIntent);
finish();
```

oppure solo `finish()`

I Task

- Ad un applicazione viene associato almeno un task
 - numero che identifica una sequenza di activity
- Le activity si alternano sul display
 - c'è uno stack di Activity
- Se non specificato le activity condividono lo stesso task
- Per lanciare una Activity in un nuovo task devono fare due cose:
 - dichiarare un attributo taskaffinity
 - `android:taskAffinity=":Other"`
 - settare nell'Intent un flag specifico
 - `Intent.FLAG_ACTIVITY_NEW_TASK`

