Föreläsning 2

- Activity
- Fragment
- Referenser till Views
- Händelsehantering
- Workshop
  Design och implementering av program med
  * Activity för att initiera program och hålla Fragment-objekt
  * UI och händelsehantering i Fragment-objekt
  * Controller-klass med logik

Viktiga UI-klasser

View
Superklass för alla UI-komponenter, synliga (controls / widgets) och osynliga som layouts.

ViewGroup
Subklass till View. Kan innehålla flera Views. Ett exempel är LinearLayout

Fragment
Ungefär som en JPanel i java. Innehåller ett eget UI med komponenter och händelsehantering.

Activity
Ungefär som en JFrame i java. Representerar ett fönster på skärmen.

Activity

- Ett fönster i applikationen
- Applikationen startar i en Activity
- Innehåller UI ofta bestående av Fragment
- Innehåller referens till resurser
- Klassen ska ärva Activity

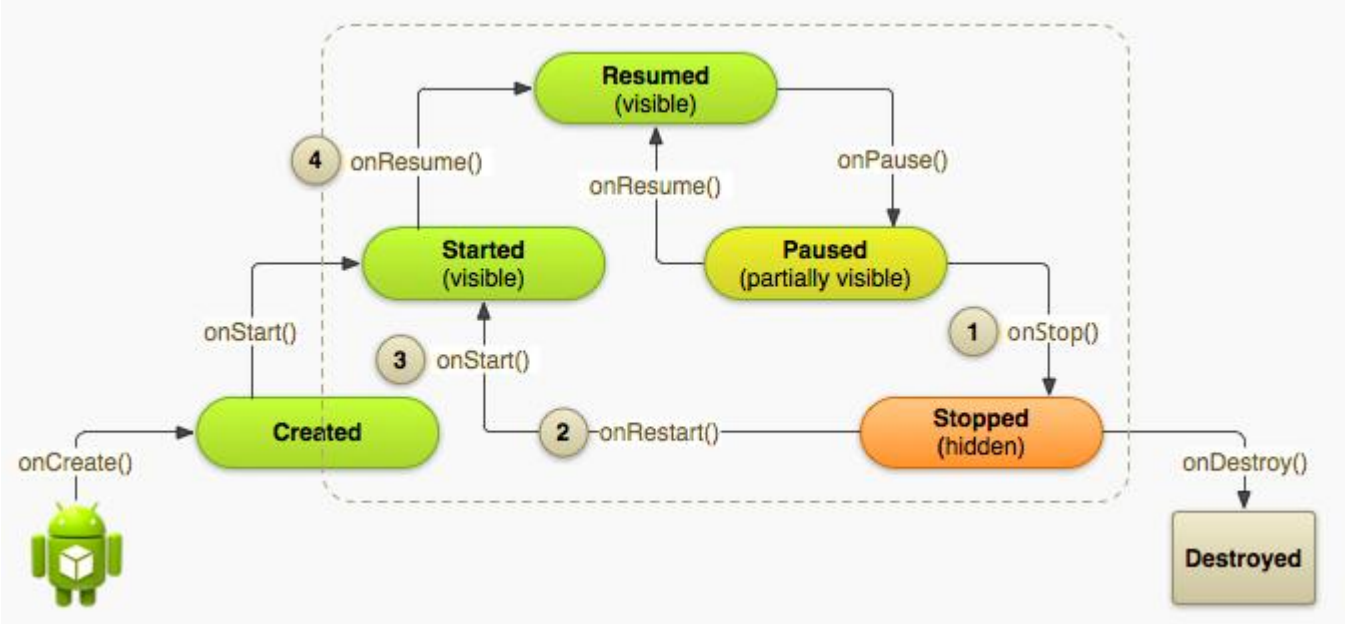
```
public class App extends Activity {

}
```

- En Aktivity ska registreras i Manifestet

```
:
<application ….>
    :
    <activity …>
        <intent-filter>
        :
        </intent-filter>
    </activity>
    :
</application>
```

