

Corso di Laboratorio di Programmazione per Sistemi Mobile e Tablet

Docente: Dr. Mauro Dragoni

Dipartimento di Ingegneria e Scienze dell'Informazione
Anno Accademico 2022/2023

in questa lezione...

- View e controlli utente
- Navigazione
 - Menu
 - ActionBar
 - Toolbar

Le View

- Svolgono due attività principali:
 - mostrano un aspetto grafico;
 - gestiscono eventi relativi all'interazione con l'utente
- Classificazione principale:
 - Layout
 - Controlli utente
 - AdapterView
- Ogni View e' dotata di un identificativo univoco all'interno del file di Layout dell'Activity che contiene la View stessa.

```
<TextView  
  android:id="@+id/nome"  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  . . .  
  . . .  
>
```

Le View

```
Button pulsante = (Button) findViewById(R.id.pulsante);  
pulsante.setOnClickListener(  
    new OnClickListener()  
    {  
        @Override  
        public void onClick(View v)  
        {  
            /*  
             * Gestione dell'evento di click su questa View  
             * */  
        }  
    }  
); // chiusura del metodo setOnClickListener
```

1. Recupero View.
2. Definizione dell'oggetto che gestisce la pressione del pulsante.
3. Registrazione del listener.

Le View

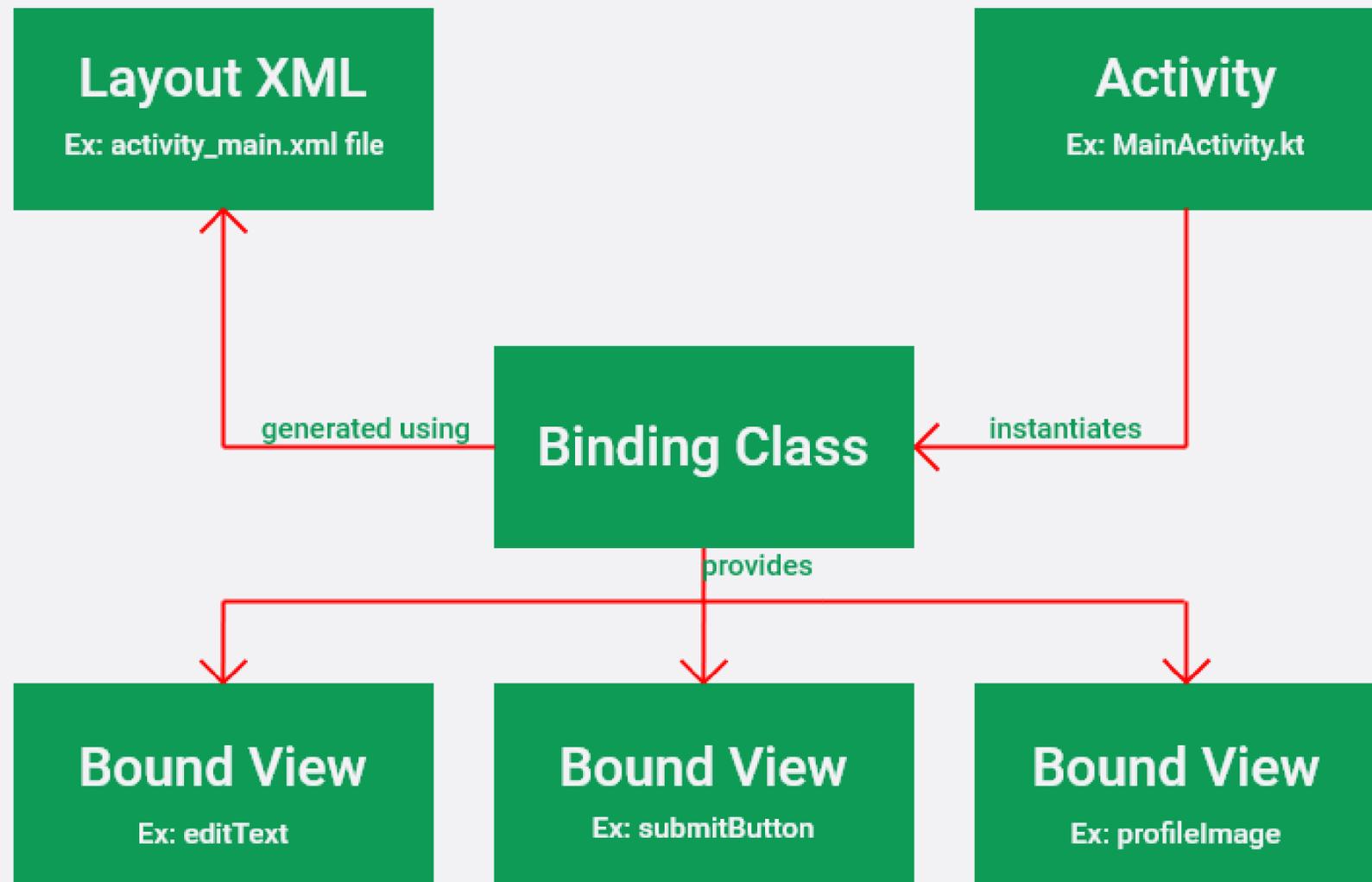
```
Button pulsante = (Button) findViewById(R.id.pulsante);  
pulsante.setOnClickListener(  
    new OnClickListener()  
    {  
        @Override  
        public void onClick(View v)  
        {  
            /*  
             * Gestione dell'evento di click su questa View  
             * */  
        }  
    }  
); // chiusura del metodo setOnClickListener
```

