



# Mobile Programming

Prof. De Prisco

## Prova scritta del esempio 9

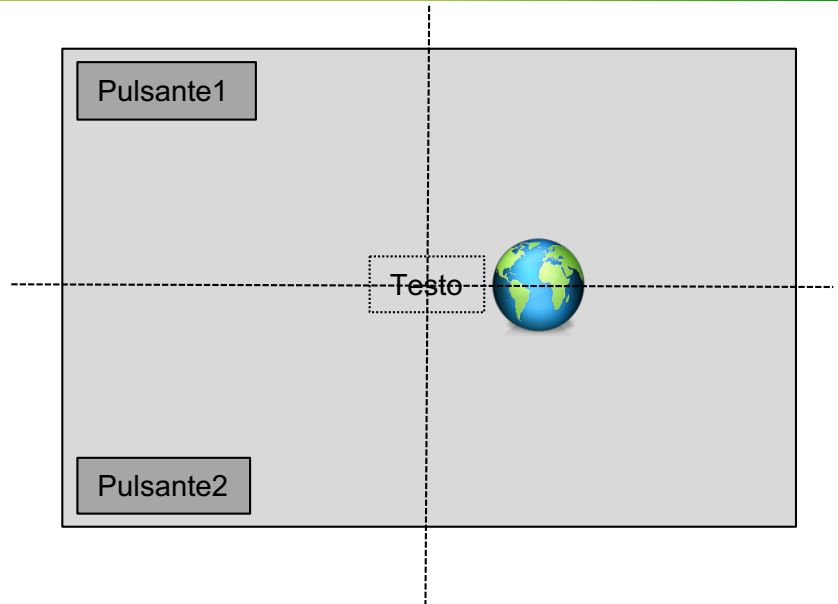
NOME: \_\_\_\_\_

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

Si scriva un frammento di codice XML che descriva il layout raffigurato a fianco scegliendo opportunamente gli elementi da utilizzare (le linee punteggiate non sono elementi del layout ma indicano il centro del layout).



Che cosa è un ListView? Che cosa è un Adapter? In che modo ListView e Adapter interagiscono?

## Quesito 3

Prof. DE PRISCO

Che cosa è il backstack? Supponendo che l'activity A sia l'unica in esecuzione (l'unica presente nel backstack), che la stessa app lancia una nuova activity B che a sua volta lancia l'activity C, che lancia l'activity D, quale è il backstack a questo punto? Cosa succede se dall'activity D si preme il pulsante di "back"?

Cosa si deve fare se si vuole fare in modo che dall'activity D, si torni direttamente ad A quando si preme il pulsante di back?

## Quesito 4

Prof. DE PRISCO

Un dispositivo Android può funzionare sia in modalità portrait (verticale) landscape (orizzontale). Quando un dispositivo Android viene ruotato si passa dall'una all'altra modalità. Per gestire in maniera appropriata tali passaggi, di cosa si deve preoccupare il programmatore?

Un'activity ActivityA ha bisogno di lanciare un'ActivityB passando a tale activity un intero x ed una stringa s. Cosa si deve fare? Dettagliare la risposta con opportuni frammenti di codice.

Il seguente abbozzo di classe NetworkTask mostra nviene utilizzato per ricevere da Internet dei dati nella forma di Stringa. Nella parte di ricezione (non dettagliata) viene eseguito un ciclo per 100 iterazioni e ad ogni itetrazione viene ricevuta una stringa str mentre la stringa received\_data viene utilizzata per memorizzare tutti i dati ricevuti che alla fine verranno visualizzati nel TextView textResponse. Nel layout c'è una progressBar utilizzata per mostrare in quale iterazione ci troviamo. Questo codice non funziona. Perché? Come possiamo risolvere il problema?

```
class NetworkTask extends AsyncTask<Integer, Integer, String> {
```

```
    @Override
```

```
    protected void onPreExecute() {
```

```
        progressBarDownload.setVisibility(ProgressBar.VISIBLE);
```

```
    }
```

```
    @Override
```

```
    protected String doInBackground(Integer...values) {
```

```
        //Do something
```

```
        String received_data = "";
```

```
        for (int i=0; i<100; i++) {
```

```
            progressBarDownload.setProgress(i);
```

```
            //Ricevi i dati nella stringa str
```

```
            received_data += str;
```

```
        }
```

```
        return received_data;
```

```
    }
```

```
    @Override
```

```
    protected void onPostExecute(String data) {
```

```
        progressBarDownload.setVisibility(ProgressBar.INVISIBLE);
```

```
        textResponse.setText(data);
```

```
    }
```

```
}
```

Il seguente snippet di codice esegue prima un'animazione e poi rimuove l'oggetto dalla view parent (gli oggetti image, animation e parentView sono stati in precedenza opportunamente inizializzati):

```
...  
image.startAnimation(animation);  
parentView.removeViewAt(0);  
....
```

Tuttavia l'effetto è quello di rimuovere immediatamente l'immagine senza dare il tempo all'animazione di essere eseguita. Perché accade ciò? Come si può ovviare al problema?



Per cosa sono utili i Frammenti?

Si descriva l'utilizzo di un sensore spiegando cosa si deve fare per utilizzarlo. Arricchire la spiegazione con frammenti di codice.

Si descrivano brevemente i Services, i BroadcastReceiver ed i ContentProvider