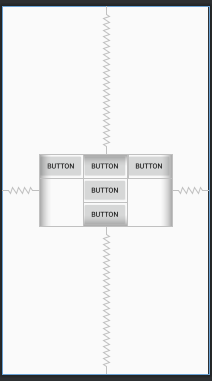
**Prova scritta del 01/2018  
1 – Disegnare nell’apposito spazio sulla destra (che corrisponde al RelativeLayout esterno) i widgets specificati dal seguente codice XML.**

?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"/>  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"/>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"/>  
  
 </LinearLayout>  
  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Button"/>  
  
 </LinearLayout>  
  
</RelativeLayout>

****

**2 – Il seguente codice incomplete è un CustomAdapter per una lista customizzata di oggetti Object. Ogni oggetto Object possiede i getter getString() e getInt(). Il file di layout “list\_element” contiene 2 TextView con i seguenti identificativi: “stringa” e “intero”. Completare il CustomAdapter per creare la view di ogni singolo elemento.**

@Override  
public View getView(int position, View v, ViewGroup parent) {

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

TextView tv\_1 = (TextView) findViewById(R.id.stringa);  
 TextView tv\_2 = (TextView) findViewById(R.id.intero);

Object o = getItem(position);

tv\_1.setText(o.getString());  
 tv\_2.setText(o.getInt());

return v;

}

**3 – Si spieghi il meccanismo del backstack. In relazione a tale meccanismo che differenza c’è fra un’activity e un frammento?**

Più activity possono coesistere tra loro e sono organizzate in un backstack. Esempio: solitamente un task (insieme di activity con le quali l’utente interagisce) parte dalla Home. Quando l’utente clicca su un’icona e lancia una nuova activity l’app è mostrata a schermo (foreground). Nel momento in cui vengono lanciate nuove activity, la corrente viene messa nel backstack e ci si può tornare col pulsante back.

Un task con le sue activity può essere spostato in background (quando l’utente inizia un nuovo task o clicca su Home per esempio) e le sue activity sono messe in stato di Stop ma il loro backstack resta intatto.

Il backstack considera solo lea activity, per i fragments dobbiamo gestirlo manualmente, per inserire i cambiamenti nel backstack bisogna creare un metodo addToBackStack(). Nel caso in cui non chiameremo il metodo addToBackStack() i cambiamenti andranno persi.

**4 – Si completi il disegno sotto stante il ciclo di vita di un’activity. Si descriva un’operazione che è solitamente effettuata in onStart(), con la corrispondente operazione effettuata in onStop(), e un operazione solitamente effettuata in onResume() con la corrispondente operazione effettuata in onPause().**

onCreate, onStart, onResume, onPause, onStop, onDestroy.

Un’operazione che solitamente viene effettuata in on

Un’operazione che solitamente viene effettuata in onResume è la registrazione dei listener realativi ai sensori, con il rilascio nel metodo onPause.

**5 – Si forniscano degli spezzoni di codice per il lancio di una nuova activity con un Intent esplicito e con un intent implicito. Si spieghi cosa è necessario per lanciare l’intent implicito.**

Per lanciare una nuova activity con l’utilizzo di un Intent esplicito bisogna indicare la classe della activity che si vuole lanciare:  
Intent i = new Intent(getApplicationContext(), SecondaActivity.class);  
startActivity(i);

Per lanciare una nuova activity con l’utilizzo di un intent implicito, non server indicare la classe dell’activity, poiché android ne sceglierà una adatta in basse all’action, type, uri, category.  
Esempio:  
Intent geoIntent = new Intent(Uri.parse(getApplicationContext(), “geo:0,0?q=”+address));  
startActivity(geoIntent);

**6 – Descrivi I tipi utilizzati per i parametri nella classe AsyncTask <Type1, Type2, Type3>. Qual è il ruolo di Type1, Type2, Type3? Fornire un esempio per rendere chiara la risposta.**

Questi tre tipi di dato saranno rispettivamente il tipo di dato accettato in input dai metodi doInBackground, onProgressUpdate, onPostExecute.