The diagram illustrates the code with three numbered callouts: 1 points to the variable 'pulsante', 2 points to the 'onClick' method body, and 3 points to the 'setOnClickListener' method call.

Altri esempi di eventi che possono essere gestiti:

- Cambiamento del focus
- Pressione di una chiave hardware
- Gestione del Touch
- Evento di LongClick

Le View - Binding



Abilitare il ViewBinding

```
android {  
    ...  
    buildFeatures {  
        viewBinding = true  
    }  
}
```

Le View - Binding

```
public class MainActivity extends AppCompatActivity {
    ActivityMainBinding activityMainBinding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        activityMainBinding = ActivityMainBinding.inflate(getLayoutInflater());

        View view = activityMainBinding.getRoot();
        setContentView(view);

        activityMainBinding.buttonSubmit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String str = activityMainBinding.editText.getText().toString();
                if(str.isEmpty()) {
                    Toast.makeText(this, "Please enter something", Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(this, "You entered " +
                        activityMainBinding.editText.getText().toString(),
                        Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}
```

Controlli utente

- Sottoclassi delle View

- Esempi di controlli utente
 - TextView
 - EditText
 - Button
 - CheckBox
 - Radio

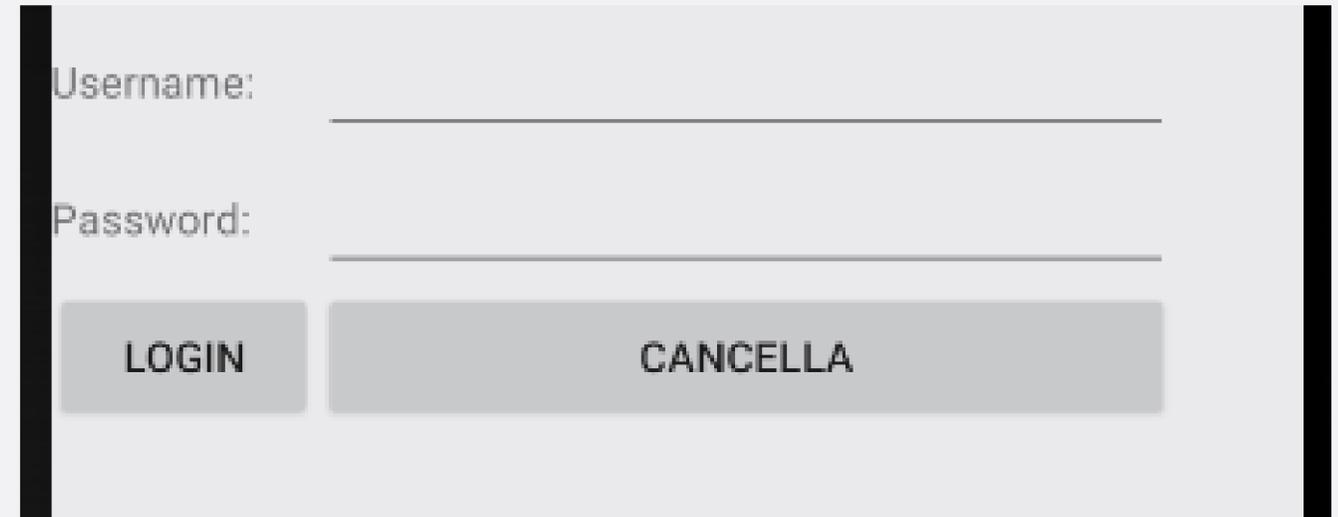
Controlli utente - utilizzo

- Nota: definizione dei Button:

```
<Button
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:onClick="login"
android:text="Login"/>
<Button
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:onClick="cancel"
android:text="Cancella"/>
```

