

Android Undervisningsmateriale

Indhold

Android Undervisningsmateriale	1
Android Undervisningsmateriale	6
Platform Versioner 2023	6
1. Installation af Android og Eclipse	8
Java JDK (Java Development Kit)	8
Nu om dage er Eclipse og Android ADT med når Android SDK hentes	8
Eclipse	8
Android SDK (Android Software Development Kit).....	8
Android ADT (Android Development Tools)	9
Android Emulator	9
Nyt Android projekt.....	9
Android Devices og ADB	10
ADB Shell	12
Få emulatoren til at åbne hurtigere i Windows	13
Vis Android enheden på PC-skærmen	14
Nyt Projekt med Eclipse.....	15
Problemer.....	16
Flyt et projekt til andet workspace	16
Android Studio	18
2. Introduktion	19
Klassen R	19
3. Aktivitets livscyklus og udskrift i Logcat.....	21
Udskriv i Logcat når eventhåndlere kaldes.....	21
Log niveauer	24
LogCat i nyere Android Studio	27
4. Brugerflade.....	30
En Button oprettet i Java	30
Et layout oprettet i Java.....	30

Id oprettet i ressourcer.....	31
Brugerflade i xml	32
Øvelse Brugerflade	33
5. Eventhandling på bruger-interface.....	34
Brugerflade til afprøvning af eventhandling.....	34
Eventhandler internt i Activity-klassen.....	35
Øvelser Eventhandler internt i Activity-klassen.....	36
Eventhandler i extern klasse.....	36
Øvelse Eventhandler i extern klasse	37
Læs fra og skriv til tekstboks.....	37
Eventhandler i anonym indre klasse.....	37
Øvelse Eventhandler i anonym indre klasse	39
Eventhandler i Indre klasse	39
Øvelse Eventhandler i Indre klasse	39
Eventhandler angivet i layout.....	40
Øvelse Eventhandler angivet i layout	41
Flere knapper med samme eventhandler.....	41
Øvelser	44
Opgave regnemaskine – forskellige handlere for hver knap	44
Opgave regnemaskine – samme handler for alle knapper	44
6. Ressourcer.....	46
Demo af Ressourcer	46
Skift sprog.....	51
Billeder i forskellig opløsning.....	54
Layout til landscape og portrait.....	55
Øvelse ressourcer	56
7. Layout.....	57
En enkelt knap på skærmen	57
LinearLayout.....	58
Portrait og landscape layout ved vending af skærm.....	59
Blokering af vending af skærm	60
Skift af layout restarter aktiviteten.....	60
LinearLayout med dynamisk indhold.....	60

Blanding af statisk og dynamisk layout.....	61
Nestede linearlayouts.....	62
FrameLayout	64
RelativeLayout.....	65
Positioner relativ til container	66
Positioner relativ til andre widgets.....	66
Evaluering af id.....	66
TableLayout.....	67
GridLayout.....	68
ScrollView.....	71
Øvelser	73
Opgave Regnemaskine med TableLayout og RadioButtons.....	73
Regnemaskine med tal-knapper i GridLayout.....	73
8. Dialog, Toast og Snackbar.....	74
Dialog	74
Toast.....	79
Snackbar	79
9. Widgets og Views	81
Gravity.....	82
Button	84
RadioButton og RadioGroup.....	85
Øvelse.....	89
Opgave Regnemaskine med TableLayout og RadioButtons	91
10. ListView	92
10.1 ListLayout og Adaptere.....	92
Simple Lister.....	92
ListView med ListActivity	92
ListView med almindelig Activity.....	94
ListView med radiobuttons	95
ListView med checkbuttons.....	96
Spin Control / Drop down list	98
.....	98
Opgave Regnere med operatorer i Spinner.....	101

Opgave Spinner med objekter.....	101
Opgave To Do liste med ToDo-objekter i Spinneren.....	101
10.2 Customized ListView.....	103
11. Intents og Activities	107
Vis web-side i browser.....	107
Oprettelse af kalender-element	109
Aktivitet, der kalder en anden aktivitet.....	110
activity_main.xml	110
activity_second.xml.....	111
Aktivitet der åbner en af flere aktiviteter og får data tilbage	114
Deprecated startActivityForResult	119
En start-aktivitet, der vises kort og går derefter videre til en menu-aktivitet, hvor der kan vælges et program(aktivitet) der ønskes udført	121
Øvelser med flere aktiviteter.....	124
Øvelse To-Do Liste	125
12. Fragmenter.....	126
En static Fragment.....	127
To Fragmenter udskiftes i placeholder	130
13. Grafik	136
13.1 Grafik og onTouchEvent	136
13.2 Gentagent kald af onDraw og autonome bevægelser	140
14. Asynkrone programmer.....	145
Udførelse af en asynkron task	145
Gentagent udførelse af task	147
Blokere hovedtråden.....	148
Blokering i en ny tråd som opdaterer GUI	149
Blokering i ny tråd og overgivelse til hovedtråd at opdatere GUI.....	150
AsyncTask	152
AsyncTask med opdatering af GUI undervejs	154
Problemer med AsyncTask	155
15. Menuer	156
Menu oprettet i XML	156
Menuer oprettet i Java	158

16.	Sensors	169
17.	Persistent Data	172
	En aktivitets livscyklus	173
	Bundle	175
	Eksempel	175
	Opgave	177
	Persistence	177
	Shared Preferences	177
	Eksempel	178
	Et objekt gemt i en string i Shared Persistence	180
	Internal Storage.....	181
	External Storage	183
	Eksempel	183
18.	SQLite Databases.....	184
	Eksempel	184
	Network Connection	186
	Android Backup Service	186
	Titanium Backup.....	186
	Andre emner i persistent data.....	187
	Content Provider	187
	Statiske filer i applikationen	187
19.	Kontakter og sikkerhed i mobiltelefon	188
	Bede om brugerens tilladelse til at programmet tilgår kontakter	191
20.	Sensorer	200
	Udskriv de tre Accelerometer-værdier.....	201
	Tråde	205
21.	Netværk.....	205
	Java.....	205
	Android Studio og Java	206
	Noter	207
	Java versus C#	208
	Flytte et project i Android Studio	210
	Noter	210

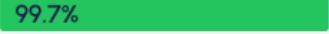
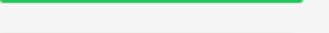
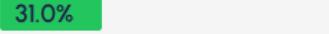
Fejl.....	211
5 Answers.....	211
Resource is out of sync with the filesystem	212
Answer	212
Android sdk main.out.xml parsing error.....	213
R cannot be resolved.....	214

Android Undervisningsmateriale

Platform Versioner 2023

Følgende diagram indeholder den kumulative fordeling af hver Android-version. Det viser procentdelen af enheder på markedet, som vil kunne køre en app, hvis der vælges den pågældende version.

Cumulative Distribution

Platform Version	API	Distribution
Android 4.4 (KitKat)	19	 99.7%
Android 5.0 (Lollipop)	21	 99.2%
Android 5.1 (Lollipop)	22	 98.9%
Android 6.0 (Marshmallow)	23	 97.4%
Android 7.0 (Nougat)	24	 95.1%
Android 7.1 (Nougat)	25	 93.6%
Android 8.0 (Oreo)	26	 92.1%
Android 8.1 (Oreo)	27	 89.9%
Android 9.0 (Pie)	28	 83.8%
Android 10.0 (Q)	29	 71.9%
Android 11.0 (R)	30	 54.1%
Android 12.0 (S)	31	 31.0%
Android 13.0 (T)	33	 14.7%

1. Installation af Android og Eclipse

Java JDK (Java Development Kit)

Download jdk7 fra

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1637583.html>

Installer Java.

Gå til installationsmappen og kopier stien til bin mappen f.eks. C:\Programmer\Java\jdk1.7.0\bin og tilføj denne til de eksisterende indgange i miljø-variablen Path adskilt af semikolon.

(Start | Højreklik på Denne Computer | Egenskaber | Avanceret | Miljøvariabler | Path – Edit)

Nu om dage er Eclipse og Android ADT med når Android SDK hentes

Download fra

<http://developer.android.com/sdk/index.html>

adt-bundle-windows-x86_64-20140321.zip eller adt-bundle-windows-x86-20140321.zip. Udpak, hvorved der kommer mapperne sdk og eclipse. Gå ind i mappen eclipse og start programmet eclipse.exe.

Så er man kørende og der kan hoppes over følgende 3 punkter, Eclipse, Android SDK og Android ADT.

Eclipse

Download Eclipse fra

<http://www.eclipse.org/downloads/>

Udpak Eclipse i den ønskede mappe f.eks. C:\AndroidEclipse\Eclipse\

Jeg havde problemer med for lange fil-mappe-sti-navne og valgte så at installere en ældre version.

Android SDK (Android Software Development Kit)

Download Android installer <http://developer.android.com/sdk/index.html>

og udfør denne. Vælg evt. at placere installationen i mappen ved siden af Eclipse, f.eks.

C:\AndroidEclipse\Android\

Når der er udpakket bliver Android SDK Manager automatisk startet op. Her kan der vælges at installere yderligere packages for de forskellige versioner af Android mm. Sørg for at Android SDK Tools og Android SDK Platform Tools bliver installeret.

Gå til mappen for installationen af Android SDK og ind i mappen tools f.eks. C:\AndroidEclipse\Android\tools. Kopier stien og indsæt også denne i Path miljø-variablen adskilt af semikolon. Gør det samme med C:\AndroidEclipse\Android\platform-tools

Start Eclipse.

Vælg workspace. Der kan vælges det foreslæde eller browses til en anden mappe. Der kan vælges en hvilken som helst mappe til Workspace. Et Workspace er en mappe hvor systemet holder styr på de oprettede projekter. Så et projekt kan ikke bare flyttes til et andet Workspace.

Vælg menu Windows | Preferences | Android og i SDK Location browse til mappen for installationen af Android SDK f.eks. C:\AndroidEclipse\Android\.

Tryk OK.

Android ADT (Android Development Tools)

Forbliv i Eclipse.

Vælg menuen Help | Install New Software

I Work With indtast: <https://dl-ssl.google.com/android/eclipse/>

Klik checkbox for Developer Tools.

Tryk Next, Accept og Finish.

Android Emulator

I Eclipse vælges Window | AVD Manager.

Tryk New.

Vælg et Navn f.eks. JansEmulator, Target f.eks. Android 2.3.3. – API Level 10 og Size 512 MB.

Tryk Create AVD.

Og evt. Start.

Nyt Android projekt

Vælg File | New ... Project ... Android Project.

(Det hele varierer noget i rækkefølge alt efter versionen af Eclipse)

Indtast Projekt Name og Tryk Next.

Vælg Target og tryk Next.

Vælg Application Name, det samme som Project Name.

Vælg packet – minimum to navne adskilt af punktum.

Vælg navn på start-Activity-klassen.

Vælg Minimum SDK svarende til Target version.

Tryk Finish og udfør programmet.

Android Devices og ADB

Hvis man ønsker at afprøve programmerne på en riktig Running Android Device i stedet for en Android Virtual Device, skal der installeres det korrekte software. Et er at der er en USB-driver, der kan se device'en som fil-lager. Noget andet er at Eclipse kan se enheden.

Det er egentlig ADB (Android Debug Bridge) der skal kunne se enheden.

Programmet adb.exe ligger i SDK platform-tools og ved at udføre følgende kommando i en command-prompt, kan man liste hvilke enheder, den kan se.

```
C:\>adb.exe devices
```

Jeg måtte installere Samsung Kies programmet for at få den rigtige driver til Samsung-enheden, selv om computeren kunne se enhedens filesystem. Kies installerer en driver, som nok kan findes og installeres uden at installere Kies.

Når man forsøger at udføre et Android program på en ny enhed, kan der komme en fejlmeddeelse at man skal udføre

```
C:\>adb.exe uninstall <package-navn>
```

Placering af adb.exe

```
C:\Users\jaj\AppData\Local\Android\Sdk\platform-tools\adb.exe
```

Hvis der kommer en fejl som denne:

```
Unable to open connection to ADB server: java.io.IOException: Can't find adb server on port 5037, IPv4 attempt:
```

Kan det være nødvendigt at restarte adb.

Adb-serveren kan stoppes og startes med følgende kommandoer.

```
C:\> adb kill-server
```

C:\>adb start-server

ADB Shell

Man kan også bruge ADB til at udføre Linux-shell kommandoer på Android enheden

```
C:\>adb.exe shell  
shell@android:/ $ ls  
...  
sdcard  
...  
  
shell@android:/ $ cd sdcard  
  
shell@android:/sdcard $ ls  
  
...  
Pictures  
...  
  
shell@android:/sdcard $ cd Pictures  
  
shell@android:/sdcard/Pictures $ ls  
  
...  
Screenshots  
...  
  
shell@android:/sdcard/Pictures $ cd Screenshots  
  
shell@android:/sdcard/Pictures/Screenshots $ ls  
  
...  
Screenshot_2000-02-20-15-07-27.png  
...
```

Få emulatoren til at åbne hurtigere i Windows

- Install "Intel x86 Emulator Accelerator (HAXM)" => SDK-Manager/Extras
- Install "Intel x86 Atom System Images" => SDK-Manager/Android 2.3.3
- Go to the Android SDK root folder and navigate to
extras\intel\Hardware_Accelerated_Execution_Manager.
Execute file IntelHaxm.exe to install.
- Create AVD with "Intel atom x86" CPU/ABI
- Run emulator and check in console that HAXM running

Packages

Name	API	Rev.
Android 1.5 (API 5)		
Extras		
Android Support Library	10	
Google AdMob Ads SDK	8	
Google Analytics SDK	2	
Google Cloud Messaging for Android Library	3	
Google Play services	3	
Google Play APK Expansion Library	2	
Google Play Billing Library	2	
Google Play Licensing Library	2	
Google USB Driver	7	
Google Web Driver	2	
Intel x86 Emulator Accelerator (HAXM)	2	

Show: Updates/New Installed Obsolete Select [New](#)

Sort by: API level Repository [Deselect All](#)

Done loading packages.

Also don't forget install this one

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Android 4.3 (API18)			
<input type="checkbox"/>	<input type="checkbox"/>	Documentation for Android SDK	18	2	<input type="checkbox"/> Not installed
<input type="checkbox"/>	<input type="checkbox"/>	SDK Platform	18	2	<input checked="" type="checkbox"/> Installed
<input type="checkbox"/>	<input type="checkbox"/>	Samples for SDK	18	1	<input type="checkbox"/> Not installed
<input type="checkbox"/>	<input type="checkbox"/>	ARM EABI v7a System Image	18	2	<input checked="" type="checkbox"/> Installed
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Intel x86 Atom System Image	18	1	<input checked="" type="checkbox"/> Installed
<input type="checkbox"/>	<input type="checkbox"/>	Google APIs	18	3	<input type="checkbox"/> Not installed
<input type="checkbox"/>	<input type="checkbox"/>	Sources for Android SDK	18	1	<input type="checkbox"/> Not installed

P.S. during AVD creation add emulation memory: Hardware/New/Device ram size/set up value 512 or more

Vis Android enheden på PC-skærmen

SCRPCY (Screen Copy) fra scrcpy.org, er vist den bedste i 2023.

Droid-at-screen. Opdaterer lidt langsomt, så noget der bevæger sig på skærmen bliver noget hakket.

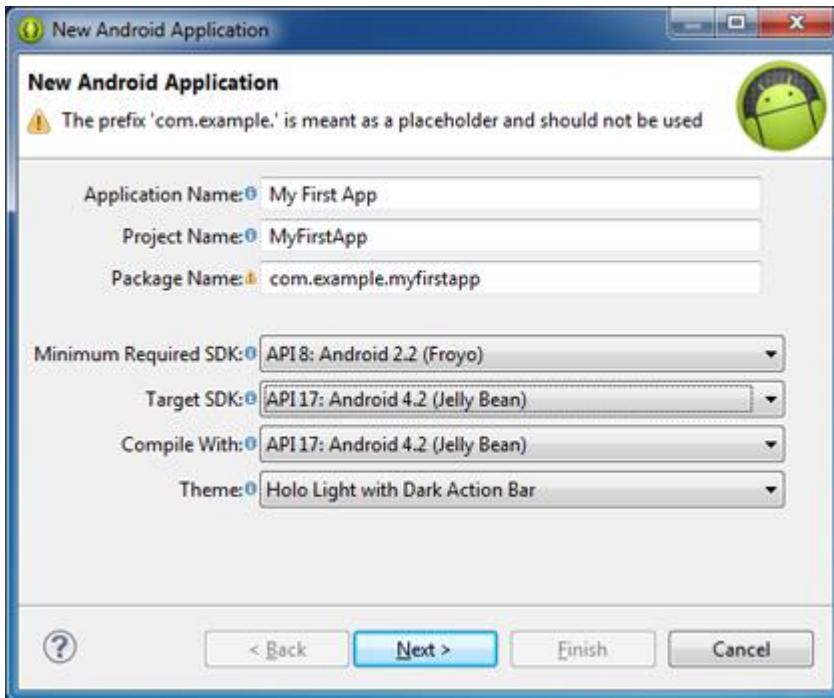
<http://droid-at-screen.ribomation.com/download/>

Eller brug TeamViewer, den opdaterer noget hurtigere.

Eller VySor <https://www.vysor.io/download/>

Nyt Projekt med Eclipse

Klik på New  i toolbaren eller New i File-menuen, vælg mappen Android, og vælg **Android Application Project**.



Forklaringerne er hentet fra <http://developer.android.com/training/basics/firstapp/creating-project.html#Eclipse>.

- **Application Name** is the app name that appears to users. For this project, use "My First App."
- **Project Name** is the name of your project directory and the name visible in Eclipse.
- **Package Name** is the package namespace for your app (following the same rules as packages in the Java programming language). Your package name must be unique across all packages installed on the Android system. For this reason, it's generally best if you use a name that begins with the reverse domain name of your organization or publisher entity. For this project, you can use something like "com.example.myfirstapp." However, you cannot publish your app on Google Play using the "com.example" namespace.
- **Minimum Required SDK** is the lowest version of Android that your app supports, indicated using the [API level](#). To support as many devices as possible, you should set this to the lowest version available that allows your app to provide its core feature set. If any feature of your app is possible only on newer versions of Android and it's not critical to the app's core feature set, you can enable the feature only when running on the versions that support it (as discussed in [Supporting Different Platform Versions](#)). Leave this set to the default value for this project.
- **Target SDK** indicates the highest version of Android (also using the [API level](#)) with which you have tested with your application.

As new versions of Android become available, you should test your app on the new version and update this value to match the latest API level in order to take advantage of new platform features.

- **Compile With** is the platform version against which you will compile your app. By default, this is set to the latest version of Android available in your SDK. (It should be Android 4.1 or greater; if you don't have such

a version available, you must install one using the [SDK Manager](#)). You can still build your app to support older versions, but setting the build target to the latest version allows you to enable new features and optimize your app for a great user experience on the latest devices.

- **Theme** specifies the Android UI style to apply for your app. You can leave this alone.

Tryk Next nogle gange, vælg eventuelt navn på Activity, MyMainActivity, og tryk Finish.

Der vil være en fejl fra starten(måske bare nogle sekunder) indtil der er kompileret første gang, fordi klassen R først oprettes da.

Et nyt projekt vil indeholde følgende:

Src: MyMainActivity.java under stien som den angivne package.

Gen: Kommer R.java til at ligge. Den har konstanter til følgende Id'er: R.layout.main, R.drawable.icon, R.string.app_name, R.string.hello, og får flere hen ad vejen.

Res: Drawable-hdpi: icon.png 4147 bytes

Res: Drawable-mdpi: icon.png 2574 bytes

Res: Drawable-ldpi: icon.png 1723 bytes

Res: Layout: Main.xml, som er layoutet til aktiviteten.

Res: Values: Strings.xml, som er tekstdefinitioner, der gør at man kunne skifte til et andet sprog.

Problemer

"The project cannot be build until Build Path error is resolved"

Højreklik på projektet og vælg Build Path og Cofigure Build Path....

Gå i fanen Libraries og fjern de der har en rød fejlmærkning. Tilføj evt. manglende libraries.

Flyt et projekt til andet workspace

Eclipse registrerer hvilke projekter, der er oprettet i et workspace. Derfor kan et projekt ikke bare flyttes hen i et andet workspace.

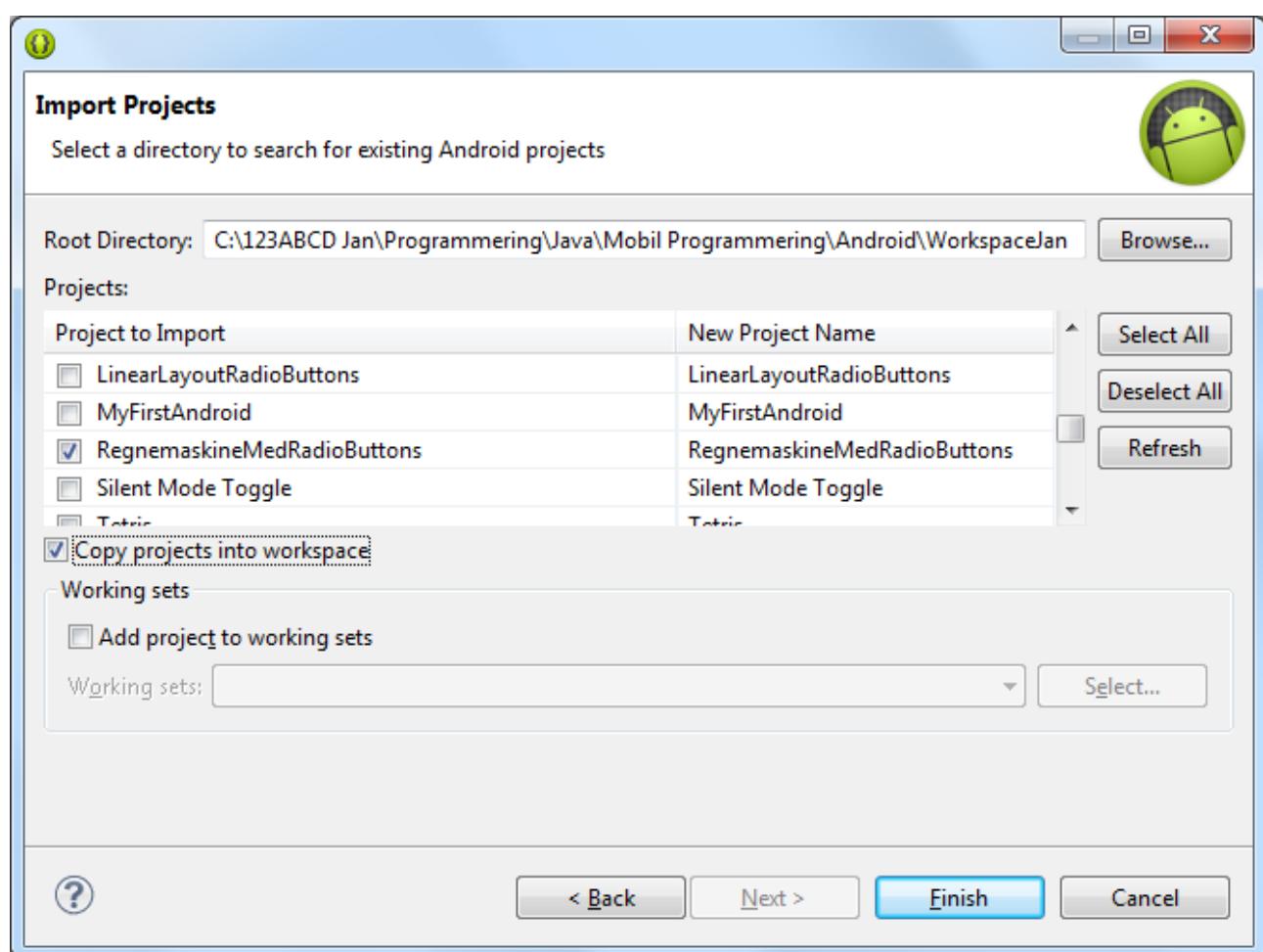
Hvis man ønsker at kopiere et projekt fra et andet workspace over i det workspace man er i, vælges File | New... | Other... | Android | Android Project from Existing Code.

Tryk Next og vælg som Root Directory den mappe som er workspacet, hvor det eksisterende projekt ligger. Der vises så de projekter som ligger i det aktuelle workspace og man kan klikke det af der ønskes kopieret.

Checkboksen Copy projects into workspace afgør om projektet bare bliver liggende hvor det er og bliver registreret i det nuværende workspace eller om der lægges en kopi i det nuværende workspace.

Tryk Finsish.

Så har man et nyt projekt det nye sted, baseret på det eksisterende projekt.



Selve Import i Eclipse er der ikke så stor begejstring for hos erfarene programmører.

Android Studio

Download og installer Android Studio

<https://developer.android.com/studio/install.html>

Sørg for at JAVA_HOME peger på den nyeste jdk.

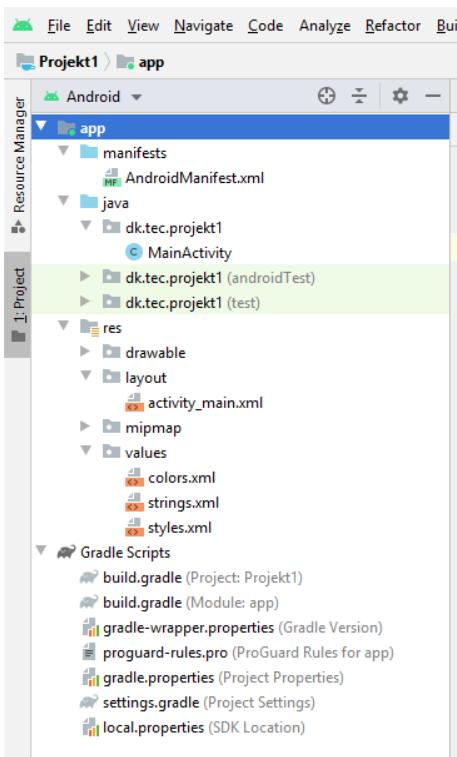
Select **Start menu > Computer > System Properties > Advanced System Properties**. Then open **Advanced tab > Environment Variables** and add a new system variable **JAVA_HOME** that points to your JDK folder, for example **C:\Program Files\Java\jdk1.8.0_77**.

2. Introduktion

De væsentligste filer i et nyoprettet Anroid-projekt er MainActivity.java og activity_main.xml.

Klassen MainActivity er start-stedet for applikationen med metoden onCreate() som bliver kaldt.

activity_main.xml er den brugerflade der som udgangspunkt bliver vist.



Klassen R

Klassen R.java indeholder identifikationer af alle resourcer, såsom id på knapper og tekst-kontroller i brugerfladen, layout.xml-filer, billeder, strings, colors mv.

Den er svær at finde i en moderne Android Studio, men det har man heller ikke brug for. Man skal kunne benytte den.

I Android Studio 4.1 kunne jeg ikke finde den mere.

```
Button btnMyButton = (Button) findViewById(R.id.btnMyButton);
```

The screenshot shows the Kode IDE interface with the following details:

- Project Tree:** Shows the project structure under "Kode". Key items include ".gradle", ".idea", "app", "build", "generated", "not_namespaced_r_class_sources", "debug", "processDebugResources", "r", "dk", and "outputs".
- Code Editor:** Displays the "R.java" file content. A yellow banner at the top states: "Files under the 'build' folder are generated and should not be edited." The code itself is a generated class "R" with static final fields for various resource IDs, such as "abc_fade_in" and "abc_fade_out".
- Toolbar:** Includes standard IDE icons for File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help.

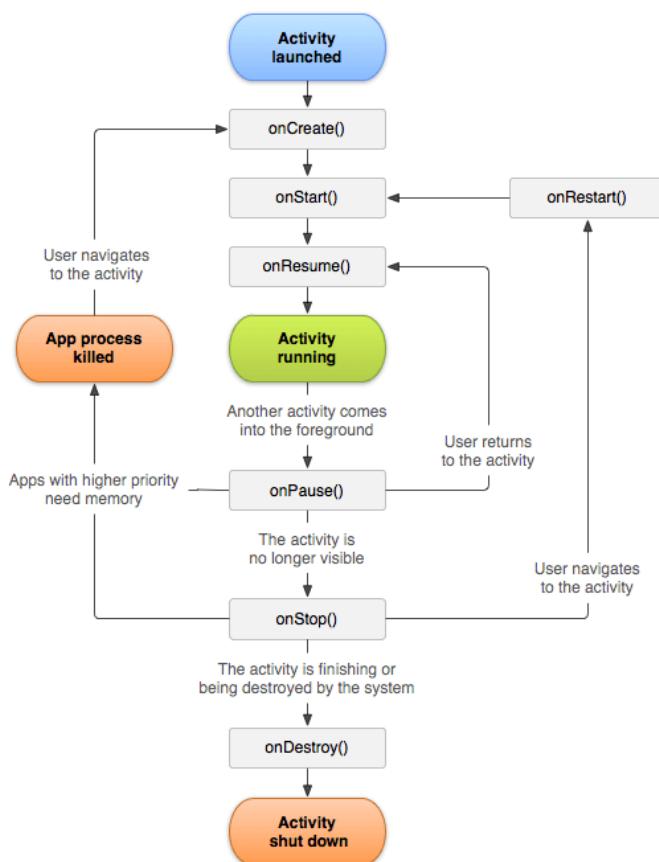
```
/* AUTO-GENERATED FILE. DO NOT MODIFY.
 *
 * This class was automatically generated by the
 * aapt tool from the resource data it found. It
 * should not be modified by hand.
 */
package dk.tec.jaj.myapplication;

public final class R {
    public static final class anim {
        public static final int abc_fade_in=0x7f01000
        public static final int abc_fade_out=0x7f0100
        public static final int abc_grow_fade_in_from
        public static final int abc_popup_enter=0x7f0
        public static final int abc_popup_exit=0x7f01
```

3. Aktivitets livscyklus og udskrift i Logcat

En Android-app består af en eller flere Activities her kaldt aktivtiteter. En aktivitet kan vise en brugerflade og reagere på brugerens handlinger.

Når hovedaktiviteten, ofte kaldt MainActivity, afsluttes, afsluttes app'en. Men enhver aktivitet i app'en har følgende livscyklus.



Udskriv i Logcat når eventhåndlere kaldes

Med Log-klassen kan der udskrives i Logcat. Med metoden `Log.e()` udskrives på niveauet error, og der angives to parametre tag og msg.

Et nyt Android-projekt har en MainActivity kun med eventhåndleren `onCreate()` overridet.

Der kan overrides andre eventhåndlere ved direkte i klassen at skrive begyndelsen på en metodes navn, f.eks. `onStart...` og lade intellisense hjælpe og så trykke enter på `onStart()`. Så genereres metode-erklæringen.

Herunder er der oprettet alle de eventhåndlere, som er angivet i diagrammet. Der skal så demonstreres hvordan de forskellige eventhåndlere bliver kaldt.

```

package dk.tec.projekt1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Log.e(">>DEBUG", "onCreate");
    }

    @Override
    protected void onStart() {
        super.onStart();

        Log.e(">>DEBUG", "onStart");
    }

    @Override
    protected void onRestart() {
        super.onRestart();

        Log.e(">>DEBUG", "onRestart");
    }

    @Override
    protected void onResume() {
        super.onResume();

        Log.e(">>DEBUG", "onResume");
    }

    @Override
    protected void onPause() {
        super.onPause();
        Log.e(">>DEBUG", "onPause");
    }

    @Override
    protected void onStop() {
        super.onStop();

        Log.e(">>DEBUG", "onStop");
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();

        Log.wtf(">>DEBUG", "onDestroy");
    }
}

```

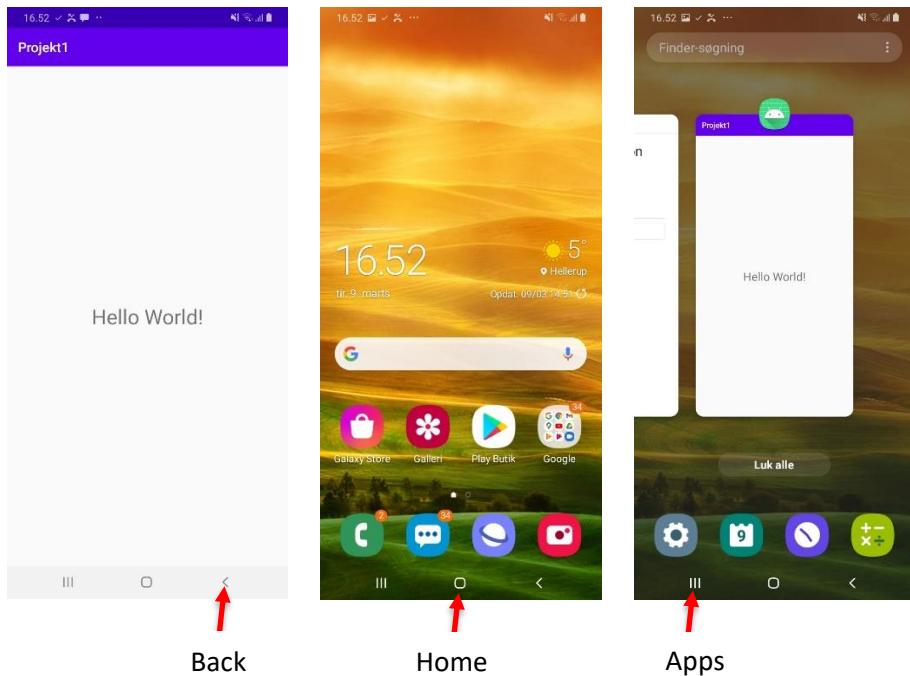
Nu startes App'en og der gøres forskelligt med den.

Ved at trykke på Home, sættes App på pause.

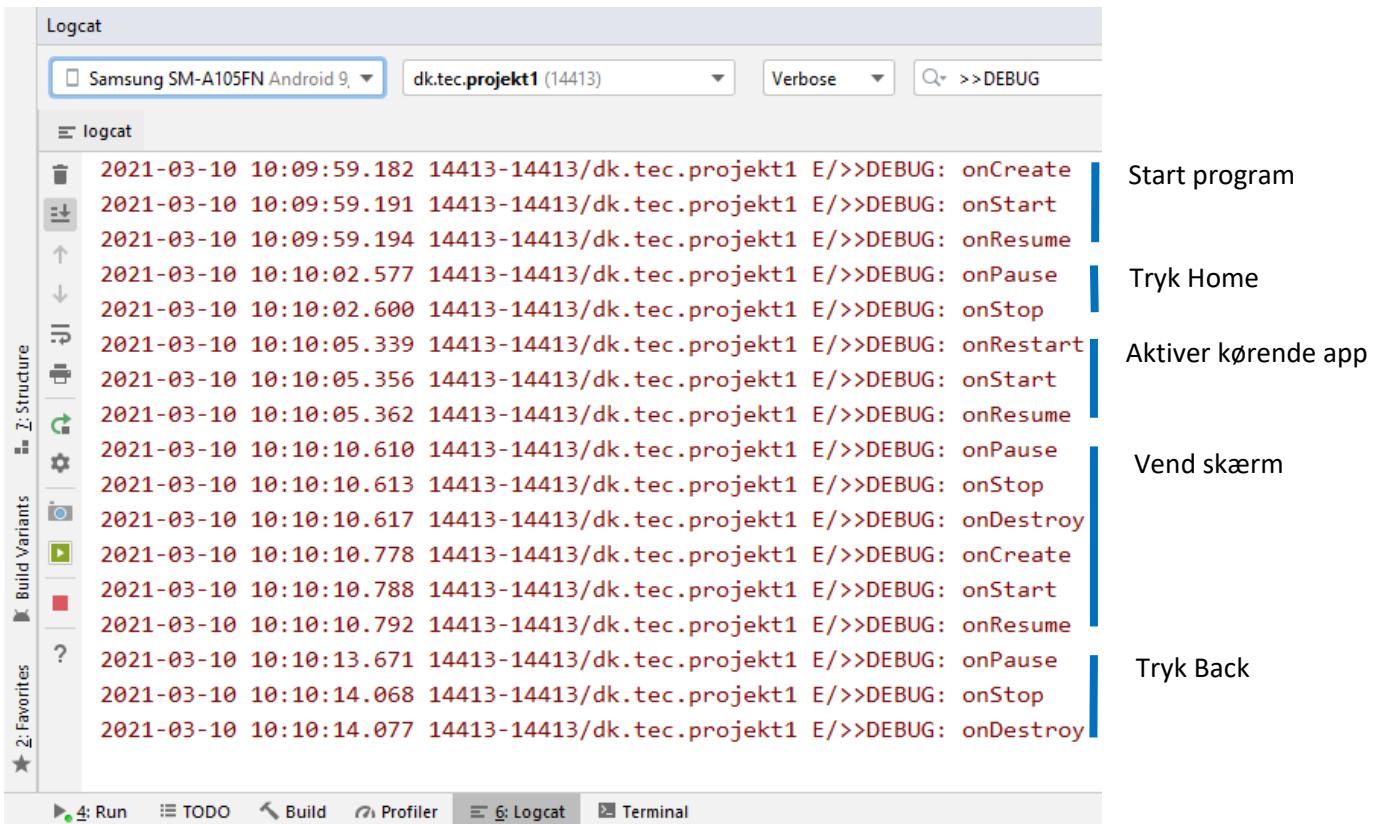
Tryk på Apps vises kørende apps og der trykkes på den aktuelle, som aktiveres igen.

Vending af skærmen gør at aktiviteten genstartes, altså helt nedlægges og startes op igen.

Tryk på Back afslutter aktiviteten og dermed app'en.



I Logcat filtreres der på teksten ">>DEBUG", så der kun ses det der udskrives med Log.e().



Log niveauer

Klassen Log har flere udskrivnings-metoder, der afgør niveauet.

Her under afprøves de alle på en gang.

```
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    Log.v(">>DEBUG", "Verbose");
    Log.d(">>DEBUG", "Debug");
    Log.i(">>DEBUG", "Info");
    Log.w(">>DEBUG", "Warning");
    Log.e(">>DEBUG", "Error");
    Log.wtf(">>DEBUG", "What the F...");
```

I Logcat kan der vælges niveau for det viste. Hvis der er valgt Verbose, vises alt. Og der kommer rigtig meget i Logcat. Denne dropdown gør det muligt at vælge niveau, sådan at Verbose er alle niveauer og Assert er laveste niveau.

The screenshot shows the Android Studio Logcat window. The device is set to "Samsung SM-A105FN Android 9". The log level is set to "Verbose". The log output shows various system and application logs. A search bar at the top right contains the text ">>DEBUG".

```
2021-03-09 16:31:27.648 15369-15369/dk.tec.projekt1 I/MainActivity: Accessing hidden method Landroid/view/View$OnTouchListener;{Ldk/tec/projekt1/TouchListener;}
```

Other log entries include:

- 2021-03-09 16:31:27.650 15369-15369/dk.tec.projekt1 I/MainActivity: Accessing hidden method Landroid/view/View\$OnTouchListener;{Ldk/tec/projekt1/TouchListener;}
- 2021-03-09 16:31:27.740 15369-15369/dk.tec.projekt1 E/MainActivity: BUG: Verbose
- 2021-03-09 16:31:27.740 15369-15369/dk.tec.projekt1 E/MainActivity: DEBUG: Debug
- 2021-03-09 16:31:27.740 15369-15369/dk.tec.projekt1 I/>>DEBUG: Info
- 2021-03-09 16:31:27.740 15369-15369/dk.tec.projekt1 W/>>DEBUG: Warning
- 2021-03-09 16:31:27.740 15369-15369/dk.tec.projekt1 E/>>DEBUG: Error
- 2021-03-09 16:31:27.741 15369-15369/dk.tec.projekt1 E/>>DEBUG: What the F...
- 2021-03-09 16:31:27.772 15369-15369/dk.tec.projekt1 D/OpenGLRenderer: Skia GL Pipeline
- 2021-03-09 16:31:27.777 15369-15369/dk.tec.projekt1 D/EmergencyMode: [EmergencyManager] android createPack

Ved at filtrere på, som herunder ">>DEBUG", kan man se kun det er udskrevet med Log med tagget >>DEBUG.

The screenshot shows the Android Studio Logcat window with the filter set to ">>DEBUG". The log output is much more concise, only showing messages with the >>DEBUG tag.

```
V/MainActivity: >>DEBUG: Verbose
```

Other log entries include:

- D/MainActivity: >>DEBUG: Debug
- I/MainActivity: >>DEBUG: Info
- W/MainActivity: >>DEBUG: Warning
- E/MainActivity: >>DEBUG: Error
- E/MainActivity: >>DEBUG: What the F...

Så kan der skiftes til næste niveau Debug. Så vises Verbose ikke.

The screenshot shows the Android Studio Logcat window with the filter set to "Debug". The log output is even more concise, only showing messages with the "Debug" tag.

```
D/MainActivity: >>DEBUG: Debug
```

Other log entries include:

- I/MainActivity: >>DEBUG: Info
- W/MainActivity: >>DEBUG: Warning
- E/MainActivity: >>DEBUG: Error
- E/MainActivity: >>DEBUG: What the F...

Og Info. Så vises kun Info og niveauer der under.

The screenshot shows the Android Studio Logcat window. The device is set to "Samsung SM-A105FN Android 9" and the package is "dk.tec.projekt1". The log level is set to "Info". The log output shows several entries:

```
2021-03-09 16:17:27.622 14550-14550/dk.tec.projekt1 I/>>DEBUG: Info
2021-03-09 16:17:27.622 14550-14550/dk.tec.projekt1 W/>>DEBUG: Warning
2021-03-09 16:17:27.622 14550-14550/dk.tec.projekt1 E/>>DEBUG: Error
2021-03-09 16:17:27.622 14550-14550/dk.tec.projekt1 E/>>DEBUG: What the F...
```

Og Warn og det der er under.