Activity - lifecycle



En Activity kan vara i tre olika states:

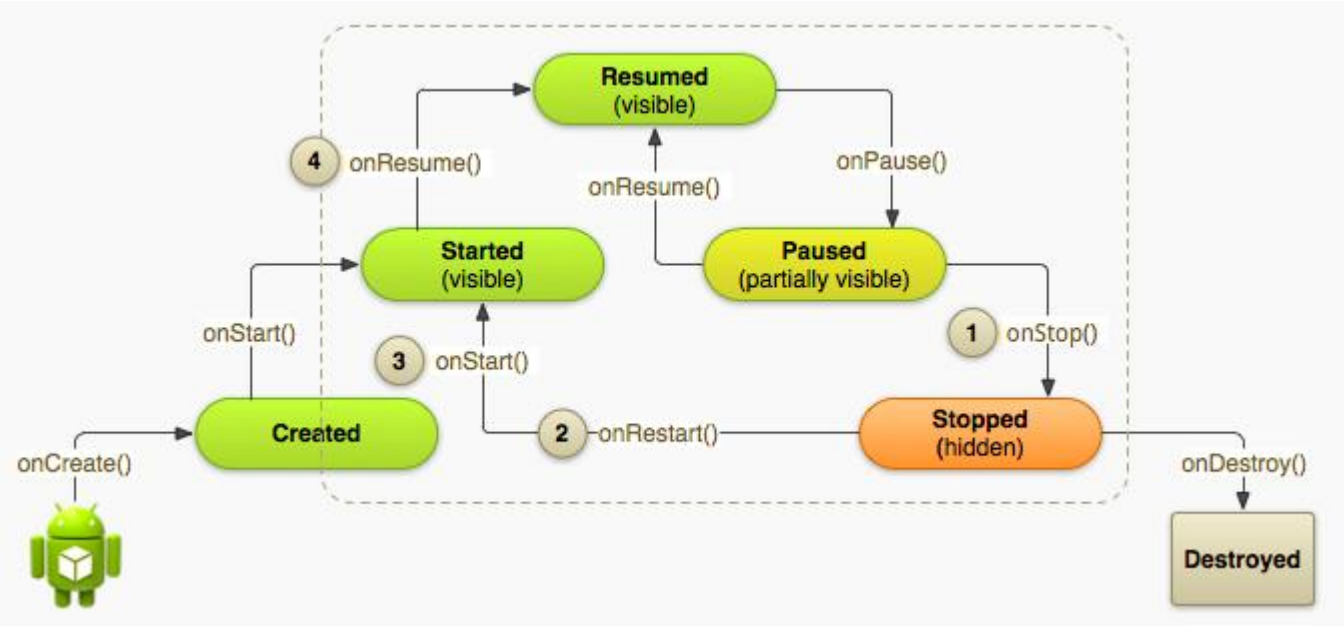
- **Active** - synlig
  onCreate(), onStart() och onResume() är anropade
- **Paused** – delvis synlig
  Efter Active, då onPause() anropats
- **Stopped** – osynlig
  Efter Paused, då onStop() anropats

Activity - lifecycle

onCreate(Bundle savedInstance)
Initialisera Activity och skapa UI
Eventuellt aktivera resurser som avaktiverats vid avbrott av Activity (savedInstance!=null)

onRestart(), onStart()
Aktivera resurser som avaktiverats av onStop()

onResume()
Aktivera resurser som avaktiverats i onPause()