- Significa che nel codice dell'Activity dovremo avere un metodo così definito:

```
public void login(View arg0)
```



Controlli utente - utilizzo



A screenshot of a login form. It features two input fields: 'Username:' and 'Password:'. Below the fields are two buttons: 'LOGIN' and 'CANCELLA'.

```
public void login(View v)
{
    EditText username = (EditText) findViewById(R.id.username);
    EditText password = (EditText) findViewById(R.id.password);
    String account_username = username.getText().toString();
    String account_password = accounts.get(username.getText().toString());
    . . .
    . . .
}
```

Menu

- Le tipologie di menu' piu' comuni sono due:
 - **Options menu:** menu principale dell'applicazione che contiene voci relative ad operazioni di interesse generale.
 - **Context menu:** menu invocabile su ogni singolo componente dell'interfaccia utente.
- I menu sono **risorse** della nostra applicazione ed il loro layout va definito nella sottocartella "res/menu".

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
  <item
    android:id="@+id/MENU_1"
    android:title="Nuova nota"/>
  <item
    android:id="@+id/MENU_2"
    android:title="Elenco note"/>
</menu>
```

Attivazione di un OptionsMenu

- **Override del metodo `onCreateOptionsMenu` all'interno dell'Activity**

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    // RECUPERA IL RIFERIMENTO DEL SERVIZIO DI INFLATER
    MenuInflater inflater = getMenuInflater();

    // AZIONE DI INFLATING, CIOE' DELLA CREAZIONE DEL MENU' VERA E PROPRIA
    inflater.inflate(R.menu.main, menu);

    // SE LA FUNZIONE RESTITUISCE true ALLORA IL MENU' DIVIENE ATTIVO
    return true;
}
```

- **Il metodo `onCreateOptionsMenu` viene invocato solamente UNA volta alla creazione dell'Activity.**

Gestione delle voci di un OptionsMenu

- **Override del metodo `onOptionsItemSelected` all'interno dell'Activity**

```
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    int id=item.getItemId();
    switch(id)
    {
    case R.id.MENU_1:
        /*
        Codice di gestione della voce MENU_1
        */
        break;
    case R.id.MENU_2:
        /*
        Codice di gestione della voce MENU_2
        */
    }
    return false;
}
```

Creazione di un ContextMenu

- I concetti di base espressi finora relativamente alla gestione di un OptionsMenu, sono validi anche per i ContextMenu. Infatti, le operazioni da effettuare per la definizione di un ContextMenu sono le seguenti:
 1. definizione della struttura del menu' contestuale dentro la cartella "res/menu"
 2. override del metodo `onCreateContextMenu` dove vengono eseguite le stesse operazioni del metodo `onCreateOptionsMenu`
 3. definizione delle risposte ai click sul menu nel metodo `onContextItemSelected`
 4. **registrazione dell'elemento al quale associare il menu' contestuale**

Registrazione di un ContextMenu

- Esecuzione del metodo `registerForContextMenu(View v)` solitamente nel metodo `onCreate` dell'Activity.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Button btn = (Button) findViewById(R.id.btnShow);
    registerForContextMenu(btn);
}
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.main_menu, menu);
}
```

Registrazione di un ContextMenu

- Esecuzione del metodo `registerForContextMenu(View v)` solitamente nel metodo `onCreate` dell'Activity.

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();

    if(v.getId() == R.id.first_button)
        inflater.inflate(R.menu.contextmenu1, menu);
    else if(v.getId() == R.id.second_button)
        inflater.inflate(R.menu.contextmenu2, menu);
}
}
```

E dove li mettiamo i menu'?

