



# Mobile Programming

**Prof. De Prisco**

## Prova scritta del

13/01/2021

### MODALITA' TELEMATICA

Per rispondere ai quesiti della prova utilizza dei fogli bianchi. Un foglio (anche una sola faccia di un foglio) per ogni quesito. Etichetta il foglio/faccia con il tuo nome in alto a sinistra e con il numero del quesito al quale il foglio/faccia si riferisce:

Alla fine della prova, in fase di consegna dovrai mostrare tutti i fogli alla telecamera per permettere al docente di controllare quanto scritto.

Terminata la connessione dovrai scannerizzare i fogli, o fare delle foto in cui il testo scritto risulti leggibile.

Successivamente dovrai creare un unico file, possibilmente PDF, con tutti i fogli IN ORDINE, cioè dal primo quesito all'ultimo. Se per qualche quesito non hai risposto, ovviamente puoi saltarlo in fase di digitalizzazione.

Infine dovrai caricare il file pdf tramite il gruppo Teams creato per l'esame, aspettando una conferma di ricezione da parte del docente.

COGNOME, NOME	QUESITO X

Scrivi il codice XML per un file di layout che prevede un viewgroup radice che sfrutta tutto lo spazio disponibile e all'interno del quale sono posizionati i seguenti elementi, che prenderanno solo lo spazio necessario per essere disegnati:

1. 2 pulsanti allineati uno di seguito all'altro e, come insieme, centrati sia orizzontalmente che verticalmente nel viewgroup radice, con testo "Pulsante1" e "Pulsante2" e listeners "azione1" e "azione2"
2. Un textview sopra i 2 pulsanti, centrato orizzontalmente nel viewgroup radice, con testo "Sopra"
3. Un textview sotto i 2 pulsanti, centrato orizzontalmente nel viewgroup radice, con testo "Sotto"

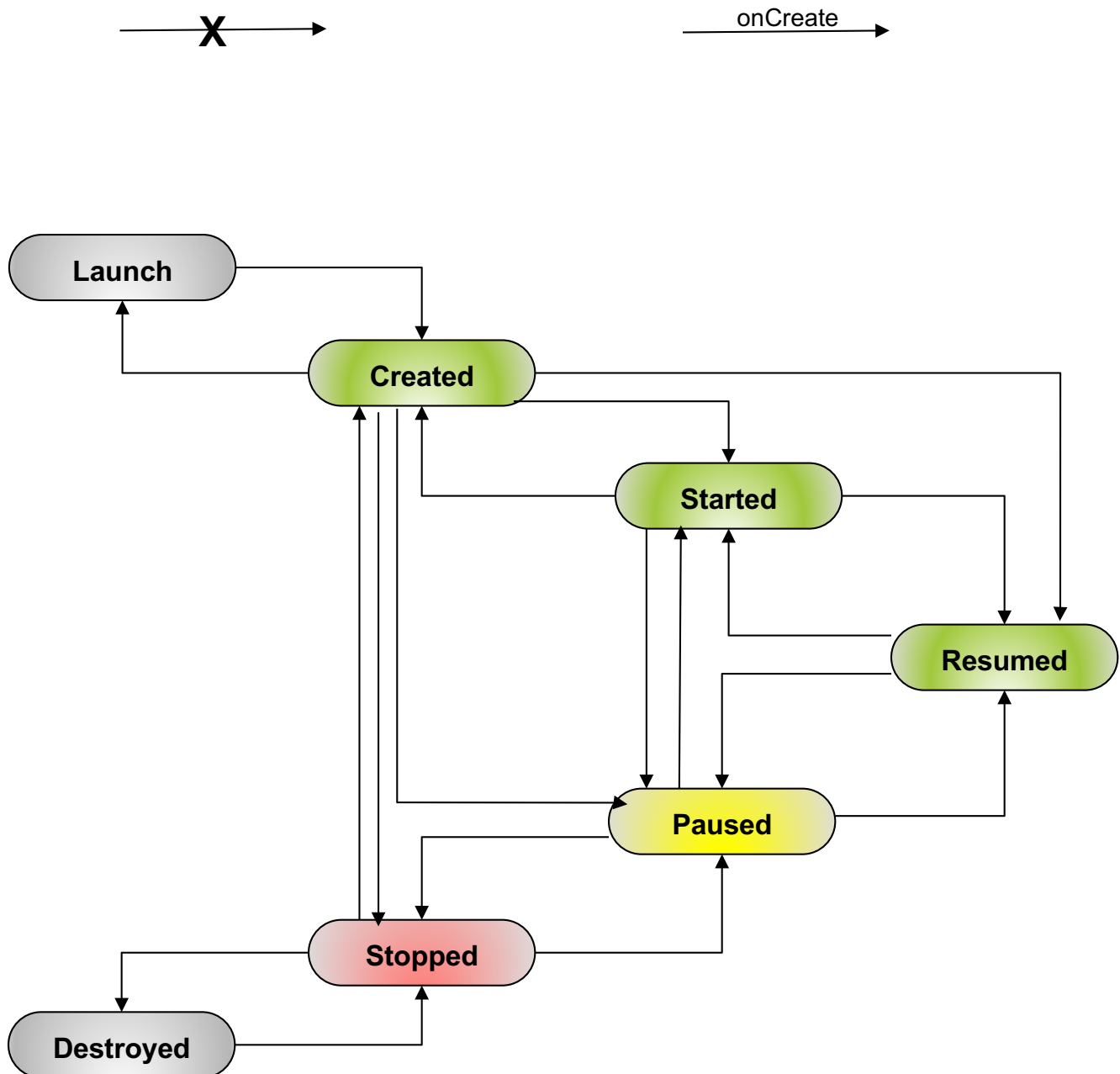
Si consideri il metodo `OnItemClickListener` associato ad un `listview`. Quale è l'informazione principale che il sistema operativo fornisce, come parametro di input al metodo, quando viene invocato tale metodo in seguito alla selezione (click), da parte dell'utente, di un elemento della lista?