The screenshot shows the Android Studio Logcat window. The device is set to "Samsung SM-A105FN Android 9" and the package is "dk.tec.projekt1". The log level is set to "Warn". The log output shows several entries:

```
2021-03-09 16:20:51.520 14803-14803/dk.tec.projekt1 W/>>DEBUG: Warning
2021-03-09 16:20:51.520 14803-14803/dk.tec.projekt1 E/>>DEBUG: Error
2021-03-09 16:20:51.521 14803-14803/dk.tec.projekt1 E/>>DEBUG: What the F...
```

Og Error, som gælder både Log.e og Log.wtf (What a Terrible Failure ☺)

The screenshot shows the Android Studio Logcat window. The device is set to "Samsung SM-A105FN Android 9" and the package is "dk.tec.projekt1". The log level is set to "Error". The log output shows several entries:

```
2021-03-09 16:20:51.520 14803-14803/dk.tec.projekt1 E/>>DEBUG: Error
2021-03-09 16:20:51.521 14803-14803/dk.tec.projekt1 E/>>DEBUG: What the F...
```

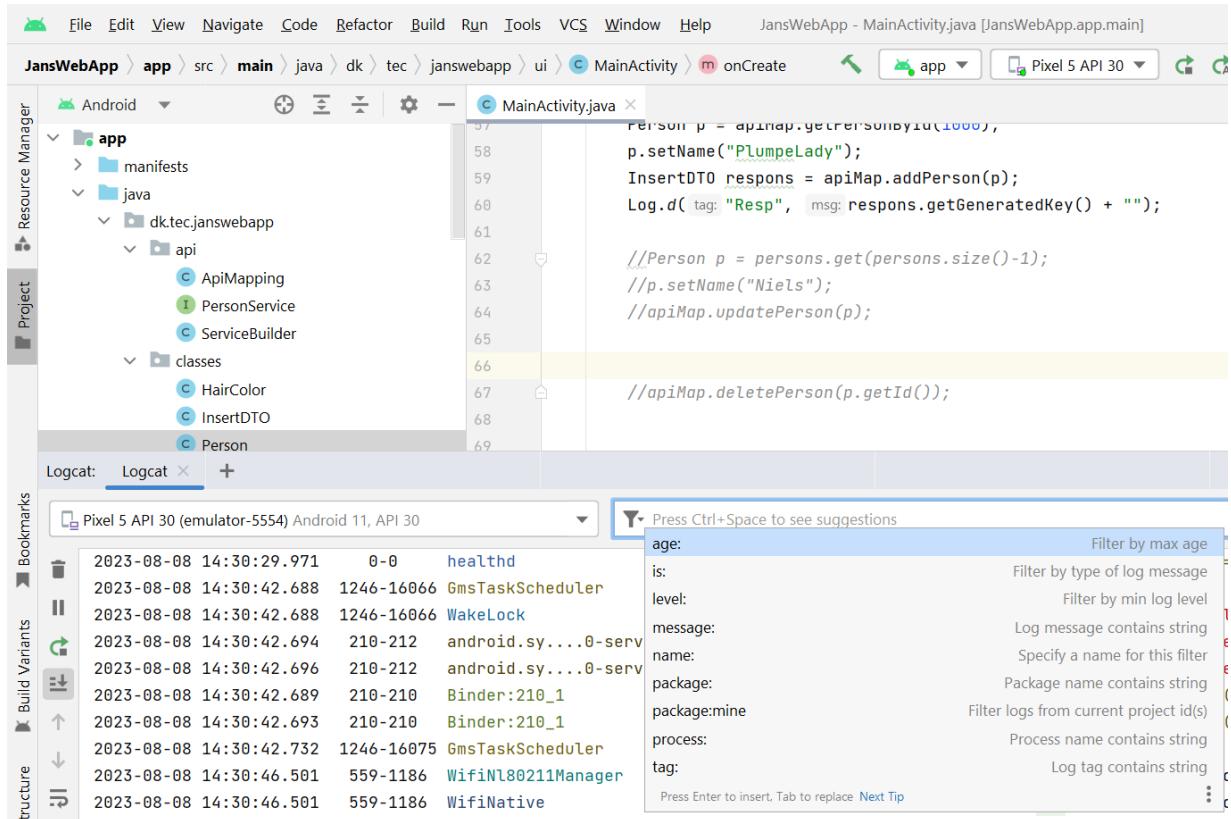
LogCat i nyere Android Studio

Der er nu ikke en dropdown med level og en tekst til søgning, men et sted til det hele.

Der er flere muligheder, såsom logiske operatorer og regular expressions.

<https://developer.android.com/studio/debug/logcat>

I LogCats højre dropdown tryk ctrl-space, så kommer der forslag.



Vælg f.eks. level:

MainActivity.java

```

    person p = apiMap.getPersonById(1000);
    p.setName("PlumpeLady");
    InsertDTO respons = apiMap.addPerson(p);
    Log.d( tag: "Resp", msg: respons.getGeneratedKey() + "");

    //Person p = persons.get(persons.size()-1);
    //p.setName("Niels");
    //apiMap.updatePerson(p);

    //apiMap.deletePerson(p.getId());

```

Logcat: Logcat

Pixel 5 API 30 (emulator-5554) Android 11, API 30

----- beginning of system
----- beginning of kernel
----- beginning of main

level: debug

- debug Filter by DEBUG or higher
- assert Filter by ASSERT or higher
- error Filter by ERROR or higher
- info Filter by INFO or higher
- verbose Filter by VERBOSE or higher
- warn Filter by WARN or higher

Og dernæst debug

MainActivity.java

```

    person p = apiMap.getPersonById(1000);
    p.setName("PlumpeLady");
    InsertDTO respons = apiMap.addPerson(p);
    Log.d( tag: "Resp", msg: respons.getGeneratedKey() + "");

    //Person p = persons.get(persons.size()-1);
    //p.setName("Niels");
    //apiMap.updatePerson(p);

    //apiMap.deletePerson(p.getId());

```

Logcat: Logcat

Pixel 5 API 30 (emulator-5554) Android 11, API 30

level: debug

level:	Filter by min log level
level:	Filter by max age
age:	Filter by type of log message
android:	Log message contains string
is:	Specify a name for this filter
com:	Package name contains string
message:	Filter logs from current project id(s)
netmgr:	Process name contains string
name:	
netmgr:	
package:	
process:	
wifi:	
tag:	Log tag contains string
com:	

2023-08-08 14:32:56.214 210-212 android.system.OService android: Filter by min log level
2023-08-08 14:32:56.209 210-210 Binder:210_1 android: Filter by max age
2023-08-08 14:33:00.005 753-753 KeyguardClockSwitch com.android.keyguard: Filter by type of log message
2023-08-08 14:33:25.683 451-451 netmgr netmgr: Log message contains string
2023-08-08 14:33:25.683 451-451 netmgr netmgr: Specify a name for this filter
2023-08-08 14:33:26.504 559-1186 WifiNL80211Manager com.android.wifi: Package name contains string
2023-08-08 14:33:26.504 559-1186 WifiNative com.android.wifi: Filter logs from current project id(s)
2023-08-08 14:33:27.118 460-460 wifi_forwarder system: Process name contains string
2023-08-08 14:33:27.118 460-460 wifi_forwarder system:

Dernæst ctrl-space og vælg tag:

Og skriv så teksten efter tag: >>DEBUG (eller rettere her Resp, hvis det skal passe med den viste Log.d() i koden)

The screenshot shows the Android Studio interface with the following details:

- Top Bar:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help, JansWebApp - MainActivity.java
- Toolbar:** Includes icons for file operations, search, and navigation.
- Project Structure:** Shows the project tree under "app".
 - app (selected)
 - manifests
 - java
 - dk.tec.janswebapp
 - api
 - ApiMapping
 - PersonService
 - ServiceBuilder
 - classes
 - HairColor
 - InsertDTO
 - Person
- Main Activity:** The code editor shows MainActivity.java with the following content:

```
57     Person p = apiMap.getPersonById(1);
58     p.setName("PlumpeLady");
59     InsertDTO respons = apiMap.addPerson(p);
60     Log.d( tag: "Resp", msg: respons.getId());
61
62     //Person p = persons.get(persons.size() - 1);
63     //p.setName("Niels");
64     //apiMap.updatePerson(p);
65
66     //apiMap.deletePerson(p.getId());
67
68
69
```
- Logcat:** The logcat tab is selected, showing the following output:
 - Pixel 5 API 30 (emulator-5554) Android 11, API 30
 - beginning of system
 - level:debug tag: >>DEBUG

4. Brugerflade

Brugerflader oprettes almindeligvis i xml, men det er også muligt at gøre det i Java. Widgets er controller der kan vises på skærmen, og er klasser der arver fra View. Det design der er angivet i xml vil resultere i oprettelse af tilsvarende objekter Java-koden. Denne kode kan ikke ses, men der kan findes referencer til alle, der har en id i xml-koden.

Derfor er det også muligt at oprette brugerfladen eller dele af den i Java-koden.

En Button oprettet i Java

Her oprettes og vises en Button i Java-koden. Den fylder hele skærmen. Dette gøres ved at kalde setContentView() og angive det der skal vises.

```
public class MyMainActivity extends Activity
{
    Button btnMyButton;
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

        btnMyButton = new Button(this); //this - Context som Activity arver fra
        btnMyButton.setText("My Button");
        setContentView(btnMyButton);
    }
}
```

Et layout oprettet i Java

Der kan også oprettes et layout, som fyldes med views. Så kan der være flere ting på skærmen.

Når layoutet er fyldt med views, sættes denne til at blive vist med metoden setContentView().

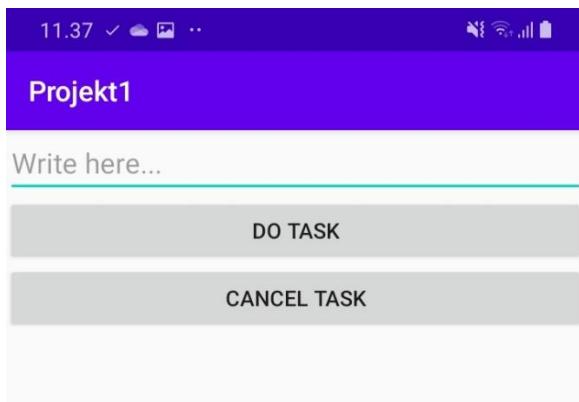
```
TableLayout tl = new TableLayout(this);

EditText txtEdit = new EditText(this);
txtEdit.setHint("Write here...");
txtEdit.setId(117);
tl.addView(txtEdit);

Button btnDoTask = new Button(this);
btnDoTask.setText("Do Task");
tl.addView(btnDoTask);

Button btnCancel = new Button(this);
btnCancel.setText("Cancel Task");
tl.addView(btnCancel);

setContentView(tl);
```

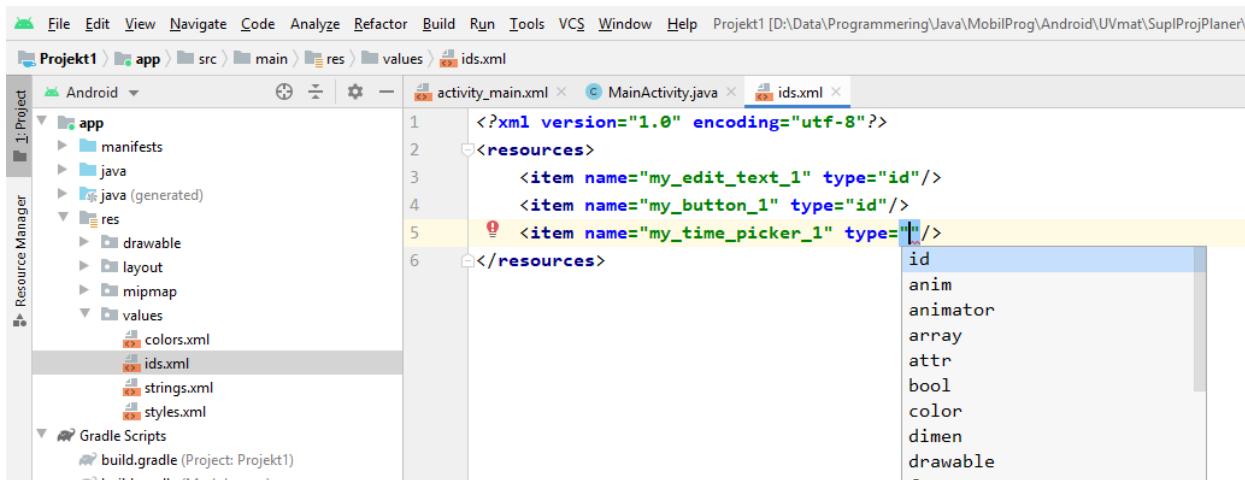


Id oprettet i ressourcer

I nyere udgaver af Android Studio, kommer der en advarsel, hvis der sættes en konstant int som id. Dette er fordi der kan komme konflikter i tilfælde af at flere har samme int. Der ønskes en int der er defineret som en ressource.

Der oprettes ressource-id'er ved at lave en xml fil under res – values, her ids.xml.

Der oprettes items med name og type="id", så får de en id i klassen R.



Og kan bruges som vist her under.

```

EditText txtEdit = new EditText(this);
txtEdit.setId(R.id.my_edit_text_1);

```

Grunden til at der sættes et id på txtEdit er, at et view der har en id vil bibeholde sit indhold, når skærmen vendes. Når skærmen vendes vil aktiviteten begynde forfra, og evt. indhold som brugeren har tastet vil glemmes. Indholdet huskes imidlertid når viewet har en id.

Et view der er oprettet i xml, og hvor brugeren skal gøre noget, indtaste eller vælge, skal altid have et id for at viewet kan tilgås i java-koden. Et view der er oprettet programmatisk, vil jo altid kunne tilgås i java-koden. Så id'et kunne virke overflødig. Men det har altså den effekt at indhold/valg huskes.

Brugerflade i xml

Den samme brugerflade kan laves activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Write here..."/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id	btnDoTask"
        android:text="Do Task"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id	btnCancelTask"
        android:text="Cancel Task"/>
</TableLayout>
```

Og vises med setContentView() i Java koden.

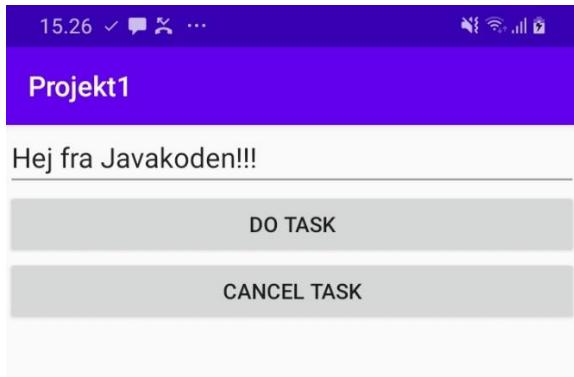
```
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    EditText txtEdit2 = findViewById(R.id.txtEdit);
    Button btnDoTask2 = findViewById(R.id.btnDoTask);
    Button btnCancelTask2 = findViewById(R.id.btnCancelTask);

    txtEdit2.setText("Hej fra Javakoden!!!");
```

Fordi views har en id kan objekterne findes og lægges i respektive variable.

Der kan nu gøres ting med dem. Knapperne kan tilskrives eventhåndlere (det gøres ikke her) og txtEdit2 kan der skrives noget i her fra Java-koden.



Øvelse Brugerflade

Øvelse tilføj en knap til layoutet

Givet ovenstående kode, hvor layoutet er defineret i activity_main.xml.

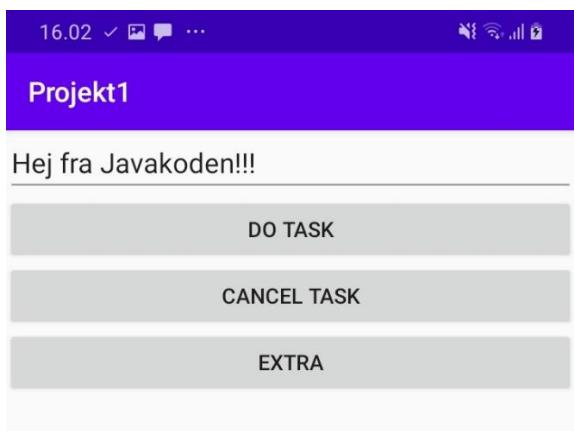
I activity_main.xml:

- Giv tagget TableLayout en id, f.eks. theLayout.

I Java-koden,

- Find TableLayoutet.
- Opret en ny Button med teksten Extra.
- Tilføj den til TableLayout.

Så det ser ud som her under.



5. Eventhandling på bruger-interface

Dette kapitel om handler eventhandling ved tryk på knapper. Men princippet gælder alle eventhåndlere på bruger-handlinger.

Eventhandling i Java gøres ved at overgive et objekt af en klasse, der implementerer et bestemt interface, til det objekt, der genererer eventet.

Hvis en Button skal kunne kalde en click-eventhandler, skal den tilskrives et objekt af en klasse, der implementerer interfacet OnClickListener, og dermed metoden OnClick() . Button vil så kalde denne metode når der klikkes på den.

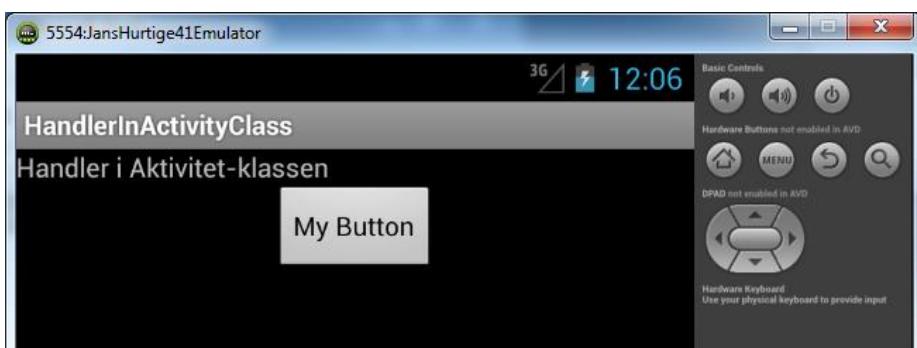
Man kan så lægge den kode der ønskes udført ved klik på Button i metoden onClick().

At oprette en klasse der implementerer OnClicklistener kan ske på flere måder.

Brugerflade til afprøvning af eventhandling

For at afprøve disse oprettes følgende layout i activity_main.xml (altså ændre i det eksisterende layout).

Titlen svarer til navnet på projektet. Dette er defineret i string.xml.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:id="@+id/txtMyText"
    />
    <Button
        android:text="My Button"
        android:id="@+id/btnMyButton"
    >
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
    />
</LinearLayout>

```

XML-koden kan formatteres pænt ved shift-ctrl-F.

Eventhandler internt i Activity-klassen

Ved at lade aktiviteten selv implementere OnClickListener, vil den selv have metoden onClick() og referencen til objektet af klassen fås med keyword this, der gives som argument til den aktuelle Buttons metode setOnClickListener().

Ulempen er at der jo kun er en metode onClick() som skal bruges til alle Buttons på brugerfladen. Men det er der råd for, da den udløsende Button kommer som argument til onClick().

```

public class HandlerInActivityClass extends Activity implements OnClickListener
{
    Button btnMyButton;
    TextView txtMyText;

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

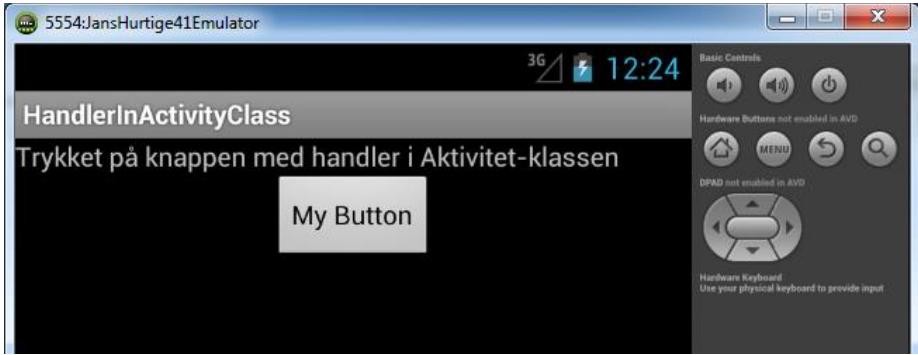
        setContentView(R.layout.main);

        txtMyText = ((TextView) findViewById(R.id.txtMyText));

        btnMyButton = (Button) findViewById(R.id.btnMyButton);
        btnMyButton.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        txtMyText.setText("Trykket på knappen med handler i aktivitet-klassen");
    }
}

```



Øvelser Eventhandler internt i Activity-klassen

Øvelse antal tryk

Opret et Android project. Læg en Button på brugerfladen og skriv på den, hvor gang der trykkes på den, hvor mange gange der er trykket.

Øvelse Skift Farver

Opret en attribut af navnet Skift og af typen boolean, som kan styre at txtMyText skifter i mellem to baggrundsfarver hver gang der trykkes på btnMyButton.

Eventhandler i extern klasse

Eventhandleren onClick() ligger i en navngiven klasse f.eks. OnClickHandler1. Dette princip muliggør en særskilt eventhandler til hvert event, så f.eks. hver Button kan få sin egen eventhandler, ved at oprette flere klasser, der implementerer OnClickListener og dermed metoden onClick() i.

Ulempen er at klassen, eller rettere et objekt af klassen ikke kender til objektet, hvor den aktuelle Button ligger i. Derfor oprettes en constructor, som modtager en reference til denne klasse, hvorved den kan tilgå ressourcerne og som her skrive noget i et TextView i brugerfladen.

```
public class HandlerInExternalClass extends Activity
{
    Button btnMyButton;
    TextView txtMyText;

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

        setContentView(R.layout.main);

        txtMyText = ((TextView) findViewById(R.id.txtMyText));

        btnMyButton = (Button) findViewById(R.id.btnMyButton);
        btnMyButton.setOnClickListener(new Handler(this));
    }
}
```

```

    }

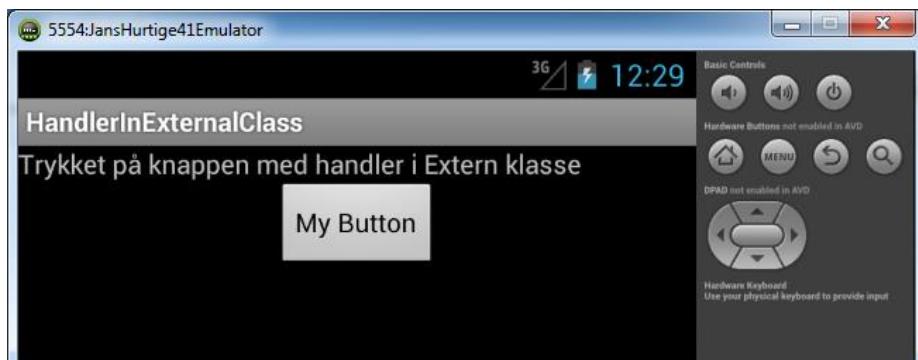
}

class Handler implements OnClickListener
{
    private HandlerInExternalClass theActivity;

    public Handler(HandlerInExternalClass activity)
    {
        theActivity = activity;
    }

    public void onClick(View v)
    {
        theActivity.txtMyText.setText("Trykket på knappen med handler i Extern klasse");
    }
}

```



Øvelse Eventhandler i extern klasse

Læs fra og skriv til tekstboks

Opret et Android-projekt som har to Buttons og en EditText i brugerfladen.

Opret to eksterne klasser som eventhåndlere, en til hver Button.

Sørg for at de to klasser har en constructor, der kan modtage en reference til Activity-objektet(this), sådan at de kan få fat i denne når en eventhandler kaldes.

Den ene Button skal ved tryk gemme det som brugeren har skrevet i EditText, og den anden Button skal skrive det tidligere læste tilbage i EditText.

Eventhandler i anonym indre klasse

Når eventhåndleren skal sættes på en Button, skal der angives et objekt af en klasse der implementerer interfacet OnClickListener.

I dette tilfælde bliver der oprettet et objekt af en anonym klasse, der erklæres i samme moment. Så kan det hele defineres som argument til Buttons metode setOnClickListener()

En anonym klasse har intet navn. Den defineres samtidig med at der oprettes et objekt af den, så der kan ikke laves flere objekter af den.

Fordelen er at hver Button får sin egen eventhandler. Omvendt kan en eventhandler kun bruges til én Button.

Her ønskes oprettet et objekt med eventhandleren onClick(). Derfor skal den anonyme klasse implementere interfacet OnClickListener. Det er dette navn der bruges som konstruktør-navn efter keyword new. Efter dette følger klassens definition.

```
public class HandlerInAnonymousClass extends Activity
{
    Button btnMyButton;
    TextView txtMyText;

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

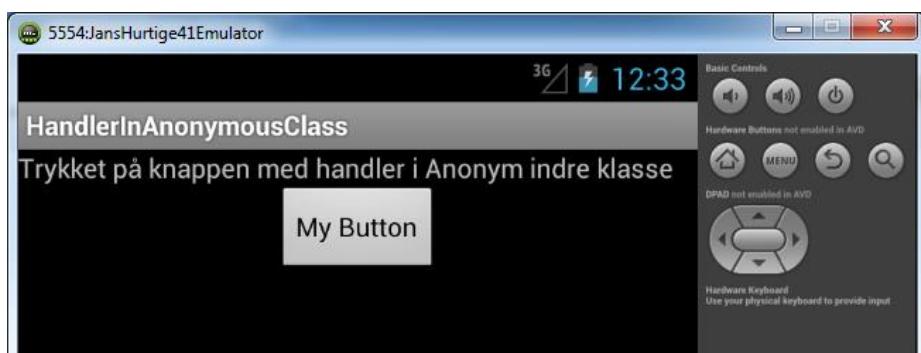
        setContentView(R.layout.main);

        txtMyText = ((TextView) findViewById(R.id.txtMyText));

        btnMyButton = (Button) findViewById(R.id.btnMyButton);

        btnMyButton.setOnClickListener(new OnClickListener()
        {
            public void onClick(View v)
            {
                txtMyText.setText("Trykket på knappen med handler i anonymous-class");
            }
        });
    }
}
```

Hvis txtMyText var en lokal variabel i onCreate(), og den anonyme klasse skal bruge den, skal txtMyText have modifier final, hvilket får den anonyme klasse til at gemme sin egen kopi af txtMyText.



Objektet af den anonyme indre klasse, kan laves som et lambda-udtryk. Her kræves at Java-versionen er mindst 8 (1.8).

Øvelse Eventhandler i anonym indre klasse

Øvelse simpel regnemaskine

Opret et projekt som har 3 EditText og en Button med teksten Adder.

Brugeren skal kunne indtaste to tal i hver sin EditText,
og ved tryk på Button skal summen af de to tal vises i den 3. EditText.

Eventhandleren skal ligge i en anonym indre klasse.

Eventhandler i Indre klasse

Som eventhandler oprettes en indre klasse i aktivitetsklassen.

Fordelen her er at der kan oprettes lige så mange indre klasser som man ønsker af forskellige evenhåndlere, den enkelte klasse kan omvendt bruges til flere Buttons, og at indre klasser kender alt i den klasse de ligger i.

```
public class HandlerInInnerClass extends Activity
{
    Button btnMyButton;
    TextView txtMyText;

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

        setContentView(R.layout.main);

        txtMyText = (TextView) findViewById(R.id.txtMyText);

        btnMyButton = (Button) findViewById(R.id.btnMyButton);
        btnMyButton.setOnClickListener(new Handler());
    }

    class Handler implements OnClickListener
    {
        public void onClick(View v)
        {
            txtMyText.setText("Trykket på knappen med handler i Indre klasse");
        }
    }
}
```

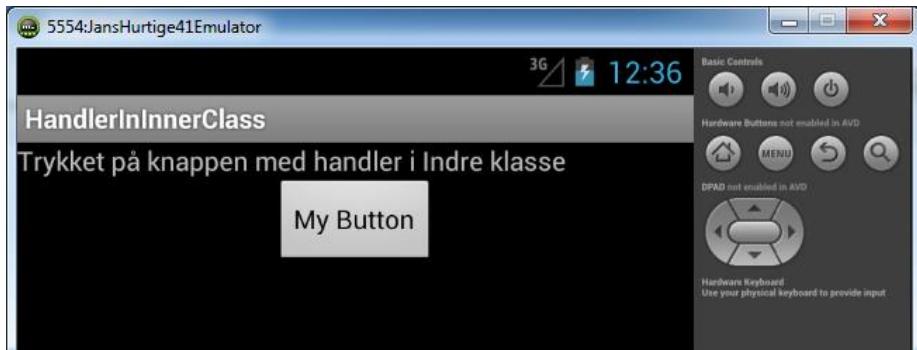
Øvelse Eventhandler i Indre klasse

Simpel regnemaskine

Opret et projekt som har 3 EditText og en Button med teksten Subtraher.

Brugeren skal kunne indtaste to tal i hver sin EditText,
og ved tryk på Button skal det ene tal trækkes fra det andet, og resultatet vises i den 3. EditText.

Eventhandleren skal ligge i en indre klasse.



Eventhandler angivet i layout

I xml-layoutet angives hvilken metode der skal kaldes ved eventet. Dette er meget nemt, men har den ulempe at metoden kan hedde hvad som helst og derfor kan det være ulydigt hvilke metoder der er eventhandlere.

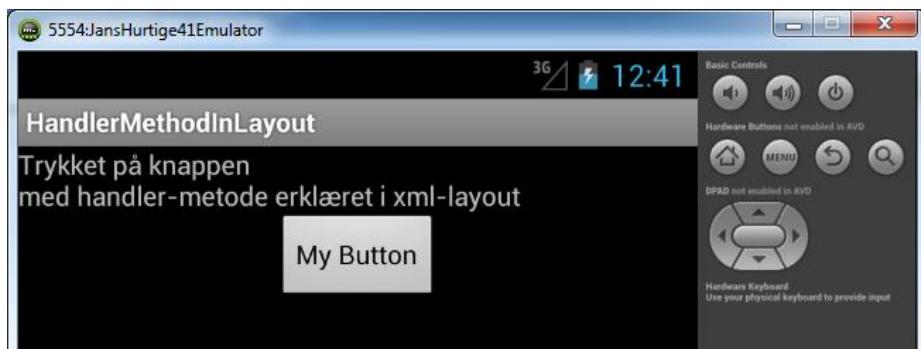
```
<Button  
    android:text="My Button"  
    android:id="@+id/btnMyButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal"  
    android:onClick="myOnClickMethod"  
/>
```

```
public class HandlerMethodInLayout extends Activity  
{  
    Button btnMyButton;  
    TextView txtMyText;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState)  
    {  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.main);  
  
        txtMyText = ((TextView) findViewById(R.id.txtMyText));  
        btnMyButton = (Button) findViewById(R.id.btnMyButton);  
    }  
}
```

```

public void myOnClickMethod(View v)
{
    txtMyText.setText(
        "Trykket på knappen \nmed handler-metode erklæret i xml-layout");
}

```



Man kunne så spørge om det er muligt at sætte denne property i programkoden

```
android:onClick="myOnClickMethod"
```

Men nej! Denne property foranlediger det samme som hvis der i onCreate() havde stået følgende

```

btnMyButton.setOnClickListener(new OnClickListener()
{
    public void onClick(View v)
    {
        myOnClickMethod(v);
    }
});
```

Øvelse Eventhandler angivet i layout

Russisk Roulette

Opret en attribut med et Random-objekt i, hvorfra der hentes et tal imellem 0 og 5 hver gang der trykkes på btnMyButton. Hvis tallet er 3 skal der udskrives teksten DU ER DØD og med rød baggrundsfarve. Ellers skal der udskrives DU SLAP MED LIVET I BEHOLD, skiftende imellem baggrundsfarverne blå og grøn, hver gang det er denne tekst, der udskrives.

Farver kan angives med en int, som er 4 bytes af AARRGGBB (AA er ugenemsigtighed) og hexadecimalt angives med Ox foran tallet.2

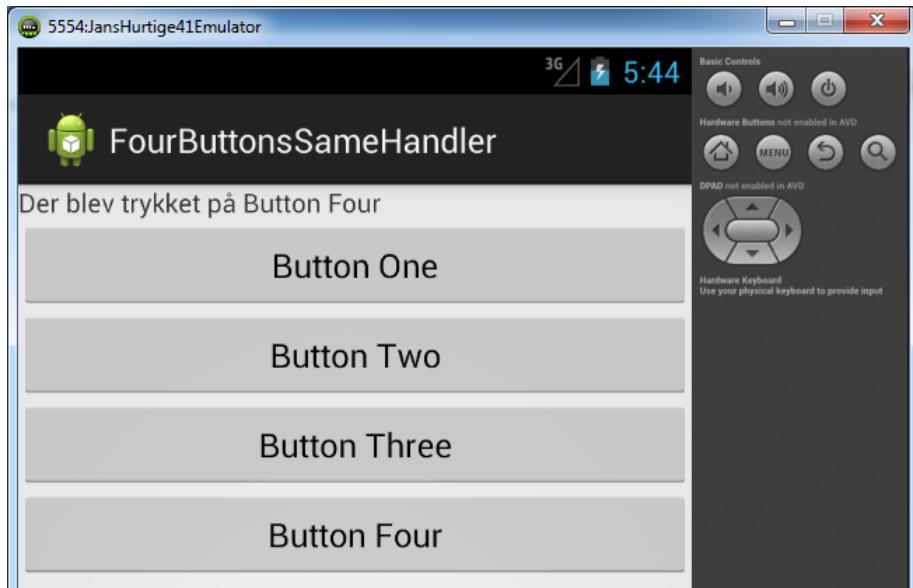
Flere knapper med samme eventhandler.

Følgende layout opretter i en LinearLayout som er orientation:vertical, en TextView og der under 4 Buttons med eventhåndlere, der udskriver en besked når der trykkes på en af dem.

```

<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"
    tools:context=".MainActivity"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello_world"
        android:id="@+id/txtMyText" />
    <Button
        android:id="@+id/btnMyButtonOne"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button One" />
    <Button
        android:id="@+id/btnMyButtonTwo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button Two" />
    <Button
        android:id="@+id/btnMyButtonThree"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button Three" />
    <Button
        android:id="@+id/btnMyButtonFour"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button Four" />
</LinearLayout>

```



```

public class MainActivity extends Activity implements OnClickListener
{
    Button btnMyButtonOne;
    Button btnMyButtonTwo;
    Button btnMyButtonThree;
    Button btnMyButtonFour;

    TextView txtMyText;

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

        btnMyButtonOne = (Button) findViewById(R.id.btnMyButtonOne);
        btnMyButtonOne.setOnClickListener(this);

        btnMyButtonTwo = (Button) findViewById(R.id.btnMyButtonTwo);
        btnMyButtonTwo.setOnClickListener(this);

        btnMyButtonThree = (Button) findViewById(R.id.btnMyButtonThree);
        btnMyButtonThree.setOnClickListener(this);

        btnMyButtonFour = (Button) findViewById(R.id.btnMyButtonFour);
        btnMyButtonFour.setOnClickListener(this);

        txtMyText = ((TextView) findViewById(R.id.txtMyText));
    }

    @Override
    public void onClick(View v)
    {

        // Hvilken Button vha. Id
        if(v.getId() == R.id.btnMyButtonOne)
            txtMyText.setText("Der blev trykket på Button One");

        // Hvilken Button vha. reference
        if(v == findViewById(R.id.btnMyButtonTwo))
            txtMyText.setText("Der blev trykket på Button Two");

        // Hvilken Button vha. reference
        if(v == btnMyButtonThree)
            txtMyText.setText("Der blev trykket på Button Three");

        // Hvad står der på Button?
        if(((Button)v).getText().equals("Button Four"))
            txtMyText.setText("Der blev trykket på Button Four");
    }
}

```

Eventhandlerne genkender den trykkede knap på 4 forskellige måder. Egentlig kun 3 måder, da de to er Button-objektets reference.

Øvelser

Opgave regnemaskine – forskellige handlere for hver knap

Opret et Android projekt med en brugerflade der ligner det viste billede.



Ved indtastning af de to øverste tal og dernæst tryk på en operator-knap, skal de to tal beregnes og resultatet skrives forneden.

Der benyttes LinearLayout og knapperne kan ligge i en nested LinearLayout.

De fire operator-knapper skal have hver sin måde på at få tildelt en onClick() handler, hvilket kan være i Activity-klassen selv, en ekstern klasse, en intern klasse, en anonym intern klasse eller angivet i layoutet.

Opgave regnemaskine – samme handler for alle knapper

Opret et Android projekt med en brugerflade der ligner det viste.



Der skal kunne indtastes to tal, og ved tryk på en operator-knap, skal tallene beregnes ud fra den valgte regneart, og resultatet vises nederst.

Der skal bruger LinearLayout med nestede LinearLayouts til de fire rækker med flere Views i hver. Brug evt. layout_weight til bredde og placering af TextView og EditText og en evt. tom TextView til højre. Brug også text_alignment.

De fire operator-knapper skal have samme eventhandler. Brug en switch konstruktion i eventhandleren.

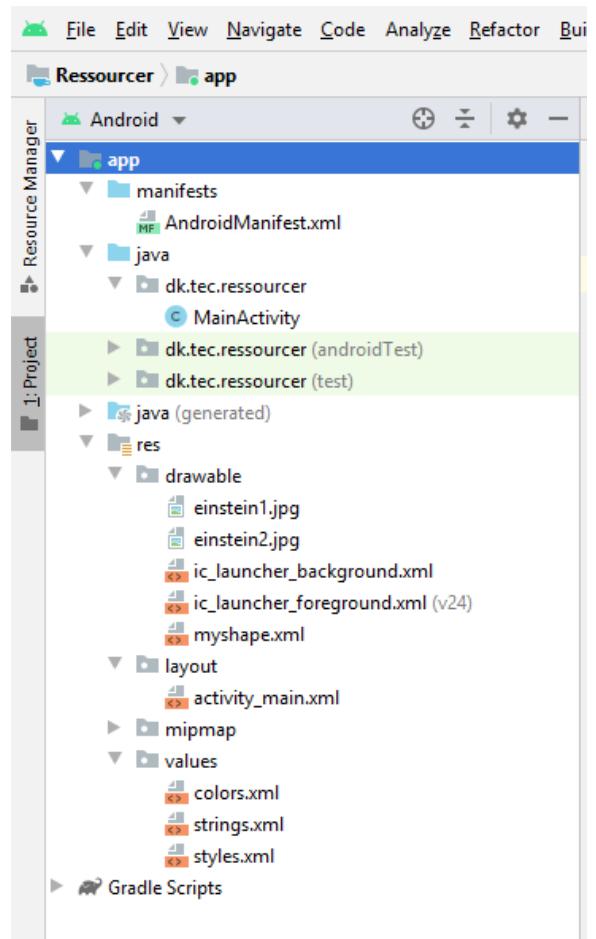
6. Ressourcer

Alt det der ligger under mappen res kaldes ressourcer.

I mappen drawable ligger der billeder og grafiske xml-filer. Der kan være andre drawable-xx mapper med forskellige oplosninger af det samme billede. Den benyttede styres af skærmopløsningen.

I mappen layout ligger xml filer med design af brugerflader. Der kan også være andre layout-yy mapper med designet tilpasset portrait eller landscape.

I mappen mipmap ligger der flere af de samme launcher icons med forskellige oplosninger. Den benyttede styres af skærmopløsningen.



I mappen values ligger som udgangspunkt tre filer, colors.xml med farve-konstanter, strings.xml med string-konstanter og styles.xml med grupperinger af styles til Views i brugerfladen.

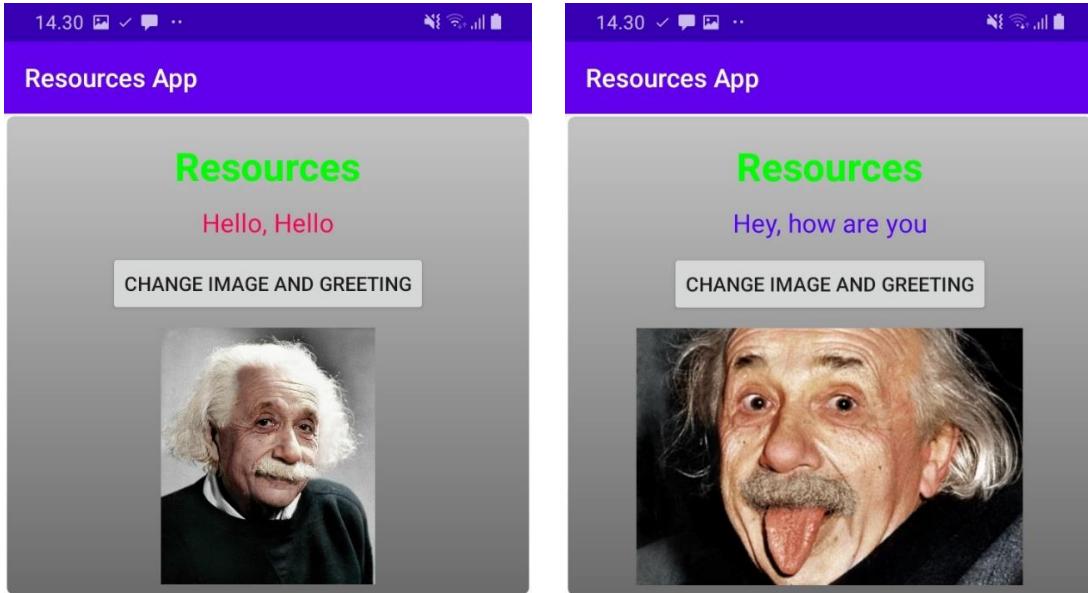
Der kan være flere values-zz mapper, hvor zz er landekoder, så der f.eks. kan skiftes sprog efter hvad enheden er instillet til.

Demo af Ressourcer

I det følgende projekt demonstreres brug af strings.xml, colors.xml, styles.xml og drawable, og hvordan de kan bruges både i xml-layout og i Java-kode.

Skærmorientering, billedopløsning og landekoder kommer senere.

Knappen skifter billedet samt tekst og farve på hilsenen.



AndroidManifest.xml

Der refereres til ic_launcher og ic_launcher_round til forskellige enheder og forskellige oplosninger.

En string.xml tekst bruges som navn foroven i den blaа bjælke.

Der sættes en style som theme for hele applikationen.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="dk.tec.ressourcer">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

strings.xml

De definerede tekster bruges i den rækkefølge der ses på skærbilledet

```
<resources>
    <string name="app_name">Resources App</string>
    <string name="header">Resources</string>

    <string name="greetingOne">Hello, Hello</string>
    <string name="greetingTwo">Hey, how are you</string>

    <string name="btn_text">Change Image and Greeting</string>
</resources>
```

colors.xml

De første 3 color definitioner bruges i den style som bruges som bases theme for applikationen.

De sidste 2 color definitioner er farver til hver greeting-tekst.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>

    <color name="greetingColorRed">#FF0055</color>
    <color name="greetingColorBlue">#5500FF</color>
</resources>
```

styles.xml

Den første style er den theme som giver applikationen basis-farver.

Den anden style er nogle attributter som kunne sættes direkte på et TextView start-tag, og bruges i layoutet til overskriften.

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="colorPrimary">@color/colorPrimary</item>
        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="colorAccent">@color/colorAccent</item>
    </style>

    <style name="HeaderStyle" parent="TextAppearance.AppCompat">
        <item name="android:textColor">#00FF00</item>
        <item name="android:textSize">30dp</item>
        <item name="android:textStyle">bold</item>
        <item name="android:layout_margin">10dp</item>
    </style>
</resources>
```

activity_main.xml

LinearLayout får som background en grafik myshape.xml som ligger i drawable. Filen ses længere nede.

TextView til overskriften får den style som er defineret i styles.xml.

TextView til greeting får color greetingColorRed og string greetingOne.

Button får som tekst en string btn_text.

