

Activiteiten

Klasse

Een app moet minstens 1 activiteit bevatten. Deze activiteit ("MainActivity" in het voorbeeld hieronder) wordt afgeleid van de basisklasse "Activity".

```

package android.vb01;

import android.app.Activity;
import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}

```

Afgeleide klassen

Een activiteit kan bvb. van 1 van de volgende basisklassen afgeleid worden: TabActivity, ListActivity, ExpandableListActivity, ActionBarActivity, FragmentActivity. Elke activiteit van deze basisklassen heeft specifieke eigenschappen en methodes.

Lifecycle functies

Een activiteit moet minstens 1 functie (methode) bevatten, de "onCreate()" functie.

In de "onCreate()" functie wordt de koppeling met de layout ingesteld en worden alle initialisaties geprogrammeerd die moeten uitgevoerd worden als de app opgestart wordt.

Op de website van Android vind je uitgebreide informatie over de levenscyclus van een activiteit.

(<http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle>)

Als een app wordt opgestart wordt eerst de "onCreate()" functie uitgevoerd. Daarna kunnen de functies "onStart()" en "onResume()" uitgevoerd worden. Eens een app is opgestart blijft ze aanwezig onder het Android besturingssysteem totdat de app wordt stopgezet. De app kan niet meer zichtbaar zijn voor de gebruiker maar is nog wel actief. Dit kan je zien als je de lijst met recente apps opent op het Android toestel.

De "onPause()" functie wordt uitgevoerd juist voordat de app naar de achtergrond verdwijnt. In deze functie kan je dan best de code opnemen om nog niet bewaarde gegevens weg te schrijven.

Als de app volledig wordt afgesloten door de finish() functie aan te roepen in de code, dan worden de functies "onStop()" en "onDestroy()" uitgevoerd.

Een app wordt opnieuw gestart als bvb. de oriëntatie van het Android toestel wijzigt. Er wordt een nieuwe "instantie" van de activiteit gestart. Je kan de vorige situatie vastleggen in de "savedInstanceState bundle".

Layout koppeling

In de onCreate() functie van de activiteit wordt met de functie (methode) setContentView(View) de scherm layout opgehaald uit de aangeduide View en getoond op het scherm.

Het argument van de functie kan een verwijzing bevatten naar een XML layout bestand of een andere View (bvb. een ListView, een LinearLayout, een ScrollView).

Deze regel moet aanwezig zijn als je een XML layout gebruikt. Er wordt een gegenereerd nummer gebruikt voor de layout (R.layout.). De naam verwijst naar de bestandsnaam.

```
setContentView(R.layout.activity_main);
```

Als je liever Java code schrijft om de layout te maken dan kan je de hoofd View gebruiken in de setContentView() functie. Een voorbeeld:

```
// main layout
llmain = new LinearLayout(this);
llmain.setId(1);
llmain.setOrientation(LinearLayout.VERTICAL);
llmain.setLayoutParams(new LayoutParams(LayoutParams.FILL_PARENT,
                                         LayoutParams.FILL_PARENT));

// content frame
flcontent = new FrameLayout(this);
flcontent.setId(2);
flcontent.setLayoutParams(new LayoutParams(LayoutParams.MATCH_PARENT,
                                             LayoutParams.MATCH_PARENT));
flcontent.setBackgroundColor(Color.parseColor("#FFFFFF"));
// logo
ivlogo = new ImageView(this);
ivlogo.setImageResource(R.drawable.gradebook_logo2);
ivlogo.setLayoutParams(new LayoutParams(900, 678));
ivlogo.setPadding(150, 150, 150, 150);
flcontent.addView(ivlogo);
llmain.addView(flcontent);
setContentView(llmain);
```

Met de functie "addView()" kan je andere Views in een omsluitende View steken. In het voorbeeld is een ImageView toegevoegd aan een FrameLayout en de FrameLayout is toegevoegd aan de LinearLayout. De LinearLayout is gebruikt in de "setContentView()" functie.

Een standaard gegenereerde XML layout ziet er als volgt uit:

```
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  android:paddingBottom="@dimen/activity_vertical_margin"
  tools:context=".MainActivity" >

  <TextView android:text="@string/hello_world"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

</RelativeLayout>
```

De plaats van de View in het bestand bepaalt tot welke hoofd View de View behoort. In het voorbeeld zit de TextView in een RelativeLayout.

Menu koppeling

Als je een blanco activiteit gekozen hebt bij het aanmaken van een nieuwe app dan wordt de onderstaande code gegenereerd.

```
package android.vb01;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        if (id == R.id.action_settings) {
            return true;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

In het voorbeeld zijn 2 menu functies aanwezig: "onCreateOptionsMenu()" en "onOptionsItemSelected()".

4/4 | Android - activiteiten

Om de menu te kunnen tonen gebruik je een "menu inflater" (getMenuInflater().inflate()). Deze functie maakt een "Menu" object aan op basis van de XML gegevens die in het "menu_main.xml" bestand staan.

```
<menu 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"
      tools:context=".MainActivity">
  <item android:id="@+id/action_settings"
        android:title="@string/action_settings"
        android:orderInCategory="100"
        app:showAsAction="never" />
</menu>
```

Standaard is er 1 menu item aanwezig waarmee een nog te maken activiteit "Settings" kan uitgevoerd worden.

In een "Virtual Device" kan je de menu oproepen door op de F2 toets te drukken als de app uitgevoerd wordt. Soms kan je ook de hardware knop gebruiken als die geactiveerd is in het virtual device.

De menu kan in de "ActionBar" verschijnen of rechtsonder op het scherm.

