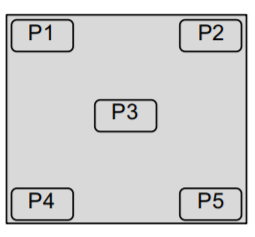
**Prova scritta del 04/2017**

**1 – Scrivere un file di layout che implementi l’interfaccia riportata in figura. Il rettangolo è un frame di 200x200px e deve essere posizionato al centro del display. I riquadri etichettati “Pn” sono dei pulsanti la cui pressione deve essere collegata alla funzione “pulsantePremuto”.**

****

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <RelativeLayout  
 android:layout\_width="200px"  
 android:layout\_height="200px"  
 android:background="#ccc"  
 android:layout\_centerInParent="true">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="P1"  
 android:onClick="pulsantePremuto"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="P2"  
 android:onClick="pulsantePremuto"  
 android:layout\_alignParentEnd="true"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="P3"  
 android:onClick="pulsantePremuto"  
 android:layout\_centerInParent="true"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="P4"  
 android:onClick="pulsantePremuto"  
 android:layout\_alignParentBottom="true"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="P5"  
 android:onClick="pulsantePremuto"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentEnd="true"/>  
  
 </RelativeLayout>  
  
</RelativeLayout>

**2 – Che cosa è un Adapter customizzato per un ListView? Si fornisca una spiegazione possibilmente con frammenti di codice.**

Quando si vuole creare una ListView di oggetti personalizzati abbiamo bisogno di definire un Adapter customizzato. Un adapter customizzato ci permette di inserire nella lista oggetti personalizzati, e di decidere come visualizzarli. Esempio per ogni l’elemento della lista abbiamo un’immagine una stringa e un intero.

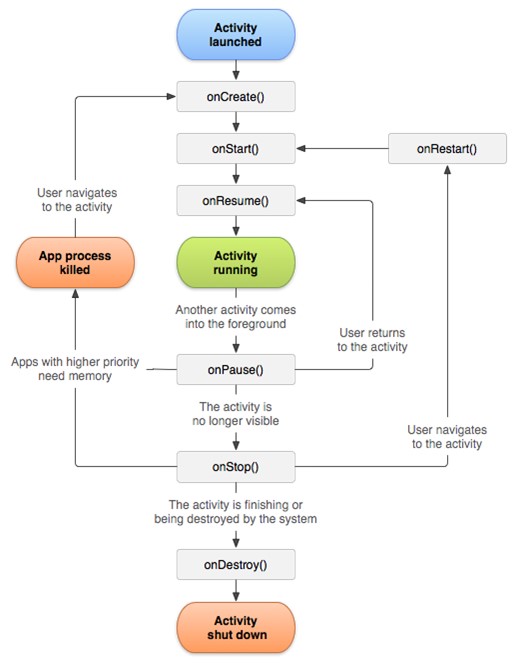
All’interno del metodo getView del customer Adapter andiamo a descrivere come visualizzare ogni oggetto della ListView.

Public View getView(int position, View v, ViewGroup parent) {

If (v == null) {  
 v = inflater.inflate(R.layout.list\_element, null);  
 }

Contatto c = getItem(position);  
 TextView name = (TextView) v.findViewById(R.id.elem\_lista\_name);  
 TextView telefono = (TextView) v.findViewById(R.id.elem\_lista\_tel);  
 ImageView foto = (ImageView) v.findViewById(R.id.elem\_lista\_foto);  
 name.setText(c.getName());  
 telefono.setText(c.getTel());  
 foto.setImageDrawable(c.getPicture());  
 return v;  
}

**3 – Si descriva il ciclo di vita di un’activity.**

****

**4 – Si scrivano opportuni frammenti di codice per il lancio di una nuova activity con un intent esplicito. L’activity lanciata restituisce un risultato numerico che deve essere recuperato nell’activity originale.**

ActivityA  
Intent i = new Intent(getApplicationContext(), ActivityB.class);  
startActivityForResult(I, 0);

Activity B  
int x = 10;  
intent I = new Intent(getAppicationContext(), ActivityA.class);  
i.putExtra(“Intero”, x);  
setResult(RESULT\_OK, i);  
}

**5 - Android permette al programmatore di specificare le dimensioni usando varie unità di misura:  
- dp (density-independent pixels)  
- sp (scale-independent pixels)  
- pt, points (1/72 di pollice)  
- px, pixel reali  
- mm, millimetri  
- in, pollici (inches)  
Perché? Si discuta delle differenze fra queste varie possibilità.**

Vengono utilizzate diverse unità di misura per adattare i widget a diversi dispositivi. Dp, sp, pt sono screen density ovvero dipendono da quanti pixel ci sono nel display (dpi), in particolare tra dp e sp l’unica differenza è che gli sp scalano in base alla preferenza dell’utente sulla grandezza del font.  
Px, mm, in rappresentano la grandezza reale del display

**6 – Si spieghi il meccanismo di misurazione e posizionamento che avviene sull’albero delle view quando ci sono delle modifiche al layout.**

-Fase di Misurazione (onMeasure())

Nella fase di misurazione c’e un processo di negozzazione tra parent e figli.