ImageView får drawable einstein1.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    android:padding="10dp"
    android:background="@drawable/myshape"
```

```

tools:context=".MainActivity">

<TextView
    style="@style/HeaderStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/header" />

<TextView
    android:id="@+id/textGreeting"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20dp"
    android:textColor="@color/greetingColorRed"
    android:text="@string/greetingOne" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="@string/btn_text"
    android:id="@+id/btnChangeImage" />

<ImageView
    android:layout_width="wrap_content"
    android:layout_height="200dp"
    android:id="@+id/imgEinstein"
    android:src="@drawable/einstein1" />
</LinearLayout>

```

myshape.xml

Et grafisk billede som ligger i drawable. Det bruges som baggrund i layoutet og giver en grå flade, der bliver mørkere nedad.

```

<?xml version="1.0" encoding="UTF-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <stroke
        android:width="2dp"
        android:color="#FFFFFF" />
    <gradient
        android:endColor="#DDBBBBBB"
        android:startColor="#DD555555"
        android:angle="90" />
    <corners
        android:bottomRightRadius="7dp"
        android:bottomLeftRadius="7dp"
        android:topLeftRadius="7dp"
        android:topRightRadius="7dp" />
</shape>

```

MainActivity.java

I onClick() skiftes der billede, tekst og farve på greeting.

Teksten til greeting kan først hentes og så sættes på TextView.

Det kan også sættes direkte ved at angive id'et på den aktuelle string. (Det er derfor det ikke kan lade sig gøre at sætte teksten i form af et heltal uden først at konvertere det til en string).

I onCreate() er der også vist hvordan der fås fat i selve en Drawable. Det er ikke så relevant her, men bruges hvis der skal paintes en Drawable i grafik i onPaint());

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener
{
    ImageView imgViewEinstein;
    TextView txtGreeting;

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

        txtGreeting = findViewById(R.id.textGreeting);
        imgViewEinstein = findViewById(R.id.imgEinstein);

        Button btnChangeImage = findViewById(R.id.btnChangeImage);
        btnChangeImage.setOnClickListener(this);

        Drawable d1 = ResourcesCompat.getDrawable(getResources(), R.drawable.einstein1, null);
        Drawable d2 = ContextCompat.getDrawable(this, R.drawable.einstein2);
        imgViewEinstein.setImageDrawable(d1);
    }

    @Override
    public void onClick(View view)
    {
        imgViewEinstein.setImageResource(R.drawable.einstein2);

        String greetingText = getResources().getString(R.string.greetingTwo);
        txtGreeting.setText(greetingText);
        // Eller
        txtGreeting.setText(R.string.greetingTwo);

        int greetingColor = ContextCompat.getColor(this, R.color.greetingColorBlue);
        txtGreeting.setTextColor(greetingColor);
    }
}
```

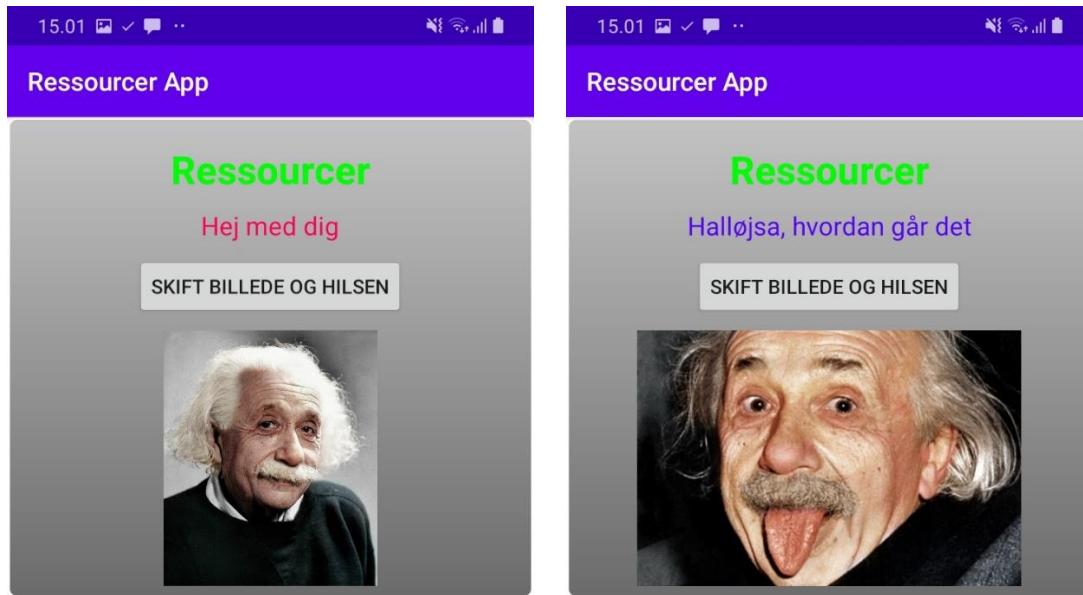
Skift sprog

Der kan oprettes flere strings.xml i hver sin mappe, som så vælges af systemet ud fra hvilket sprog enheder er sat til.

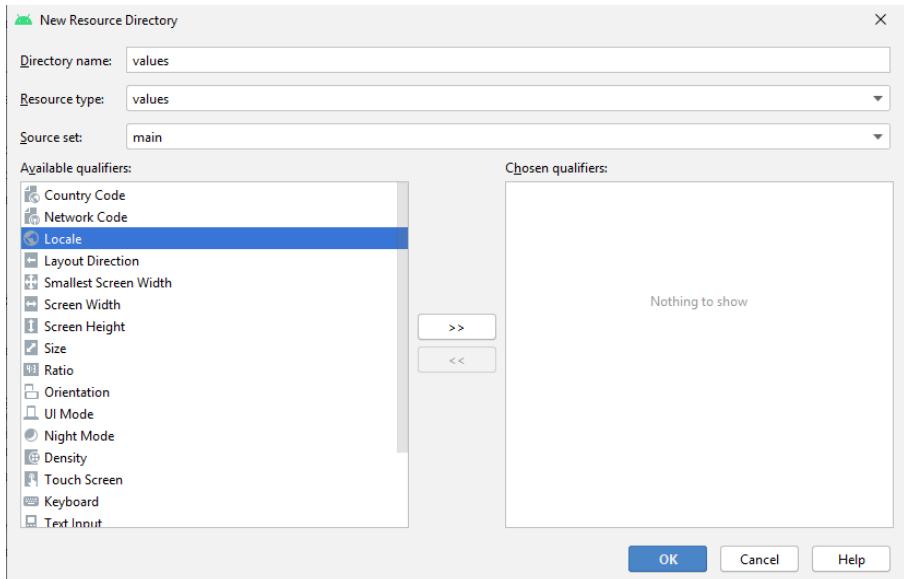
Her oprettes en values-da med en strings.xml med danske tekster, og når enheden er sat til dansk, vil den danske blive brugt.



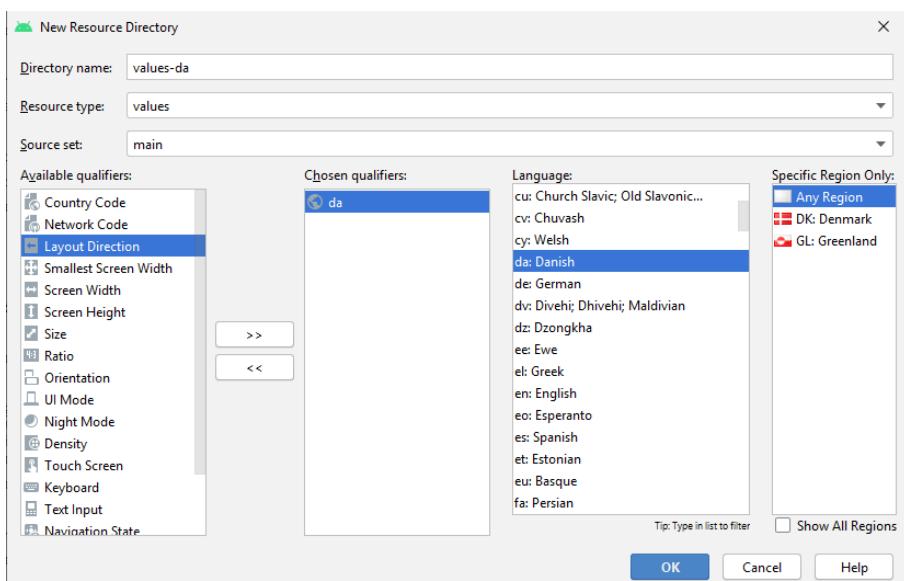
Det ses her at der vises danske tekster på brugerfladen



Mappen values-da oprettes ved at højreklikke på mappen res og vælge New Resource Directory, vælge values og Locale og klikke denne over i højre vindue.



Så kommer der en mulighed for at vælge da:Danish og mappen får navnet values-da.



I mappen values-da skal der nu lægges en string.xml ind i magen til string.xml i values, med den forskel at teksterne er danske i stedet for engelske

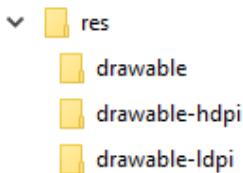
```
<resources>
    <string name="app_name">Ressourcer App</string>
    <string name="header">Ressourcer</string>
    <string name="greetingOne">Hej med dig</string>
    <string name="greetingTwo">Halløjsa, hvordan går det</string>
    <string name="btn_text">Skift Billede og Hilsen</string>
</resources>
```

I Android Studio ser det igen anderledes ud end den fysiske filstruktur. Og det kan ikke rigtig lade sig gøre at oprette en strings.xml fil her, da den ikke spørger om hvilken mappe den skal ligge i, og at man derfor ikke kan lave en fil af samme navn. Så jeg oprettede den nye strings.xml ved at kopiere den i stifinderen og rette den til.

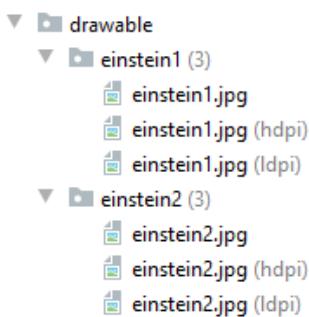


Billeder i forskellig oplosning

Mappen drawable indeholder billeder og grafiske xml-filer. Det er muligt at lave mapper til billeder i hver sin oplosning. Altså have det samme billede i flere oplosninger, så valget af billede afgøres af skærmens oplosning. Dette gøres ved at oprette mapper der hedder f.eks. drawable-ldpi og drawable-hdpi.

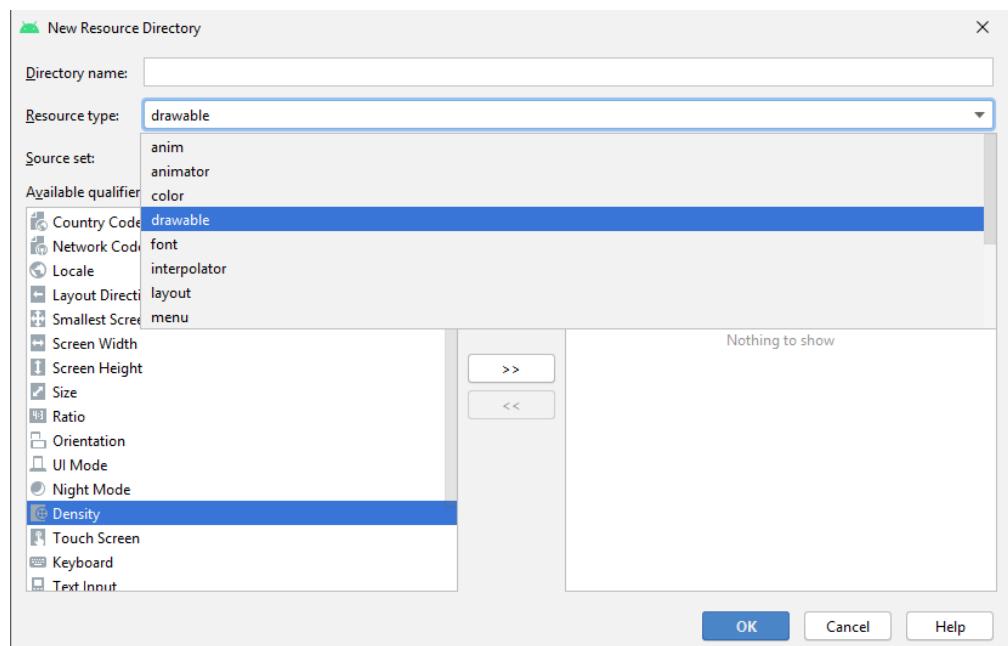


Det ser anderledes ud i Android Studio end i det fysiske filhierarki. Når der trækkes en fil ind i drawable bliver man spurgt om hvilken mappe den skal ligge i. De filer der hedder det samme bliver grupperet sammen og med mappenavn i parentes. Det er lige det omvendte af det fysiske, hvor filerne er grupperet i mapper efter oplosning.

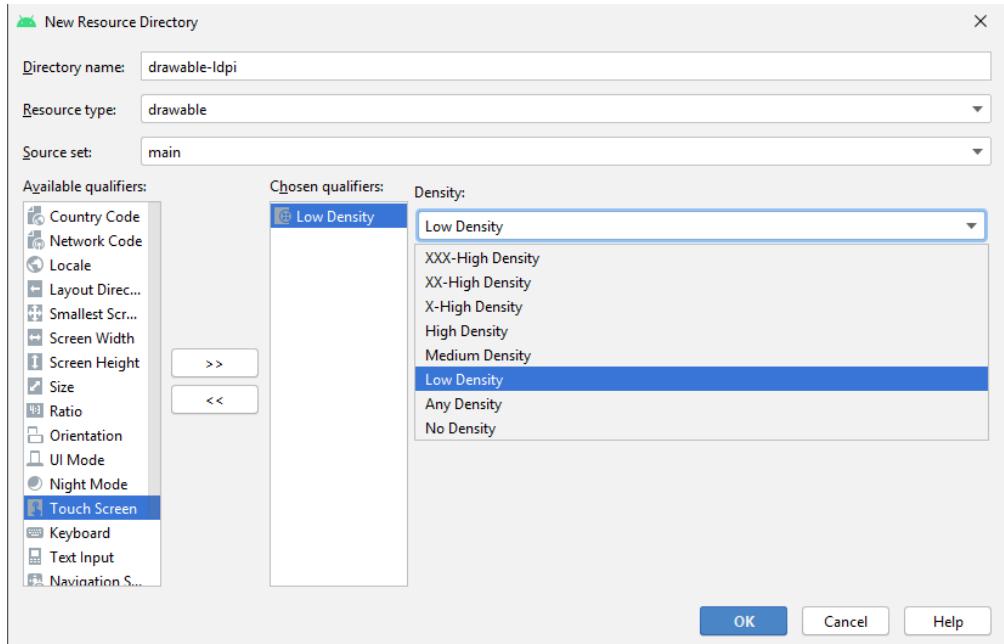


Mapperne oprettes ved at højreklikke på mappen res og vælge New Resource Directory.

Der vælges Resource type drawable og Available qualifier Density, som klikkes over i højre felt.



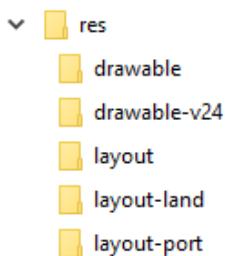
Nu kan der vælges Density og navnet bliver automatisk som det valgte.



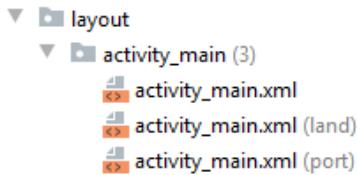
Layout til landscape og portrait

Mappen layout indeholder design af brugerflader i xml-filer. Det er muligt at lave flere mapper med layout-filer, der hedder det samme, men bruges henholdsvis når skærmen er i landscape mode eller portrait mode. Hvis den aktuelle layout ikke findes i landscape eller portrait bruges mappen layout. Der findes også en square.

Det ser ud som følgende i den fysiske filstruktur, men anderledes i Android Studio.



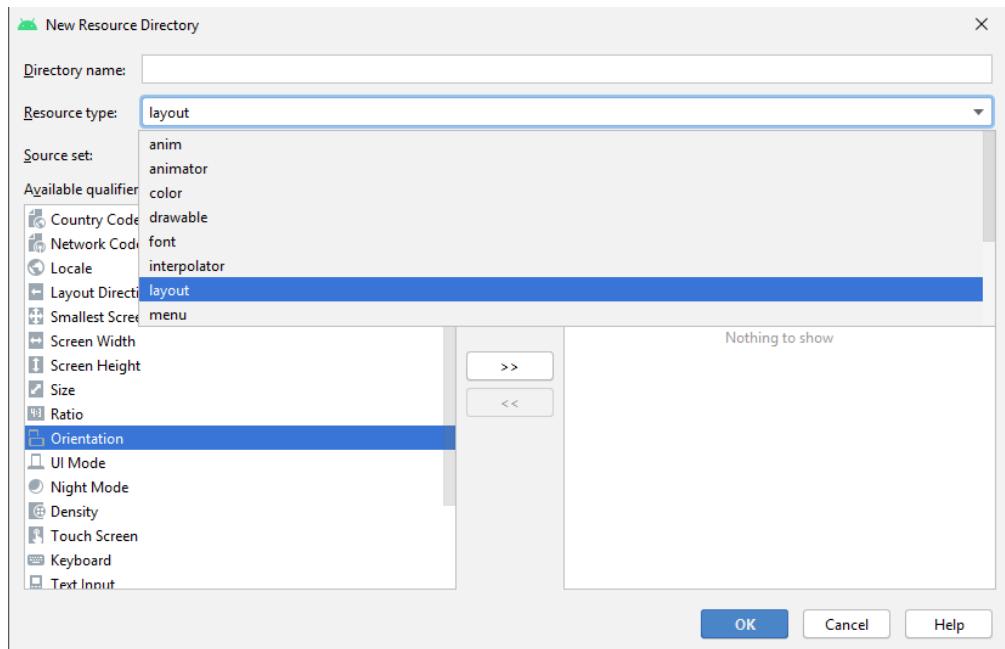
Hvis der under alle tre mapper ligger en activity_main.xml, vil det se sådan ud i Android studio.



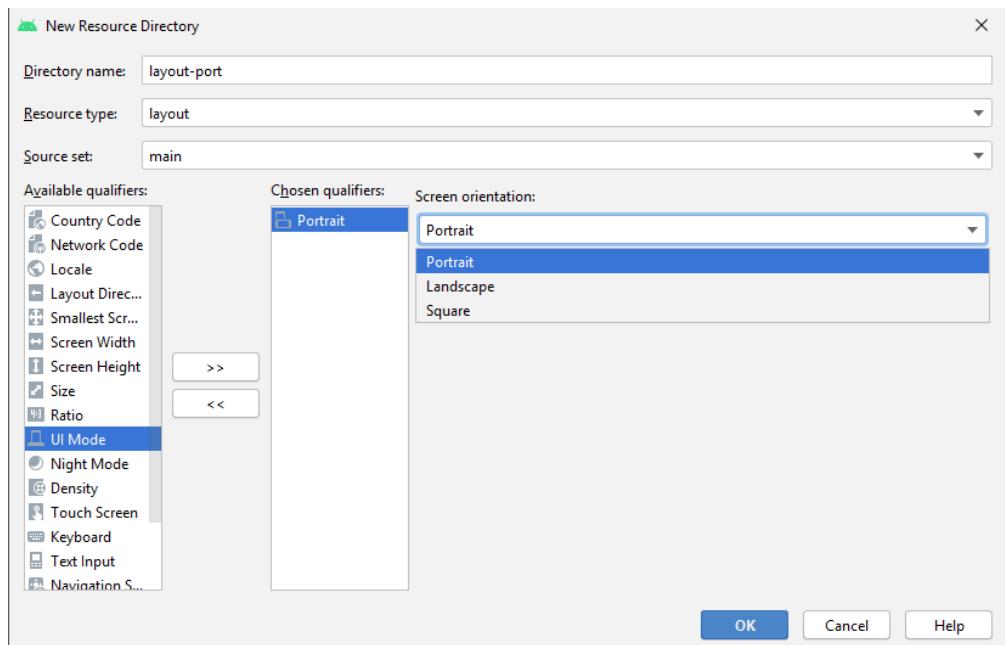
Hvis man i Android Studio renamer en af filerne, følger de andre med. Default layout udgaven er egentlig overflødig, når der både findes en landscape og en portrait.

For at oprette mapperne højre-klikkes der på mappen res og vælges New Resource Directory.

Der vælges Resource type layout, og der klikkes på Orientation i Available qualifier, som klikkes over i højre vindue.



Nu kan der vælges Portrait eller Landscape, hvilket også generer det rigtige navn layout-port.



Øvelse ressourcer

Opret en brugerflade med tekster og farver fra ressourcer.
Indret det sådan at der skiftes sprog i henhold enhedens sprog.

7. Layout

Der er to packages som indeholder klasser til at danne brugerinterface i Android.

Package android.widget indeholder mest synlige, UI elementer til skærmen. F.eks. Button.

Package android.view indeholder basis-klasser for UI elementerne. F.eks. View, og ViewGroup, som arver fra View.

Button arver fra View indirekte igennem TextView, da der jo kan vises tekst på en Button.

LinearLayout arver fra ViewGroup

Hvis der skal vises mere end en enkelt widget på skærmen, må de lægges i en layout-manager, f.eks. en LinearLayout.

Skærmens layout kan dannes i <layout>.xml-filer, men det er også muligt at oprette widget-objekter, og tilføje disse til layout, i Java koden, altså en dynamisk oprettelse af brugerfladen.

En enkelt knap på skærmen

Her oprettes en Button i Java-koden.

Den bliver vist i stedet for det layout, der er oprettet i main.xml, og den vil fyldde hele skærmen.

Hver gang der trykkes på knappen, vil tidspunktet, som vises på den, blive opdateret.

```
public class EnStorButtonActivity extends Activity {  
  
    Button btnMyButton;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        btnMyButton = new Button(this);  
        setContentView(btnMyButton);  
  
        btnMyButton.setOnClickListener(  
            new OnClickListener() {  
                public void onClick(View v) {  
                    upDateTime();  
                }  
            }  
        );  
    }  
  
    private void upDateTime()  
    {  
        btnMyButton.setText((new Date()).toString());  
    }  
}
```

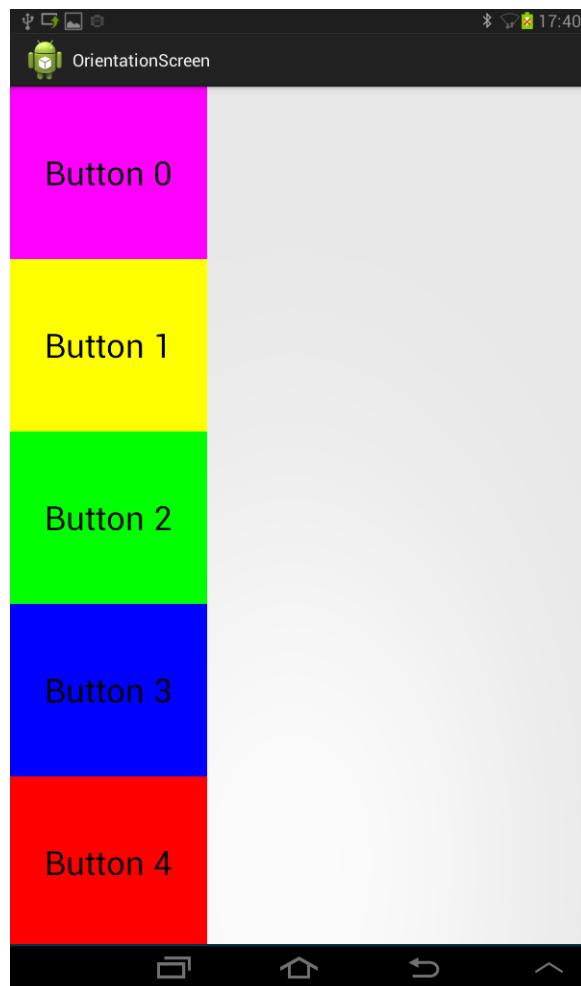


LinearLayout

Ved linear layout lægges Views efter hinanden med retning i henhold til attriubutten orientation.

I følgende eksempel er orientation vertical.

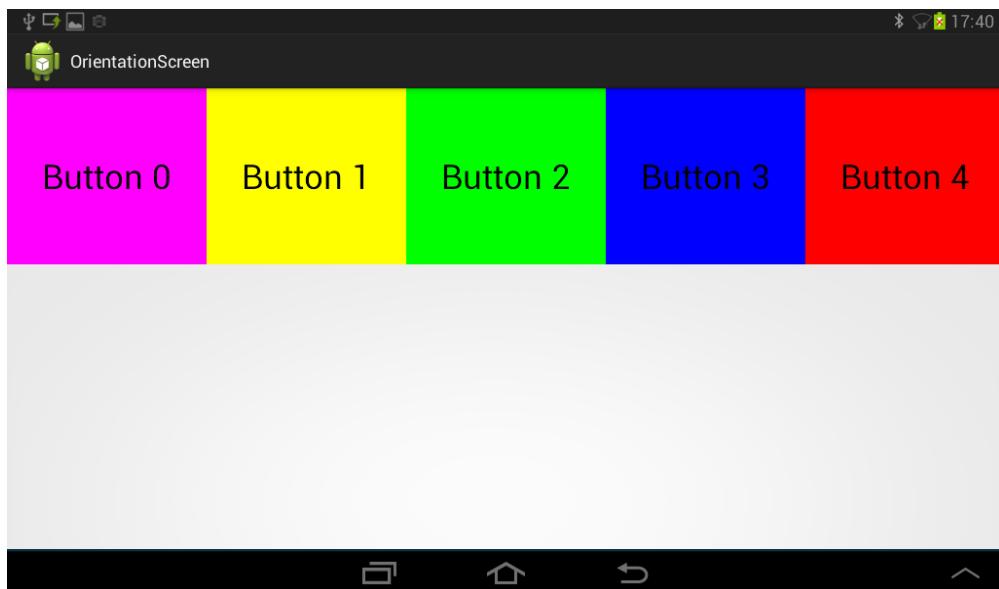
```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:orientation="vertical" >  
  
    <Button  
        android:id="@+id/button0"  
        android:layout_width="205sp"  
        android:layout_height="180sp"  
        android:textSize="35sp"  
        android:background="#FF00FF"  
        android:text="Button 0" />  
  
    ...
```



Og samme eksempel med orientation horizontal.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal" >
    <Button
        android:id="@+id/button0"
        android:layout_width="205sp"
        android:layout_height="180sp"
        android:textSize="35sp"
        android:background="#FF00FF"
        android:text="Button 0" />
    ...

```



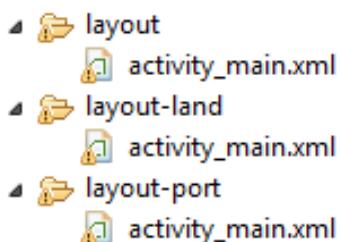
Portrait og landscape layout ved vending af skærm

Hvis de to eksempler gemmes med samme filnavn, activity_main.xml, og lægges i henholdsvis mappen layout-land og layout-port, vil Android selv finde ud af at vælge den aktuelle, når skærmen er i stilling portræt eller landscape.

Mapperne layout-land og layout-port findes ikke i et nyoprettet projekt, men skal tilføjes med nøjagtig disse navne.

Hvis en eller begge af mapperne ikke findes vil Android defaulte til layout-filen i mappen layout.

Alle layout-mapperne ligger i mappen res.



Og vises som sædvanlig med setContentView()

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

Blokering af vending af skærm

Hvis det ønskes at fastlåse skærmen i enten portrait-mode eller landscape-mode, fordi at det kun er en af dem der giver mening, kan dette gøres i Manifest.xml med koden

```
<activity
    ...
    android:screenOrientation="portrait" >
</activity>
```

Skift af layout restarter aktiviteten

Når enheden vendes sådan at der skiftes mellem portrait og landscape, vil den pågældende aktivitet starte forfra. Dette kan være uhensigtsmæssigt, hvis status i aktiviteten er ændret siden den startede, f.eks. ved et spil, hvor spillet så vil begynde forfra.

Dette kan løses ved at gemme f.eks. spil-objektet i en statisk attribut, og så spørge når aktiviteten starter om objektet er null. Hvis det er null, oprettes der et nyt spil-objekt, og hvis det ikke er null, arbejdes der videre med objektet. Når spillet så afsluttes skal objektet sættes til null, sådan at der oprettes et nyt spil-objekt hver gang spillet starter.

LinearLayout med dynamisk indhold

Her vises brugen af flere widgets i en LinearLayout. Det hele oprettet dynamisk.

LinearLayout har default orientation=horizontal, hvilket gør at widgets lægges efter hinanden horisontalt.

```
public class DynamiskLayoutActivity extends Activity
{
    @Override
```

```

public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);

    LinearLayout linLayout = new LinearLayout(this);
    linLayout.setOrientation(LinearLayout.VERTICAL);

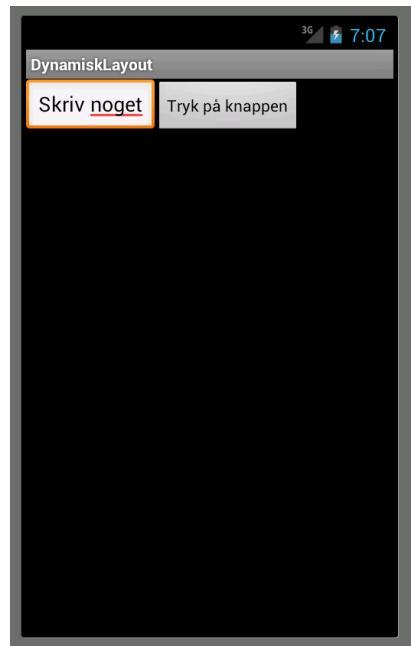
    EditText txtMyText = new EditText(this);
    txtMyText.setText("Skriv noget");

    Button btnMyButton = new Button(this);
    btnMyButton.setText("Tryk på knappen");

    linLayout.addView(txtMyText);
    linLayout.addView(btnMyButton);

    setContentView(linLayout);
}
}

```



Blanding af statisk og dynamisk layout

Her benyttes den LinearLayout, der er oprettet i <layout>.xml-filen.
orientation er sat til vertical, hvilket gør at widgets placeres efter hinanden vertikalt.

Der oprettes stadig i Java-koden en Button og en EditText, som lægges i den statiske, i XML oprettede, layout.

XML:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:id="@+id/linLayout">
</LinearLayout>

```

Java:

```

public class DynamiskLayoutActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);

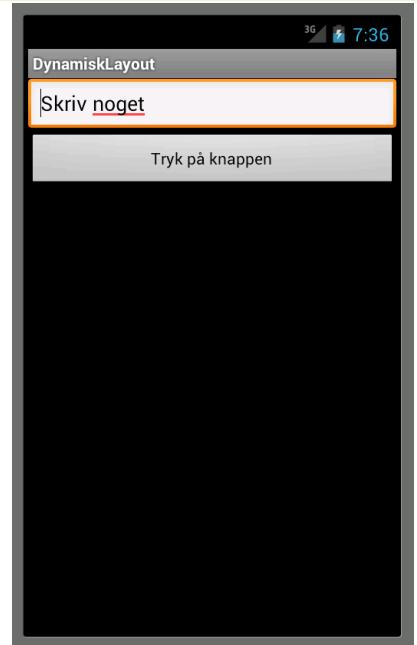
        setContentView(R.layout.main);

        //LinearLayout linLayout = new LinearLayout(this);
        LinearLayout linLayout =
            (LinearLayout) findViewById(R.id.linLayout);

        EditText txtMyText = new EditText(this);
        txtMyText.setText("Skriv noget");

        Button btnMyButton = new Button(this);
        btnMyButton.setText("Tryk på knappen");
    }
}

```



```

        linLayout.addView(txtMyText);
        linLayout.addView(btnMyButton);

        //setContentView(linLayout);
    }
}

```

LinearLayout orientation kunne også sættes i koden med kommandoen

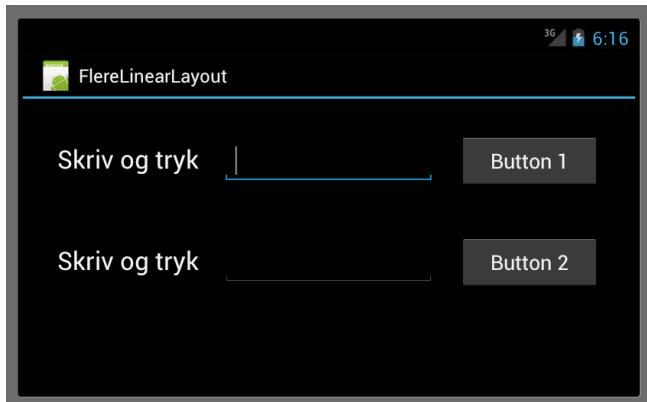
```
linLayout.setOrientation(LinearLayout.VERTICAL);
```

Nestede linearlayouts

Det er muligt at kombinere LinearLayouts, sådan at der fås rækker af widgets under hinanden, eller som det nok ikke er så almindeligt, kolonner af widgets ved siden af hinanden.

Efterfølgende vises resultatet af, og main.xml koden, som opretter to LinearLayout med orientation horizontal inde i en LinearLayout med orientation vertical. Derved kan der oprettes to rækker af widgets under hinanden.

Der er desuden brugt nogle ekstra properties som padding, som er luft udenom en widget, margin, som er luft inde i en container, samt weight, hvormed bredden af widgets ved siden af hinanden får hver deres del. Her får TextView 2 dele, EditText 3 dele og Button 2 dele af pladsen. Et bedre overblik kunne evt. fås ved at bruge procenter, f.eks. 30, 40 og 30 procent. Altid sørge for at summen er 100, så kan man nemt se på hver enkelt, hvilken del den får.



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:padding="10dp">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="10dp">
            <TextView
                android:layout_width="0dp"

```

```

        android:layout_height="wrap_content"
        android:text="Skriv og tryk"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_margin="10dp"
        android:layout_weight="2"/>
    <EditText
        android:id="@+id/editText1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="3"/>
    <Button
        android:id="@+id/button1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Button 1"
        android:layout_margin="10dp"
        android:layout_weight="2"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="10dp">
    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Skriv og tryk"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:layout_margin="10dp"
        android:layout_weight="2"/>
    <EditText
        android:id="@+id/editText2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="3"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:text="Button 2"
        android:layout_margin="10dp"
        android:layout_weight="2"/>
</LinearLayout>
</LinearLayout>

```

```

public class FlereLinearLayoutActivity extends Activity
{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}

```

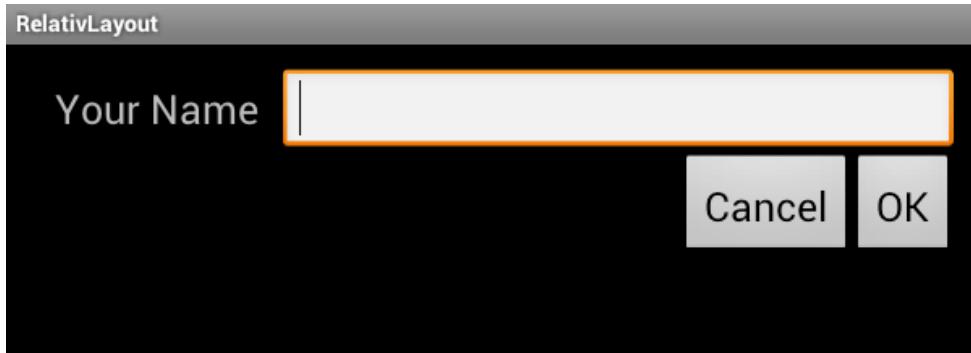
FrameLayout

I FrameLayout placeres views på en stack, hvor et view er i front og andre er bagved.



RelativeLayout

I en RelativeLayout placeres widgets i forhold til containeren samt i forhold til andre widgets.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding="15dp">
    <TextView android:id="@+id/Label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Your Name"
        android:layout_alignBaseline="@+id/entry"
        android:layout_alignParentLeft="true"
        android:padding="15dp"
        android:textSize="25dp"/>
    <EditText
        android:id="@+id/entry"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@+id/Label"
        android:layout_alignParentTop="true"
        android:textSize="25dp"/>
    <Button
        android:id="@+id/ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/entry"
        android:layout_alignRight="@+id/entry"
        android:text="OK"
        android:padding="15dp"
        android:textSize="25dp"/>
    <Button
        android:id="@+id/cancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toLeftOf="@+id/ok"
        android:layout_alignTop="@+id/ok"
        android:text="Cancel"
        android:padding="15dp"
        android:textSize="25dp"/>
</RelativeLayout>
```

Positioner relativ til container

Først må en eller flere widgets placeres i forhold til containeren(RelativeLayout).

Følgende relationer kan sættes = true/false:

- android:layout_alignParentTop
- android:layout_alignParentBottom
- android:layout_alignParentLeft
- android:layout_alignParentRight
- android:layout_centerHorizontal
- android:layout_centerVertical
- android:layout_centerInParent

Placeringen er baseret på den samlede plads som en widget fylder, dvs. dens naturlige størrelse plus eventuel padding.

Positioner relativ til andre widgets

Fire properties kontrollerer positionen af en widget i forhold til andre widgets:

- android:layout_above
- android:layout_below
- android:layout_toLeftOf
- android:layout_toRightOf

Der findes desuden fire properties der kontrollerer en widget's alignment/tilpasning til andre widgets:

- android:layout_alignTop
- android:layout_alignBottom
- android:layout_alignLeft
- android:layout_alignRight
- android:layout_alignBaseline

Den sidste layout_alignBaseline property er anvendelig til at placere labels og tekstfelter sådan at teksten synes naturlig. Da EditText har en boks omkring sig, som en TextView ikke har, vil layout_alignTop gøre at tekstlinien på de to ikke er på samme højde.

For at en widget kan placeres i forhold til en anden widget, må den anden widget have en id som den ene widget kan referere til.

Hvis widget B skal placeres til højre for widget A, skal denne have en id, f.eks. id="@+id/widget_a" og widget B skal så have property android:layout_toRightOf = "@id/widget_a"

Evaluering af id

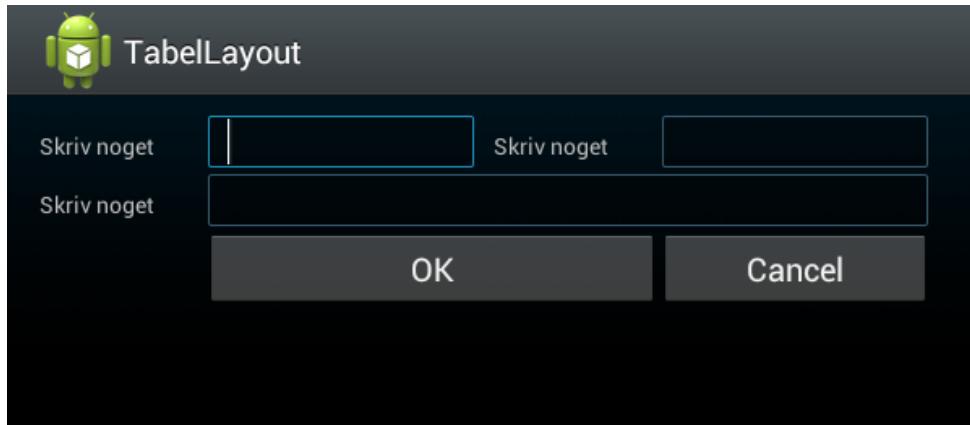
Før Android 1.6 kunne der kun refereres til et id, der var defineret tidligere i XML-filen, hvilket jo kunne være problematisk, fordi der så kun kunne refereres til widgets, der lå højere oppe på skærmen.

Dette er løst med at indføre en "two passes process" sådan at alle id er registreret inden referencerne evalueres.

TableLayout

I en TableLayout lægges der widgets i TableRows. Hver widget ligger i en kolonne. Der er mindst lige så mange kolonner, som den række der har flest widgets. Men der kan godt være flere, da property layout_column, som angiver hvilken kolonne-nummer en widget skal placeres i, kan angives med et nummer der ikke eksisterede ellers, og layout_span, som angiver hvor mange kolonner en widget skal spænde over, kan have en værdi, der får den aktuelle widget til at fortsætte ud i kolonner, der ellers ikke eksisterede.

Et view kan spænde over flere columns, men kan ikke spænde over flere rows.



Property ems="7" angiver at bredden skal svare til 7 karakterer af den aktuelle font-size. En em er altså font-størrelsen, bredden af et M. Dette stammer fra CSS.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp">
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Skriv noget"
            android:ems="7"
            android:paddingLeft="10dp"/>
        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="7" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Skriv noget"
            android:ems="7"
            android:paddingLeft="10dp"/>
        <EditText
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:ems="7"/>
    </TableRow>
    <TableRow>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Skriv noget"
            android:ems="7"
            android:paddingLeft="10dp"/>
    </TableRow>
```

```

        android:layout_height="wrap_content"
        android:text="Skriv noget"
        android:paddingLeft="10dp" />

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_span="3" />
</TableRow>
<TableRow>
    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="OK"
        android:layout_column="1"
        android:layout_span="2" />
    <Button
        style="?android:attr/buttonStyleSmall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Cancel"
        android:layout_column = "3" />
</TableRow>
</TableLayout>

```

GridLayout

GridLayout kan virke mere avanceret end TableLayout, men der er ikke nogen nem mulighed for at lade kolonnernes bredde være lige stor, og heller ikke lade kolonnerne fyldte hele bredden af skærmen.

Den er bedst når der fyldes indhold i cellerne som er lige store.

I GridLayout kan views både spænde over flere rækker og flere kolonner, i modsætning til TableLayout, der kun kan spænde over flere kolonner.

GridLayout har columnCount og rowCount, så der på forhånd sættes hvor mange rækker og kolonner gridden skal have.

I de views der indsættes angives positionen med layout_column og layout_row.
Hvis de skal fyldes mere end en celle bruges layout_rowSpan og layout_columnSpan.

Der kan bruges layout_gravity="fill_vertical" og layout_gravity="fill_horizontal"
til at få views til at fyldes det angivne span.

Ellers er det de største views der bestemmer hvor høje og brede cellerne skal være.

Regnemaskine



```
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal|center_vertical"
    android:columnCount="3"
    android:rowCount="5" >
    <TextView
        android:id="@+id/txtRegn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_column="0"
        android:layout_row="0"
        android:layout_columnSpan="3"
        android:text="Regnemaskine"
        android:textSize="25sp" />
    <Button
        android:layout_column="0"
        android:layout_row="1"
        android:text="1" />
    <Button
        android:layout_column="1"
        android:layout_row="1"
        android:text="2" />
    <Button
        android:layout_gravity="fill_vertical"
        android:layout_column="2"
        android:layout_row="1"
        android:layout_rowSpan="3"
        android:text="6" />
    <Button
        android:id="@+id/button4"
        android:layout_column="0"
        android:layout_row="2"
        android:text="3" />
    <Button
        android:id="@+id/button5"
        android:layout_column="1"
        android:layout_row="2"
        android:text="4" />
    <Button
        android:layout_gravity="fill_horizontal"
        android:layout_column="0"
        android:layout_columnSpan="2"
        android:layout_row="3"
        android:text="5" />
</GridLayout>
```

ScrollView

Følgende program viser en række farvebjælker, som har højden 100pixels. Alt efter skærmstørrelsen kan de være der, når skærmen er lodret, men måske ikke når skærmen er vandret.

Derfor sættes der et ScrollView uden om det hele.

Læg mærke til at der er brugt tags af typen View, som er den klasse alle widgets arver fra. Den kan have en størrelse og en farve. Dernæst en TextView, der viser farve-værdien som hex-RGB. Hvert par er lagt i en TableRow i en TableLayout.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android=
        "http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="0">
        <TableRow>
            <View
                android:layout_height="100px"
                android:background="#000000"/>
            <TextView android:text="#000000"
                android:padding="5px"
                android:layout_gravity=
                    "center_vertical" />
        </TableRow>
        <TableRow>
            <View
                android:layout_height="100px"
                android:background="#440000" />
            <TextView android:text="#440000"
                android:paddingLeft="5px"
                android:layout_gravity="center_vertical" />
        </TableRow>
        <TableRow>
            <View
                android:layout_height="100px"
                android:background="#884400" />
            <TextView android:text="#884400"
                android:paddingLeft="5px"
                android:layout_gravity="center_vertical" />
        </TableRow>
        <TableRow>
            <View
                android:layout_height="100px"
                android:background="#AA8844" />
            <TextView android:text="#AA8844"
                android:paddingLeft="5px"
                android:layout_gravity="center_vertical" />
        </TableRow>
        <TableRow>
            <View
                android:layout_height="100px"
                android:background="#FFAA88" />
            <TextView android:text="#FFAA88"
                android:paddingLeft="5px"
                android:layout_gravity="center_vertical" />
        </TableRow>
    </TableLayout>
</ScrollView>
```



```

        android:layout_gravity="center_vertical" />
    </TableRow>
    <TableRow>
        <View
            android:layout_height="100px"
            android:background="#FFFFAA" />
        <TextView android:text="#FFFFAA"
            android:paddingLeft="5px"
            android:layout_gravity="center_vertical" />
    </TableRow>
    <TableRow>
        <View
            android:layout_height="100px"
            android:background="#FFFFFF" />
        <TextView android:text="#FFFFFF"
            android:paddingLeft="5px"
            android:layout_gravity="center_vertical" />
    </TableRow>
</TableLayout>
</ScrollView>
```

Når skærmen vendes er der ikke længere plads i højden til alle farvebjælker. Her træder ScrollView til og der kan trækkes op og ned på skærmen. Der vises en scrollbar i højre side, når der trækkes, men den forsvinder



med det samme igen, så det er ikke muligt at tage et skærmbillede med den på.