- ~~AppBar~~
- ActionBar
- Toolbar

ActionBar

- L'ActionBar puo' essere definita come una cornice programmabile destinata a contenere opzioni di navigazione generali relative all'intera applicazione.
- Esempi:
 - Actions (voci del OptionsMenu collocati nella ActionBar)
 - Navigazione tramite Tabs
 - Navigazione con menu' a tendina
 - Campi di ricerca
 - ...

ActionBar

- L'ActionBar puo' essere utilizzata solamente per versioni di Android superiori alla 3 (quindi API Level 11).
- Se si sta' programmando per versioni superiori ricordarsi di impostare all'interno del manifesto dell'applicazione i parametri **targetSdkVersion** o **minSdkVersion** almeno a 11.
- Disporre di un tema di tipo "olografico", quindi nel manifesto dell'applicazione il parametro **android:theme** deve essere impostato a **Theme.Holo** o una sua sottoclasse.

Comandi nella ActionBar



1. Icona dell'applicazione
2. Actions
3. Action Overflow

Comandi nella ActionBar



Icona e titolo dell'applicazione

```
<application  
android:allowBackup="true"  
android:icon="@drawable/ic_launcher"  
android:label="@string/app_name"
```

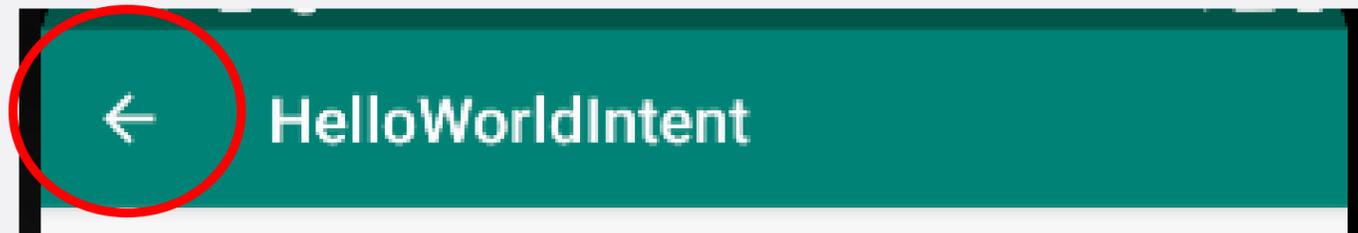
```
...  
...
```

Comandi nella ActionBar



Actions ed Action Overflow: utilizzo dell'attributo ***showsAsAction*** nella definizione del layout del menu'.

ActionBar – navigare all'indietro



- Per abilitare la navigazione all'indietro e' sufficiente implementare i seguenti due passi nel codice della Activity per la quale si vuole inserire la freccia:

1. Modifica del manifesto:

```
<activity android:name=".SecondActivity"  
android:label="Seconda Activity"  
android:parentActivityName=".MainActivity"/>
```

2. Inserire nel metodo onCreate:

```
getActionBar().setDisplayHomeAsUpEnabled(true);
```

oppure

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

Comandi nella ActionBar

- ifRoom***: mostra le icone sull'ActionBar compatibilmente con lo spazio disponibile
- never***: non mostra le voci del menu sull'ActionBar ma solo nel OptionsMenu
- always***: il layout viene forzato a mostrare le voci in ActionBar. E' un valore sempre sconsigliabile, meglio optare per ***ifRoom***
- withText***: Oltre all'icona viene mostrato in ActionBar anche il testo, solitamente collegato all'attributo *android:title*

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
<item
android:id="@+id/MENU_1"
android:showAsAction="ifRoom"
android:title="Nuovo"
android:icon="@android:drawable/ic_menu_add"/>
<item
android:id="@+id/MENU_2"
android:showAsAction="ifRoom|withText"
android:title="Elenco"
android:icon="@android:drawable/ic_menu_agenda"/>
</menu>
```

Toolbar

- Discende dalla ActionBar con alcune caratteristiche:
 - puo' essere inserita dove si vuole nel layout della propria Activity
 - di fatto e' un oggetto di tipo ViewGroup
 - **raccomandata** per contenere, oltre a strumenti di navigazione, anche altre view (es. casella di testo)
 - **IMPORTANTE:** bisogna esplicitare nel tema della Activity la scomparsa della ActionBar

Approfondimenti

Sul sito ufficiale:

- [Menus](#)
- [App bar](#)

Altri riferimenti

- [Defining the ActionBar](#)
- [Using the App Toolbar](#)