I figli definiscono una measuredSize ovvero quanto grande la view dovrebbe essere, che poi si trasformera nella size reale: quanto grande la view sarà in realtà.

Ogni view può esprimere la propria preferenza usando la classe ViewGroup.LayoutParams.

La misurazione avviene nel metodo onMeasure

-Fase di Layout (onLayout())

La View Parent decide la grandezza e la posizione dei figli in accordo alle misure fatte nella fase precedente

-Fase di Disegno (onDraw())

La view viene disegnata col metodo onDraw(), quando c’è un cambiamento si utilizza invalidate() chiamando onDraw su tutta la view e requestLayout che ripete l’intero processo su tutto l’albero

**7 – Per un oggetto AsyncTask, a cosa serve il metodo onProgressUpdate(Integer… progress)? Viene eseguito nel thread creato per il task asincrono o nel main thread? Perché?**

onProgressUpdate serve per aggiornare il main thread in base al lavoro del thread di background. Viene eseguito nel main thread in riposta al publishProgress nel doInBackground, perché Android non permette ai thread in background di interagire con l’interfaccia utente.

**8 - Quando un’activity viene distrutta perde lo stato. Descriverne brevemente i modi per salvare permanentemente i dati di un’activity.**

Per salvare permanentemente i dati di un’activity bisogna usare i Data Storage ovvero

Shared Preferences

-dati privati con coppa chiave-valor

File

-File privati dell’app

-File pubblici

Database SQLite

-Dati strutturati in database privati

**9 – Si consideri l’utilizzo di un sensore, Android permette di specificare la velocità di campionamento quando si registra il listener del sensore:  
boolean registerListener(SensorEventListener listener, Sensor sensor, int samplingPeriodUs). Android fornisce anche dei valori predeterminati da usare per il parametro samplingPeriodUs:  
- SensorManager.SENSOR\_DELAY\_NORMAL (0.2sec)  
- SensorManager.SENSOR\_DELAY\_GAME (0.02sec)  
- SensorManager.SENSOR\_DELAY\_UI (0.06sec)  
- SensorManager.SENSOR\_FASTEST (0sec)**

**Si discuta del ruolo di questo parametro spiegando perché si dovrebbe scegliere una frequenza più lenta o più veloce e delle conseguenze della scelta.**

DELAY\_NORMAL è consigliabile per il cambio di orientamento dello schermo

DELAY\_GAME è consigliabile per i giochi

DELAY\_UI è consigliabile per l’interfaccia utente

FASTEST permette di ottenere i dati dal sensore il più velocemente possibile

Un vantaggio è che avendo una frequenza di campionamento maggiore abbiamo dei dati più precisi, lo svantaggio è che si consuma più batteria

**10 – Si discuta del meccanismo dei frammenti. Si faccia un esempio per illustrarne l’utilità.**

Un Fragment è una porzione di Activity, non si tratta solo di un gruppo di controlli o di una sezione del layout. Può essere definito più come una specie di sub-activity con un suo ruolo funzionale molto importante ed un suo ciclo di vita.

Un Fragment non può vivere senza un’Activity. Come detto il Fragment ha il suo ciclo di vita fortemente collegato con quello dell’Activity di appartenenza

Un’activity può ospitare vari frammenti, i frammenti possono essere inseriti e rimossi durante l’esecuzione. SI possono creare UI con molti frammenti anche in funzione della grandezza dello schermo.

La filosofia di progettazione che risiede dietro i frammenti riguarda lo sviluppo di un interfaccia utente dinamica, in particolare capace di adattarsi sia a schermi grandi che a schermi piccoli

Un fragment può comunicare comunicare con l’activity creando dei metodi di callback.

Un esempio nell’utilita dei fragment e un app che mostra articoli di giornale un frammento conterra la lista degli articoli e l’altro la descrizione

• Se lo schermo è piccolo

– sarà visibile solo uno dei frammenti(quello contente la lista)

• cliccando un titolo si passerà dal frammento lista al frammento descrizione

• Se lo schermo è grande

**– saranno visualizzati entrambi i frammenti uno a fianco all’altro**