Øvelser

Opgave Regnemaskine med TableLayout og RadioButtons

Se slutningen af kap 9

Regnemaskine med tal-knapper i GridLayout

Opret en regnemaskine med 10 tal-knapper, samt +, -, *, / -knapper og en = -knap.

Ved at trykke på knapperne skal der skrives i display, og ved tryk på = skal resultatet findes og skrives i display.

8. Dialog, Toast og Snackbar

Dialog

En Dialog kan ikke returnere noget, men har callback-metoder i sig som udfører det, der skal ske, når brugeren evt. giver sit tekst-input og gør sit valg. Det vil sige at den kode, der opretter og viser Dialogen, udføres videre, så der kan ikke her laves noget, der er afhængigt af hvad der vælges i dialogen, fordi dette sker langt senere. Dialogen vises indtil brugeren gør sit valg, selv om programmet udføres videre.

Det er muligt at oprette brugerinterface på en Dialog hvor brugeren kan give sit input, og derudover kan der være op til 3 knapper at vælge imellem, som kalder en af callback-metoderne og afslutter dialogen.

Først et eksempel på en dialog hvor der vælges imellem to muligheder, nemlig at slette noget eller ikke slette noget.

Dialog 1



```
AlertDialog.Builder alert = new AlertDialog.Builder(this);

alert.setTitle("Delete Person");
alert.setMessage("Ønskes det at slette den aktuelle Person");

alert.setPositiveButton("Ja", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int whichButton)
    {
        // Slet den aktuelle
        // og vend evt. tilbage til en anden side
    }
});

alert.setNegativeButton("Nej", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int whichButton)
    {
        // Slet ikke noget
    }
});

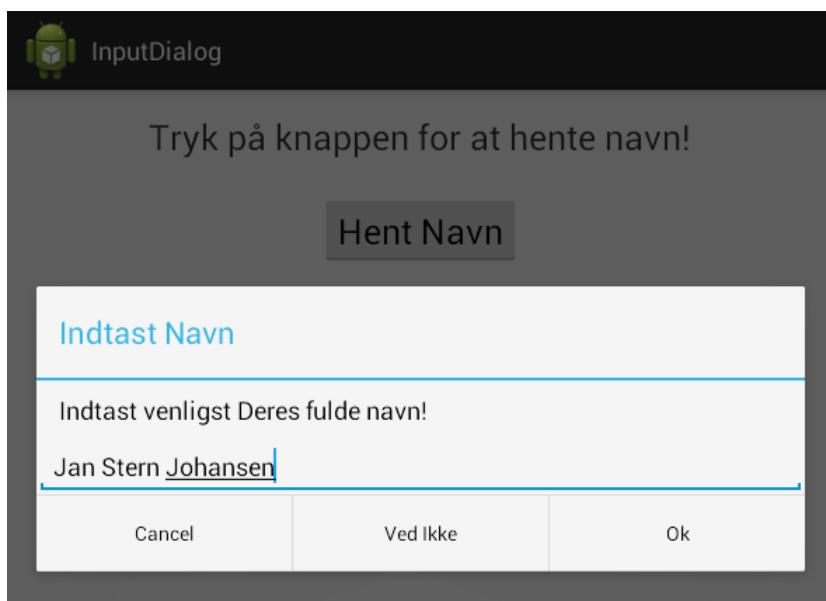
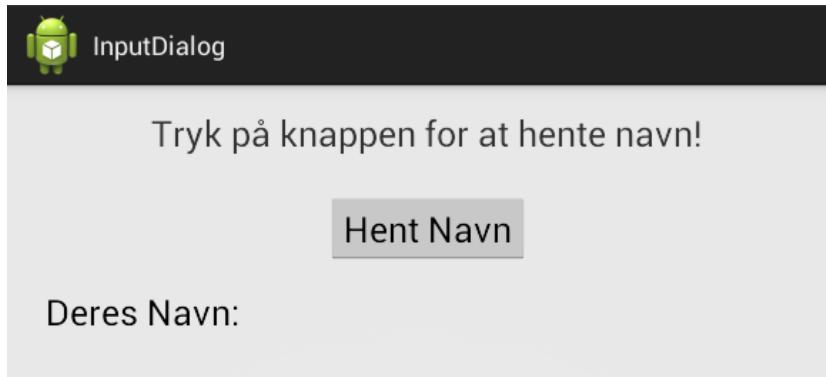
alert.show();
```

Dialog 2

Et mere avanceret eksempel på en Dialog, hvor brugeren skal indtaste sit navn og trykke på en af knapperne.

Her vil Dialogens knappers callback-metoder, skrive i den brugerflade, hvor Dialogen oprettes fra.

Dialogen oprettes når der trykkes på knappen Hent Navn.




```

package dk.tec.jaj.inputdialog;

import ...

public class MainActivity extends Activity implements OnClickListener
{
    Button btnHent;
    TextView txtResultat;

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

        btnHent = (Button)findViewById(R.id.btnHent);
        btnHent.setOnClickListener(this);
        txtResultat = (TextView)findViewById(R.id.txtResultat);
    }

    @Override
    public void onClick(View v)
    {
        AlertDialog.Builder alert = new AlertDialog.Builder(this);

        alert.setTitle("Indtast Navn");
        alert.setMessage("Indtast venligst Deres fulde navn!");

        final EditText input = new EditText(this);
        alert.setView(input);

        alert.setPositiveButton("Ok", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int whichButton)
            {
                String value = input.getText().toString();
                txtResultat.setText(value);
            }
        });

        alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int whichButton)
            {
                txtResultat.setText("De skulle skrive Deres navn!!!");
            }
        });

        alert.setNeutralButton("Ved Ikke", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int whichButton)
            {
                txtResultat.setText("Ved De ikke hvad De hedder??");
            }
        });

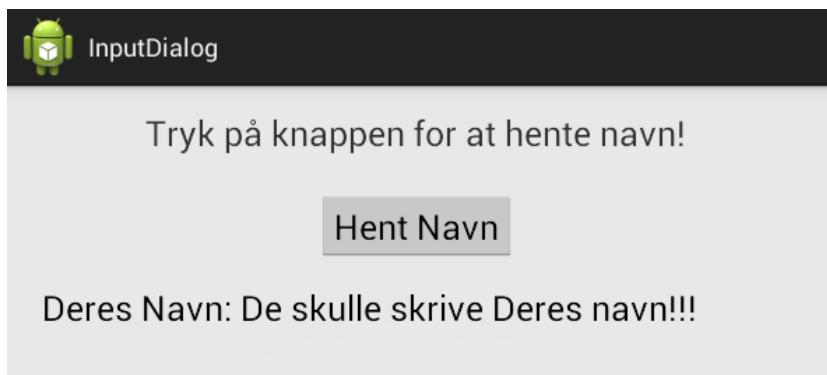
        alert.show();
    }
}

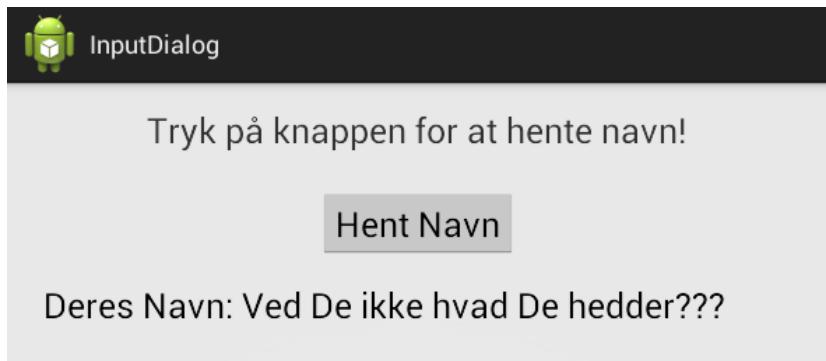
```

```

<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"
    android:gravity="center_horizontal"
    android:padding="16dp"
    tools:context="dk.tec.jaj.inputdialog.MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="Tryk på knappen for at hente navn!" />
    <Button
        android:id="@+id	btnHent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:textSize="25sp"
        android:text="Hent Navn" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:padding="@dimen/activity_vertical_margin">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="25sp"
            android:text="Deres Navn: " />
        <TextView
            android:id="@+id/txtResultat"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="25sp"
            android:text="" />
    </LinearLayout>
</LinearLayout>

```





Toast

```
Toast.makeText(MainActivity.this, "De skulle skrive Deres navn!!!", Toast.LENGTH_LONG).show();
```



Snackbar

Snackbar minder meget om Toast, men kan have en action i form af en knap, der kan trykkes på. Problemet er bare at der kan kun være én action, så hvis der udføres noget, når der trykkes på action, er der kun den mulighed ikke at trykke på den, hvis der ikke skal gøres noget. Derfor må den sættes til at forsvinde af sig selv med Snackbar.LENGTH_LONG eller Snackbar.LENGTH_SHORT og så skal man altså være lidt hurtig hvis man, som her, vil fortryde.

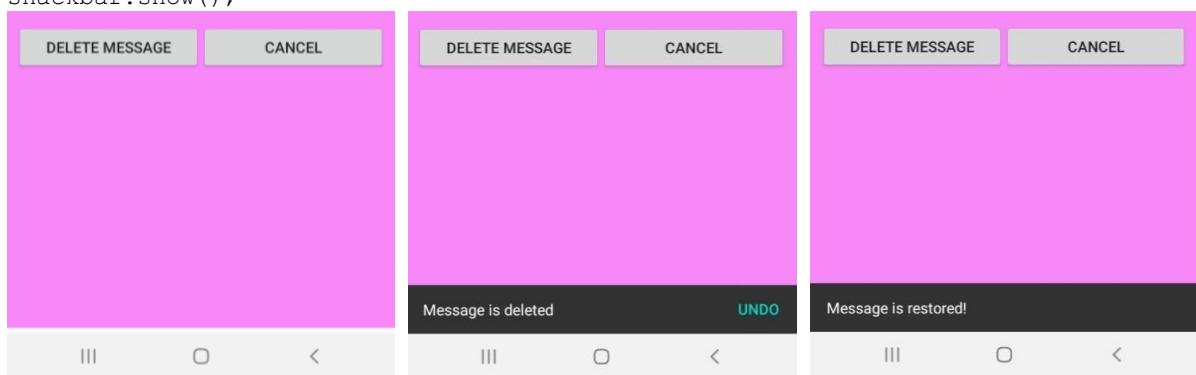
```
Snackbar snackbar = Snackbar
    .make(linLayout, "Message is deleted", Snackbar.LENGTH_LONG)
    .setAction("UNDO", new View.OnClickListener() {
        @Override
        public void onClick(View view) {
```

```

        .make(linLayout, "Message is restored!", Snackbar.LENGTH_SHORT);
        snackbar1.show();
    });
}

snackbar.show();

```

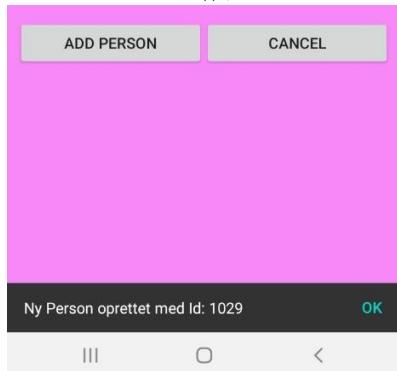


Det kan også være en Snackbar, med `Snackbar.LENGTH_INDEFINITE`, som vises indtil der bliver trykket på den. Der kan så, som her, ske intet andet end at den forsvinder. Der kunne også ske noget som så skal ske under alle omstændigheder, som at skifte til en anden side. Der er altså ikke tale om noget valg.

```

Snackbar snackbar = Snackbar
    .make(linLayout, "Ny Person oprettet med Id: " + newId, Snackbar.LENGTH_INDEFINITE)
    .setAction("OK", new View.OnClickListener() {
        @Override
        public void onClick(View view) {
        }
    });
snackbar.show();

```



9. Widgets og Views

En widget er en komponent, der kan vises på skærmen, og med sådanne opbygges et bruger-interface.

En widget arver fra klassen View, som er basis-klassen for brugerinterface-komponenter. En View fylder en rektangulær plads på skærmen og er ansvarlig for at tegne denne på skærmen og håndtering af events.

Klassen ViewGroup, som arver fra View, er basis-klassen for layouts, der er usynlige containere, som kan indeholde andre View- eller ViewGroups og definere deres layout-egenskaber.

Der er ca. 95 klasser, som arver fra klassen View.

Herunder er vist arvstrukturen med nogle af de mest anvendte views.

```
java.lang.Object
    android.view.View
        android.support.v7.widget.Space
        android.view.ViewGroup
            android.widget.RelativeLayout
            android.widget.LinearLayout
                android.widget.ActionMenuView
                android.widget.RadioGroup
                android.widget.TableLayout
            android.widget.FrameLayout
                android.widgetScrollView
                android.widget.DatePicker
            android.widget.AdapterView
                android.widget.AbsSpinner
                android.widget.Spinner
        android.widget.ImageView
            android.widget.ImageButton
        android.widget.AnalogClock
        android.widget.TextView
            android.widget.EditText
            android.widget.Textclock
        android.widget.Button
            android.widget.CompoundButton
            android.widget.RadioButton
            android.widget.CheckBox
            android.widget.Switch
            android.widget.ToggleButton
```

Gravity

`android:layout_gravity` is the **Outside** gravity of the View. That means, to specify the direction in which the View should touch its parent's border.

`android:gravity` is the **Inside** gravity of that View. This means, in which direction its contents should align.

HTML/CSS Equivalents:

`android:layout_gravity` = `float` in CSS

`android:gravity` = `text-align` in CSS

Easy trick to remember: Take "layout-gravity" as "Lay-outside-gravity"

```
<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:layout_margin="20sp"
    android:gravity="center_horizontal"
    tools:context="com.example.testings.MainActivity" >

    <TextView
        android:layout_width="500sp"
        android:layout_height="wrap_content"
        android:background="#AAAAAA"
        android:textSize="25sp"
        android:textStyle="bold"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal"
        android:text="Hello World" />
</LinearLayout>
```

Layoutets

```
    android:gravity="center_horizontal"
```

sætter alt indhold i layoutet indholdet til at være centreret horisontalt.

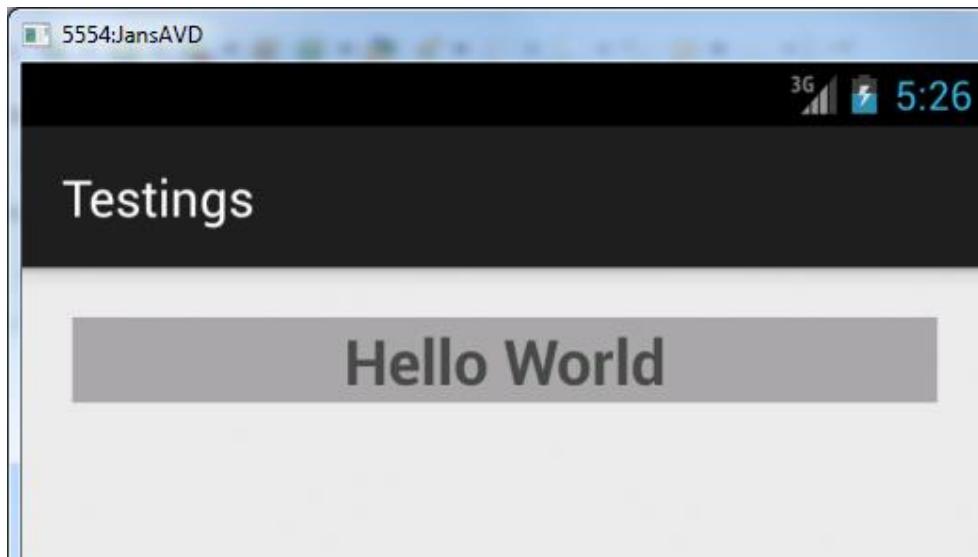
Det har samme virkning over for textviewet som dets

```
    android:layout_gravity="center_horizontal"
```

Hvor textviewets

```
android:gravity="center_horizontal"
```

sætter dets tekst til at være centreret



Button

Hvis der direkte sættes baggrundsfarve på en Button, bliver den grim. Den mister kanten med de afrundede hjørner og mister den visuelle virkning ved tryk på den.



```
btnButton.setBackgroundColor(0xFF00FF00);
```

Baggrundsfarven kan derimod sættes ved at hente baggrunden og så sætte farve på den.



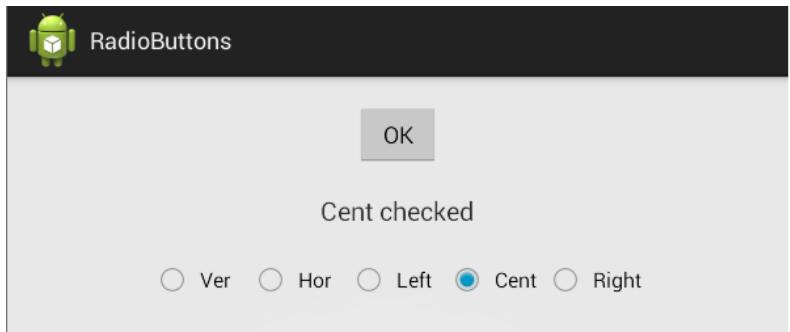
```
btnButton.getBackground().setColorFilter(  
    new LightingColorFilter(0xFF00FF00, 0xFFAA0000));
```

Eller sådan.



```
btnButton.getBackground().setColorFilter(  
    0xFF00FF00, PorterDuff.Mode.MULTIPLY);
```

RadioButton og RadioGroup



RadionButtons skal omgives af en RadioGroup for at de kan virke med hinanden.

RadioGroup virker samtidig som et layout, der kan være horizontal eller vertical, samt højre, venstre og centerstilles.

I RadioGroup kan der spørges om hvilken RadioButton, der er checked. Der kan også oprettes eventhandler OnCheckedChangeListener på RadioGroup. Så det eneste som id på den enkelte RadioButton skal bruges til er, ikke at finde den i Java-koden, men at kunne sammenligne med den id, som RadioGroup fortæller, der er checked. Der kan også på RadioGroup sættes hvilken RadioButton der skal være checked fra starten, ved at angive dens id. Check fra starten kan også sættes på den enkelte RadioButton.

```
<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:padding="20dp"
    android:orientation="vertical"
    tools:context="com.example.radiobuttons.MainActivity" >

    <Button
        android:id="@+id/btnOK"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="OK" />
    <TextView
        android:id="@+id/txtChecked"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_margin="20dp"
        android:textSize="20sp"
        android:text="Ver checked" />

    <RadioGroup
        android:id="@+id/radioGroup1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:checkedButton="@+id/rdbVer">
        >
        <RadioButton
            android:id="@+id/rdbVer"
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="Ver" />
        <RadioButton
            android:id="@+id/rdbHor"
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:text="Hor" />
        <RadioButton
            android:id="@+id/rdbLeft"
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:text="Left" />
        <RadioButton
            android:id="@+id/rdbCenter"
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:text="Cent" />
        <RadioButton
            android:id="@+id/rdbRight"
            android:layout_width="75dp"
            android:layout_height="wrap_content"
            android:text="Right" />
    </RadioGroup>
</LinearLayout>
```

```

public class MainActivity extends Activity implements OnCheckedChangeListener, OnClickListener
{
    RadioGroup rdg;
    Button btnOK;
    TextView txtChecked;

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

        rdg = (RadioGroup) findViewById(R.id.radioGroup1);
        rdg.setOnCheckedChangeListener(this);

        Button btnOK = (Button) findViewById(R.id.btnOK);
        btnOK.setOnClickListener(this);

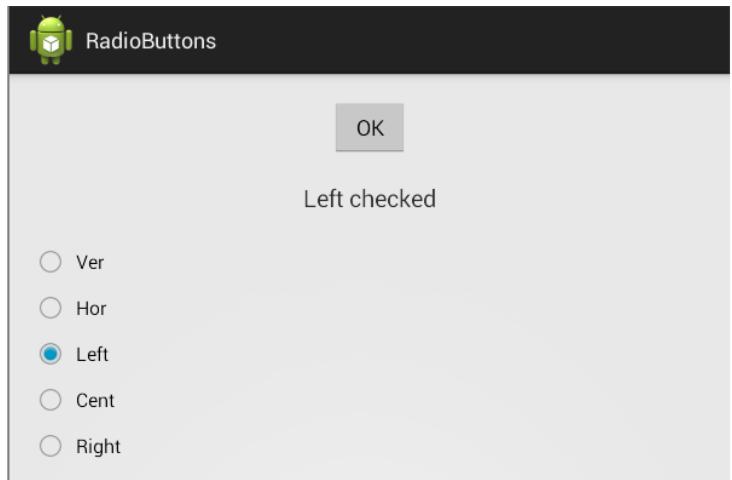
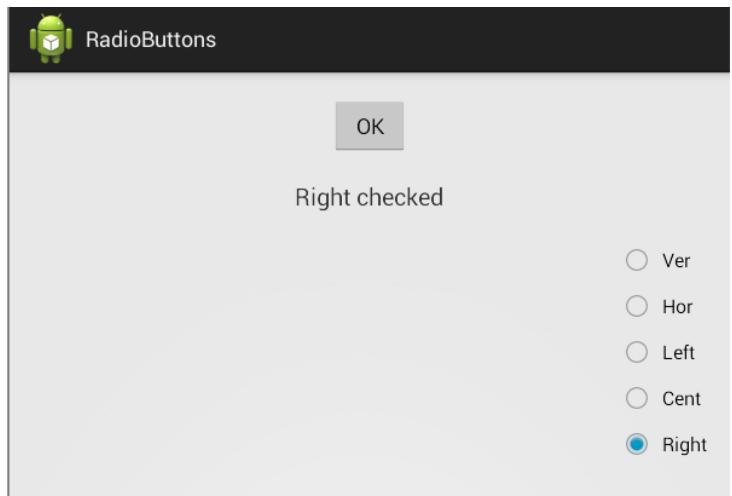
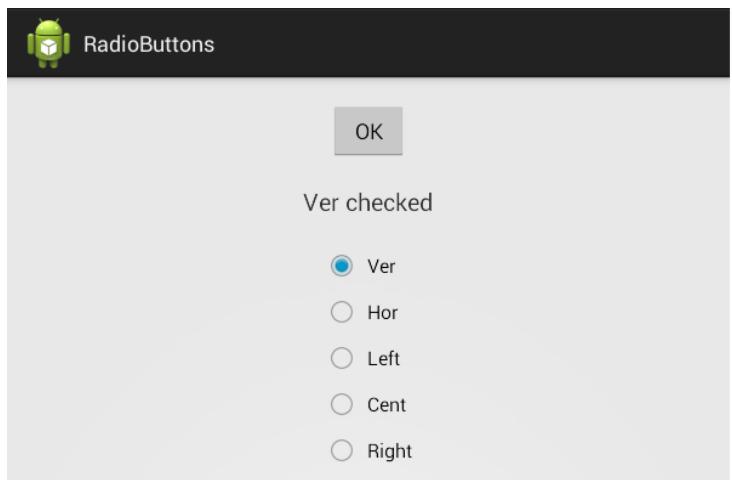
        txtChecked = (TextView) findViewById(R.id.txtChecked);
    }

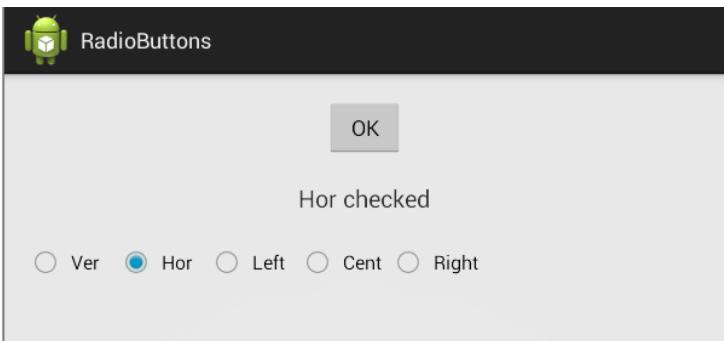
    @Override
    public void onCheckedChanged(RadioGroup group, int checkedId)
    {
        testChecked(checkedId);
    }

    @Override
    public void onClick(View v)
    {
        int checkedId = rdg.getCheckedRadioButtonId();
        testChecked(checkedId);
    }

    private void testChecked (int checkedId)
    {
        switch(checkedId)
        {
            case R.id.rdbHor:
                rdg.setOrientation(LinearLayout.HORIZONTAL);
                txtChecked.setText("Hor checked");
                break;
            case R.id.rdbVer:
                rdg.setOrientation(LinearLayout.VERTICAL);
                txtChecked.setText("Ver checked");
                break;
            case R.id.rdbLeft:
                rdg.setGravity(Gravity.LEFT);
                txtChecked.setText("Left checked");
                break;
            case R.id.rdbCenter:
                rdg.setGravity(Gravity.CENTER);
                txtChecked.setText("Cent checked");
                break;
            case R.id.rdbRight:
                rdg.setGravity(Gravity.RIGHT);
                txtChecked.setText("Right checked");
                break;
        }
    }
}

```





Man kan også spørge på de enkelte radiobuttons om de er checked

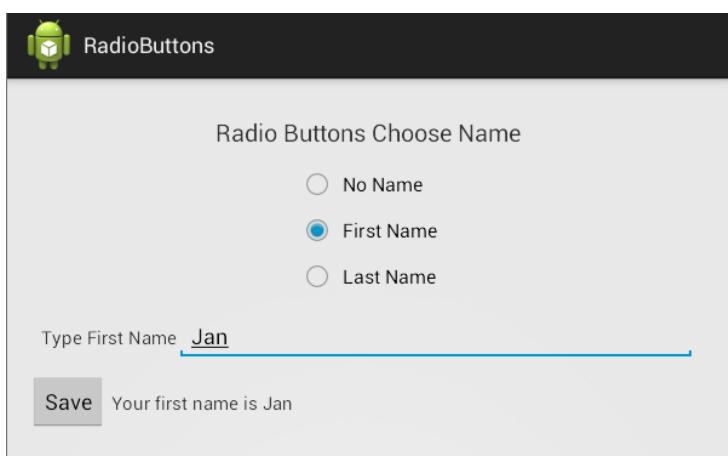
```
if(((RadioButton)findViewById(R.id.rdbCenter)).isChecked())...
```

Øvelse

Opret et program med brugerfladen som vist herunder.

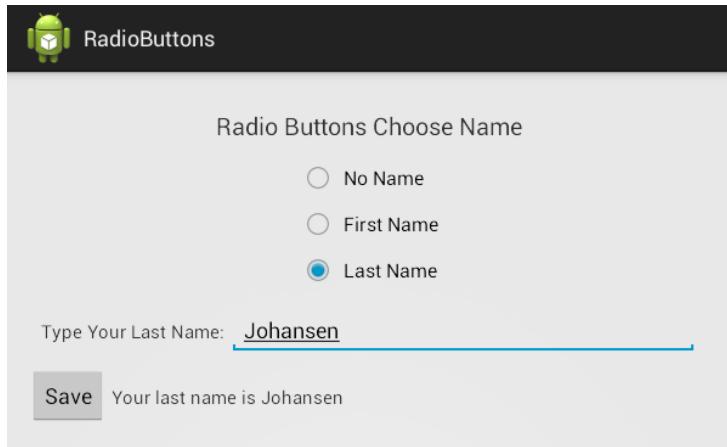
Når der vælges radiobutton med teksten First Name, skal brugerfladen vise Type First Name: og editview til at indtaste fornavn.

Ved tryk på Save skal der ved siden af denne vises teksten Your first name is <det indtastede>



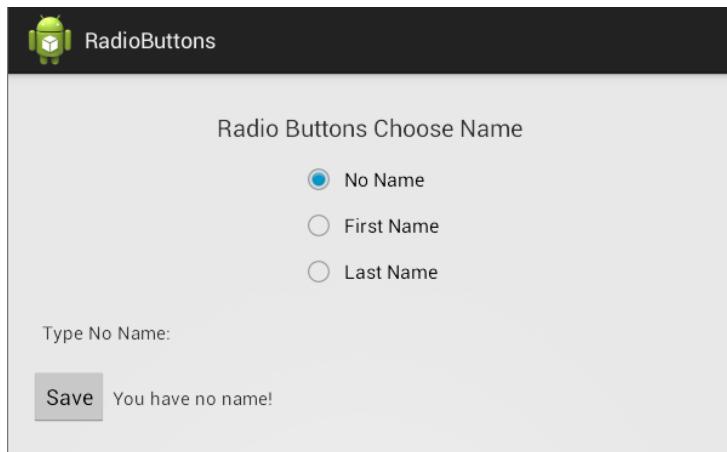
Når der vælges radiobutton med teksten Last Name, skal brugerfladen vise Type Last Name: og vise editview til at indtaste efternavn.

Ved tryk på Save skal der ved siden af denne vises teksten Your last name is <det indtastede>.

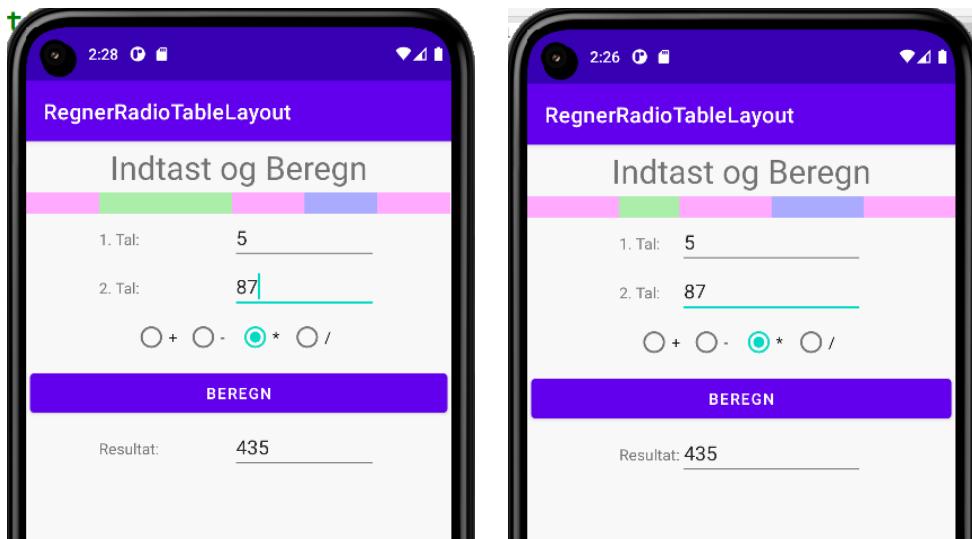


Når der vælges radiobutton med teksten No Name, skal brugerfladen vise Type No Name: og skjule editview til indtastning.

Ved tryk på Save skal der ved siden af denne vises teksten You have no name!



Opgave Regnemaskine med TableLayout og RadioButtons



Skriv et Android program som kan beregne to tal med de fire regnearter som vist på skærbillederne.

Der skal indtastes to tal, regnearten vælges på radiobuttons og beregningen udføres når der trykkes på knappen Beregn.

Layoutet skal laves med TableLayout.

Lidt forklaring til TableLayout: På de viste skærbilleder er der en række af 5 TextViews uden tekst, men med farver, for at oprette og vise 5 kolonner. I det venstre skærbillede er alle kolonner stretched, og i det højre skærbillede er alle stretched på nær anden kolonne. Som udgangspunkt bliver hver kolonne den bredde, som det bredeste indhold der er i kolonnen. Med stretched kolonner fordeles overskydende plads på kolonnerne, så rækken fylder hele parents bredde, altså her hele skærm bredden.

Der ikke sat en egentlig bredde på noget i de andre kolonner, så uden stretched går det helt galt.

Det lader også til at width på alle views er underordnet om det er wrap_content eller match_parent.

10. ListView

10.1 ListLayout og Adaptere

Simple Lister

ListView med ListActivity

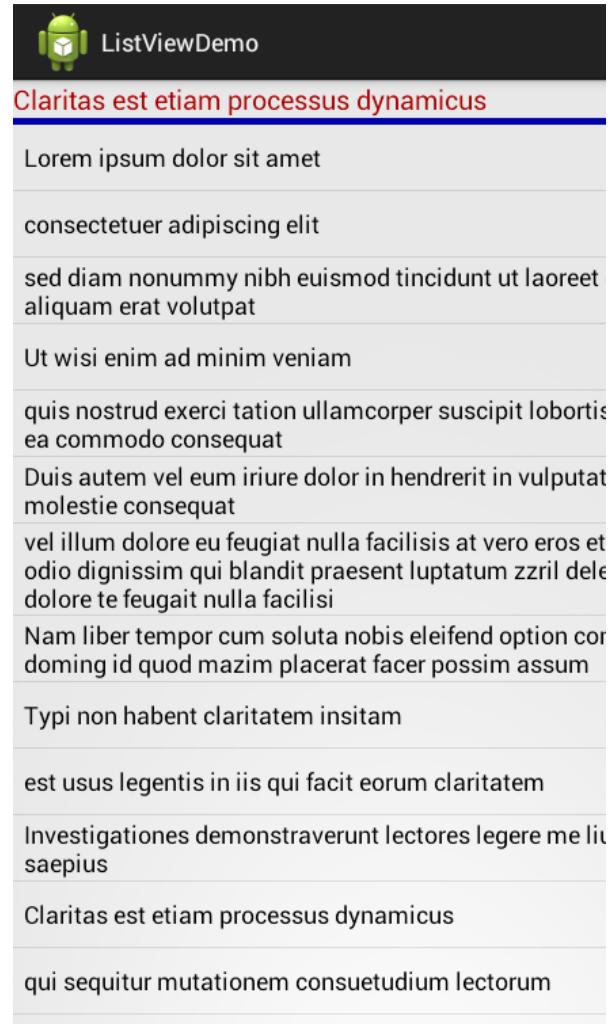
Følgende projekt viser en liste fyldt af items, der hver indeholder en string fra et string-array.

Når der klikkes på et item, vises dets tekst i toppen af skærmen.

Layoutet består af en TextView, der viser det valgte item, og af en ListView, der er fyldt af items indeholdende strings fra string-arrayet.

Læg mærke til at ListView har en predefineret id, som kan bruges af en ListActivity, som aktiviteten arver fra.

```
<LinearLayout  
    xmlns:android=  
        "http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent" >  
    <TextView  
        android:id="@+id/txtSelectedText"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:textSize="20sp"  
        android:textColor="#AA0000"  
        android:text="Her skrives den valgte item"/>  
    <View android:layout_height="5dp"  
        android:layout_width="match_parent"  
        android:background="#0000AA"/>  
    <ListView  
        android:id="@android:id/list"  
        android:layout_width="fill_parent"  
        android:layout_height="fill_parent"  
        android:drawSelectorOnTop="false"/>  
</LinearLayout>
```



Et ListView kan fyldes i en aktivitet, der arver fra Activity. Men det kan også gøres lidt mere automatisk ved at arve fra en ListActivity.

Hvis man benytter ListActivity, kan skærmen fyldes med items uden overhovedet at anvende et layout. Men så er der jo heller ikke andet på skærmen end listen.

Med det viste layout bliver der lagt et TextView over listen, og denne fyldes så i en ListActivity, uden explicit at nævne den.

Koden virker som følger.

Der kunne bare oprettes et array med nogle ord i. Men for sjov oprettes en string med en større tekst i.

Denne tekst splittes på karaktererne (.), (,) og (;) og lægges i arrayet strItems.

For at gøre det lidt pænt, fjernes indledende mellemrum.

Med setListAdapter() tilføjes items til layoutets ListView, indeholdende teksterne i arrayet, så der bliver samme antal items som der er strings i arrayet.

Argumentet android.R.layout.simple_list_item_1 er et XML-layout, som indeholder et TextView med Id = text1, og som bruges til hvert item.

onListItemClick() kaldes når der vælges et item, og dets indhold skrives foroven på skærmen.

```
public class ListViewDemoActivity extends ListActivity
{
    TextView txtSelectedText;
    String loremIpsum =
        "Lorem ipsum dolor sit amet, consectetur adipiscing elit, " +
        "sed diam nonummy nibh euismod tincidunt ut labore magna " +
        "aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud " +
        "exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea " +
        "commodo consequat. Duis autem vel eum iriure dolor in hendrerit in " +
        "vulputate velit esse molestie consequat, vel illum dolore eu feugiat " +
        "nulla facilisis at vero eros et accumsan et iusto odio dignissim qui " +
        "blandit praesent luptatum zzril delenit augue duis dolore te feugait " +
        "nulla facilisi. Nam liber tempor cum soluta nobis eleifend option " +
        "congue nihil imperdiet doming id quod mazim placerat facer possim " +
        "assum. Typi non habent claritatem insitam; est usus legentis in " +
        "iis qui facit eorum claritatem. Investigationes demonstraverunt " +
        "lectores legere me lius quod ii legunt saepius. Claritas est etiam " +
        "processus dynamicus, qui sequitur mutationem consuetudium lectorum. " +
        "Mirum est notare quam littera gothica, quam nunc putamus parum claram, " +
        "anteposuerit litterarum formas humanitatis per seacula quarta decima " +
        "et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, " +
        "fiant sollemnes in futurum.";

    String[] strItems;
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);

        strItems = loremIpsum.split("[.,;]");

        //Fjern indledende spaces. For-each duer ikke her
        for(int i = 0; i < strItems.length; i++)
            strItems[i] = strItems[i].trim();

        setContentView(R.layout.activity_list_view_demo);
        .setListAdapter(new ArrayAdapter<String>(
            this, android.R.layout.simple_list_item_1, strItems));
        txtSelectedText = (TextView)findViewById(R.id.txtSelectedText);
    }

    public void onListItemClick(ListView parent, View v, int position, long id)
    {
```

```
        txtSelectedText.setText(strItems[position]);  
    }  
}
```

Hvis følgende kode udelades, sådan at layoutet ikke bruges, vil der blot vises en liste på skærmen.

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

ListView med almindelig Activity

Hvis man skal lave nogenlunde samme funktionalitet som i forrige projekt men ved brug af en almindelig Activity, skal ListView fyldes udtrykkelig med items og der skal tilknyttes en eventhandler til at håndtere valg af items. (Det kunne jo være at der skulle bruges flere lister eller man bare vil programme lidt mindre mystisk.)

ListView skal have en erklæret id, så den ændres som det eneste i samme layout som før.

```
...  
<ListView  
    android:id="@+id/LstTheList"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:drawSelectorOnTop="false"/>  
...
```

For at oprette en eventhandler, som før var onListItemClick() og lå i ListActivity, må der nu implementeres OnItemClickListener som har metoden onItemClick(). Man kunne forvente at de havde været samme metode, men de har samme parameterliste.

ListView findes på dets id og med dets metode setAdapter() sættes samme adapter som før på ListActivity.setAdapter(). Her hedder metoderne altså også noget forskelligt.



Til sidst sættes ListView OnItemClickListener.

```
public class ListViewDemo2Activity extends Activity implements OnItemClickListener  
{  
    TextView txtSelectedText;  
    String loremIpsum =  
        "Lorem ipsum dolor sit amet, consectetur adipiscing elit, " +  
        "sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna";  
  
    String[] strItems;  
    @Override  
    public void onCreate(Bundle icicle)  
    {  
        super.onCreate(icicle);  
    }
```

```

strItems = loremIpsum.split("[.,;]");
//Fjern indledende spaces. For-each dør ikke her
for(int i = 0; i < strItems.length; i++)
strItems[i] = strItems[i].trim();

setContentView(R.layout.activity_list_view_demo2);

ListView lstTheList = (ListView) findViewById(R.id.lstTheList);

lstTheList.setAdapter(new ArrayAdapter<String>(
this, android.R.layout.simple_list_item_1, strItems));

lstTheList.setOnItemClickListener(this);

txtSelectedText = (TextView) findViewById(R.id.txtSelectedText);
}

@Override
public void onItemClick(AdapterView<?> parent, View v, int position, long id)
{
    txtSelectedText.setText(strItems[position]);
}
}

```

ListView med radiobuttons

Det er muligt at sætte radio-buttons på listen, sådan at det kan ses hvilket item der sidst er valgt.

Dette gøres ved at ændre ListView property ChoiceMode, som kan gøres i layoutet eller i Java-koden, som her.

Desuden skal adapteren have et andet layout til hvert item,

R.layout.simple_list_item_single_choice



Der kan nu spørges om hvilket item der er checked med `ListView.getCheckedItemPosition()`.