Un esempio è il seguente:

class LoadIconTask extends AsyncTask <Integer, Integer, Bitmap> {  
 private Integer index = 1;

@Override  
 protected void onPreExecute() {  
 ProgressBar.setVisibility(ProgressBar.VISIBLE);  
 }

@Override  
 protected Bitmap doInBackground(Integer… ids) {  
 Bitmap img;  
 img = BitmapFactory.decodeResource(  
 getResources(), img\_ids[0]);

for(int I = 0; I < 10; i++)   
 publishProgress(i);

return img;  
}

@Override  
 protected void onProgressUpdate(Integer… values) {  
 partialLoad();  
 progressBar.setStatus(values[0]);  
 if(values[0] > 75)  
 Toast.newToast(getApplicationContext(),  
 “Caricamento quasi finite”,  
 Toast.LENGHT\_LONG);  
 }

@Override  
 protected void onPostExecute(Bitmap result) {  
 ProgressBar.setStatus(0);  
 ProgressBar.setVisibility(ProgressBar.INVISIBLE);  
 imageView.setImageBitmap(result);  
 }

**7 – In che modo (o modi) varie activity che fanno parte della stessa app possono condividere dati? Si discuta dei vantaggi e svantaggi di ciascuno dei modi descritti.**

Varie activity che fanno parte della stessa app possono condividere dati tramite i data storage ovvero:  
- SharedPreferences: prevede di salvare e recuperare dati usando coppie di chiave-valore. Questi dati verranno salvati all’interno di un file xml e saranno accessibili solo dalle activity dell’applicazione.   
Il vantaggio e che non ha bisogno di stream, lo svantaggio e che utilizzando una coppia chiave valore per molti dati diventa scomodo.  
- File: per ogni app il sistema operativo prevede una directory privata, solo l’app può accedere a questa directory e se l’app viene disinstalla la directory viene cancellata. Per creare, scrivere, leggere un file utilizziamo gli stream, al file possono essere assegnata anche modalità come MODE\_APPEND. Il vantaggio e che è una soluzione ottima per salvare molti dati, lo svantaggio risiede nell’utilizzo degli stream.  
- Database: i dati vengono salvati all’interno di un database, Android fornisce un supporto per database SQL. Il vantaggio di utilizzare un database e che i dati vengono salvati in tabelle, mentre lo svantaggio e che si può lavorare sui dati solo tramite SQL.

**8 – Il seguente frammento di codice mostra un OnTouchListener per un MotionEvent. Si completi il codice facendo in modo che la variabile counter (si assumi che tale variabile sia accessibile globalmente) contenga sempre il numero di dita che stanno toccando lo schermo.**

public boolean onTouch(View v, MotionEvent event) {  
   
 switch (event.getActionMasked()) {  
 case MotionEvent.*ACTION\_DOWN*:  
 counter = 1;  
 break;  
 case MotionEvent.*ACTION\_POINTER\_DOWN*:  
 counter++;  
 break;  
 case MotionEvent.*ACTION\_UP*:  
 counter = 0;  
 break;  
 case MotionEvent.*ACTION\_POINTER\_UP*:  
 counter--;  
 break;  
  
 case MotionEvent.*ACTION\_MOVE*: {  
 break;  
 }  
 }  
 return true;  
}

**9 – Si spieghi come avviene la misurazione e il posizionamento delle view di un layout. Perché in alcuni casi i metodi v.getWidth e v.getHeight, dove v è una view del layout, usati in onCreate() restituiscono 0?**

**10 – Che cos’è un Toast customizzato? Si spieghi come implementare un Toast customizzato.**

Un toast customizzato è un breve messaggio temporaneo personalizzato che serve al programmatore per dare un feedback a colui che andrà ad utilizzare l’applicazione.

Public void showCustomToast(View v) {  
 Toast toast = new Toast(getApplicationContext());  
 toast.setGravity(Gravity.CENTER\_VERTICAL, 0 , 0);  
 toast.setDuration(Toast.LENGTH\_LONG);  
 toast.setView(  
 getLayoutInflater().inflate(R.layout.custom\_toast, null));  
 toast.show();  
}