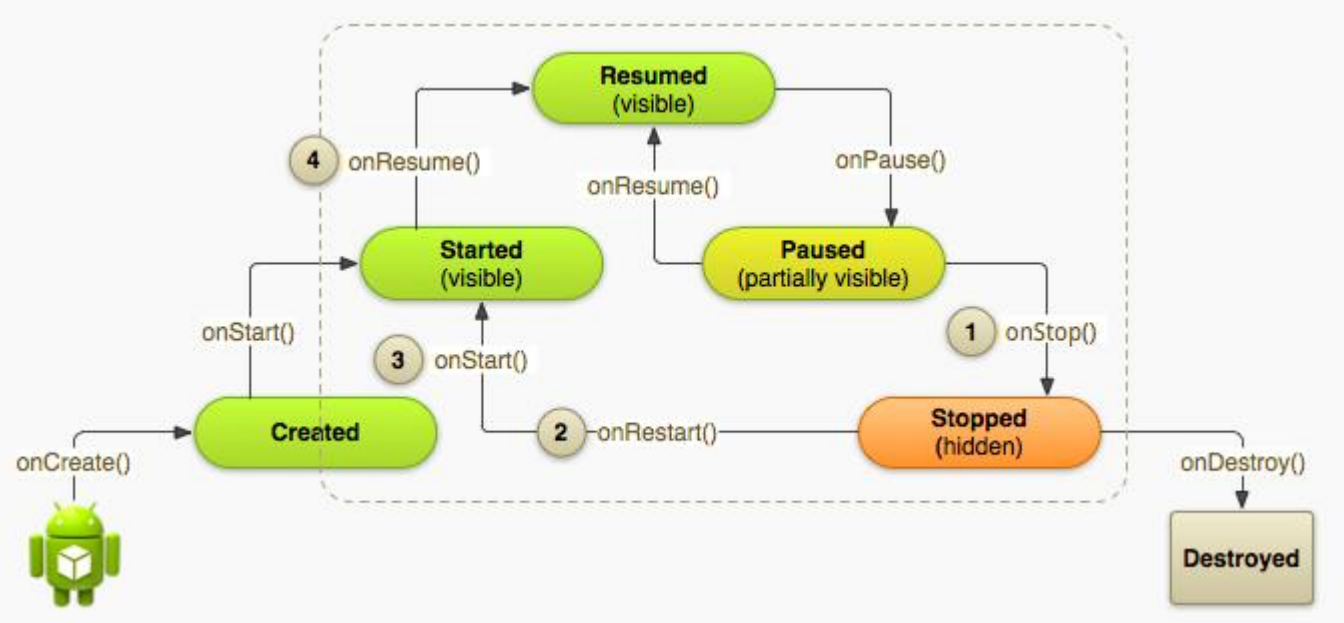
Activity - lifecycle

onSaveInstanceState(Bundle savedInstance)
Spara undan UI state, levereras till bl.a. onCreate

onPause()
Avaktivera resurser som ej behövs. Spara nödvändig data. Snabb exekvering krävs!

onStop()
Aktivera resurser som ej avaktiverats i onPause

onDestroy()
slutstädning



Activity – UI, Views och händelser

- Ett fönster i applikationen
- Applikationen startar i en Aktivity
- Innehåller UI ofta bestående av Fragment

```
public class AnActivity extends Activity {
    private Button btnHello;
    private TextView tvInfo;

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

    private void initializeComponents() {
        tvInfo = (TextView)findViewById(R.id.tvInfo);
        btnHello = (Button)findViewById(R.id.btnHello);
    }
```

- onCreate, onRestart, onStart och onResume inleds med ett anrop till superklassens motsvarighet av metoden.
- onSaveInstanceState, onPause, onStop och onDestroy avslutas med ett anrop till superklassen motsvarighet av metod.