I eventhandleren haves position i forvejen som argument, men man har måske brug for at spørge et andet sted i koden, f.eks. i en button eventhandler.

```

public class ListViewDemo3Activity extends Activity implements OnItemClickListener
{
    ListView lstTheList;
    TextView txtSelectedText;
    String loremIpsum =
        "Lorem ipsum dolor sit amet, consectetur adipiscing elit, " +
        "sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna";

    String[] strItems;
}

```

```

@Override
public void onCreate(Bundle icicle)
{
    super.onCreate(icicle);

    strItems = loremIpsum.split("[.,;]");
    for(int i = 0; i < strItems.length; i++)
        strItems[i] = strItems[i].trim();

    setContentView(R.layout.activity_list_view_demo3);

    lstTheList = (ListView)findViewById(R.id.lstTheList);

    lstTheList.setChoiceMode(ListView.CHOICE_MODE_SINGLE);

    lstTheList.setAdapter(new ArrayAdapter<String>(
this, android.R.layout.simple_list_item_single_choice, strItems));

    lstTheList.setOnItemClickListener(this);

    txtSelectedText = (TextView)findViewById(R.id.txtSelectedText);
}

@Override
public void onItemClick(AdapterView<?> parent, View v, int position, long id)
{
    txtSelectedText.setText(strItems[position] +
" [Position " + lstTheList.getCheckedItemPosition() + "]");
}

```

ListView med checkbuttons

Det er også muligt at sætte checkbuttons på hvert item i listen.

Listen skal have ChoiceMode

`ListView.CHOICE_MODE_MULTIPLE`

og adapteren skal have layoutet

`android.R.layout.simple_list_item_multiple_choice`



Nu kan der jo være checked flere items, så derfor kaldes listens getCheckedItemPositions(), som returner et SparseBooleanArray. Det er en collection af key/value par, som er int og boolean. Her vil der i dette tilfælde på en plads være en 1/true, en 2/true og en 4/true. Det er lidt usikkert om de ikke valgte også er med, altså 0/false og 3/false. Derfor er det nødvendigt med if-betingelsen i gennemløbet af denne collection.

I øvrigt vil det her i dette eksempel, være den seneste items tekst, der er trykket på, som bliver vist, også selv om det var en uncheck.

```
public class ListViewDemo4Activity extends Activity implements OnItemClickListener
{
    ListView lstTheList;
    TextView txtSelectedText;
    String loremIpsum =
        "Lorem ipsum dolor sit amet, consectetur adipiscing elit, " +
        "sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna " +
        "aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud";
    String[] strItems;
    @Override
    public void onCreate(Bundle icicle)
    {
        super.onCreate(icicle);
        strItems = loremIpsum.split("[.,;]");
        for(int i = 0; i < strItems.length; i++)
            strItems[i] = strItems[i].trim();
        setContentView(R.layout.activity_list_view_demo4);

        lstTheList = (ListView) findViewById(R.id.lstTheList);

        lstTheList.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);

        lstTheList.setAdapter(new ArrayAdapter<String>(
            this, android.R.layout.simple_list_item_multiple_choice, strItems));

        lstTheList.setOnItemClickListener(this);

        txtSelectedText = (TextView) findViewById(R.id.txtSelectedText);
    }

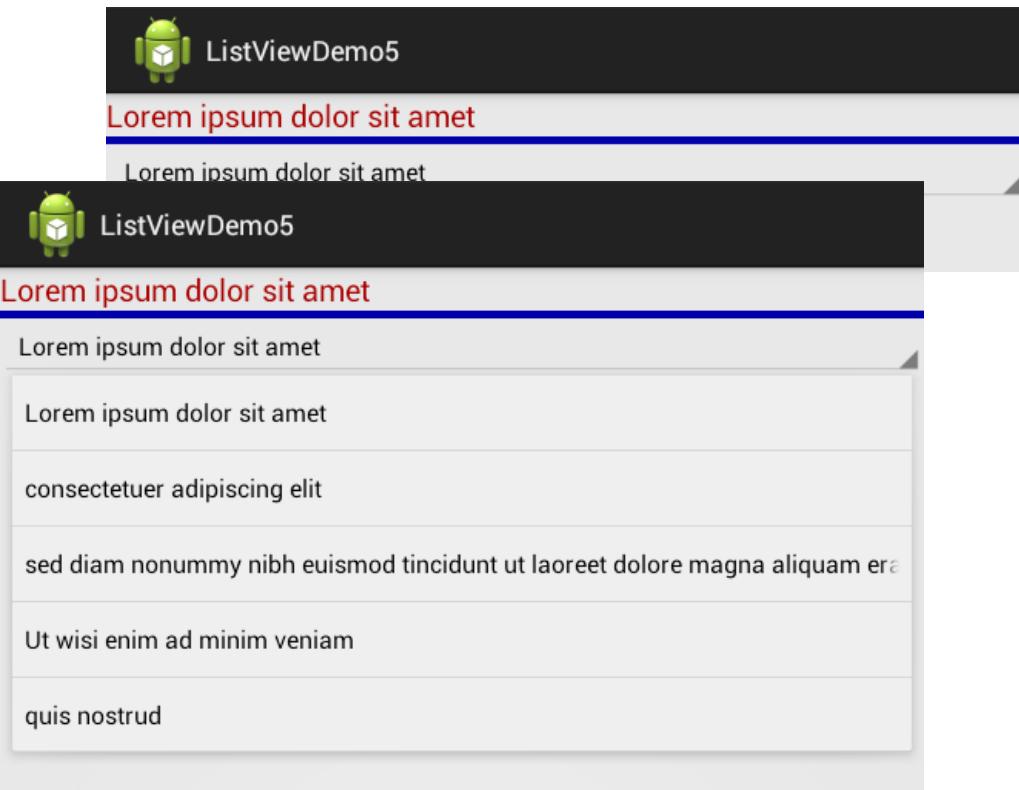
    @Override
    public void onItemClick(AdapterView<?> parent, View v, int position, long id)
    {
        txtSelectedText.setText(strItems[position]);
        SparseBooleanArray positions = lstTheList.getCheckedItemPositions();
        for(int i = 0; i < positions.size(); i++)
        {
            if(positions.valueAt(i) == true)
                txtSelectedText.setText(txtSelectedText.getText()
                    + " - Pos " + positions.keyAt(i));
        }
    }
}
```

Spin Control / Drop down list

For at oprette en drop-down-list, placeres en Spinner i layoutet, i stedet for den tidlige ListView.

Når der trykkes på denne Spinner, folder den sig ud. Og når der trykkes på en item i listen, folder den sig ind igen.

```
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent" >  
    <TextView  
        android:id="@+id/txtSelectedText"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:textSize="20sp"  
        android:textColor="#AA0000"  
        android:text="Her skrives den valgte item"/>  
    <View android:layout_height="5dp"  
        android:layout_width="match_parent"  
        android:background="#0000AA"/>  
    <Spinner  
        android:id="@+id/spnTheSpinner"  
        android:layout_width="fill_parent"  
        android:layout_height="wrap_content"  
        android:drawSelectorOnTop="true"/>  
</LinearLayout>
```



Java-koden minder meget om de tidlire eksempler og ændres i henhold til følgende.



Aktiviteten implementerer OnItemSelectedListener, som har metoderne onItemClick() og onNothingSelected().

ArrayAdapteren skal både have android.R.layout.simple_spinner_item og android.R.layout.simple_spinner_dropdown_item.

Til en afveksling er der i onItemClick() benyttet argumentet parent, som er den det er sket på, ved at kalde parent.getSelectedItem().toString() fås teksten i det valgte item.

```
package jaj.tec.dk.spinnerapp;

import android.app.Activity;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.TextView;

public class MainActivity extends Activity
    implements AdapterView.OnItemSelectedListener
{
    Spinner spnTheSpinner;
    TextView txtSelectedText;

    String[] strItems = {"Tobias", "Kim", "Lars", "Rune", "Morten",
        "Jon", "Lirim", "Alexander", "Jan", "Patrick", "Flemming"};

    @Override
    public void onCreate(Bundle icicle) {
        super.onCreate(icicle);

        setContentView(R.layout.activity_main);

        spnTheSpinner = (Spinner) findViewById(R.id.spnTheSpinner);
        spnTheSpinner.setOnItemSelectedListener(this);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(
            this, android.R.layout.simple_spinner_item, strItems);
        adapter.setDropDownViewResource(
            android.R.layout.simple_spinner_dropdown_item);

        spnTheSpinner.setAdapter(adapter);

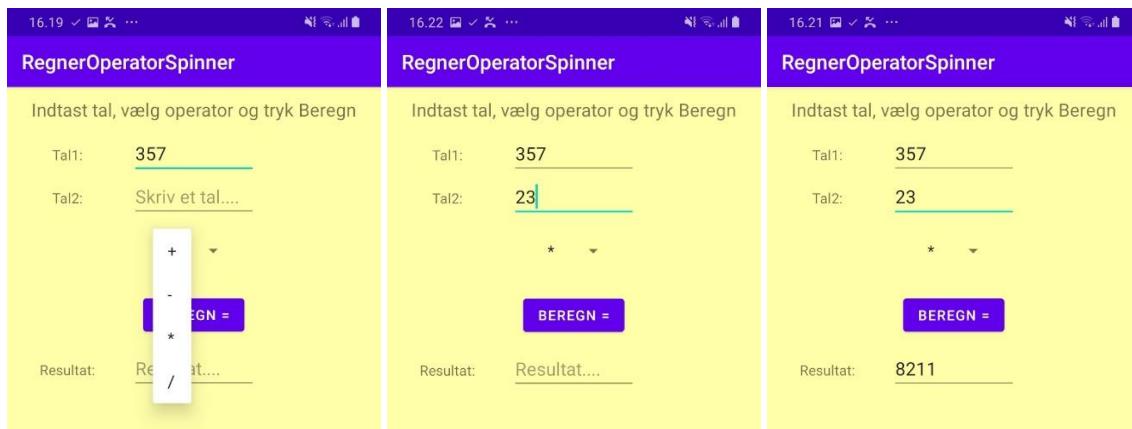
        txtSelectedText = (TextView) findViewById(R.id.txtSelectedText);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View v, int position, long id)
```

```
{  
    txtSelectedText.setText(" Pos: " + (position+1) + " "  
    + parent.getSelectedItem().toString());  
}  
  
@Override  
public void onNothingSelected(AdapterView<?> arg0) {  
    txtSelectedText.setText("");  
}  
}
```

Opgave Regner med operatorer i Spinner

Skriv et Android program som vist her under, hvor der vælges operator i en Spinner og beregnes i henhold til dette, når der trykkes Beregn.



Opgave Spinner med objekter

Opret en klasse Person med Fornavn og Efternavn.

Opret et array af Person-objekter og fyld en Spinner med disse.

Gør det sådan at der i spinneren vises fulde navne.

Opgave To Do liste med ToDo-objekter i Spinneren.

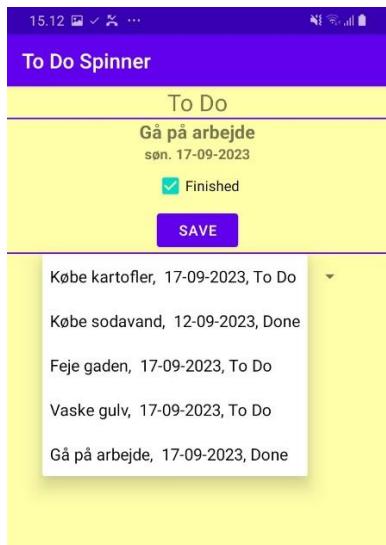
Skriv et Android program, hvor der gemmes ToDo objekter i en Spinner eller en ListView.

ToDo-klassen skal indeholde en tekst, en dato og en boolean som angiver om den er udført.

Der oprettes programmatisk en række ToDo-objekter i Spinneren.

Når der vælges en ToDo i Spinneren skal detaljerne vises foroven, hvor det skal være muligt at mærke den aktuelle som uført. Ved tryk på Save skal den aktuelle listen opdateres i henhold til om den er udført eller ej.

I denne version skal det ikke være muligt at oprette nye ToDo, men det kan der udvides med hvis der er tid.



10.2 Customized ListView

Undervisningsmaterialet kap. 10.2

Her oprettes der et layout som hvert item kan bygge på.

Der oprettes en Adapterklasse som arver fra BasicAdapter, som sørger for at danne hvert item med det rette indhold.

MainActivity.java

```
package dk.tec.jaj.customizedlistview;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ListView;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity
{
    ListView lstMad;

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

        List<Mad> mads = new ArrayList<Mad>();
        mads.add(new Mad("Flæsk og Kål", "God for den hårdarbejdende landmand"
                , R.drawable.flaeiskogkaal));
        mads.add(new Mad("Grønkål og Hamburgerryg", "Husmoderens special"
                , R.drawable.glaeseret_hamburgerryg_med_groenlangkaal));
        mads.add(new Mad("Frisk grønkål og Kål", "Meget meget frisk og grøn kål"
                , R.drawable.groenLangkaal_med_frisk_groenkaal));
        mads.add(new Mad("Kål pølse og Kål", "Grønlangkål og langpølse"
                , R.drawable.kalpolsesoggronlangkal));
        mads.add(new Mad("Medister og Langgrønkål", "Den kører langt på"
                , R.drawable.medisterpoelse_med_groenLangkaal_original));
        mads.add(new Mad("Super Grønkål", "98 oktan. Den kører langt på"
                , R.drawable.opskriфт_groenkaal));
        mads.add(new Mad("Råt for usødet grønkål", "Den er sund"
                , R.drawable.raat_groenkaal));
    }
}
```



```

        MadAdapter madAdapter = new MadAdapter(mads, this);

        lstMad = findViewById(R.id.LstMadKaal);
        lstMad.setAdapter(madAdapter);
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:text="Kålmadtyper samt enkelte andre noget"
        />
    <ListView
        android:id="@+id/lstMadKaal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>

```

MadAdapter.java

```

package dk.tec.jaj.customizedlistview;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

public class MadAdapter extends BaseAdapter
{
    int tal;
    private final List<Mad> mads;
    private MainActivity mainActivity;

    public MadAdapter(List<Mad> mads, MainActivity mainActivity)
    {
        this.mads = mads;
    }
}

```

```

        this.mainActivity = mainActivity;
    }

@Override
public int getCount() {
    return mads.size();
}

@Override
public Object getItem(int i) {
    return null;
}

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(final int pos, View view, ViewGroup viewGroup)
{
    LayoutInflator inflater = LayoutInflater.from(mainActivity);
    View v = inflater.inflate(R.layout.custom_item,null);

    Mad mad = mads.get(pos);

    ImageView imgView = v.findViewById(R.id.imageView);
    imgView.setImageResource(mad.getBilledNr());
    TextView txtOverskrift = v.findViewById(R.id.txtOverskrift);
    txtOverskrift.setText(mad.getOverskrift());
    TextView txtBeskrivelse = v.findViewById(R.id.txtBeskrivelse);
    txtBeskrivelse.setText(mad.getBeskrivelse());
}

/////////////////////////////////////////////////////////////////
Button btnDelete = v.findViewById(R.id.btnDelete);
btnDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view)
    {
        AlertDialog.Builder alert = new AlertDialog.Builder(mainActivity);
        alert.setTitle("Delete Food?");
        alert.setMessage("Ønsker De at slette den ret?");

        alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton)
            {
                int posToRemove = pos;
                Mad remove = mads.remove(posToRemove);
                notifyDataSetChanged();
            }
        });

        alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
        {
            public void onClick(DialogInterface dialog, int whichButton)
            {

            }
        });
    }
});

alert.show();
}
});
```

```
//////////  
    return v;  
}  
}  
}
```

custom_item.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:orientation="vertical"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    >  
    <LinearLayout  
        android:orientation="horizontal"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content">  
        <ImageView  
            android:id="@+id/imageView"  
            android:layout_width="0px"  
            android:layout_height="match_parent"  
            android:layout_weight="1"  
            app:srcCompat="@mipmap/ic_launcher_round"  
            android:scaleType="centerCrop"/>  
        <LinearLayout  
            android:layout_width="0px"  
            android:layout_height="wrap_content"  
            android:layout_weight="2.5"  
            android:orientation="vertical">  
            <TextView  
                android:id="@+id/txtOverskrift"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:textSize="@android:dimen/app_icon_size"  
                android:text="TextView" />  
            <TextView  
                android:id="@+id/txtBeskrivelse"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:text="TextView" />  
            <Button  
                android:id="@+id/btnSlet"  
                android:layout_width="match_parent"  
                android:layout_height="wrap_content"  
                android:text="Slet Ret" />  
        </LinearLayout>  
    </LinearLayout>  
    <View  
        android:layout_width="match_parent"  
        android:layout_height="10dp"  
    />  
    <View  
        android:layout_width="match_parent"  
        android:layout_height="1dp"  
        android:background="#0000ff"  
    />  
    <View  
        android:layout_width="match_parent"
```

```
    android:layout_height="10dp"
/></LinearLayout>
```

Mad.java

```
package dk.tec.jaj.customizedlistview;

public class Mad
{
    String overskrift;
    String beskrivelse;
    int billedNr;

    public Mad(String overskrift, String beskrivelse, int billedNr) {
        this.overskrift = overskrift;
        this.beskrivelse = beskrivelse;
        this.billedNr = billedNr;
    }

    public String getOverskrift() {
        return overskrift;
    }

    public void setOverskrift(String overskrift) {
        this.overskrift = overskrift;
    }

    public String getBeskrivelse() {
        return beskrivelse;
    }

    public void setBeskrivelse(String beskrivelse) {
        this.beskrivelse = beskrivelse;
    }

    public int getBilledNr() {
        return billedNr;
    }

    public void setBilledNr(int billedNr) {
        this.billedNr = billedNr;
    }
}
```

11. Intents og Activities

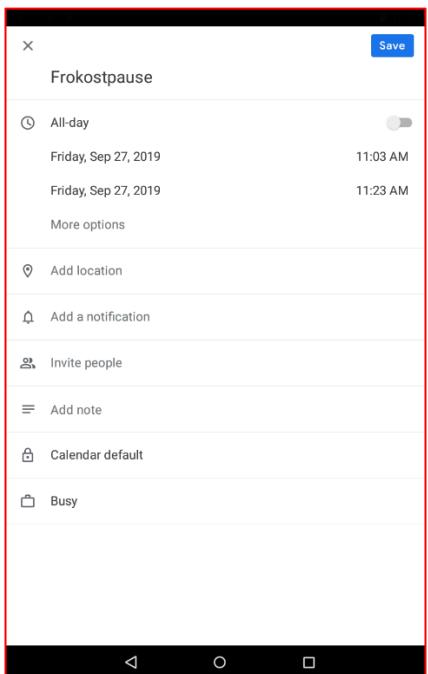
Vis web-side i browser

```
Intent intent = new Intent(Intent.ACTION_VIEW);
intent.setData(Uri.parse("http://www.google.com"));
startActivity(intent);
```



Oprettelse af kalender-element

```
Calendar cal = Calendar.getInstance();
Intent intent = new Intent(Intent.ACTION_EDIT);
intent.setType("vnd.android.cursor.item/event");
intent.putExtra("beginTime", cal.getTimeInMillis());
cal.add(Calendar.MINUTE, 20);
intent.putExtra("endTime", cal.getTimeInMillis());
intent.putExtra("title", "Frokostpause");
intent.putExtra("allDay", false);
startActivity(intent);
```



Aktivitet, der kalder en anden aktivitet

For at starte en anden aktivitet, skal der oprettes et Intent-objekt, som peger på den aktuelle aktivitet.

Den nye aktivitet startes med metoden `startActivity()` og med intentet som argument.

```
Intent secondIntent = new Intent(MainActivity.this, SecondActivity.class);
startActivity(secondIntent);
```

I intentet kan der overføres data til den nye aktivitet. Dette gøres ved at angive key/value-par.

I følgende eksempel er key: "Hilsen" og value: "Hello from MainActivity"

```
secondIntent.putExtra("Hilsen", "Hello from MainActivity");
```

Dette gøres inden kaldet til metoden `startActivity()`.

Der kan overføres alle simple typer, string og objekter af klasser der implementerer `Serializable`.

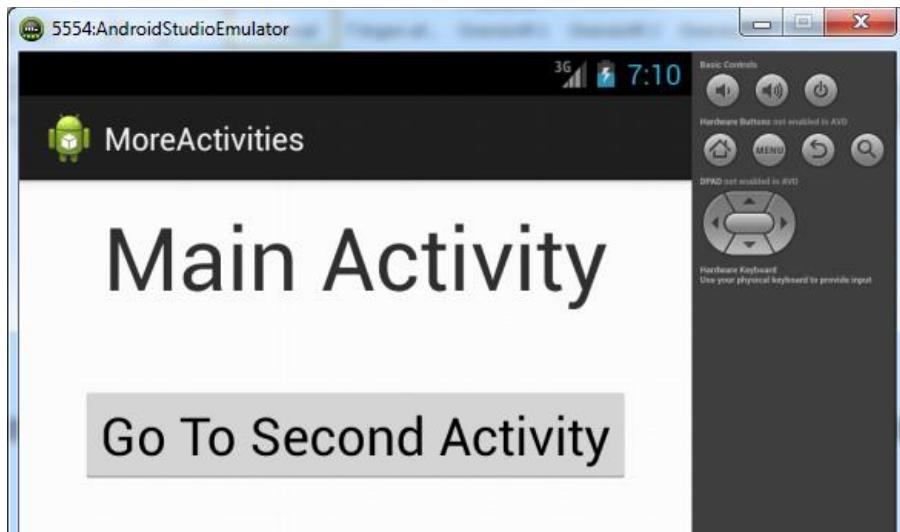
I den kaldte aktivitet kan data så hentes ud af intentet.

activity_main.xml

```
<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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Main Activity"
        android:textSize="50sp"
        android:layout_gravity="center_horizontal"
        android:padding="10sp"
        android:id="@+id/textView"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go To Second Activity"
        android:id="@+id/btnGoToSecond"
        android:layout_gravity="center_horizontal"
        android:textSize="30sp"
        android:layout_marginTop="30dp"/>
</LinearLayout>
```



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

        Button btnGoSecond = (Button) findViewById(R.id.btnGoToSecond);
        btnGoSecond.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View arg0)
            {
                Intent secondIntent =
new Intent(MainActivity.this, SecondActivity.class);

                secondIntent.putExtra("Hilsen", "Hello from MainActivity");

                startActivity(secondIntent);
            }
        });
    }
}
```

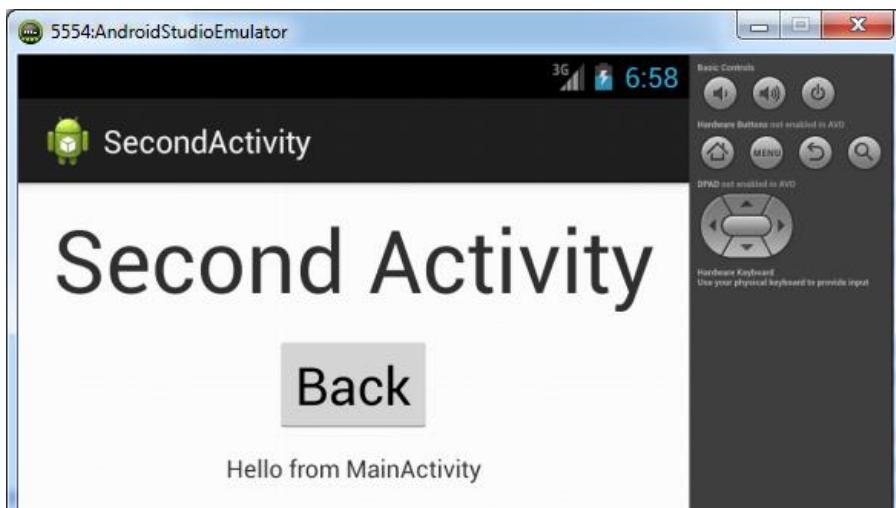
activity_second.xml

```
<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"
    tools:context=".SecondActivity" >
```

```

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:padding="10sp"
    android:text="Second Activity"
    android:textSize="50sp" />
<Button
    android:id="@+id	btnBackFromSecond"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30sp"
    android:layout_gravity="center_horizontal"
    android:text="Back" />
<TextView
    android:id="@+id/tvFromMain"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="10sp"
    android:layout_gravity="center_horizontal"
    android:text="" />
</LinearLayout>

```



Den nye aktivitet vises og der vendes tilbage når den afsluttes. Dette kan ske med Back-knappen på enheden eller med at kalde metoden `finish()`.

Hvis den nye aktivitet ønsker at læse data fra den kaldende aktivitet, må det benyttede Intent-objekt findes og data læses i denne.

Dette gøres med

```
Intent callingIntent = getIntent();
```

```
String hilsen = callingIntent.getStringExtra("Hilsen");
```

SecondActivity.java:

```
public class SecondActivity extends Activity
{
    TextView tvFromMain;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);
        Button btnBackFromSecond = (Button) findViewById(R.id.btnBackFromSecond);
        btnBackFromSecond.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                SecondActivity.this.finish(); // Eller bare finish()
            }
        });
    }

    tvFromMain = (TextView) findViewById(R.id.tvFromMain);
    Intent callingIntent = getIntent();
    String hilser = callingIntent.getStringExtra("Hilsen");
    tvFromMain.setText(hilser);

    // Eller
    ((TextView) findViewById(R.id.tvFromMain))
    .setText(getIntent()           .getStringExtra("Hilsen"));
}

@Override
public void onBackPressed()
{
    //super.onBackPressed();

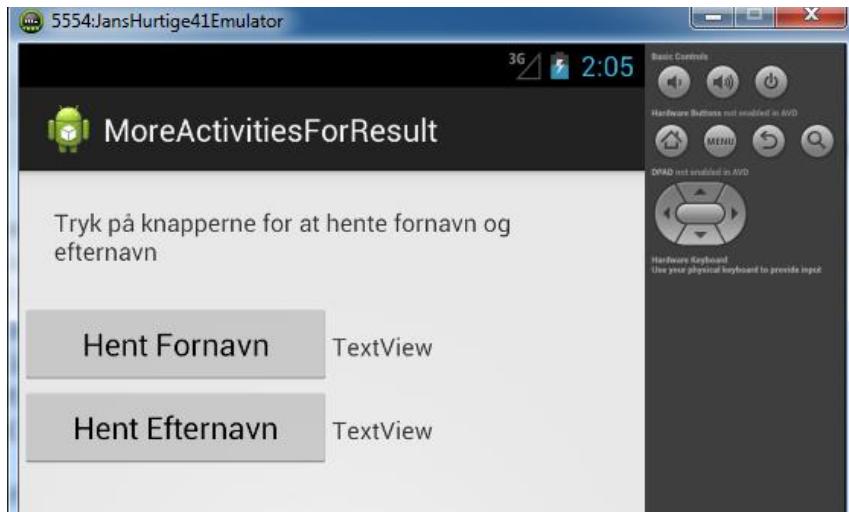
    // Kode der ønskes udført inden der returneres
    finish();
}
```

Aktivitet der åbner en af flere aktiviteter og får data tilbage

For at demonstrere at en aktivitet åbner en af flere andre aktiviteter for at få data tilbage når de lukker, oprettes følgende layout-xml til main-aktiviteten.

```
<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"
    tools:context=".MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="20sp"
        android:text="Tryk på knapperne for at hente fornavn og efternavn" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <Button
            android:id="@+id/btnHentFornavn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Hent Fornavn" />
        <TextView
            android:id="@+id/tvFornavn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="TextView" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
        <Button
            android:id="@+id/btnHentEfternavn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Hent Efternavn" />
        <TextView
            android:id="@+id/tvEfternavn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="TextView" />
    </LinearLayout>
</LinearLayout>
```

Dette giver følgende udseende.



MainActivity.java

Når en aktivitet kaldes for at få data tilbage, når den afsluttes, benyttes metoden `startActivityForResult()`.

For at kunne vide hvilken aktivitet, der vender tilbage skal der gives en `request-code` med. Derfor oprettes der en konstant til hver aktivitet, `Fornavn` og `Efternavn`, der skal kaldes.

Når den aktuelle aktivitet vender tilbage, kaldes eventhandleren `onActivityResult()`. Det er en eventhandler som findes i `Activity`-klassen, og som derfor bare skal overrides.

`onActivityResult()` bliver kaldt med tre argumenter,

- `requestCode`, som angiver hvilken aktivitet, der er vendt tilbage,
- `resultCode`, som den aktuelle aktivitet har sat og angiver f.eks. `OK` eller `Cancel`,
- `data`, som er en `Intent` hvor returdata ligger.

```
public class MainActivity extends Activity implements OnClickListener {
    Button btnHentFornavn, btnHentEfternavn;
    TextView tvFornavn, tvEfternavn;

    final static int requestCodeFornavn = 0;
    final static int requestCodeEfternavn = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnHentFornavn = (Button)findViewById(R.id.btnHentFornavn);
        btnHentFornavn.setOnClickListener(this);
        btnHentEfternavn = (Button)findViewById(R.id.btnHentEfternavn);
        btnHentEfternavn.setOnClickListener(this);

        tvFornavn = (TextView)findViewById(R.id.tvFornavn);
        tvEfternavn = (TextView)findViewById(R.id.tvEfternavn);
```

```

}

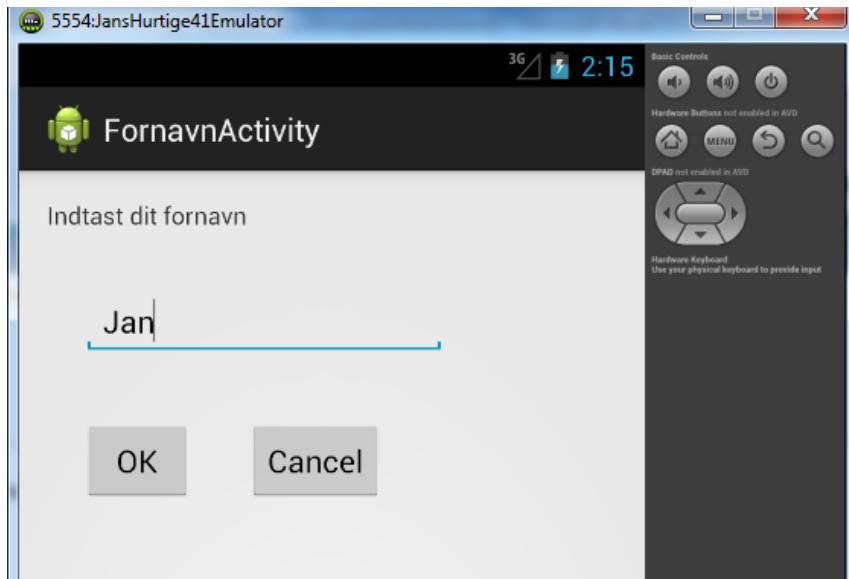
@Override
public void onClick(View v)
{
    if(v == btnHentFornavn)
    {
        Intent intentFornavn = new Intent(this, FornavnActivity.class);
        startActivityForResult(intentFornavn, requestCodeFornavn);
    }
    if(v == btnHentEfternavn)
    {
        Intent intentEfternavn = new Intent(this, EfternavnActivity.class);
        startActivityForResult(intentEfternavn, requestCodeEfternavn);
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);

    if(requestCode == requestCodeFornavn)
    {
        if(resultCode == RESULT_OK)
        {
            tvFornavn.setText(data.getStringExtra("Fornavn"));
        }
        else if(resultCode == RESULT_CANCELED)
        {
            tvFornavn.setText("Fornavn Cancelled");
        }
    }
    if(requestCode == requestCodeEfternavn)
    {
        if(resultCode == RESULT_OK)
        {
            tvEfternavn.setText(data.getStringExtra("Efternavn"));
        }
        else if(resultCode == RESULT_CANCELED)
        {
            tvEfternavn.setText("Efternavn Cancelled");
        }
    }
}
}

```

Den ene af de to aktiviteter, FornavnActivity, kunne se sådan ud. (layout.xml er ikke vist)



FornavnActivity.java

For at returnere data, hentes den Intent, som den er blevet kaldt med, og der tilføjes data med putExtra().

Derefter kaldes metoden setResult(), hvor der angives resultCode vha. eksisterende konstanter f.eks. RESULT_OK, samt den Intent, hvor der er indsatturdate.

Og der afsluttes.

```
public class FornavnActivity extends Activity implements OnClickListener
{
    Button btnFornavnOK, btnFornavnCancel;
    EditText tvFornavn;

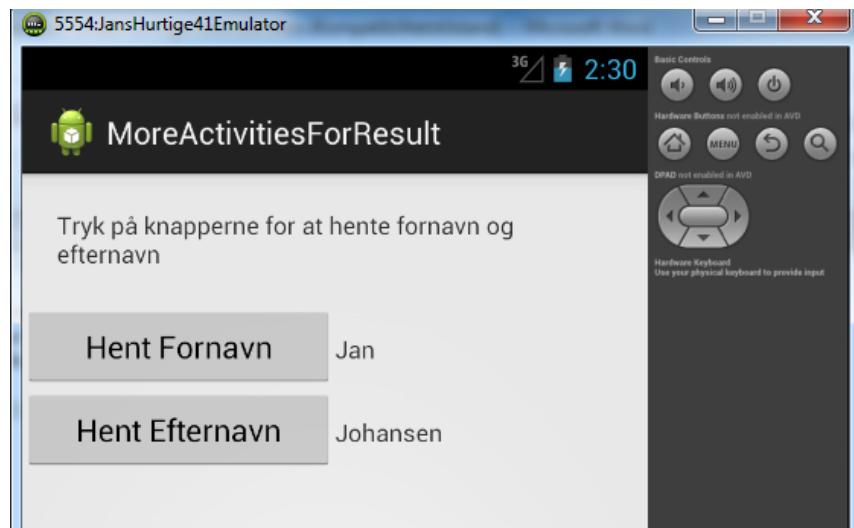
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fornavn);

        btnFornavnOK = (Button)findViewById(R.id.btnFornavnOK);
        btnFornavnOK.setOnClickListener(this);
        btnFornavnCancel = (Button)findViewById(R.id.btnFornavnCancel);
        btnFornavnCancel.setOnClickListener(this);
        tvFornavn = (EditText)findViewById(R.id.tvIndtastFornavn);
    }

    @Override
    public void onClick(View v)
    {
        if(v == btnFornavnOK)
        {
            Intent i = getIntent();
            i.putExtra("Fornavn", tvFornavn.getText().toString());
        }
    }
}
```

```
        setResult(RESULT_OK, i);
    }
    finish();
}
}
```

Og sådan kan resultatet så se ud, når der er vendt tilbage til den kaldende aktivitet.



Deprecated startActivityForResult

Metoden `startActivityForResult` er deprecated.

Dette medfører at der oprettes en callback-metode til hvert kald af en anden aktivitet.

Følgende kode er med `startActivityForResult`.

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

        Button btnGoToSecond = findViewById(R.id.btnGoToSecond);

        btnGoToSecond.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Intent intent = new Intent(MainActivity.this, SecondActivity.class);
                intent.putExtra("TextToSecond", "Hej fra Main");
                startActivityForResult(intent, 1);
            }
        });
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data)
    {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 1 && resultCode == Activity.RESULT_OK)
        {
            String message = data.getStringExtra("TextFromSecond");
            Log.v("Tag", message);
        }
    }
}
```

Dette ændres til det følgende, hvor der oprettes et objekt af typen `ActivityResultLauncher<Intent>`, som indeholder callback-metoden `onActivityResult()`.

På dette objekt kaldes metoden `launch()`, som derved bliver afløseren for `startActivityForResult()`.

```
public class MainActivity extends AppCompatActivity
{
    ActivityResultLauncher<Intent> secondActivityLauncher;

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

```

Button btnGoToSecond = findViewById(R.id.btnGoToSecond);
txtFromSecond = findViewById(R.id.txtFromSecond);

secondActivityLauncher = registerForActivityResult(
new ActivityResultContracts.StartActivityForResult(),
new ActivityResultCallback<ActivityResult>()
{
    @Override
    public void onActivityResult(ActivityResult result)
    {
        if (result.getResultCode() == Activity.RESULT_OK) {
            // There are no request codes
            Intent data = result.getData();
            String message = data.getStringExtra("TextFromSecond");
            Log.v("Tag",message);
        }
    }
)) ;

btnGoToSecond.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Intent intent = new Intent(MainActivity.this,SecondActivity.class);
        intent.putExtra("TextToSecond", "Hej fra Main");
        secondActivityLauncher.launch(intent);
    }
});
}
}

```

En start-aktivitet, der vises kort og går derefter videre til en menu-aktivitet, hvor der kan vælges et program(aktivitet) der ønskes udført

Der oprettes et nyt projekt og main.xml sættes til at indeholde nogle billeder, som følgende.

```
<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"
    android:background="#FF000000"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="20sp"
        android:layout_gravity="center_horizontal"
        android:textSize="50sp"
        android:textColor="#ff96ff13"
        android:text="Android Menu" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200sp"
        android:padding="5sp"
        android:id="@+id/imageView"
        android:src="@drawable/images_android2"
        android:layout_gravity="center_horizontal"/>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="200sp"
        android:id="@+id/imageView2"
        android:src="@drawable/image_android1"
        android:layout_gravity="center_horizontal"/>
</LinearLayout>
```

Det kunne se sådan ud.



For at vise denne start-skærm benyttes metoden Thread.sleep(). Den kan ikke kaldes direkte i onCreate(), der brugerfladen første vises når onCreate() er slut.

Derfor oprettes en ny tråd, efter at skærmbilledet er sat til at vises. I tråden kaldes sleep() og derefter vises den nye menu-aktivitet.

MainActivity.java:

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;

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

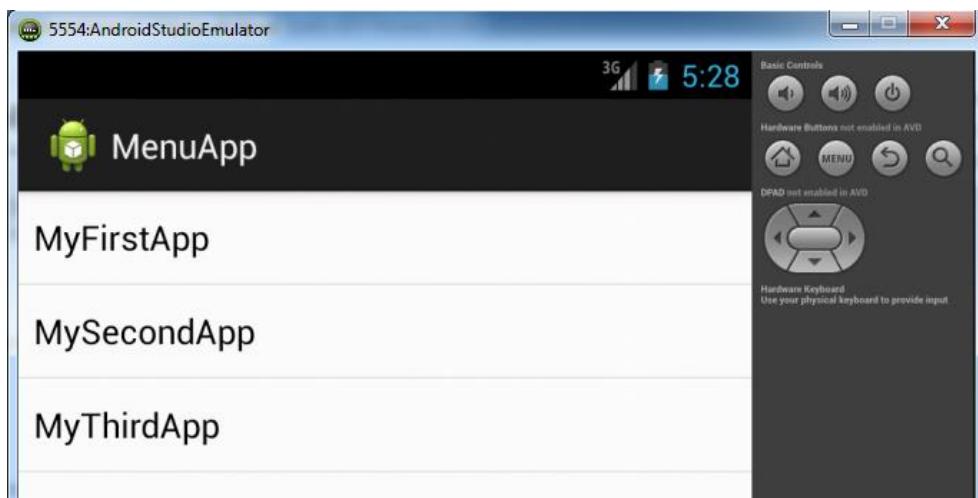
        Thread th = new Thread()
        {
            @Override
```

```

        public void run()
    {
        super.run();
        try {
            Thread.sleep(5000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        finally
        {
            Intent menuIntent = new Intent(MainActivity.this,
                    MainActivity.class);
            startActivity(menuIntent);
        }
    };
    th.start();
}
}

```

Den nye menu-aktivitet er en ListActivity. Til denne skal der ikke laves en layout.xml, da den kun kan vise en liste af items.



Der tilføjes til en en ArrayAdapter, som kan vises items svarende til elementerne i et string-array.

Eventhandleren onListItemClick() overrides. Denne kaldes bl.a. med argumentet position af det item som er klikket og som svarer til elementet i string-arrayet.

Ved at benytte denne string og sætte den sammen med package-navnet kan den tilsvarende klasse findes vha. Class.forName() og derefter kan oprettes et Intent og skiftes til den aktuelle aktivitet.