A quale problematica si va incontro quando si usa un `listview` customizzato con dei listener su singoli elementi del layout customizzato e come possiamo risolverla?

```
        return v;  
    }  
}
```

Il seguente diagramma riporta gli stati del ciclo di vita di un'activity con delle frecce che indicano dei passaggi da uno stato a un altro.

Indicare per ogni freccia, un passaggio "impossibile" con una X sulla freccia, e un passaggio possibile con il nome del metodo che implementa il passaggio vicino alla freccia



Si consideri il seguente codice di un'Activity

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        Intent i = new Intent(this, AltraActivity.class);  
        i.putExtra("Stringa", "Sono una stringa");  
        i.putExtra("Intero", 137);  
        startActivity(i);  
    }  
}
```

Scrivere la classe AltraActivity che ha come file di layout "altro\_layout" contenente due TextView con identificativi "tvStringa" e "tvNumero" e far visualizzare in tali texview la stringa e l'intero passati dalla MainActivity tramite l'intent.

State sviluppando un app “DoppioGioco” che utilizza un frammento Tavolo (da gioco) che implementa opportunamente il tavolo da gioco (i dettagli del gioco non sono importanti). Il file di layout prevede il posizionamento di 2 frammenti Tavolo, in due contenitori i cui **id** sono rispettivamente **idContenitore1** e **idContenitore2**.

Completate il metodo `onCreate` creando i due frammenti Tavolo e posizionandoli nei rispettivi contenitori.

L'oggetto Tavolo prevede un metodo **`setAltroTavolo(Tavolo t)`** che prende in input un Tavolo **`t`**. Tale metodo serve perché ogni modifica fatta al tavolo che verrà fatta anche al tavolo **`t`**. In altre parole se si chiama **`t1.setAltroTavolo(t2)`**, ogni modifica fatta al tavolo **`t1`** verrà fatta anche al tavolo **`t2`** – questo è codificato nel frammento stesso e non bisogna preoccuparsi di come avviene.

Dopo aver creato e posizionato i frammenti, fare in modo che ognuno dei due tavoli da gioco siano uno la copia dell'altro.

NOTA: Non devi scrivere codice per la classe Tavolo, ma solo per il metodo `onCreate` dell'app.

```
public class Tavolo extends Fragment implements View.OnClickListener {  
    //PER QUESTA CLASSE NON BISOGNA SCRIVERE NIENTE  
    //occorre solo utilizzarla nel metodo onCreate  
}
```

```
public class MainActivity extends Activity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

A cosa serve il metodo **invalidate()** definito per un oggetto View? Descrivi una circostanza in cui occorre chiamare tale metodo e spiega che cosa succede come conseguenza della chiamata.

Come funziona la classe `AsyncTask<Type1, Type2, Type3>`?

Quali sono i 5 metodi che occorre (o che si possono) sovrascrivere? Fornisci una spiegazione.



Spiega come costruire un'animazione di una view, e descrivi la struttura di un file XML che definisce un'animazione.

A cosa serve la classe `Cursor`? Fornisce un esempio in cui è utile utilizzare tale classe.

Oltre alle Activity, Android ha altre 3 componenti fondamentali: Broadcast, Content Provider e Services.

Si fornisca una descrizione generale dei Services spiegando, in modo generico, i principi di funzionamento di un Service.