Activity – UI, Views och händelser

- Ett fönster i applikationen
- Applikationen startar i en Aktivity
- Innehåller UI ofta bestående av Fragment

```
public class AnActivity extends Activity {
    :
    private void registerListeners() {
        btnHello.setOnClickListener(new BL());
    }

    private class BL implements OnClickListener {
        int index=0;
        String[] info = {"Hello","GoodBye"};

        public void onClick(View v) {
            index = (index+1) % 2;  // index = 0,1,0,1,0,1 osv
            tvInfo.setText(info[index]);
        }
    }
}
```

Fragment

- Som en panel i java
- Har eget UI och händelsehantering
- Bör vara väl inkapslad
- Bör ej innehålla logik
- Har tillgång till referens till sin Activity, metoden getActivity()
- Klassen ska ärva Fragment

```
public class Frag extends Fragment {

}
```

Fragment - lifecycle

En Fragment har metoder som motsvarar metoderna i Activity-klassen. Dessutom tillkommer det några metoder. OnCreateView finns som regel i alla Fragment (ej i fragments som inte är synliga).

- **onAttach(Activity activity)**
  eventuellt lagra referens till activity-klassen
- **onCreate(Bundle savedInstanceState)**
  initialisera
- **public View onCreateView(LayoutInflater inflater,**
  ViewGroup container,
  Bundle savedInstanceState)
  skapa UI, Metod som finns i synliga fragments
- **onStart()** och **onResume()**
  Eventuellt aktivera resurser

Superklassens motsvarighet anropas i början av metoderna. Gäller ej onCreateView!

Fragment - lifecycle

En Fragment har metoder som motsvarar metoderna i Activity-klassen. Dessutom tillkommer det några metoder. OnCreateView finns som regel i alla Fragment (ej i fragments som inte är synliga).

- **onPause()**
  Avaktivera resurser
- **onSaveInstanceState(Bundle savedInstance)**
  Spara undan UI state, levereras till bl.a. onCreate och OnCreateView
- **onStop()**
  Aktivera resurser som ej avaktiverats i onPause
- **onDestroy()**
  slutstädning
- **onDetach()**

Superklassens motsvarighet anropas i slutet av metoderna.

Fragment – i en Activity

Fragment kan placeras i en layout med fragment-taggen. Så här kan Activity-klassens layoutfil se ut:

```
<LinearLayout
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:orientation="vertical" >

    <fragment
        android:id="@+id/viewer_fragment"
        android:name="se.mah.tsroax.staticfragment.ViewerFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <fragment
        android:id="@+id/input_fragment"
        android:name="se.mah.tsroax.staticfragment.InputFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

Fragment – UI, Views och händelser

- En del av ett fönster i applikationen
- Innehåller UI och händelsehantering

```
public class AFragment extends Fragment {
    private Button btnHello;
    private TextView tvInfo;

    public View onCreateView(LayoutInflater inflater,
                             ViewGroup container,
                             Bundle savedInstanceState) {
        View view =  inflater.inflate(R.layout.viewer,
                                      container, false);
        initializeComponents(view);
        initializeResources();
        return view;
    }

    private void initializeComponents(View view) {
        tvInfo = (TextView)view.findViewById(R.id.tvInfo);
        btnHello = (Button)view.findViewById(R.id.btnHello);
    }
```

Fragment – UI, Views och händelser

- En del av ett fönster i applikationen
- Innehåller UI och händelsehantering

```
public class AFragment extends Fragment {
    private Controller controller;
    :
    private void registerListeners() {
        btnHello.setOnClickListener(new BL());
    }

    public void setInfo(String str) {
        tvInfo.setText(str);
    }

    private class BL implements OnClickListener {
        public void onClick(View v) {
            controller.newInfo();
        }
    }
}
```

FragmentManager

Activityklassen kan hantera sina Fragment med FragmentManager-objektet. T.ex. går det att få referens till olika Fragment som finns i Activityn.

```
FragmentManager fm = getFragmentManager();
ViewerFragment viewer =
  (ViewerFragment)fm.findFragmentById(R.id.viewer_fragment);
InputFragment input =
  (InputFragment)fm.findFragmentById(R.id.input_fragment);
```

Workshop

Ett sten, sax och påse-spel bestående av

- En Activity-klass (MainActivity)
- Två Fragment-klasser (InputFragment och ViewerFragment)
- En controller-klass (RPSController)
- En datorspelare-klass (RPSPlayer)