```

public class MenuActivity extends ListActivity
{
    String[] appNames = {"MyFirstApp", "MySecondApp", "MyThirdApp"};
    @Override
    protected void onCreate(Bundle savedInstanceState)

```

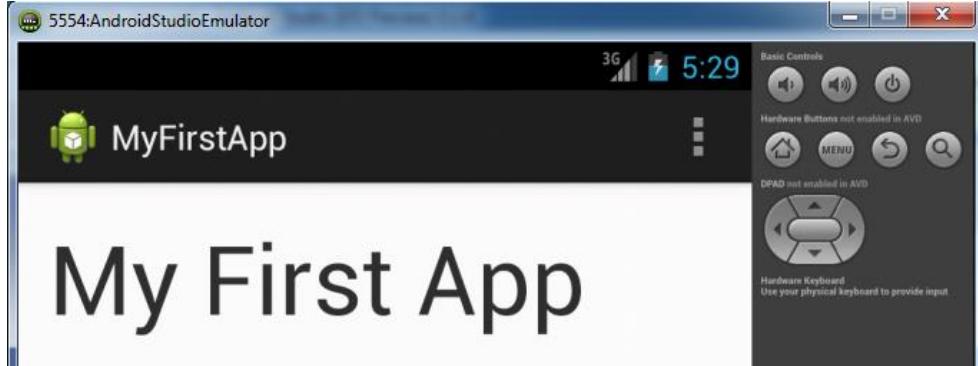
```

    {
        super.onCreate(savedInstanceState);
        setListAdapter(new ArrayAdapter<String>(this,
                R.layout.simple_list_item_1, appNames));
    }

    @Override
    protected void onListItemClick(ListView l, View v, int position, long id)
    {
        super.onListItemClick(l, v, position, id);
        Class c = null;
        try {
            c = Class.forName("dk.tec.jaj.menuapp." + appNames[position]);
            //skal have package med
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        Intent i = new Intent(this, c);
        startActivity(i);
    }
}

```

Der skiftes så til denne aktivitet.



Øvelser med flere aktiviteter

1. Et program med MainAct hvor der indtastes to tal, som sendes til RegneAct, hvor de vises. Brugeren skal nu angive summen af de to tal, som sendes tilbage til MainAct og vises, sammen med at der udskrives om der er regnet rigtigt.
2. Et program med MainAct, hvor der indtastes to tal, som sendes til RegneAct, hvor de vises. Brugeren skal nu trykke på en af 4 knapper, som henholdsvis vil udregne addition, subtraktion, division eller multiplikation. Resultatet og den benyttede regneart skal sendes tilbage til MainAct, hvor det vises.
3. MainAct henter navn fra NameAct, hvor bruger indtaster sit navn, samt Skonummer fra ShoeAct, hvor bruger indtaster sit skonummer. Begge dele vises så på MainAct.

4. MainAct henter et tal mellem 1 og 5 fra LevelAct, hvor dette angives med 5 radiobuttons, samt en checkbox, der angiver dead or alive. Det angivne udskrives på MainAct.
5. På MainAct indtastes en farvekode, som vises som en farve på ShowColorAct.

Øvelse To-Do Liste

Vis en to-do liste på første brugerflade.

Opret en to-do på en anden brugerflade, og tilføj den til listen på første brugerflade.

Vis en To-Do

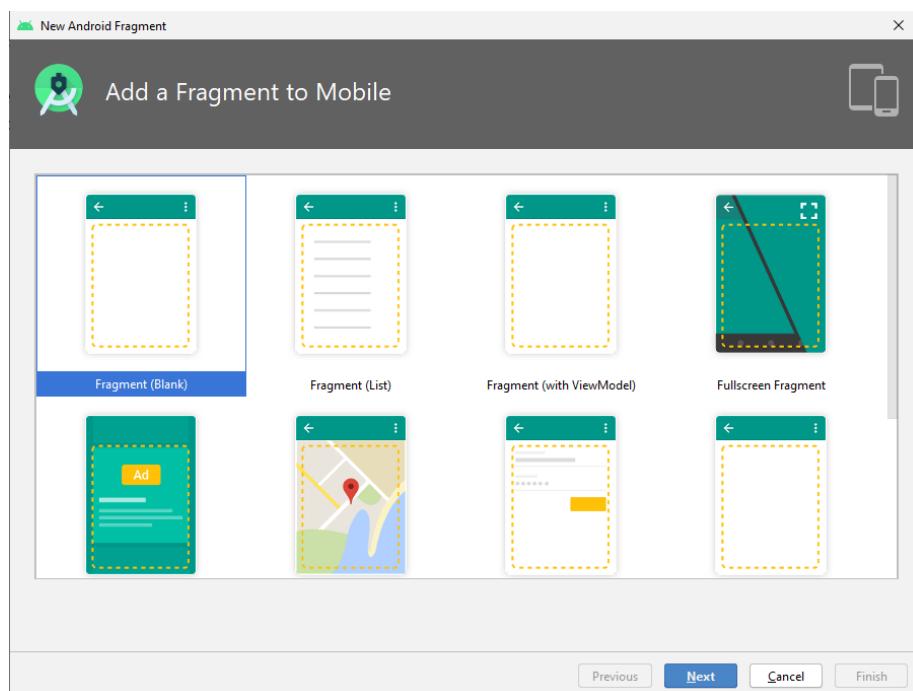
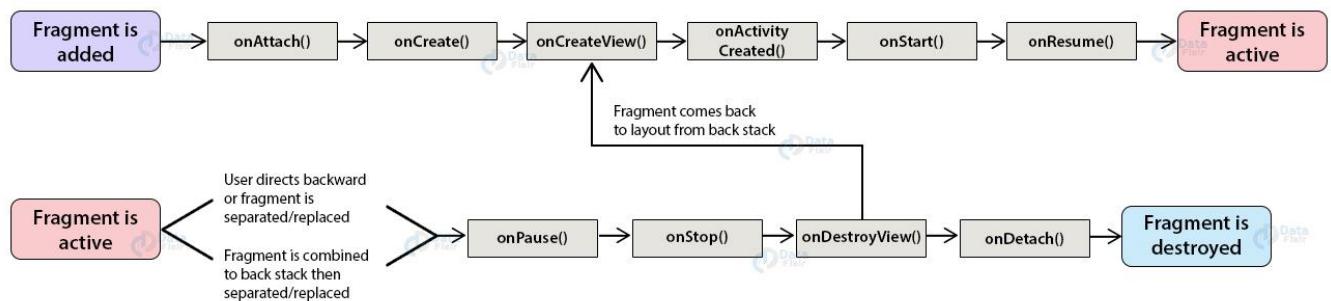
Slet en To-Do

Rediger en To-Do

12. Fragmenter

Fragmenter er en slags mini Aktiviteter, som skal huses i Aktivititer. De giver mulighed for at udskifte dele af brugerfladen.

Android Fragment Lifecycle



En static Fragment

En ny Fragment kan oprettes i projektet med New Fragment, og der kan vælges en template. Her er det en Blank. Der er dog her fjernet alt det overflødige i java-koden, som alligevel er der.

Når en Fragment knyttes til fragment-tag i Aktiviteten er den static og kan ikke udskiftes dynamisk.

Dens widgets kan tilgås fra Aktiviteten som hvis de lå direkte i denne.



activity_main.xml

Der indsættes et fragment-tag i layoutet, der hvor Fragmentet skal være.

Med attributten name angives det fulde navn på Fragment-klassen, her dk.tec.onestaticfragment.MyFragment.

Det kræves at der angives et id på tagget, selv om det ikke eksplisit bruges i koden.

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    android:background="#b8f"
    android:gravity="center_horizontal"
    android:padding="20sp"
    tools:context=".MainActivity">

    <TextView
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Static Fragments"
        android:textColor="#000"
        android:textSize="25sp"
        android:layout_margin="20sp"
    />
<fragment
    android:id="@+id/fragmentId"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="dk.tec.onestaticfragment.MyFragment"
/>
</LinearLayout>

```

MainActivity.java

Aktiviteten skal arve fra AppCompatActivity eller FragmentActivity, hvis den skal kunne indeholder Fragmenter.

Der ud over kræves der ikke noget i koden for at huse Fragmentet.

Dog vises der at man kan få fat i widgets i Fragmentet fra Aktivity-koden.

```

package dk.tec.onestaticfragment;

import androidx.appcompat.app.AppCompatActivity;
import androidx.fragment.app.FragmentActivity;
import android.os.Bundle;

public class MainActivity extends FragmentActivity // AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        TextView txtInFragment = findViewById(R.id.txtInFragment);
        txtInFragment.setText("Written from Activity");
    }
}

```

fragment_my.xml

Layoutet i et Fragment laves på samme måde som i Aktivitet.

```

<?xml version="1.0" encoding="utf-8"?>
<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:background="@color/colorPrimaryDark"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    tools:context=".MyFragment">

    <TextView
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:layout_margin="20sp"
        android:textColor="#ff0"
        android:text="Hello from blank Fragment" />

<TextView
    android:id="@+id/txtInFragment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:layout_margin="20sp"
    android:textColor="#ff0"
/>
</LinearLayout>

```

MyFragment.java

MyFragment arver fra Fragment og skal override metoden onCreateView(), som skal returnere et View dannedt af det tilhørende layout. Dette gøres vha. LayoutInflater parameteren.

Brugefladen kan selvfølgelig også dannes i selve java-koden.

```

package dk.tec.onestaticfragment;

import android.os.Bundle;
import androidx.fragment.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class MyFragment extends Fragment
{
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState)
    {
        View v = inflater.inflate(R.layout.fragment_my, container, false);
        return v;
    }
}

```

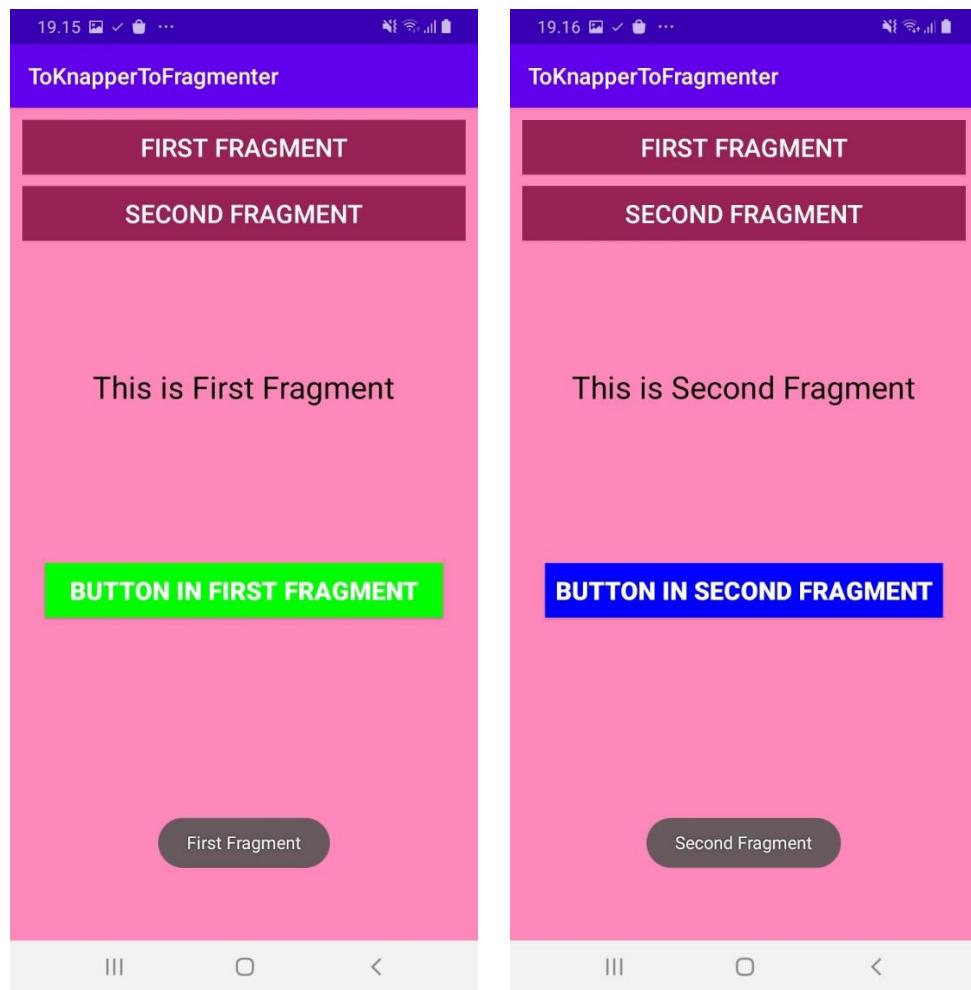
To Fragmenter udskiftes i placeholder

Fra <https://abhiandroid.com/ui/fragment>

I det viste eksempel er der øverst to knapper og herunder et Fragment.

Når der trykkes på den ene knap vises det ene Fragment, og når der trykkes på den anden knap vises det andet Fragment.

I hvert Fragment er der en knap som viser en Toast med henholdsvis teksten First Fragment og Second Fragment.



colors.xml

Der bruges nogle farver i de forskellige layouts i programmet

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
```

```

<!-- color's used in our project -->
<color name="main_background">#f8b</color>
<color name="black">#000</color>
<color name="green">#0f0</color>
<color name="white">#fff</color>
<color name="blue">#00f</color>
<color name="button_background_color">#925</color>
</resources>

```

activity_main.xml

I layoutet activity_main.xml er der under knapperne en FrameLayout med en id. Det er i denne Fragmenterne bliver lagt skiftesvis.

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    android:background="@color/main_background"
    android:padding="10sp"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnShowFirstFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/button_background_color"
        android:text="First Fragment"
        android:textColor="@color/white"
        android:textSize="20sp" />
    <Button
        android:id="@+id/btnShowSecondFragment"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:background="@color/button_background_color"
        android:text="Second Fragment"
        android:textColor="@color/white"
        android:textSize="20sp" />
    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="10dp" />
</LinearLayout>

```

MainActivity.java

I MainActivity.java kaldes metoden loadFragment() med den aktuelle Fragment, som der skal skiftes til, når der trykkes på en af knapperne. Ved hjælp af FragmentManager erstattes Fragmentet i FrameLayoutet.

```

package dk.tec.toknappertofragmenter;

import android.app.Fragment;
import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.os.Bundle;

```

```

import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity
{
    Button btnShowFirstFragment, btnShowSecondFragment;

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

        btnShowFirstFragment = (Button) findViewById(R.id.btnShowFirstFragment);
        btnShowSecondFragment = (Button) findViewById(R.id.btnShowSecondFragment);

        btnShowFirstFragment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                loadFragment(new FirstFragment());
            }
        });

        btnShowSecondFragment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                loadFragment(new SecondFragment());
            }
        });
    }

    private void loadFragment(Fragment fragment)
    {
        FragmentManager fm = getFragmentManager();
        FragmentTransaction fragmentTransaction = fm.beginTransaction();
        fragmentTransaction.replace(R.id.frameLayout, fragment);
        fragmentTransaction.commit();
    }
}

```

fragment_first.xml

Fragmneternes layout er begge kun en tekst og en knap.

```

<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="dk.tec.toknappertofragmenter.FirstFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:text="This is First Fragment"
        android:textColor="@color/black"
        android:textSize="25sp" />
    <Button

```

```

        android:id="@+id	btnInFirstFragment"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:background="@color/green"
        android:text="Button In First Fragment"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold" />
    </RelativeLayout>

```

FirstFragment.java

Fragment-klasserne arver fra Fragment.

Når en Fragment loades, kaldes onCreateView(), som inflater layoutet og returnerer det resulterende View.

Der bliver også lige lavet en eventhandler på Fragmentets knap, som viser en Toast, når der trykkes på knappen.

```

package dk.tec.toknappertofragmenter;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

public class FirstFragment extends Fragment
{
    View view;
    Button btnInFirstFragment;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState)
    {
        view = inflater.inflate(R.layout.fragment_first, container, false);

        btnInFirstFragment = (Button) view.findViewById(R.id.btnInFirstFragment);
        btnInFirstFragment.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                Toast.makeText(getActivity(), "First Fragment", Toast.LENGTH_LONG).show();
            }
        });
        return view;
    }
}

```

fragment_second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    tools:context="dk.tec.toknappertofragmenter.SecondFragment">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:text="This is Second Fragment"
        android:textColor="@color/black"
        android:textSize="25sp" />
    <Button
        android:id="@+id	btnInSecondFragment"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginLeft="20dp"
        android:layout_marginRight="20dp"
        android:background="@color/blue"
        android:text="Button In Second Fragment"
        android:textColor="@color/white"
        android:textSize="20sp"
        android:textStyle="bold" />
</RelativeLayout>
```

SecondFragment.java

```
package dk.tec.toknappertofragmenter;

import android.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.Toast;

public class SecondFragment extends Fragment
{
    View view;
    Button btnInSecondFragment;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState)
    {
        view = inflater.inflate(R.layout.fragment_second, container, false);
        btnInSecondFragment = (Button) view.findViewById(R.id.btnInSecondFragment);
        btnInSecondFragment.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                Toast.makeText(getActivity(), "Second Fragment", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

```
        return view;
    }
}
```

13. Grafik

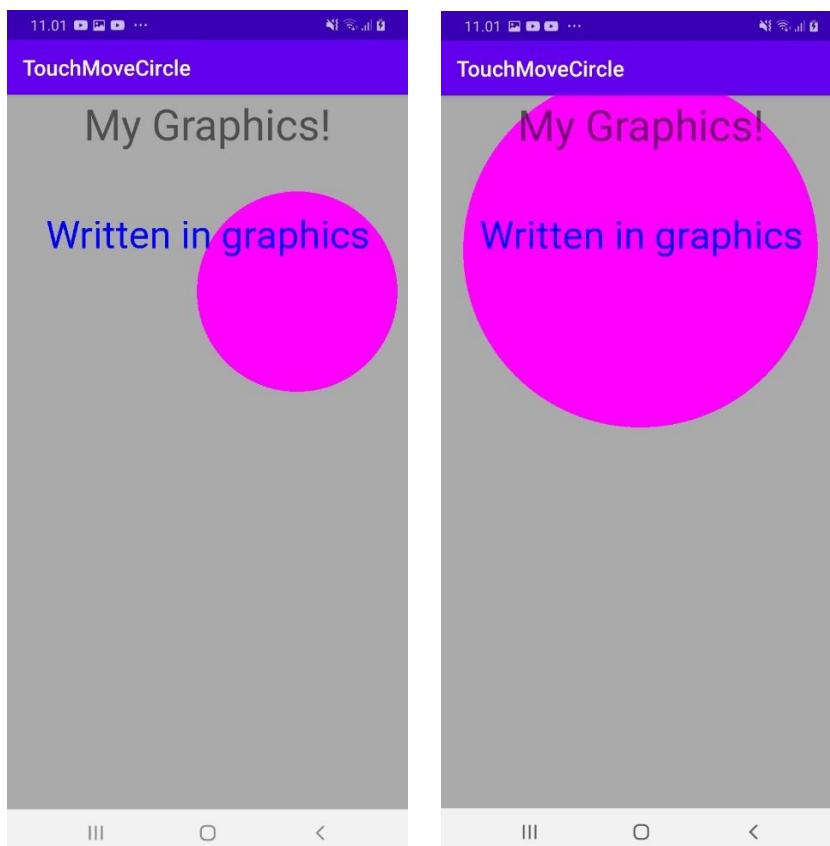
Grafik og flytning af objekter

13.1 Grafik og onTouchEvent

I det følgende er der grafisk tegnet en cirkel, som kan flyttes med fingeren, og størrelsen kan ændres med to fingre.

Der tegnes på skærmen ved at oprette en klasse MyGraphics som arver fra View, hvor metoden onDraw() overrides og der tegnes i denne.

Et objekt af MyGraphics lægges ud i brugerfladen.



I layoutet benyttes FrameLayout, som er en stack med TextView i front og objektet MyGraphics, som tilføjes til at ligge senere, vil ligge bagved.

Framelayoutet får en id, så der kan tilføjes til den i Java-koden.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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:id="@+id/graphics_layout" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:textSize="40sp"
        android:text="@string/graphics_title" />

</FrameLayout>

```

I MainActivity oprettes et objekt af MyGraphics og tilføjes til FrameLayout.

Index 0 i kaldet af addView angiver at view'et lægges bagerst.

```

package com.example.touchmovecircle;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.FrameLayout;

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

        MyGraphics myGraphics = new MyGraphics(this);
        FrameLayout l = findViewById(R.id.graphics_layout);
        l.addView(myGraphics, 0);
    }
}

```

I metoden onDraw() tegnes cirklen i henhold til xPos, yPos for centrum, og radius.

Der findes også størrelsen af skærmen, og dette bruges til at sætte tekststørrelsen og placering af den tekst som også tegnes grafisk i onDraw().

Bemærk at det er størrelsen på den fysiske skærm, der fastlægges. View'et kan godt være mindre end skærmen. View'ets størrelse fastlægges når det placeres i layoutet. Her vil eventhandleren onSizeChanged() kaldes

```

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh)
{
    super.onSizeChanged(w, h, oldw, oldh);
    viewHeight = h;
    viewWidth = w;
}

```

I metoden onTouch() håndteres om fingeren sættes ned på skærmen, om den flyttes og om den løftes fra skærmen. Der håndteres også om der er to fingre på skærmen.

```

package com.example.touchmovecircle;

import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Point;
import android.view.Display;
import android.view.MotionEvent;
import android.view.View;
public class MyGraphics extends View
{
    Paint paint = new Paint();
    int xPos, yPos, radius;
    int xPrevious, yPrevious;
    boolean moving = false;

    int viewWidth;
    int viewHeight;
    int textSize;
    public MyGraphics(MainActivity mainActivity)
    {
        super(mainActivity);
        setBackgroundColor(0xFFAAAAAA);
        Display display = mainActivity.getWindowManager().getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);
        viewWidth = size.x;
        viewHeight = size.y;
        textSize = viewHeight/20;

        xPos = size.x/2;
        yPos = size.y/2;
        radius = size.x/4;
    }

    @Override
    protected void onDraw(Canvas canvas)
    {
        paint.setColor(0xFFFF00FF);
        canvas.drawCircle(xPos, yPos, radius, paint);

        paint.setColor(0xFF0000FF);
        paint.setTextSize(textSize);
        paint.setTextAlign(Paint.Align.CENTER);
        canvas.drawText("Written in graphics", viewWidth/2, viewHeight/5, paint);
    }

    @Override
    public boolean onTouchEvent(MotionEvent event)
    {
        int xNew = (int)event.getX();
        int yNew = (int)event.getY();

        if(event.getAction() == MotionEvent.ACTION_DOWN)
        {
            int xDiff = xNew - xPos;
            int yDiff = yNew - yPos;

            // Er der ramt inden for cirklen?
            if(Math.sqrt(xDiff * xDiff + yDiff * yDiff) < radius)
            {
                moving = true;
                xPrevious = xNew;
                yPrevious = yNew;
            }
        }
    }
}

```

```

}else
if(event.getAction() == MotionEvent.ACTION_MOVE)
{
    if(moving)
    {
        xPos += xNew - xPrevious;
        yPos += yNew - yPrevious;
        xPrevious = xNew;
        yPrevious = yNew;
        invalidate();
    }
}else
if(event.getAction() == MotionEvent.ACTION_UP)
{
    moving = false;
}

// Er der to fingre på, så juster diametern til afstanden imellem de to fingre
if(event.getPointerCount() == 2)
{
    int x1 = (int)event.getX(0);
    int y1 = (int)event.getY(0);
    int x2 = (int)event.getX(1);
    int y2 = (int)event.getY(1);

    int x1Rel = x1 - xPos;
    int y1Rel = y1 - yPos;
    int x2Rel = x2 - xPos;
    int y2Rel = y2 - yPos;

    // Er der ramt inden for cirklen?
    if(Math.sqrt(x1Rel * x1Rel + y1Rel * y1Rel) < radius
       && Math.sqrt(x2Rel * x2Rel + y2Rel * y2Rel) < radius)
    {
        // Set radius til afstanden immellem fingrene?
        int xDiff = x1 - x2;
        int yDiff = y1 - y2;
        radius = (int)(Math.sqrt(xDiff * xDiff + yDiff * yDiff));
        invalidate();
    }
}
return true;
}
}

```

13.2 Gentagent kald af onDraw og autonome bevægelser

1. Med postInvalidateDelayed() kan onDraw() afsluttes med at bestille et kald igen efter et bestemt tidsrum.

```
protected void onDraw(Canvas canvas)
{
    einstein.setBounds(xEinstein, yEinstein, xEinstein + 100, yEinstein + 100);
    einstein.draw(canvas);

    xEinstein += xVelocity;
    yEinstein += yVelocity;

    postInvalidateDelayed(100);
}
```

2. Med en Timer kan der bestilles et kald af onDraw() kontinuerligt.

```
Timer timer = new Timer();
timer.scheduleAtFixedRate(
    new TimerTask()
    {
        @Override
        public void run()
        {
            postInvalidate();
        }
    }, 1000, 1000 );
```

3. Ved at udføre MyGraphics i en ny tråd, kan der i run() udføres en uendelig løkke, som bestiller et kald af onDraw() for hvert gennemløb.

```
public class MyGraphics extends View implements Runnable
{
    ...
    public void run()
    {
        while (!quit)
        {
            postInvalidate();

            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    ...
}
```

Thread.sleep() kaster en checked exception og derfor kræver try/catch.

Istedet kan der bruges SystemClock.sleep(), som ikke kræver try/catch.

```
public class MyGraphics extends View implements Runnable
{
    ...
    public void run()
    {
        while (!quit)
        {
            postInvalidate();
            SystemClock.sleep(100);
        }
    }
    ...
}
```

Og så skal run() udføres ved at starte en ny tråd med myGraphics objektet, som indeholder run().

```
package dk.tec;

import dk.tec.touchmovecircle.R;
import android.os.Bundle;
import android.app.Activity;
import android.widget.FrameLayout;

public class GraphicsActivity extends Activity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_graphics);

        MyGraphics myGraphics = new MyGraphics(this);
        FrameLayout l = (FrameLayout) findViewById(R.id.graphics_layout);
        l.addView(myGraphics, 0);

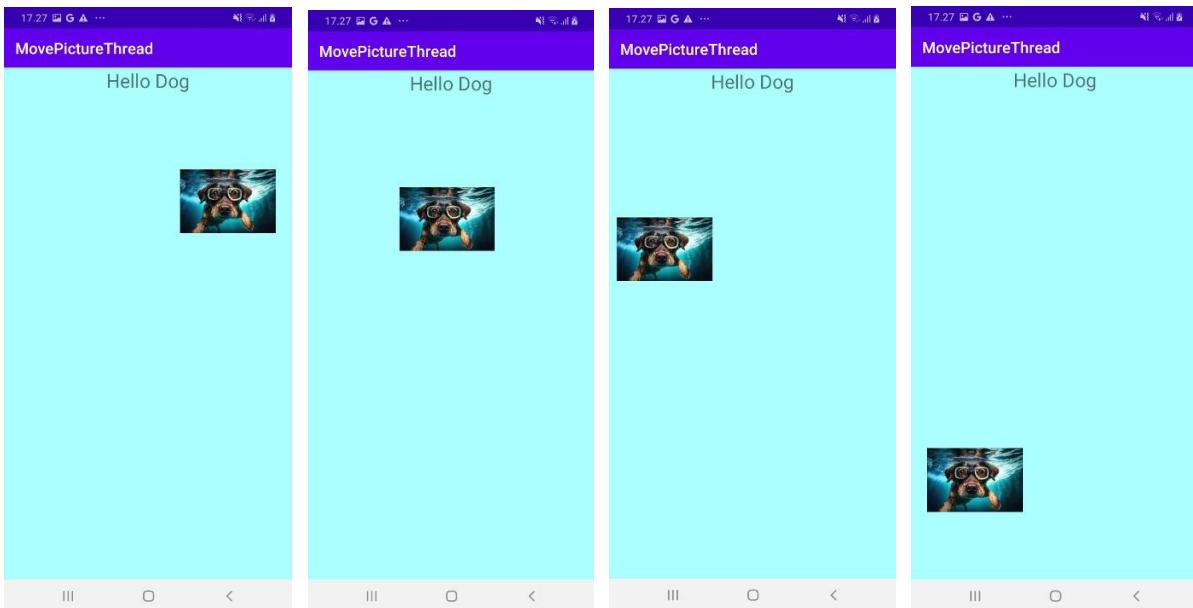
        Thread graphicsThread = new Thread(myGraphics);
        graphicsThread.start();
    }
}
```

At hente billedet ind i en variabel af typen Drawable.

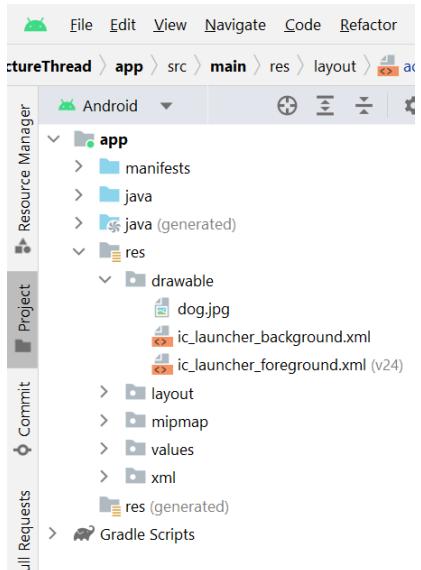
```
Drawable d = getResources().getDrawable(R.drawable.foobar, null);
```

Eksempel på billede der kører rundt på skærmen i en anden tråd.

Retningen vendes hver gang der rammes en kant.



Billedet lægges i mappen drawabel under mappen res.



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    android:gravity="center_horizontal">
```

```

    android:background="#AAFFFF"
    android:id="@+id/theLayout"
    tools:context=".MainActivity">

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="25sp"
    android:text="Hello Dog"
/>
</LinearLayout>

```

MainActivity.java

```

package dk.tec.movepicturethread;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.LinearLayout;

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

        LinearLayout theLayout = findViewById(R.id.theLayout);
        MyView mv = new MyView(this);
        theLayout.addView(mv);
        Thread t = new Thread(mv);
        t.start();
    }
}

```

MyView.java

```

package dk.tec.movepicturethread;

import android.graphics.Canvas;
import android.graphics.drawable.Drawable;
import android.view.View;

public class MyView extends View implements Runnable
{
    MainActivity main;
    Drawable drwDog;
    int xDog = 0, yDog = 0;
    int xMove = 5, yMove = 5;
    int dogWidth, dogHeight;
    int viewWidth, viewHeight;

    public MyView(MainActivity mainActivity)
    {
        super(mainActivity);
        main = mainActivity;
        drwDog = main.getResources().getDrawable(R.drawable.dog, null);
        dogWidth = drwDog.getMinimumWidth()/2;
        dogHeight = drwDog.getMinimumHeight()/2;
    }
}

```

```

@Override
protected void onSizeChanged(int w, int h, int oldw, int oldh) {
    super.onSizeChanged(w, h, oldw, oldh);
    viewWidth = w;
    viewHeight = h;
}

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    drwDog.setBounds(xDog, yDog, xDog + dogWidth, yDog + dogHeight);
    drwDog.draw(canvas);
}

@Override
public void run() {
    while(true)
    {
        xDog += xMove;
        yDog += yMove;
        if(xDog > (viewWidth - dogWidth) )
            xMove = Math.abs(xMove) * -1;
        if(xDog < 0)
            xMove = Math.abs(xMove) * 1;
        if(yDog > (viewHeight - dogHeight))
            yMove = Math.abs(yMove) * -1;
        if(yDog < 0)
            yMove = Math.abs(yMove) * 1;

        try {
            Thread.sleep(10);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        postInvalidate();
    }
}
}

```

14. Asynkrone programmer

En Android-app udføres i en tråd også kaldet GUI-tråden. Hvis der ventes på noget, vil GUI-tråden blokeres, så der ikke kan gøres noget med app'en så længe.

Når der skal ske noget andet samtidig med at hovedprogrammet/GUI-tråden udføres, og uden at blokere GUI-tråden, anvendes almindelig vis en ny tråd.

En anden tråd må ikke tilgå GUI. Hvis den gør det, vil programmet fejle.

Kode der skal udføres i en ny tråd, skal enten være et objekt af en klasse, der arver fra klassen Thread, eller det skal være et objekt af en klasse, der implementerer Runnable. I begge tilfælde vil der være en metode run(), som kaldes af tråden, og altså indeholde den kode der skal udføres i tråden.

Hvis der ikke skal ske noget løbende parallelt med GUI-tråden, men bare en forsinket hændelse, evt. en gentagen hændelse, er det ikke nødvendigt at oprette en ny tråd.

Til det formål benyttes et Handler-objekt, som kan sætte noget i gang efter et bestemt tidsrum.

Handleren skal have et Runnable-objekt, som indeholder metoden run() og den kan så kalde run() efter en angivet tid.

Udførelse af en asynkron task

I det følgende oprettes en Handler og når der trykkes på btnDoTask oprettes der en Runnable, task, som har en run() metode, der ændrer knappens tekst.

Handleren sættes til at udføre denne Runnable efter 10 sekunder ved at kalde Handlerens postDelayed().

Hvis der trykkes på btnCancel inden de 10 sekunder er gået, vil removeCallbacks() aflyse udførelsen.

```
public class Asynk1Handler extends Activity implements OnClickListener {

    Handler handler = new Handler();
    Runnable task;
    Button btnDoTask, btnCancel;

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

        TableLayout tl = new TableLayout(this);

        EditText txtEdit = new EditText(this);
        txtEdit.setText("Write something while waiting");
        tl.addView(txtEdit);

        btnDoTask = new Button(this);
        btnDoTask.setText("Do task in 10 seconds");
        tl.addView(btnDoTask);

        setContentView(tl);
    }

    public void onClick(View v) {
        if (v == btnCancel) {
            handler.removeCallbacks(task);
        } else if (v == btnDoTask) {
            task = new Runnable() {
                public void run() {
                    txtEdit.setText("The task has been done!");
                }
            };
            handler.postDelayed(task, 10000);
        }
    }
}
```

```
btnCancel = new Button(this);
btnCancel.setText("Cancel task");
tl.addView(btnCancel);

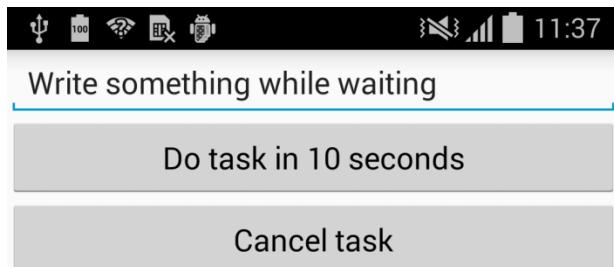
setContentView(tl);

btnDoTask.setOnClickListener(this);
btnCancel.setOnClickListener(this);
}

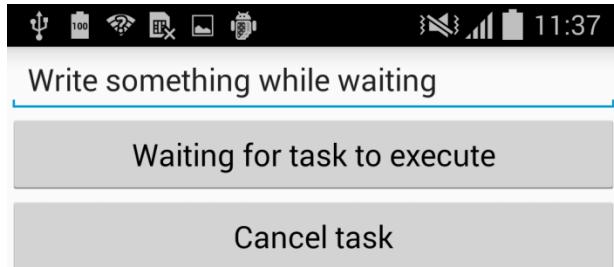
public void onClick(View v) {

    if (v == btnDoTask) {
        btnDoTask.setText("Waiting for task to execute");
        task = new Runnable() {
            public void run() {
                btnDoTask.setText("Task finished!");
            }
        };
        handler.postDelayed(task, 10000); // udfør om 10 sekunder
    } else if (v == btnCancel) {
        handler.removeCallbacks(task);
    }
}
```

Det ser sådan ud.

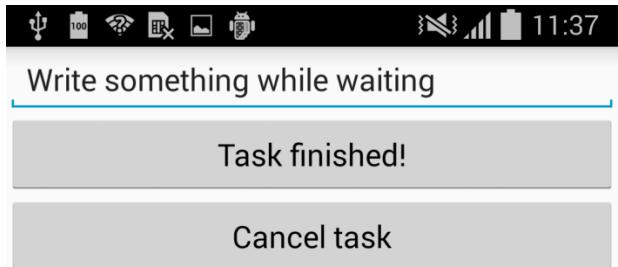


Tryk på knappen Do task..



Der kan imens der ventes på at task udføres, skrives i tekstboksen, fordi app'en ikke er blokeret.

Så går de 10 sekunder ...



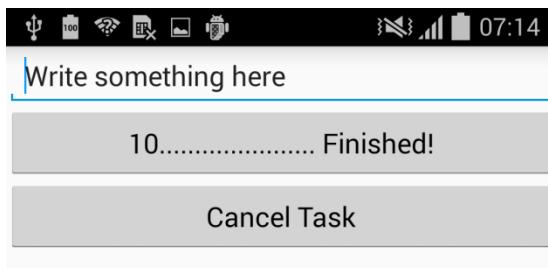
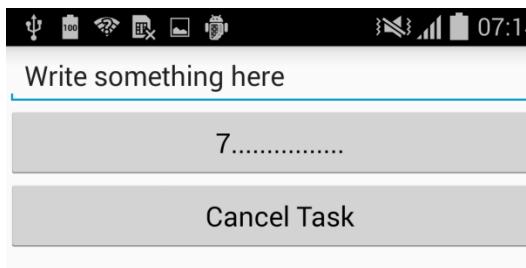
Hvis der efter tryk på Do task, trykkes på Cancel task, vil udførelsen af task blive aflyst. Så der ikke sker mere.

Gentagent udførelse af task

Hvis der inde i run(), som kaldes af handleren, bedes om en ny udførelse, vil run() blive kaldt igen indtil logikken stopper det. Her er vist at run() udføres hvert sekund i 10 sekunder.

```
if (v == btnDoTask)
{
    btnDoTask.setText("Working ...");
    task = new Runnable()
    {
        int seconds = 0;
        String strDots = "...";

        public void run()
        {
            if (seconds < 10)
            {
                seconds++;
                strDots += "...";
                btnDoTask.setText(seconds + strDots);
                handler.postDelayed(this, 1000);
            } else {
                btnDoTask.setText(seconds + strDots + " Finished!");
            }
        }
    };
    handler.postDelayed(task, 1000);
}
```



Igen kan forløbet stoppes ved at trykke på Cancel Task.

En task kan også sættes i gang med runOnUiThread(). Men her kan der ikke angives en tidsforsinkelse.

```
runOnUiThread(task);
```

Blokere hovedtråden

Hvis hovedtråden/GUI-tråden blokeres, det være sig hvad som helst der tager nogen tid, f.eks. io-operationer eller som her en simpel sleep(), vil app'en fryse.

```
public class Asynk2Thread extends Activity implements OnClickListener
{
    Button btnDoTask;

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

        TableLayout tl = new TableLayout(this);
        EditText txtEdit = new EditText(this);
        txtEdit.setText("Write something here");
        tl.addView(txtEdit);

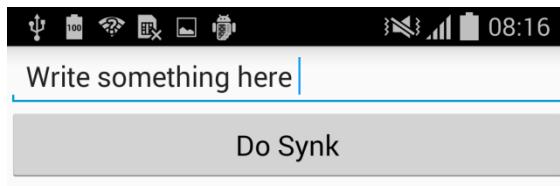
        btnDoTask = new Button(this);
        btnDoTask.setText("Do Sync");
        tl.addView(btnDoTask);

        setContentView(tl);

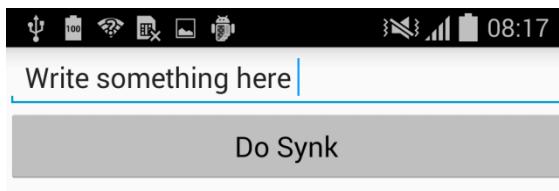
        btnDoTask.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        if (v == btnDoTask) {
            btnDoTask.setText("Working ...");
            SystemClock.sleep(10000);
            btnDoTask.setText("Finished!");
        }
    }
}
```

Programmet startes....

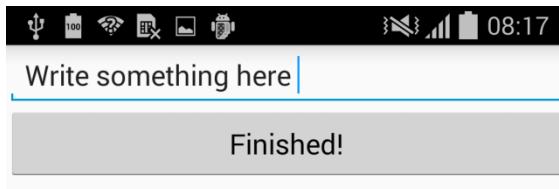


Der trykkes på knappen. Knappen "går ind" og bliver mørk, men kommer ikke ud igen og cursoren i tekstfeltet fryser.



Brugerfladen bliver nemlig ikke opdateret før eventhandleren afsluttes, så "Working" vises aldrig.

Første når sleep() afsluttes, afsluttes eventhandleren og brugerfladen opdateres.



(Jacob Nordfalk viste hvordan han åbenbart kunne gøre noget i teksboksen, som gav fejlen ANR Application Not Responding)

Blokering i en ny tråd som opdaterer GUI

I det følgende oprettes en ny tråd, som udfører blokeringen sleep(). Det er godt.

Men den opdaterer GUI og det er ikke godt.

```
public class Asynk2Thread extends Activity implements OnClickListener
{
    Button btnDoTask;

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

        TableLayout tl = new TableLayout(this);
        EditText txtEdit = new EditText(this);
        txtEdit.setText("Write something here");
        tl.addView(txtEdit);

        btnDoTask = new Button(this);
        btnDoTask.setText("Do Async");
        tl.addView(btnDoTask);

        setContentView(tl);

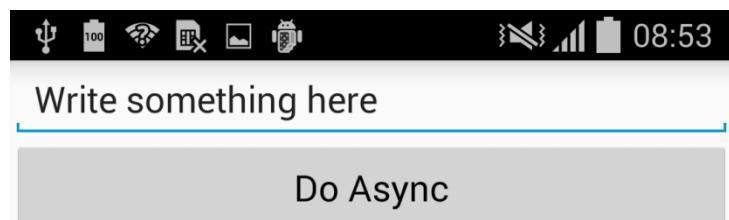
        btnDoTask.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        if (v == btnDoTask) {
```

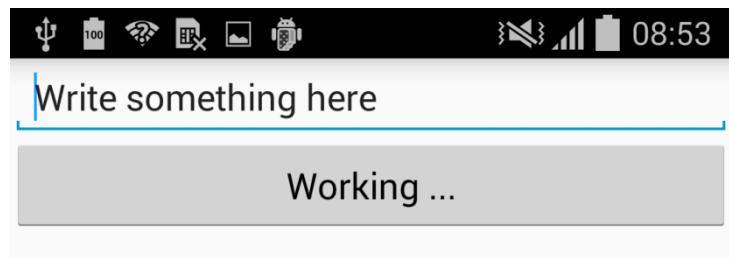
```

        btnDoTask.setText("Working ...");
        Runnable r = new Runnable() {
            @Override
            public void run()
            {
                SystemClock.sleep(10000);
                btnDoTask.setText("Finished!");
            }
        };
        new Thread(r).start();
    }
}

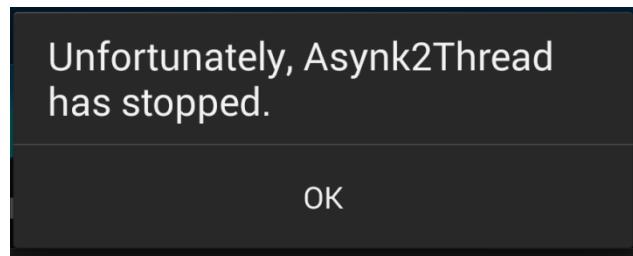
```



Der trykkes på knappen og nu kan der manipuleres i brugerfladen imens sleep() blokerer den nye tråd.



Men tråden slutter med at opdatere GUI og det må den ikke.



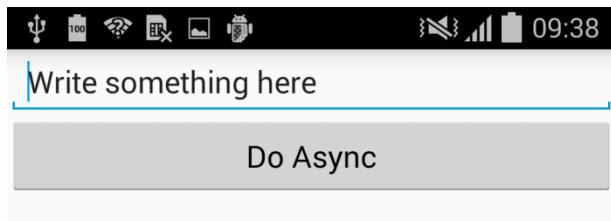
CalledFromWrongThreadException: Only the original thread that created a view hierarchy can touch its views.

Blokering i ny tråd og overgivelse til hovedtråd at opdatere GUI

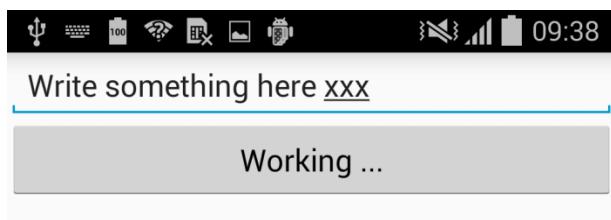
Grunden til at det kun er GUI-tråden der må manipulere GUI, er at Views ikke er trådsikre. Hvis flere tråde tager fat i samme View samtidig, kunne det give konflikt. Trådsikre Views ville blive meget komplicerede. Derfor skal alle tråde der ønsker at opdatere GUI, bede hovedtråden om at gøre det.

I stedet for at manipulere GUI i den nye tråd, oprettes der endnu en Runnable, som manipulerer GUI. Denne Runnable overgives til hovedtråden ved at kalde metoden runOnUiThread().

```
...
public void onClick(View v)
{
    if (v == btnDoTask) {
        btnDoTask.setText("Working ...");
        Runnable r = new Runnable() {
            @Override
            public void run()
            {
                SystemClock.sleep(10000);
                Runnable rr = new Runnable() {
                    @Override
                    public void run()
                    {
                        btnDoTask.setText("Finished!");
                    }
                };
                runOnUiThread(rr);
            }
        };
        new Thread(r).start();
    }
}
```

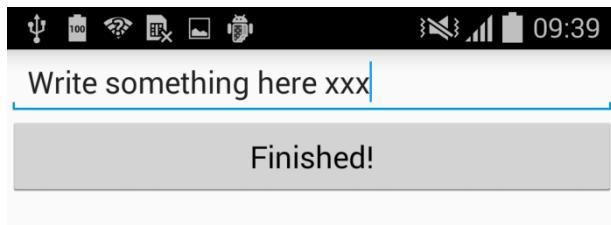


Der trykkes ...



Der kan manipuleres i brugerfladen.

Og den nye tråd kan ændre i brugerfladen via hovedtråden.



