



Mobile Programming

Prof. De Prisco

Prova scritta del esempio 1

NOME: _____

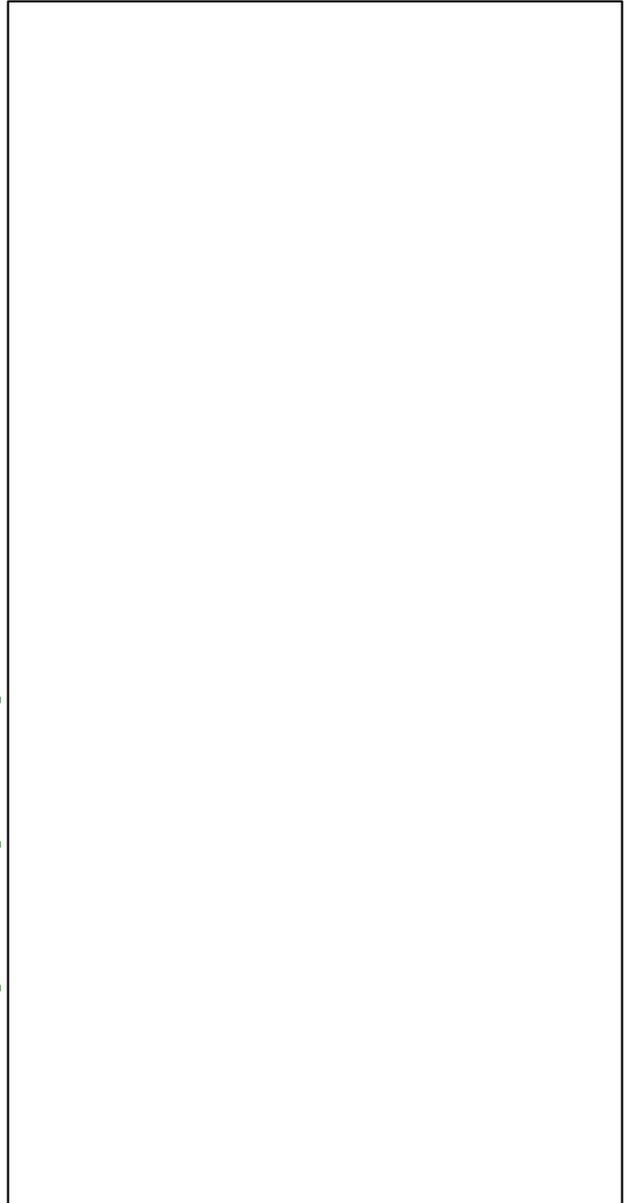
COGNOME: _____

MATRICOLA: _____

Domande	Punti
1	/10
2	/10
3	/10
4	/10
5	/10
6	/10
7	/10
8	/10
9	/10
10	/10
TOTALE	/100

Disegnare nell'apposito spazio sulla destra (che corrisponde al RelativeLayout esterno) i widgets specificati dal seguente codice XML.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Button" />
        <LinearLayout
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:orientation="vertical"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">
            <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>
    </LinearLayout>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />
</RelativeLayout>
```



Il seguente codice incompleto 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.

```
public class CustomAdapter extends ArrayAdapter<Object> {
    private int resource;
    private LayoutInflater inflater;

    public CustomAdapter(Context context, int resourceId, List<Object> objects) {
        super(context, resourceId, objects);
        resource = resourceId;
        inflater = LayoutInflater.from(context);
    }

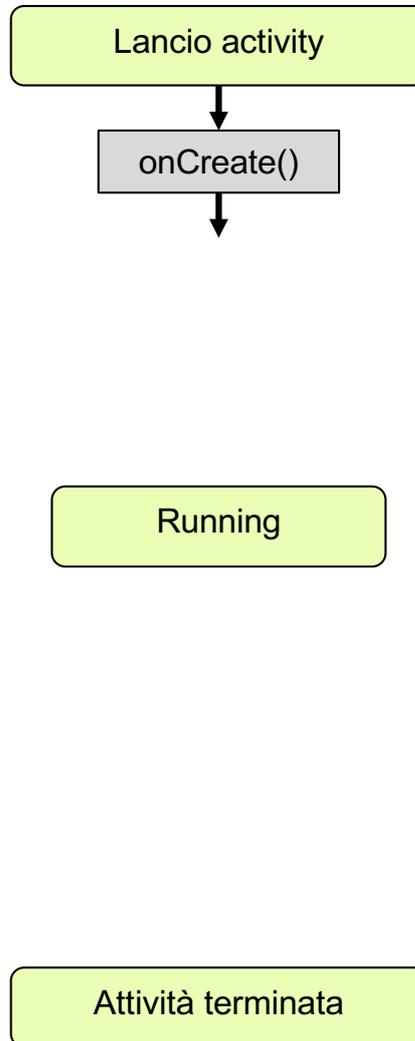
    @Override
    public View getView(int position, View v, ViewGroup parent) {
        if (v == null) {
            v = inflater.inflate(R.layout.list_element, null);
        }

        return v;
    }
}
```

Si spieghi il meccanismo del backstack.

In relazione a tale meccanismo che differenza c'è fra una activity e un frammento?

Si completi il disegno sottostante che rappresenta il ciclo di vita di una 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()`.



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.

Descrivere i tipi utilizzati per i parametri nella classe `AsyncTask<Type1, Type2, Type3>`
Quale è il ruolo di `Type1`, `Type2`, `Type3`? Fornire un esempio per rendere chiara la risposta

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.

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

```
int counter;
```

```
public boolean onTouch(View v, MotionEvent event) {  
    switch(event.getActionMasked()) {  
        case MotionEvent.ACTION_DOWN:  
  
            break;  
        case MotionEvent.ACTION_POINTER_DOWN:  
  
            break;  
        case MotionEvent.ACTION_MOVE:  
  
            break;  
        case MotionEvent.ACTION_POINTER_UP:  
  
            break;  
        case MotionEvent.ACTION_UP:  
  
            break;  
    }  
    return true;  
}
```

Quesito 9

Prof. DE PRISCO

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?

Che cosa è un Toast customizzato? Si spieghi come implementare un Toast customizzato.