AsyncTask

Klassen AsyncTask gør det lidt nemmere at udføre noget i en ny tråd og opdatere GUI.

Der skal oprettes et objekt af en klasse der arver AsyncTask og implementerer metoden doInBackground() som er abstract. Det er den metode, der kaldes af baggrundstråden.

Desuden overrides metoden onPostExecute(). Hovedtråden vil kalde denne, når doInBackground() er afsluttet.

```
public class Asynk3AsyncTask extends Activity implements OnClickListener
{
    Button btnDoTask;

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

        TableLayout tl = new TableLayout(this);
        EditText txtEdit = new EditText(this);
        txtEdit.setText("Write something here");
        tl.addView(txtEdit);

        btnDoTask = new Button(this);
        btnDoTask.setText("Do Async");
        tl.addView(btnDoTask);

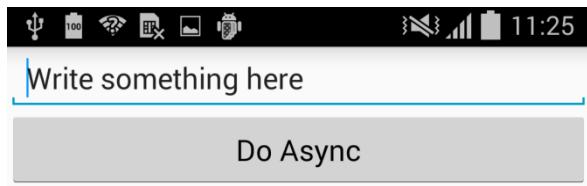
        setContentView(tl);

        btnDoTask.setOnClickListener(this);
    }

    public void onClick(View v)
    {
        if (v == btnDoTask) {
            btnDoTask.setText("Working ...");

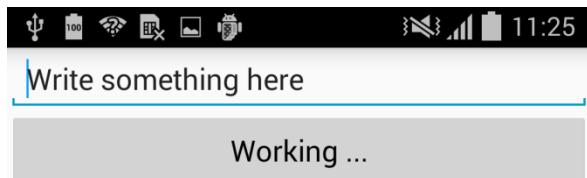
            class AsyncTask1 extends AsyncTask
            {
                @Override
                protected Object doInBackground(Object[] params)
                {
                    SystemClock.sleep(10000);
                    return null;
                }

                @Override
                protected void onPostExecute(Object result)
                {
                    btnDoTask.setText("Finished!");
                }
            }
            new AsyncTask1().execute();
        }
    }
}
```

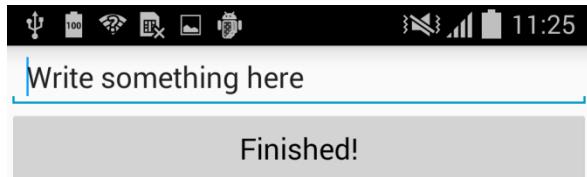


Ved at kalde AsyncTask-objektets execute() kaldes doInBackground().

Der kan manipuleres med brugerfladen.



Når doInBackground afsluttes kaldes onPostExecute().



Og så er tråden slut.

AsyncTask med opdatering af GUI undervejs

Klassen AsyncTask har yderligere en metode onProgressUpdate(), som kan opdatere GUI imens baggrundstråden kører.

Metoden doInBackground() udføres i baggrundstråden og kan ved at kalde publishProgress() sende data til metoden onProgressUpdate(), som kører i hovedtråden.

Når doInBackground() afsluttes sendes det returnerede til metoden onPostExecute(), som kører i hovedtråden.

AsyncTask.execute() kan modtage parametre, som overgives til doInBackground().

I det følgende program gives tallet 100 til execute(). I doInBackground() sættes en forløkke til at gentages disse 100 gange. Der sættes en sleep() så det hele tager 10 sekunder. For hvert gennemløb sendes tællerværdien til onProgressUpdate() ved at kalde publishProcess(). Her udskrives på knappen samt på en ProgressBar. Når doInBackground() slutter sendes "Finished" til onPostExecute().

```
public class Asynk3bAsyncTask extends Activity implements View.OnClickListener
{
    Button btnDoTask;    ProgressBar progress;

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

        TableLayout tl = new TableLayout(this);
        EditText txtEdit = new EditText(this);
        txtEdit.setText("Write something here");
        tl.addView(txtEdit);

        btnDoTask = new Button(this);
        btnDoTask.setText("Do Async");
        tl.addView(btnDoTask);

        progress = new ProgressBar(this, null, android.R.attr.progressBarStyleHorizontal);
        tl.addView(progress);

        setContentView(tl);

        btnDoTask.setOnClickListener(this);
    }

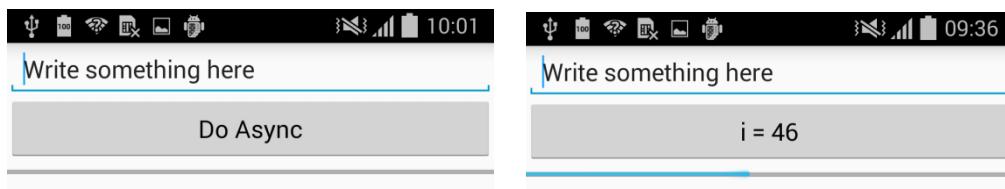
    public void onClick(View v)
    {
        if (v == btnDoTask)
        {
            btnDoTask.setText("Working ...");
            new AsyncTask()
            {
                @Override
                protected Object doInBackground(Object... params)
                {
                    int antal = (Integer) params[0];
                    for(int i = 0; i < antal; i++)
                    {
                        SystemClock.sleep(10000 / antal);
                        this.publishProgress(i);
                    }
                    return "Finished!";
                }
            }.execute(100);
        }
    }
}
```

```

@Override
protected void onProgressUpdate(Object[] progress) {
    int i = (Integer) progress[0];
    btnDoTask.setText("i = " + i);
    Asynk3bAsyncTask.this.progress.setProgress(i);
}
@Override
protected void onPostExecute(Object result)
{
    btnDoTask.setText("Result: " + result);
}
}.execute(100);
}
}
}

```

Det ser sådan ud.



Problemer med AsyncTask

- Hvis flere programmer samtidig bruger AsyncTask, vil de køre i samme baggrundstråd.
Der kan i stedet asyncTask.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR,100);
- Når skærmen vendes oprettes et nyt Aktivitetsobjekt, så de views, som AsyncTask opdaterer er de gamle, og ikke dem i den nye aktivitet.

15. Menuer

Menu oprettet i XML

I nye versioner er der ikke nogen ActionBar med prikker i.

I values – themes – themes.xml

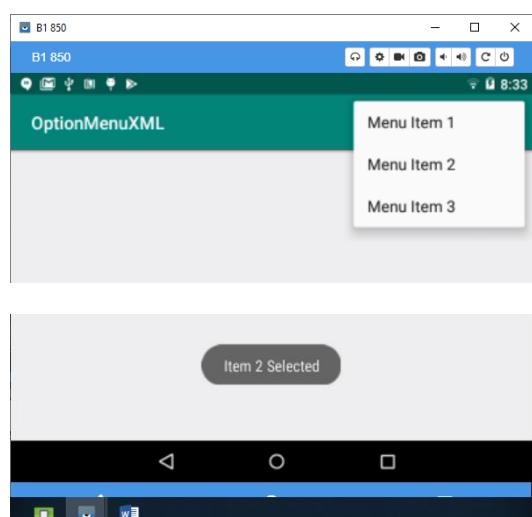
Ændres Theme.MaterialComponents.DayNight.NoActionBar

til f.eks. .DarkActionBar.

OptionMenu

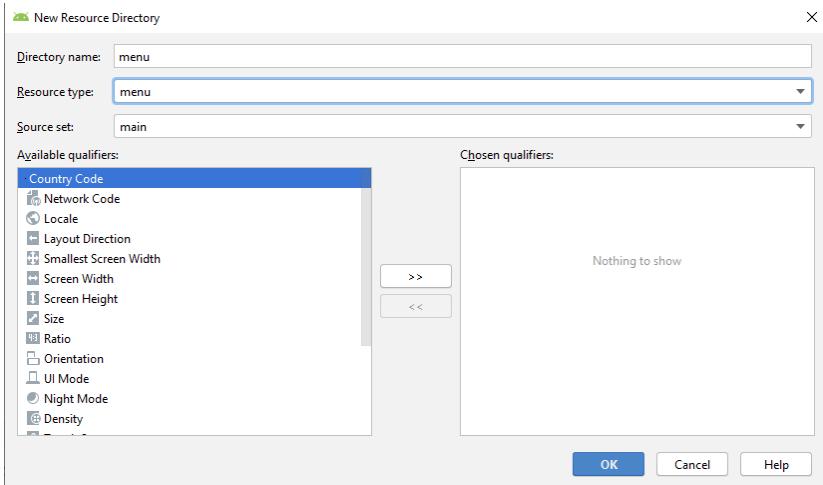
En option menu viser sig som tre prikker i højre side af app-baren øverst på skæmen.

Ved klik på de tre prikker fremkommer option menuen.



Højreklik på mappen res og vælg New | Android Resource Directory.

Vælg menu i Resource type og giv den navnet menu.



Højreklik på den nye mappe menu og vælg New | Menu resource file og giv den navnet option_menu. Fyld nedenstående kode ind i den nye fil.

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/item1"
        android:title="Menu Item 1"/>
    <item android:id="@+id/item2"
        android:title="Menu Item 2"/>
    <item android:id="@+id/item3"
        android:title="Menu Item 3"
        app:showAsAction="withText"/>
</menu>

public class MainActivity extends AppCompatActivity {

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

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.option_menu, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        switch (id){
            case R.id.item1:
                Toast.makeText(getApplicationContext(),"Item 1 Selected",
                        Toast.LENGTH_LONG).show();
                return true;
            case R.id.item2:
                Toast.makeText(getApplicationContext(),"Item 2 Selected",
                        Toast.LENGTH_LONG).show();
                return true;
            case R.id.item3:
                Toast.makeText(getApplicationContext(),"Item 3 Selected",
                        Toast.LENGTH_LONG).show();
        }
    }
}
```

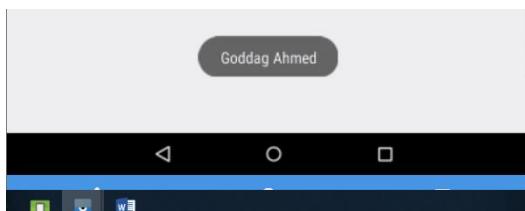
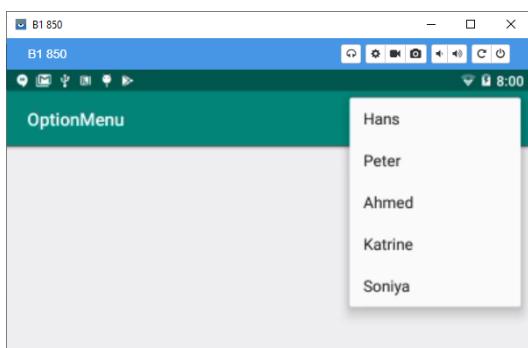
```
        Toast.LENGTH_LONG).show();  
    return true;  
default:  
    return super.onOptionsItemSelected(item);  
}  
}  
}
```

Menuer oprettet i Java

OptionsMenu

En option menu viser sig som tre prikker i højre side af app-baren øverst på skæmen.

Ved klik på de tre prikker fremkommer option menuen.



Der er ikke lavet noget i layoutet.

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu)  
    {  
        populateMenu(menu);  
        return super.onCreateOptionsMenu(menu);  
    }  
  
    private void populateMenu(Menu menu)  
    {  
        menu.add(Menu.NONE, 0, Menu.NONE, "Hans");  
        menu.add(Menu.NONE, 1, Menu.NONE, "Peter");  
        menu.add(Menu.NONE, 2, Menu.NONE, "Ahmed");  
        menu.add(Menu.NONE, 3, Menu.NONE, "Katrine");  
        menu.add(Menu.NONE, 4, Menu.NONE, "Soniya");  
    }  
}
```

```

}

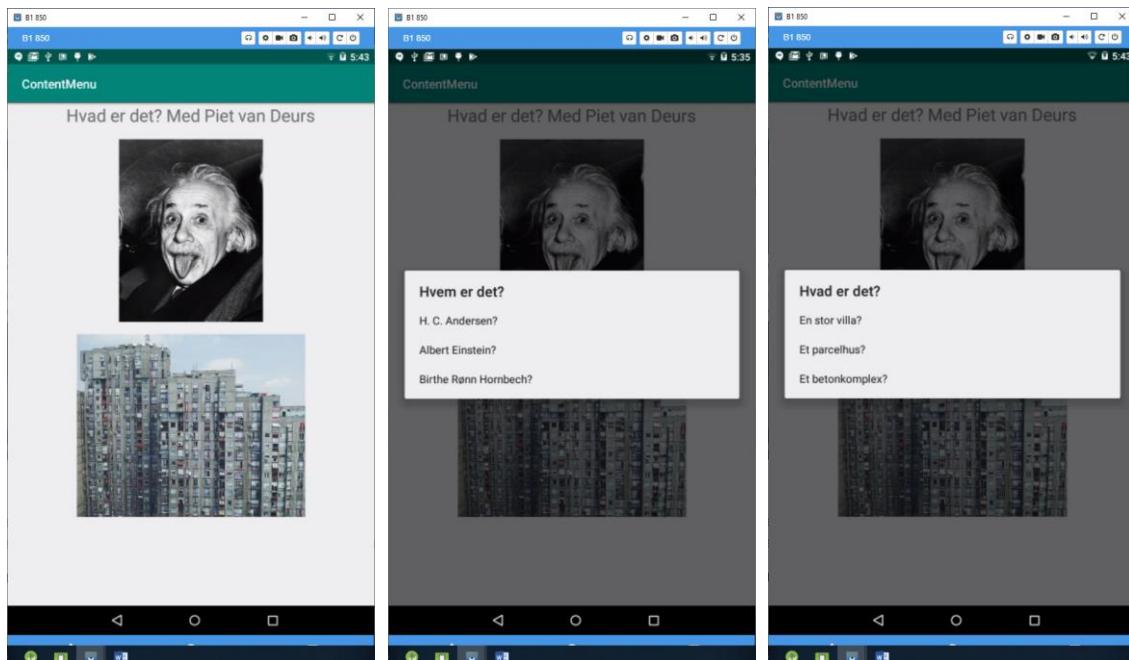
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    String str;
    switch(item.getItemId())
    {
        case 0:
            str = "Hej Hans";
            break;
        case 1:
            str = "Hello Peter";
            break;
        case 2:
            str = "Goddag Ahmed";
            break;
        case 3:
            str = "Godafoten Katrine";
            break;
        case 4:
            str = "Farvel Soniya";
            break;
        default:
            str = "No body";
    }
    Toast.makeText(this, str, Toast.LENGTH_LONG).show();

    return super.onOptionsItemSelected(item);
}
}

```

Eksempel ContextMenu

En context menu knyttes til et element på skærmen og fremkommer ved et længere tryk på det aktuelle element.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/overskrift"
        android:textSize="25sp"
        />
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="300dp"
        android:src="@drawable/einstein"
        android:layout_margin="20dp"
        android:id="@+id/imEinstein"
        />
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="300dp"
        android:src="@drawable/beograd"
        android:id="@+id/imBeograd"
        />
</LinearLayout>

public class MainActivity extends AppCompatActivity
{
    ImageView imEinstein, imBeograd;
```

```

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

    imEinstein = findViewById(R.id.imEinstein);
    imBeograd = findViewById(R.id.imBeograd);
    registerForContextMenu(imEinstein);
    registerForContextMenu(imBeograd);
}

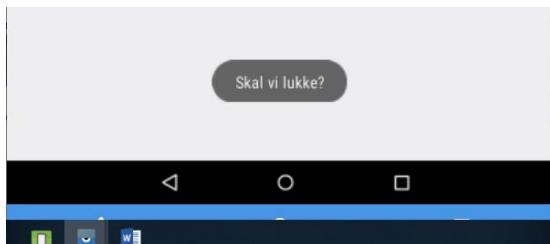
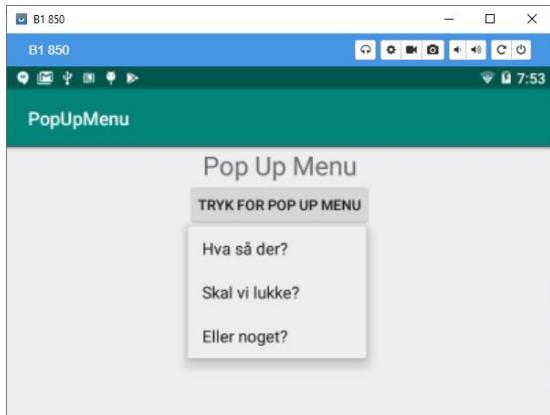
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                               ContextMenu.ContextMenuItemInfo menuInfo)
{
    if(v == imEinstein)
    {
        menu.setHeaderTitle("Hvem er det?");
        menu.add(Menu.NONE, 10, Menu.NONE, "H. C. Andersen?");
        menu.add(Menu.NONE, 11, Menu.NONE, "Albert Einstein?");
        menu.add(Menu.NONE, 12, Menu.NONE, "Birthe Rønn Hornbech?");
    }
    else if(v == imBeograd)
    {
        menu.setHeaderTitle("Hvad er det?");
        menu.add(Menu.NONE, 20, Menu.NONE, "En stor villa?");
        menu.add(Menu.NONE, 21, Menu.NONE, "Et parcelhus?");
        menu.add(Menu.NONE, 22, Menu.NONE, "Et betonkomplex?");
    }
}

@Override
public boolean onContextItemSelected(MenuItem item)
{
    Toast.makeText(this, "Du valgte: " + item.getTitle(), Toast.LENGTH_LONG).show();
    return super.onContextItemSelected(item);
}
}

```

Eksempel Pop Up Menu

Pop Up menu dannes programmatisk når det ønskes, f.eks. i onClick() på en Button.



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Pop Up Menu"
        android:textSize="25sp"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tryk for Pop Up Menu"
        android:id="@+id	btnPopUp"/>
</LinearLayout>
```

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener
{
    Button btnPopUp;

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

        btnPopUp = findViewById(R.id.btnPopUp);
```

```

        btnPopUp.setOnClickListener(this);
    }

@Override
public void onClick(View v)
{
    PopupMenu popupMenu = new PopupMenu(this, btnPopUp);
    //Der kan åpenbart ikke sættes en header i en popupmenu
    popupMenu.getMenu().add(Menu.NONE, 0, Menu.NONE, "Hva så der?");
    popupMenu.getMenu().add(Menu.NONE, 1, Menu.NONE, "Skal vi lukke?");
    popupMenu.getMenu().add(Menu.NONE, 2, Menu.NONE, "Eller noget?");

    popupMenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
        @Override
        public boolean onMenuItemClick(MenuItem item)
        {
            Toast.makeText(MainActivity.this, item.getTitle(),
                    Toast.LENGTH_LONG).show();
            return true;
        }
    });
    popupMenu.show();
}

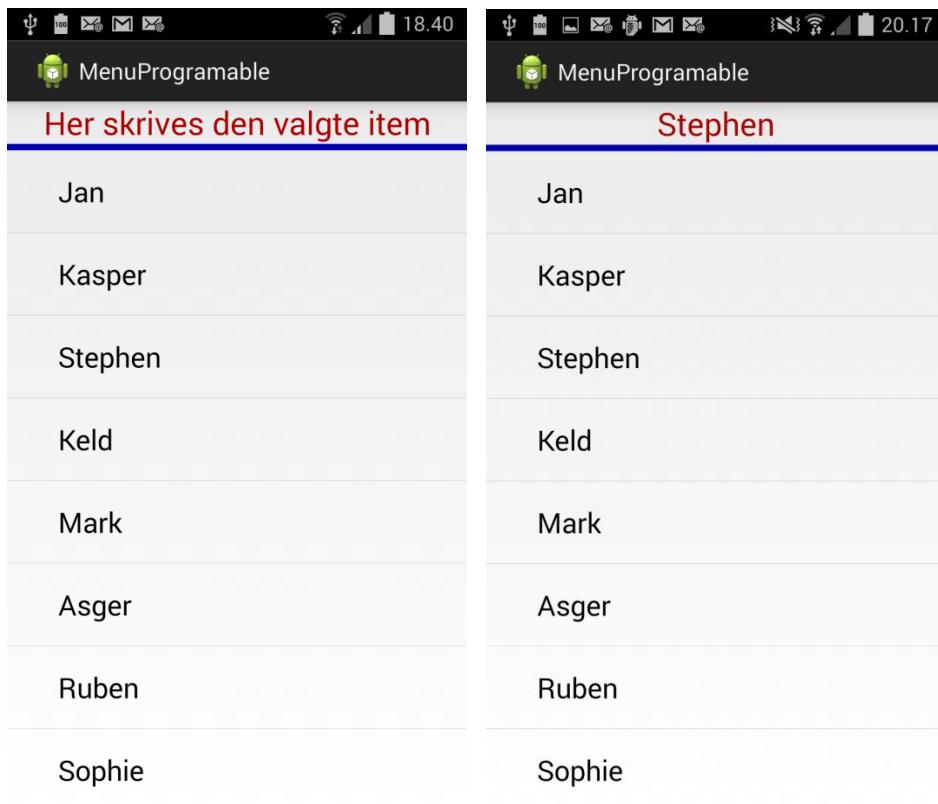
}

```

Eksempel Listview med OptionMenu og ContextMenu

For at vise brugen af menuer, laves et program, der viser en liste af navne på skærmen.

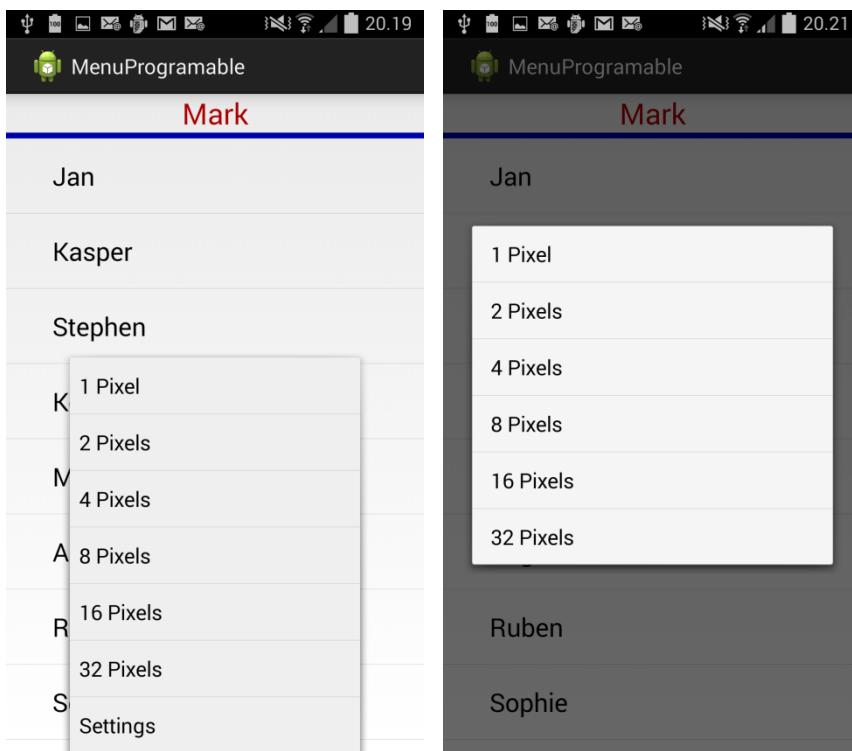
Det valgte item i listen vises foroven på skærmen.



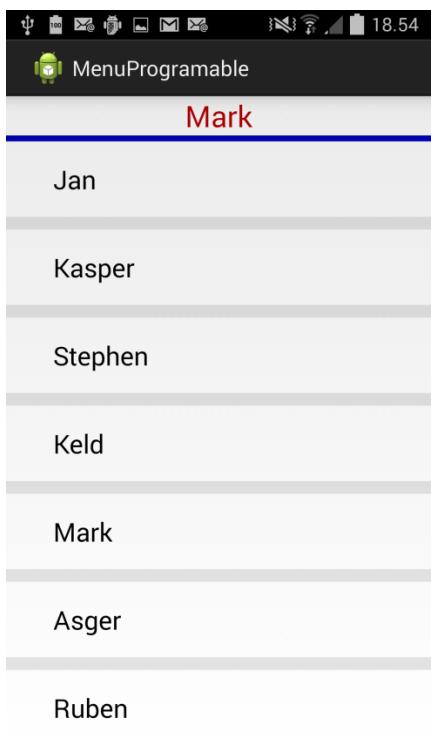
Der oprettes både en Options-Menu og en Context-Menu, som vist herunder.

Options-Menu er den, der fås frem når der trykkes på enhedens "Menu-knap", og Context-Menu er den der vises, når der trykkes og holdes på et view på skærmen, der er sat til at kunne dette. Her er det selve listen, der trykkes på og holdes.

Den første er Options-Menu og den anden er Context-Menu.



Når der vælges en af menuens items, sættes der en afstand i henhold til det valgte antal pixels imellem de enkelte items i listen af navne, som vist herunder.



En nem måde at lave en liste på er, at lade MainActivity arve fra ListActivity. Så kan der laves en liste uden at have nogen layout i xml.

Men da der ønskes at have en TextView foroven til at vise det valgte item, skal der bruges et layout (stadig med ListActivity). Dette layout skal så slutte med et ListView, og dette skal have "@android:id/list" som id. Så fungerer listen stadig som en default ting i aktviteten.

```
<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"
    tools:context="com.example.menutest.MainActivity" >
    <TextView
        android:id="@+id/txtSelectedText"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:textColor="#AA0000"
        android:gravity="center_horizontal"
        android:text="Her skrives det valgte item"/>
    <View android:layout_height="5dp"
        android:layout_width="match_parent"
        android:background="#0000AA"/>
    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:drawSelectorOnTop="false"/>
</LinearLayout>
```

I onCreate() oprettes listen ved i dette tilfælde at bruge et array af navne og kalde den arvede metode setListAdapter.

Desuden sættes der hvilket view, der skal have Context-Menuen, ved at kalde registerForContextMenu(), og det er så selve listen.

```
package dk.tec.jaj.MenuProgrammable;

public class MainActivity extends ListActivity
{
    String[] items;
    String all = "Jan Kasper Stephen Keld Mark Asger Ruben Sophie Daniel Oliver Mathias Patrick";

    TextView txtSelection;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); //Hvis denne udelades vises kun listen

        txtSelection=(TextView) findViewById(R.id.txtSelectedText);

        items = all.split(" ");
        setListAdapter(new ArrayAdapter<String>(
            this,
            android.R.layout.simple_expandable_list_item_1,
```

```

        items));
registerForContextMenu(getListView());
}

```

Metoden `onListItemClick()` skriver foroven på skærmen det valgte list-item, når der vælges et navn. Dette har ikke noget med menerne at gøre.

```

@Override
protected void onListItemClick(ListView l, View v, int position, long id)
{
    txtSelection.setText(items[position]);
}

```

Metoden `onCreateOptionsMenu()` kaldes automatisk når applikationen starter. Der overgives menu-objektet som parameter, og der fyldes så noget i det ved at kalde hjælpemetoden `populateMenu()`.

```

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    populateMenu(menu);
    return super.onCreateOptionsMenu(menu);
}

```

Metoden `onCreateContextMenu()` kaldes automatisk når applikationen starter. Der overgives menu-objektet som parameter, og der fyldes så noget i det ved at kalde hjælpemetoden `populateMenu()`.

```

@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuItemInfo menuInfo)
{
    populateMenu(menu);
    super.onCreateContextMenu(menu, v, menuInfo);
}

```

Metoden `populateMenu()` fylder den overgivne menu med items.

```

private void populateMenu(Menu menu)
{
    menu.add(Menu.NONE, 0, Menu.NONE, "1 Pixel");
    menu.add(Menu.NONE, 1, Menu.NONE, "2 Pixels");
    menu.add(Menu.NONE, 2, Menu.NONE, "4 Pixels");
    menu.add(Menu.NONE, 3, Menu.NONE, "8 Pixels");
    menu.add(Menu.NONE, 4, Menu.NONE, "16 Pixels");
    menu.add(Menu.NONE, 5, Menu.NONE, "32 Pixels");
}

```

Metoden `onOptionsItemSelected()` kaldes når der vælges i en Option-Menu og det valgte item overgives som parameter. Dette gives virkning ved at kalde hjælpemetoden `applyMenuChoice()` og overgive det valgte menu-item.

```

@Override

```

```

public boolean onOptionsItemSelected(MenuItem item)
{
    applyMenuChoice(item);
    return super.onOptionsItemSelected(item);
}

```

Metoden onContextItemSelected() kaldes når der vælges i en Context-Menu og det valgte item overgives som parameter. Dette gives virkning ved at kalde hjælpemetoden applyMenuChoice() og overgive det valgte menu-item.

```

@Override
public boolean onContextItemSelected(MenuItem item)
{
    applyMenuChoice(item);
    return super.onContextItemSelected(item);
}

```

Metoden applyMenuChoice() bruges til at give valg i begge menuer virkning, ved at indsætte en Divider imellem de enkelte navne-items i liste, af størrelse efter det valgte.

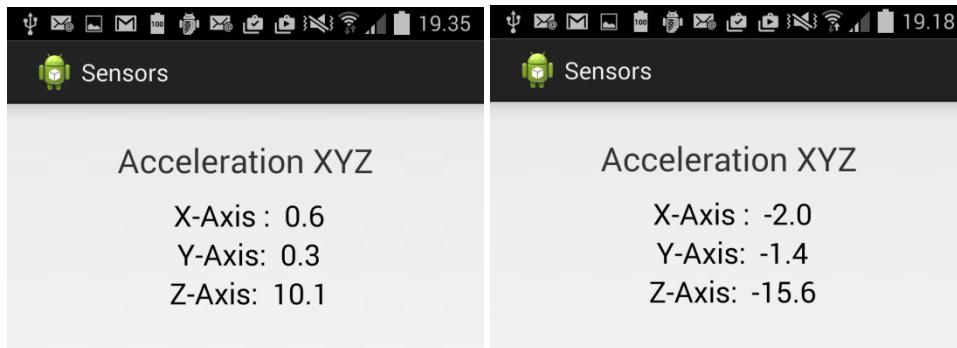
```

private void applyMenuChoice(MenuItem item)
{
    switch(item.getItemId())
    {
        case 0:
            listView().setDividerHeight(1);
            break;
        case 1:
            listView().setDividerHeight(2);
            break;
        case 2:
            listView().setDividerHeight(4);
            break;
        case 3:
            listView().setDividerHeight(8);
            break;
        case 4:
            listView().setDividerHeight(16);
            break;
        case 5:
            listView().setDividerHeight(32);
            break;
    }
}

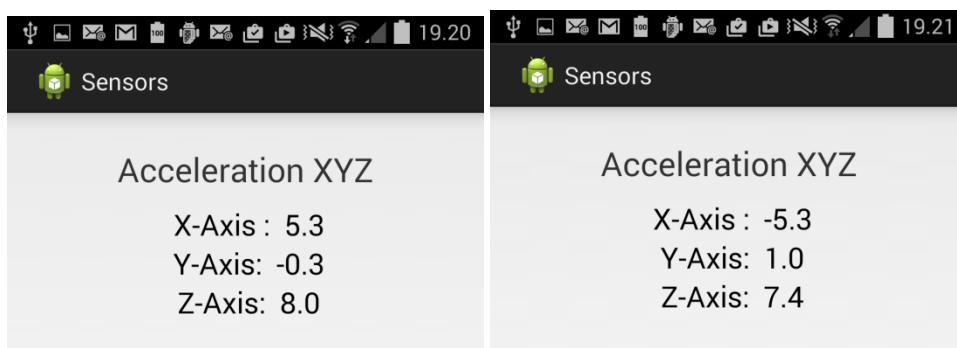
```

16. Sensors

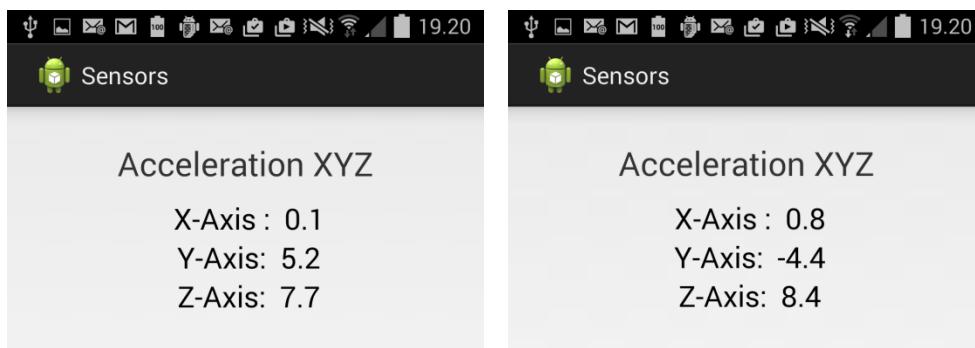
Neutral og rystelse.



Højre side op og venstre side op.



Top op og bund op.



Der udskrives i 3 TextViews. Layoutet er ikke vist.

```

public class MainActivity extends Activity implements SensorEventListener
{
    private SensorManager mSensorManager;
    TextView txtXY, txtXZ, txtZY;

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

        txtX = (TextView) findViewById(R.id.txtX_angle);
        txtY= (TextView) findViewById(R.id.txtY_angle);
        txtZ = (TextView) findViewById(R.id.txtZ_angle);

        mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mSensorManager.registerListener(this,
            mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
            SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    protected void onDestroy()
    {
        super.onDestroy();
        mSensorManager.unregisterListener(this,
            mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER));
    }

    @Override
    public void onSensorChanged(SensorEvent event)
    {
        switch (event.sensor.getType())
        {
            case Sensor.TYPE_ACCELEROMETER:
                txtX.setText(Math.round(event.values[0] *10)/10.0 + ""); // 1 decimal
                txtY.setText(Math.round(event.values[1] *10)/10.0 + "");
                txtZ.setText(Math.round(event.values[2] *10)/10.0 + "");
                break;
        }
    }

    @Override
    public void onAccuracyChanged(Sensor sensor, int accuracy) {}
}

```

Contacts

CONNECT CONTACTS

PREV CONTACT

NEXT CONTACT

Jan Stern Johansen

17. Persistent Data

Persistent data handler om hvilke teknikker der findes, som giver mulighed for at gemme data for en aktivitet, på tværs af aktiviteter i en applikation, på tværs af gentagne udførelser af samme applikation, og på tværs af applikationer.

Android applikationer har ofte brug for at kunne gemme data på en persistent måde, sådan at data ikke forsvinder med nedlæggelse af objekter eller afslutning af applikationer.

Der kan være brug for at valg på brugerflader, såsom tryk på radiobuttons, checkbuttons og indtastninger huskes ved afslutning og genoptagelse af applikationen.

Der kan være brugerfladerændringer, der er sket programmatisk og som skal huskes ved drejning af skærm.

Der kan være resultater fra f.eks. spil der ønskes gemt.

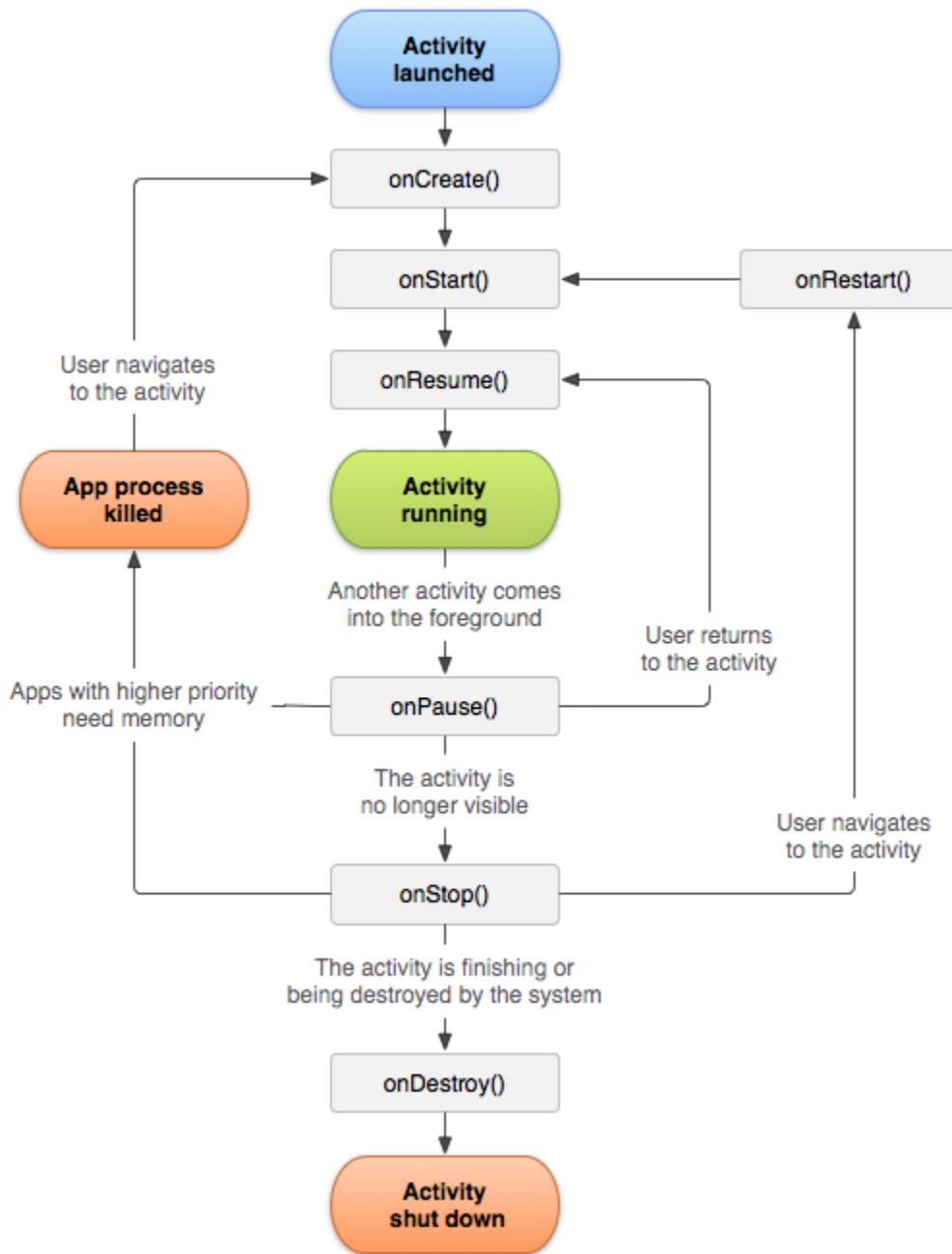
Der kan være ressourcer der er hentet på nettet eller oprettet under applikationens udførelse, der ønskes gemt til senere brug.

Der kan også være tekstfiler til visning i applikationens Hjælp, som skal være der hele tiden.

En aktivitets livscyklus

Som en begyndelse på emnet vil jeg beskæftige mig lidt med en aktivitets livscyklus og hvilke metoder der udføres under forskellige forhold.

Følgende er Androids officielle billede af dette.



Når skærmen vendes udføres følgende metoder i den angivne rækkefølge:

```
onPause()      onCreate()
onSaveInstanceState() onStart()
onStop()       onRestoreInstanceState()
onDestroy()    onResume()
```

Det ses at der er et par metoder mere end de viste i diagrammet.

Ved tryk på Back: Åbning af applikationen igen:

```
onPause()      onCreate()
onStop()       onStart()
onDestroy()    onResume()
```

Ved tryk på Home: Åbning af applikationen igen:

```
onPause()      onStart()
onSaveInstanceState() onResume()
onStop()
```

Metoderne `onCreate()`, `onSaveInstanceState()` og `onRestoreInstanceState()` er specielle ved at de modtager et objekt af typen `Bundle` som parameter.

Metoder der under alle omstændigheder udføres er `onPause()` og `onResume()`.

Godt nok bliver `onStart()` også udført i alle tilfælde, men det vil den nok ikke, hvis det handler om at der kommer en anden brugerflade ind over midlertidig.

De widgets på skærmen, der har en Id, vil beholde deres tilstand når skærmen vendes.

Men hvis applikationen afsluttes med Back, Home eller der foretages et telefonopkald, vil de miste deres tilstand når applikationen startes op igen.

Alle ændringer på skærmen, der er foretaget programmatisk, f.eks. at der er udskiftet et billede på en knap ved tryk på denne, vil under alle omstændigheder mistes.

Bundle

Hvis det kun handler om at bevare tilstanden når skærmen vendes, kan dette gøres ved at gemme og hente i Bundle-parameteren, som både gives ved `onCreate()`, `onSaveInstanceState()` og `onRestoreInstanceState()`, som alle kaldes når skærmen vendes. Her kan der gemmes key/value par.

Bundle er ikke persistent, men lever så længe applikationen kører.

Eksempel

Gem noget i `onSaveInstanceState(Bundle outState)` og hent det igen i `onCreate(Bundle savedInstanceState)` eller `onRestoreInstanceState(Bundle savedInstanceState)`

Bundle-objektet er null indtil der er puttet noget i det første gang.

Når skærmen vendes, starter Activity forfra, men inden da kaldes `onSaveInstanceState()`, hvor der kan gemmes i dens Bundle parameter og hentes igen i `onCreate()` Bundle-parameter. **Dette kan nok også gøres når app'en midlertidigt kommer ud af brugerfladen.**

Når app'en afsluttes og startes igen, vil data ikke overleve

```
public void onCreate(Bundle savedInstanceState) {  
    ...  
    if(savedInstanceState != null)  
    {  
        String str = savedInstanceState.getString("Navn", "No Name");  
        Toast.makeText(this, str, Toast.LENGTH_LONG).show();  
    }  
}  
  
protected void onSaveInstanceState(Bundle outState) {  
    ...  
    outState.putString("Navn", "Jan");  
}
```

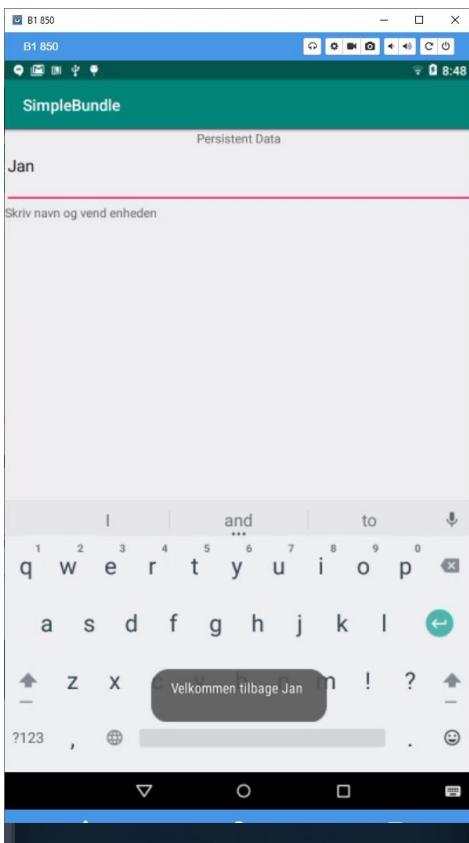
Pas på med `onSaveInstanceState()`, der findes også en med to parametre, som ikke duer her.

Hvis der ønskes bevaret tilstand under alle omstændigheder, skal metoderne `onPause()` og `onResume()` benyttes, da de altid kaldes. Og her må man så benytte andre måder end at skrive/læse Bundle-objektet, som ikke eksisterer her.

Eksempel

(D:\Data\Programmering\Java\MobilProg\Android\Persistens\PersistentData)

Det der er skrevet gemmes når enheden vendes og hentes derefter i den nystartede Activity, og skrives i en toast.



```
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center_horizontal"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Persistent Data"/>

    <EditText
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Skriv navn og vend enheden"
        />

</LinearLayout>

public class MainActivity extends AppCompatActivity
{
```

```

EditText txtName;

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

    txtName = findViewById(R.id.txtName);
    txtName.setText("");

    if(savedInstanceState != null) {
        String name = savedInstanceState.getString("Name", "No Name");
        Toast.makeText(this, "Velkommen tilbage " + name, Toast.LENGTH_LONG).show();
    }
    else
    {
        Toast.makeText(this, "Velkommen tilbage nobody!!", Toast.LENGTH_LONG).show();
    }
}

@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    String name = txtName.getText().toString();
    outState.putString("Name", name);
}
}

```

Der kan godt vises en Toast i onSaveInstanceState().

Teksten i txtName, som er skrevet i brugerfladen, kan ikke slettes i hverken onCreate() eller onSaveInstanceState(). Det kan den i en buttons click-handler.

Opgave

Skriv en app som tæller og udskriver hvor mange gange enheden vendes imellem portrait og landscape.

Persistence

5 metoder som nævnt i <http://developer.android.com/guide/topics/data/data-storage.html>

- Shared Preferences
- Internal Storage
- External Storage
- SQLite Database
- Network Connection

Shared Preferences

Kan gemme key/value par af primitiver som boolean, float, int, long og String.

Fordele:

- Indbygget og klar til at bruge på en nem måde.
- Hvilken som helst data kan gemmes ved at manipulere dem til strings.

Ulemper:

- Ikke til store data-mængder.
- Kun internal device memory.
- Manipulationer kræves ved non-primitive data.
- Farligt at offentliggøre data til andre applikationer.

Eksempel

Den aktuelle aktivitet som ønsker at gemme data henter et SharedPreferences-objekt med

```
SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);
```

Denne er privat. Kan tilgås i same applikation/package.

Eller

```
pref = getSharedPreferences(
    "MySharedFile", Context.MODE_WORLD_READABLE | Context.MODE_WORLD_WRITEABLE);
```

Denne er privat eller world og kan som world tilgås fra andre applikationer igennem opretterens context.

Eller

```
pref = getPreferences(Context.MODE_PRIVATE);
```

Denne kan kun tilgås i same aktivitet, som den er oprettet.

For at oprette key/value par hentes en editor. Med denne indsættes data ved kald af metoden put<type>() og der committes.

```
SharedPreferences.Editor editor = pref.edit();
editor.putString("EfterNavn", "Johansen");
editor.commit();
```

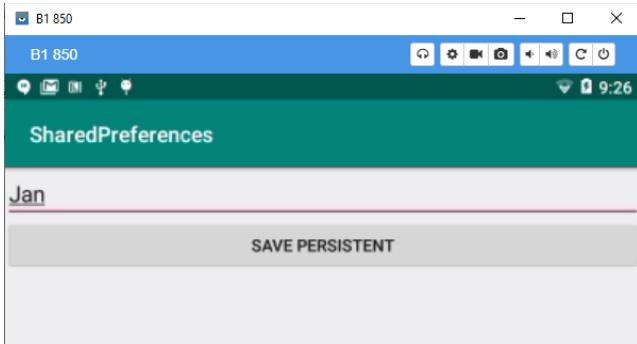
Data kan læses igen med

```
String str = pref.getString("EfterNavn", "No Efternavn");
```

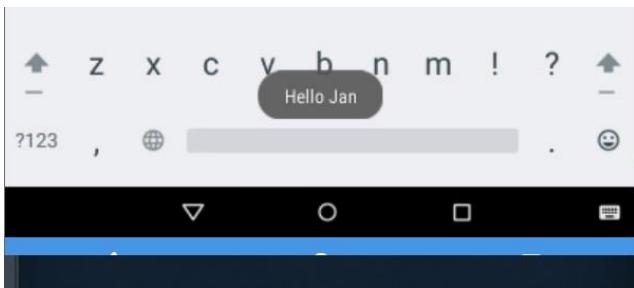
Eksempel

(D:\Data\Programmering\Java\MobilProg\Android\Persistens\SharedPreferences)

Med SharedPreferences kan app'en lukkes og startes igen, og de gemte data hentes.



Skriv navn og tryk Save. Luk app'en og start den igen.



```
<LinearLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtName" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save Persistent"
        android:id="@+id	btnSave" />

</LinearLayout>

public class MainActivity extends AppCompatActivity
{
    Button btnSave;
    EditText txtName;

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

        txtName = findViewById(R.id.txtName);
        btnSave = findViewById(R.id.btnSave);
        btnSave.setOnClickListener(new View.OnClickListener() {
```

```

@Override
public void onClick(View v)
{
    SharedPreferences pref =
        PreferenceManager.getDefaultSharedPreferences(MainActivity.this);

    SharedPreferences.Editor editor = pref.edit();
    editor.putString("Name", txtName.getText().toString());
    editor.commit();
}

SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(this);
String str = pref.getString("Name", "No Name");

Toast.makeText(this, "Hello " + str, Toast.LENGTH_LONG).show();
}
}

```

Når en anden applikation ønsker at læse preferences fra en applikation, må den gøre det igennem dennes context som kan findes ved package-navnet.

```

Context theContext = getApplicationContext().createPackageContext(
    "dk.tec.PrefDemo", Context.MODE_WORLD_WRITEABLE);
SharedPreferences pref = theContext.getSharedPreferences(
    "MyShared", Context.MODE_WORLD_READABLE);
String str = pref.getString("EfterNavn", "No Efternavn");

```

Et objekt gemt i en string i Shared Persistence

Da Shared Preferences kun kan gemme primitive data samt String, må et objekt konverteres til en String, når det skal gemmes.

Eksempel

Skriv et objekt til ud i Shared Persistence.

Der benyttes et objekt af klassen MyArea.

```

MyArea objArea = new MyArea(4, 6);

SharedPreferences pref = getPreferences(MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();

ByteArrayOutputStream byteArrayOut = new ByteArrayOutputStream();
ObjectOutputStream out = new ObjectOutputStream(byteArrayOut);
out.writeObject(objArea);

out.close();
byteArrayOut.close();

editor.putString("myObject", byteArrayOut.toString());
editor.commit();

```

Og hent objektet ind igen.

```

ByteArrayInputStream byteArrayIn =
        new ByteArrayInputStream(pref.getString("myObject", "").getBytes());

ObjectInputStream in = new ObjectInputStream(byteArrayIn);

objArea = (MyArea) in.readObject();

objArea.getSize(); //24 (4*6)

```

Internal Storage

Filer i Device Memory.

Der kan fra en tilsluttet PC kun ses det der ikke kræver root-rettigheder.

Fordele:

- Bredt anvendelig med muligheden for rettigheder.
- Kan bruges, kun af den aktuelle app, delt mellem flere, samt read-only og read-write.
- Cache kan bruges, hvor android fjerner det hvis der er behov.
- Der kan gemmes binary filer.

Ulemper:

- Begrænset plads.
- Ikke indbygget framework, så man skal selv gøre arbejdet med at tilrette datastrukturer.
- Ingen optimerede søgninger.

Eksempel

Skriv ud i en fil i internal storage.

```

FileOutputStream fos = openFileOutput("MyTextFile.txt", MODE_WORLD_READABLE);

OutputStreamWriter osw = new OutputStreamWriter(fos);
osw.write("Fra alle os til alle jer!");

osw.flush();
osw.close();

```

Læs ind fra filen i internal storage

```

FileInputStream fis = openFileInput("MyTextFile.txt");
InputStreamReader isr = new InputStreamReader(fis);

char[] inputBuffer = new char[10];
String s = "";

int charRead;

while((charRead = isr.read(inputBuffer)) > 0) //Læser det der kan være i inputBuffer
{
    String readString = String.valueOf(inputBuffer, 0, charRead);
    s += readString;
    inputBuffer = new char[10];
}

```

```
Toast.makeText(this, s, 100).show();
```

External Storage

Filer i extern memory

Fordele:

- Data både public delt imellem flere applikationer og data for en specific applikation.
- Der kan anvendes binære filer.
- Der kan sættes et stort flash memory i. Så hvis der er for lidt plads må der sættes et større kort i.
- Derfor bør applikationer bruge extern memory, fordi det ikke kan vides, hvor mange applikationer der er på telefonen.
- Der kan nemt foretages backup, da alt kan ses fra en tilsluttet PC.

Ulemper:

- Langsommere end internal storage. (forskellige hastigheder på kort)
- Intet indbygget framework, der gemmes bare i filer, så man må selv lave sin customisation.
- Ikke nogen effektiv søgning.
- Brugere har adgang til og kan modifcere data, så der skal laves fejl-check.

Eksempel

AndroidManifest.xml skal tilføjes følgende

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE">
</uses-permission>
```

Find det eksterne kort og opret en mappe til filen

```
File extCard = Environment.getExternalStorageDirectory();
File directory = new File(extCard.getAbsolutePath() + "/MyFilesDir");
directory.mkdirs();
File file = new File(directory, "MyTextFile.txt");
```

Det bliver stien /mnt/sdcard/MyFilesDir/MyTextFile.txt.

Skriv en tekst i filen.

```
FileOutputStream fos = new FileOutputStream(file);
OutputStreamWriter osw = new OutputStreamWriter(fos);
osw.write("Fra alle os til alle jer!");
osw.flush();
osw.close();
```

Læs teksten fra filen

```
FileInputStream fis = new FileInputStream (file);
InputStreamReader isr = new InputStreamReader(fis);

char[] inputBuffer = new char[10];
String s = "";
int charRead;
```

```

while((charRead = isr.read(inputBuffer)) > 0) //Læser det der kan være i inputBuffer
{
    String readString = String.valueOf(inputBuffer, 0, charRead);
    s += readString;
    inputBuffer = new char[10];
}
Toast.makeText(this, s, Toast.LENGTH_LONG).show();

```

18. SQLite Databases

Bruger internal memory til databasefil.

Fordele:

- Hurtig at sortere og hente.
- Nemt at gemme custom data.
- Constraints såsom FK og PK.

Ulemper:

- Begrænset internal memory.
- Sværere at sætte op. (Det var ikke så svært for mig og derfor fristende at bruge.
Der bruges kun små datamængder på Tetris i telefonen. Hastighed ikke kritisk, men det fylder jo i den interne lager. Dette gør så heller ikke noget i dette tilfælde.)
- Brugeren har ikke adgang til at ændre, hvis han ikke er root.
- Sårbar over for ændringer i datastrukturen.

Eksempel

Først oprettes en hjælpeklasse der arver fra SQLiteOpenHelper.

Her oprettes databasen, som er angivet i constructoren, hvis den ikke eksisterer, og onCreate() kaldes.

Hvis versionsnummeret, her 3, senere forhøjes, vil onUpgrade() kaldes.

```

class DBHelper extends SQLiteOpenHelper
{
    DBHelper(Context context)
    {
        super(context, "MyDB3", null, 3); //version 3, fejlede når jeg ændrede version.
    }

    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL("Create table Person(Id integer, Name text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)

```

```

    {
        db.execSQL("Drop table if exists Person");
    }
}

```

Nu kan databasen hentes (oprettes første gang).

```
SQLiteDatabase db = new DBHelper(this).getWritableDatabase();
```

Der indsættes en række, hvor SQL-insert kommandoen opbygges af angivne elementer.

```
ContentValues values = new ContentValues();
values.put("Id", 0);
values.put("Name", "Jan Stern Johansen");
db.insert("Person", null, values);
```

Og en række, der angives af udtrykkelig SQL-insert kommando.

```
db.execSQL("insert into Person values(1, 'Niels Peter Hansen')");
```

Inholdet læses og fyldes i en Cursor, som gennemløbes og indholdet udskrives.

```
Cursor nameCursor = db.query("Person", new String[]{"Id", "Name"},
                             null, null, null, null, null);
String strResult = "";
while(nameCursor.moveToNext())
{
    antal++;
    strResult += nameCursor.getInt(0) + " ";
    strResult += nameCursor.getString(1) + "\n";
}
Toast.makeText(this, strResult, Toast.LENGTH_LONG).show();
```

Toast:
0 Jan Stern Johansen
1 Niels Peter Hansen

Network Connection

Gemmer data via web-baserede services.

Fordele:

- Nem at dele med andre enheder og PC'er.
- Nem at foretage backup.

Ulemper:

- Ikke pålidelig.
- Svært at sætte op.
- Bruger trafik-ressourcer.
- Langsamt i forhold til lokal data.

Der kan gemmes data i online databaser, på webservices eller en hvilken som helst hjemmelavet forbindelse.

Her kommer Content Providers også ind i billedet.

Det kritiske er at man ikke kan regne med at forbindelsen altid er der. Så derfor må der for kritisk data være en mulighed for at gemme disse lokalt, når der ikke er mulighed for at gemme på nettet.

En applikation der svigter når netværket er nede, kan hurtig blive fravalgt.

Packages der bruges til netværk er java.net og android.net

Kommunikationen kan etableres med klassen ServerSocket (server) og klassen Socket (klient).

Android Backup Service

Android stiller en backup service til rådighed, sådan at hvis en applikation bliver afinstalleret og geninstalleres, kan dens persistente data hentes i skyen.

Titanium Backup

Er en applikation som kan gemme alt på telefonen/enheden i skyen og hente det igen når det behøves.

Andre emner i persistent data

Content Provider

En SQLite database er privat for applikationen, der har oprettet den. Hvis der ønskes adgang til data fra andre applikationer, kan dette ske igennem en Content Provider. Adgangen til en Content Provider sker igennem en URI(Unified Ressource Identifier). Selv om det er almindeligt at en Content Provider bruger en SQLite database som underliggende data, kan data tilgås på hvilken som helst måde, og dette er transparent for klienten.

Content Provider skal implementere visse metoder, alt efter hvad der er brug for, så som

`query()`, `insert()`, `update()`, `delete()`, `getType()` og `onCreate()`.

Metoden `query()` skal returnere en Cursor med data, og `insert()`, `update()`, `delete()` ændrer på data.

Hvis flere applikationer har adgang til Content Provider, kan det være nødvendigt at lave synkroniseret kode i denne.

Mange Android data-sources, f.eks. kontakter, giver mulighed for adgang igennem Content Provider.

Statiske filer i applikationen

Der kan i applikationens res/raw mappe lægges filer i programudviklingsfasen, som kan indlæses af programmet på udførselstidspunktet.

Filerne vil få en id i R-klassen, som gør det nemt at hente indholdet.

Eksempel

Her er der oprettet filen res/raw/mytext.txt, som indlæses og udskrives i en Toast.

```
InputStream inStream = this.getResources().openRawResource(R.raw.mytext);
BufferedReader bufReader = new BufferedReader(new InputStreamReader(inStream));

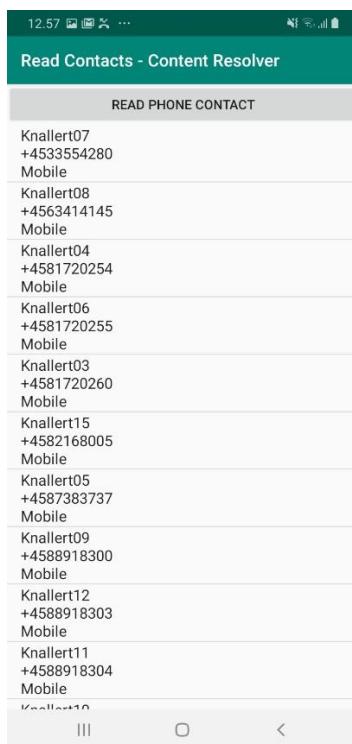
String s;
String str = "";

try {
    while((s = bufReader.readLine()) != null)
    {
        str += s;
    }
    inStream.close();
    bufReader.close();

} catch (IOException e) {
    e.printStackTrace();
}

Toast.makeText(this, str, Toast.LENGTH_LONG).show();
```

19. Kontakter og sikkerhed i mobiltelefon



Kontakterne i telefonen kan tilgås som en Content Provider via en Content Resolver.

De to linjer i metoden readPhoneContacts

```
Uri readContactsUri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
```

```
Cursor cursor = getContentResolver().query(readContactsUri, null, null, null, null);
```

åbner en cursor med indholdet fra kontakterne. Cursoren svarer til en datababel med rækker og kolonner.

Cursoren gennemløbes og der dannes en string af hver kontakts navn, nummer og telefontype. Disse indsættes i en arraylist som vises i brugerfladens ListView via en adapter.

Hvis Android versionen er nyere end 23, kan det kun lade sig gøre ved at lade brugeren tillade adgang til kontakter. (Så koden herunder vil altså kun virke på gamle versioner) Men det ordnes i næste afsnit.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="dk.tec.listcontacts">  
  
    <!-- Need below permissions.-->  
    <uses-permission android:name="android.permission.READ_CONTACTS" />  
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
```

```

<application>

activity_main.xml

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/read_phone_contacts_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read Phone Contact"/>

    <ListView
        android:id="@+id/lstContacts"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </ListView>
</LinearLayout>

public class MainActivity extends AppCompatActivity
{
    private List<String> phoneContactsList = new ArrayList<String>();
    private ArrayAdapter<String> contactsListDataAdapter;
    ListView contactsListView = null;

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

        setTitle("Read Contacts - Content Resolver");

        contactsListView = (ListView) findViewById(R.id.lstContacts);
        contactsListDataAdapter = new ArrayAdapter<String>(
           (getApplicationContext(),
            android.R.layout.simple_list_item_1,
            phoneContactsList));
        contactsListView.setAdapter(contactsListDataAdapter);

        Button readPhoneContactsButton =
                (Button) findViewById(R.id.btnReadContacts);
        readPhoneContactsButton.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                readPhoneContacts();
            }
        });
    }

    private void readPhoneContacts()

```

```

{
    phoneContactsList.clear();

    Uri readContactsUri = ContactsContract.CommonDataKinds.Phone.CONTENT_URI;

    Cursor cursor = getContentResolver().query(readContactsUri,
                                                null, null, null, null);

    if(cursor!=null)
    {
        while(cursor.moveToNext())
        {
            int displayNameIndex = cursor.getColumnIndex(
                ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME);
            String userDisplayName = cursor.getString(displayNameIndex);

            if(!userDisplayName.contains("Knallert"))
            {
                continue;
            }

            int phoneNumberIndex = cursor.getColumnIndex(
                ContactsContract.CommonDataKinds.Phone.NUMBER);
            String phoneNumber = cursor.getString(phoneNumberIndex);

            String phoneTypeStr = "Mobile";
            int phoneTypeColumnIndex = cursor.getColumnIndex(
                ContactsContract.CommonDataKinds.Phone.TYPE);

            int phoneTypeInt = cursor.getInt(phoneTypeColumnIndex);
            if(phoneTypeInt== ContactsContract.CommonDataKinds.Phone.TYPE_HOME)
            {
                phoneTypeStr = "Home";
            }
            else if(phoneTypeInt== ContactsContract.CommonDataKinds.Phone.TYPE_MOBILE)
            {
                phoneTypeStr = "Mobile";
            }
            else if(phoneTypeInt== ContactsContract.CommonDataKinds.Phone.TYPE_WORK)
            {
                phoneTypeStr = "Work";
            }

            StringBuffer contactStringBuf = new StringBuffer();
            contactStringBuf.append(userDisplayName);
            contactStringBuf.append("\r\n");
            contactStringBuf.append(phoneNumber);
            contactStringBuf.append("\r\n");
            contactStringBuf.append(phoneTypeStr);

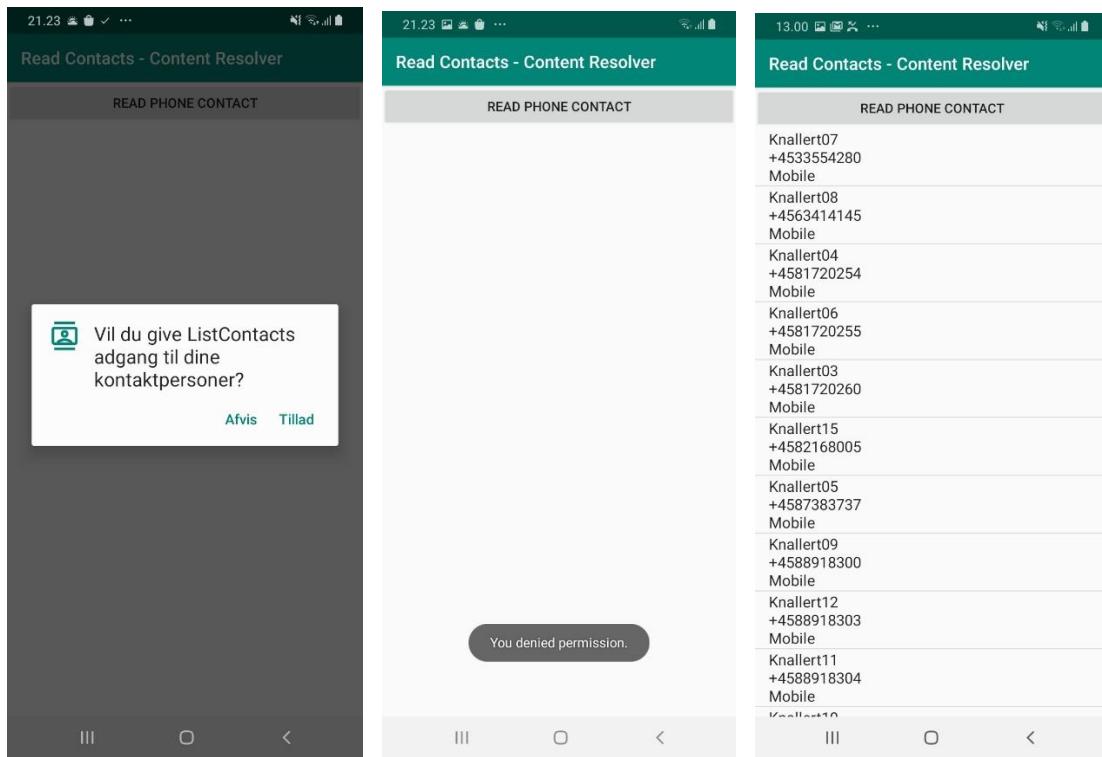
            phoneContactsList.add(contactStringBuf.toString());
        }
    }

    contactsListDataAdapter.notifyDataSetChanged();
}
}
}

```

Bede om brugerens tilladelse til at programmet tilgår kontakter

Efter Android ver 23 skal brugeren spørges om tilladelse til at tilgå kontakter



For at få en tilladelse fra brugeren til at tilgå kontakterne kaldes følgende metode

```
ActivityCompat.requestPermissions(this, requestPermissionArray, requestCode);
```

Kaldet lægges ind i en ny metode `requestPermission()` som kaldes i knappens `onClick()`. Så er det en lille smule nemmere at bruge den til flere anmodninger.

Når brugeren har gjort sit valg bliver der kaldt eventhandleren

```
onRequestPermissionsResult()
```

Her er der oplysning om hvilken tilladelse der er givet eller afvist.

Da både `requestPermissions()` og `onRequestPermissionsResult`, kan bruges til forskellige tilladelser, skal der angives en permission request kode, så man kan se hvilken request der har forårsaget kaldet af eventhandleren.

Der oprettes en konstant som angives som 2. parameter til metoden `requestPermission`.

```
private static final int PERMISSION_REQUEST_CODE_READ_CONTACTS = 1;
```

```
public void onClick(View view)
{
    requestPermission(
        Manifest.permission.READ_CONTACTS,
        PERMISSION_REQUEST_CODE_READ_CONTACTS);
}
```

```
}
```

Der kan også bedes om flere tilladelser, derfor er anden parameter i requestPermissions() et array.

```
private void requestPermission(String permission, int requestCode)
{
    String requestPermissionArray[] = {permission};
    ActivityCompat.requestPermissions(this, requestPermissionArray, requestCode);
}
```

Når der trykkes på knappen første gang efter appen er installeret og startet, vil brugeren spørges om tilladelse. Hvis brugeren afviser, spørges der igen næste gang der trykkes på knappen. Hvis brugeren giver tilladelse, spørges der ikke mere, heller ikke efter at programmet har været afsluttet og startet igen. Dette gælder altså også selv om programmet kalder requestPermissions hver gang der trykkes på knappen.

Efter kaldet af requestPermissions og brugeren har svaret enten Afvis eller Tillad, bliver onRequestPermissionsResult() kaldt. Her er der oplysning om hvilken requestCode anmodningen blev sendt med og om der blev givet Tilladelse eller Afvisning.

Da der kan bedes om flere tilladelser ad gangen, vil anden parameter permissions indeholde dem alle og tredje parameter indeholde alle tilhørende svar.

I onRequestPermissionsResult() kaldes readPhoneContacts(), hvis brugeren har givet tilladelse. Dette sker hver gang requestPermissions() kaldes efterfølgende. Så man kan bare blive ved på denne måde. Brugeren opdager det ikke når første tilladelsen er givet.

```
@Override
public void onRequestPermissionsResult(int requestCode,
    @NonNull String[] permissions, @NonNull int[] grantResults)
{
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if(grantResults.length > 0)
    {
        int grantResult = grantResults[0];

        if(grantResult == PackageManager.PERMISSION_GRANTED) {

            if (requestCode == PERMISSION_REQUEST_CODE_READ_CONTACTS) {

                readPhoneContacts();
            }
        }
        else
        {
            Toast.makeText(getApplicationContext(),
            "You denied permission.",
            Toast.LENGTH_LONG).show();
        }
    }
}
```

```
}
```

Spørge om der er givet tilladelse

Det ville måske være mere elegant at undersøge om der er givet tilladelse i stedet for at spørge om tilladelse hver gang, hvilket gives automatisk når først brugeren en gang har givet tilladelse.

Metoden ContextCompat.checkSelfPermission() returnerer et heltal, der fortæller om en given tilladelse er givet tidligere. Metoden lægges ind i en ny metode hasPhoneContactsPermission(), som kaldes i en if i onClick().

Først spørges om versionen er nyere en 23, ellers er der ingen grund til at spørge og der returneres true. Så spørges om den givne tilladelse er givet. I givet fald returneres true og ellers false.

Hvis tilladelsen er givet, kaldes readPhoneContacts() i onClick().

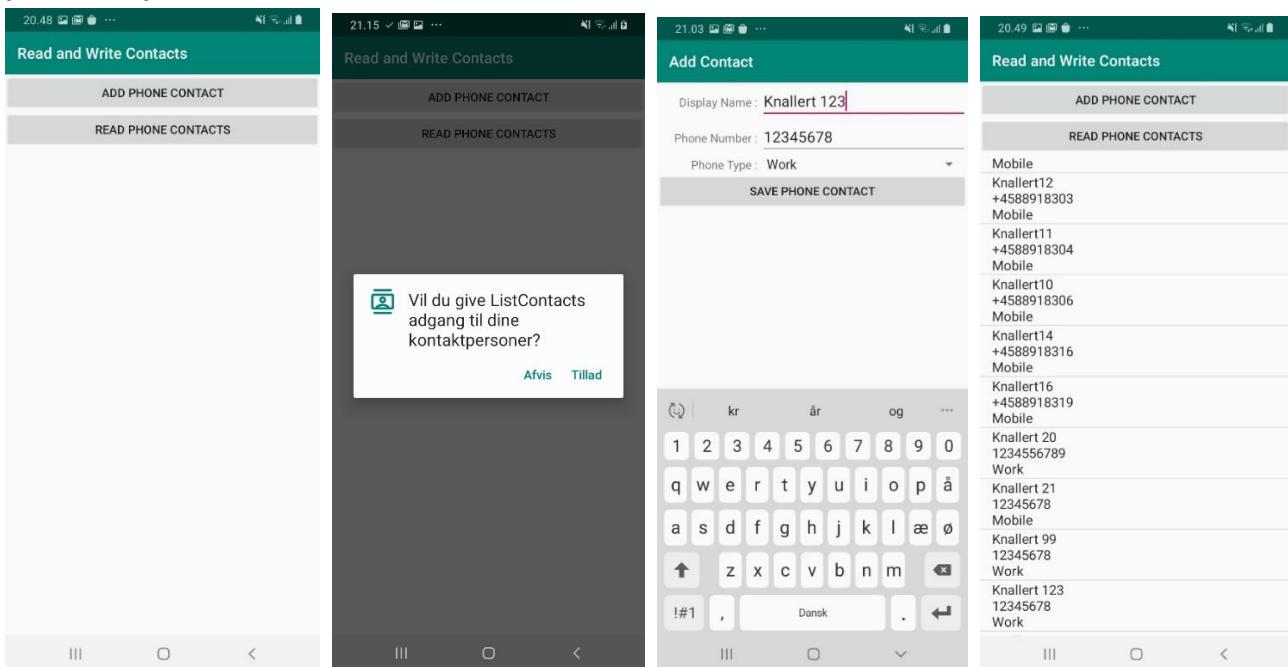
```
public void onClick(View view)
{
    if(!hasPhoneContactsPermission(Manifest.permission.READ_CONTACTS))
    {
        requestPermission(Manifest.permission.READ_CONTACTS,
                           PERMISSION_REQUEST_CODE_READ_CONTACTS);
    }else
    {
        readPhoneContacts();
    }
}

private boolean hasPhoneContactsPermission(String permission)
{
    boolean ret = false;

    if(Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) // 23
    {
        int hasPermission =
            ContextCompat.checkSelfPermission(getApplicationContext(), permission);

        if (hasPermission == PackageManager.PERMISSION_GRANTED)
        {
            ret = true;
        }
    }else
    {
        ret = true;
    }
    return ret;
}
```

Oprette en ny kontakt



```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/btnAddContact"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add Phone Contact"/>

    <Button
        android:id="@+id/btnReadContacts"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Read Phone Contacts"/>

    <ListView
        android:id="@+id/lstContacts"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </ListView>
</LinearLayout>

public class MainActivity extends AppCompatActivity
{
    private static final int PERMISSION_REQUEST_CODE_READ_CONTACTS = 1;
    private static final int PERMISSION_REQUEST_CODE_WRITE_CONTACTS = 2;

    . . .

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

        if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) != PackageManager.PERMISSION_GRANTED)
            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.READ_CONTACTS}, PERMISSION_REQUEST_CODE_READ_CONTACTS);
        else
            lstContacts.setAdapter(new ContactAdapter(this));
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)
    {
        if (requestCode == PERMISSION_REQUEST_CODE_READ_CONTACTS)
        {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
                lstContacts.setAdapter(new ContactAdapter(this));
            else
                Toast.makeText(this, "No permission to read contacts.", Toast.LENGTH_SHORT).show();
        }
        else if (requestCode == PERMISSION_REQUEST_CODE_WRITE_CONTACTS)
        {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED)
                addContact();
            else
                Toast.makeText(this, "No permission to write contacts.", Toast.LENGTH_SHORT).show();
        }
    }

    private void addContact()
    {
        Intent intent = new Intent(Intent.ACTION_INSERT_OR_EDIT);
        intent.setType(ContactsContract.Contacts.CONTENT_TYPE);
        intent.putExtra(ContactsContract.Intents.Insert.NAME, "Knallert 123");
        intent.putExtra(ContactsContract.Intents.Insert.PHONE, "12345678");
        intent.putExtra(ContactsContract.Intents.Insert.TITLE, "Work");
        startActivityForResult(intent, 1);
    }
}

```

```

{
    . . . .

    setTitle("Read and Write Contacts");

    final Button btnAddContact = (Button) findViewById(R.id.btnAddContact);
    btnAddContact.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View view)
        {
            if (!hasPhoneContactsPermission(Manifest.permission.WRITE_CONTACTS))
            {
                requestPermission(Manifest.permission.WRITE_CONTACTS,
                    PERMISSION_REQUEST_CODE_WRITE_CONTACTS);
            }
            else
            {
                Intent intent = new Intent(getApplicationContext(),
                    AddPhoneContactActivity.class);
                startActivity(intent);
            }
        }
    });
}

@Override
public void onRequestPermissionsResult(int requestCode,
    @NonNull String[] permissions,
    @NonNull int[] grantResults)
{
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);

    if (grantResults.length > 0)
    {
        int grantResult = grantResults[0];

        if (grantResult == PackageManager.PERMISSION_GRANTED) {

            if (requestCode == PERMISSION_REQUEST_CODE_READ_CONTACTS)
            {
                readPhoneContacts();
            }

            else if (requestCode == PERMISSION_REQUEST_CODE_WRITE_CONTACTS)
            {
                Intent intent = new Intent(getApplicationContext(),
                    AddPhoneContactActivity.class);
                startActivity(intent);
            }
        }
        else
        {
            Toast.makeText(getApplicationContext(),
                "You denied permission.",
                Toast.LENGTH_LONG).show();
        }
    }
}
}

```

```

activity_add_phone_contact.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="1dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:width="0dp"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Display Name : "
            android:gravity="right"/>

        <EditText
            android:id="@+id/txtDisplayName"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:width="0dp"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Phone Number : "
            android:gravity="right" />

        <EditText
            android:id="@+id/txtPhoneNumber"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:width="0dp"
            android:layout_weight="1"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Phone Type : "
            android:gravity="right" />

        <Spinner
            android:id="@+id/spnPhoneType"
            android:layout_weight="2"
            android:layout_width="0dp"
            android:layout_height="wrap_content" />

    </LinearLayout>

```

```

        android:spinnerMode="dropdown" />

    </LinearLayout>

    <Button
        android:id="@+id	btnAddContact"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save Phone Contact" />

</LinearLayout>

```

AddPhoneContactActivity.java

```

public class AddPhoneContactActivity extends AppCompatActivity
{
    private EditText txtDisplayName;
    private EditText txtPhoneNumber;
    private Spinner spnPhoneType;
    private Button btnAddContact;

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

        setTitle("Add Contact");

        txtDisplayName = (EditText) findViewById(R.id.txtDisplayName);
        txtPhoneNumber = (EditText) findViewById(R.id.txtPhoneNumber);

        spnPhoneType = (Spinner) findViewById(R.id.spnPhoneType);
        String phoneTypeArr[] = {"Mobile", "Home", "Work"};
        ArrayAdapter<String> phoneTypeSpinnerAdaptor =
            new ArrayAdapter<String>(this,
                android.R.layout.simple_spinner_item,
                phoneTypeArr);
        spnPhoneType.setAdapter(phoneTypeSpinnerAdaptor);

        btnAddContact = (Button) findViewById(R.id.btnAddContact);
        btnAddContact.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view)
            {
                Uri addContactsUri = ContactsContract.Data.CONTENT_URI;
                //content://com.android.contacts/data

                // Add an empty contact and get the generated id.
                long rowContactId = getRawContactId();

                // Add contact name data.
                String displayName = txtDisplayName.getText().toString();
                insertContactDisplayName(addContactsUri, rowContactId, displayName);

                // Add contact phone data.
                String phoneNumber = txtPhoneNumber.getText().toString();
                String phoneTypeStr = (String) spnPhoneType.getSelectedItem();
                insertContactPhoneNumber(addContactsUri,
                    rowContactId,
                    phoneNumber,

```

```

        phoneTypeStr);

Toast.makeText(getApplicationContext(),
    "New contact has been added, see it in contacts list.",
    Toast.LENGTH_LONG).show();

    finish();
}
});

}

// This method will only insert an empty data to RawContacts.CONTENT_URI
// The purpose is to get a system generated raw contact id.
private long getRawContactId()
{
    // Indsæt en tom kontakt
    ContentValues contentValues = new ContentValues();
    // contentValues er en tom HashMap

    // ContactsContract.RawContacts.CONTENT_URI:
    // content://com.android.contacts/raw_contacts
    Uri rawContactUri = getContentResolver().insert(
        ContactsContract.RawContacts.CONTENT_URI,
        contentValues);
    // rawContactUri: content://com.android.contacts/raw_contacts/206

    //String str = rawContactUri.getEncodedPath(); // = raw_contacts/206
    // Get the newly created contact raw id.
    long ret = ContentUris.parseId(rawContactUri); // 206
    return ret;
}

// Insert newly created contact display name.
private void insertContactDisplayName(Uri addContactsUri,
                                     long rawContactId,
                                     String displayName)
{
    ContentValues contentValues = new ContentValues();

    contentValues.put(ContactContract.Data.RAW_CONTACT_ID, rawContactId);

    // Each contact must has an mime type to avoid
    // java.lang.IllegalArgumentException: mimetype is required error.
    contentValues.put(
        ContactContract.Data.MIMETYPE,
        ContactContract.CommonDataKinds.StructuredName.CONTENT_ITEM_TYPE);

    // Put contact display name value.
    contentValues.put(
        ContactContract.CommonDataKinds.StructuredName.GIVEN_NAME,
        displayName);

    getContentResolver().insert(addContactsUri, contentValues);
}

private void insertContactPhoneNumber(Uri addContactsUri, long rawContactId,
                                    String phoneNumber, String phoneTypeStr)
{
    ContentValues contentValues = new ContentValues();

    contentValues.put(ContactContract.Data.RAW_CONTACT_ID, rawContactId);
    contentValues.put(ContactContract.Data.MIMETYPE,

```

```

        ContactsContract.CommonDataKinds.Phone.CONTENT_ITEM_TYPE);

// Put phone number value.
contentValues.put(ContactsContract.CommonDataKinds.Phone.NUMBER, phoneNumber);

// Calculate phone type by user selection.
int phoneContactType = ContactsContract.CommonDataKinds.Phone.TYPE_HOME;

if ("home".equalsIgnoreCase(phoneTypeStr))
{
    phoneContactType = ContactsContract.CommonDataKinds.Phone.TYPE_HOME;
} else if ("mobile".equalsIgnoreCase(phoneTypeStr))
{
    phoneContactType = ContactsContract.CommonDataKinds.Phone.TYPE_MOBILE;
} else if ("work".equalsIgnoreCase(phoneTypeStr))
{
    phoneContactType = ContactsContract.CommonDataKinds.Phone.TYPE_WORK;
}
// Put phone type value.
contentValues.put(ContactsContract.CommonDataKinds.Phone.TYPE, phoneContactType);

// Insert new contact data into phone contact list.
getContentResolver().insert(addContactsUri, contentValues);
}

```

20. Sensorer

Hvilke sensorer er der i enheden?

Layout sættes til at indeholde en ListView, som der skrives tekster ud i for hver sensor, der er på enheden.

```
<ListView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/lstSensors" />

public class MainActivity extends AppCompatActivity
{
    private SensorManager mSensorManager;
    ArrayAdapter<String> adapter;
    ListView lstSensors;

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

        lstSensors = (ListView) findViewById(R.id.lstSensors);
        ArrayList<String> list = new ArrayList<String>();

        adapter = new ArrayAdapter<String>(this,
                android.R.layout.simple_spinner_item,
                list);
        lstSensors.setAdapter(adapter);

        mSensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
        List<Sensor> deviceSensors = mSensorManager.getSensorList(Sensor.TYPE_ALL);

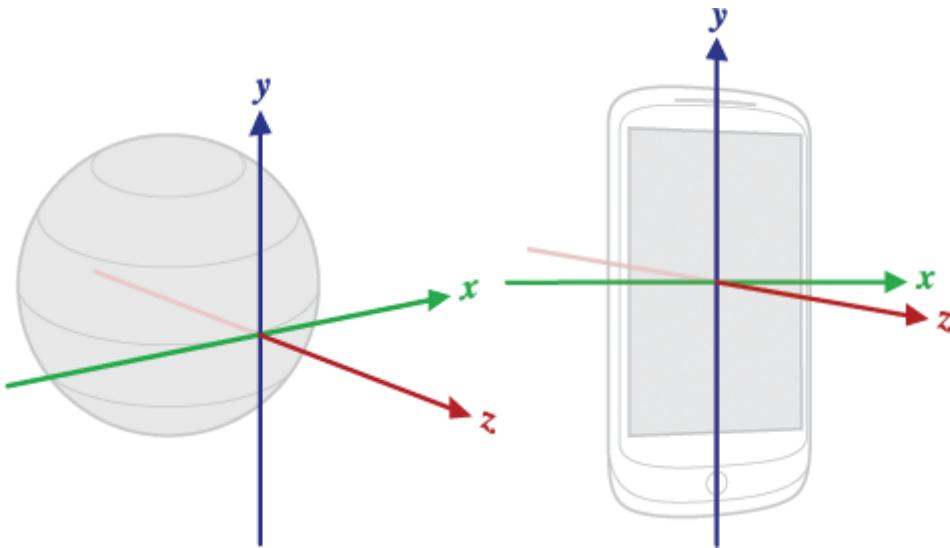
        for(Sensor s : deviceSensors)
        {
            String str = s.toString();
            int index = str.indexOf("maxRange");
            list.add(str.substring(0, index));
            list.add(" " + str.substring(index));
            adapter.notifyDataSetChanged();
        }
    }
}
```

```

Hello World!
{Sensor name="K2HH Acceleration ", vendor="STM", version=1, type=1,
 maxRange=39.2266, resolution=0.0011971008, power=0.13, minDelay=10000}
{Sensor name="CM3323E Light Sensor", vendor="Capella Microsystems, Inc.", version=1, type=5,
 maxRange=60000.0, resolution=1.0, power=0.75, minDelay=10000}
{Sensor name="SX9306 Grip Sensor", vendor="SEMTECH", version=1, type=65560,
 maxRange=5.0, resolution=5.0, power=0.75, minDelay=0}
{Sensor name="SX9306 Grip Sensor for wifi", vendor="SEMTECH", version=1, type=65575,
 maxRange=5.0, resolution=5.0, power=0.75, minDelay=0}
{Sensor name="Screen Orientation Sensor", vendor="Samsung Electronics", version=3, type=65558,
 maxRange=255.0, resolution=255.0, power=0.13, minDelay=0}

```

Udskriv de tre Accelerometer-værdier



```

<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"
    android:gravity="center_horizontal"
    tools:context="dk.tec.jaj.sensorslope.MainActivity" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="Sensors" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=" XY angle: "
            android:textAppearance="?android:attr/textAppearanceLarge" />
        <TextView

```

```

        android:id="@+id/txtXY_angle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="0"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=" XZ angle: "
            android:textAppearance="?android:attr/textAppearanceLarge" />
        <TextView
            android:id="@+id/txtXZ_angle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Large Text"
            android:textAppearance="?android:attr/textAppearanceLarge" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text=" ZY angle: "
            android:textAppearance="?android:attr/textAppearanceLarge" />
        <TextView
            android:id="@+id/txtZY_angle"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Large Text"
            android:textAppearance="?android:attr/textAppearanceLarge" />
    </LinearLayout>
</LinearLayout>

```

```

public class MainActivity extends Activity implements SensorEventListener {

    private SensorManager mSensorManager;
    private float xy_angle;
    private float xz_angle;
    private float zy_angle;

    TextView txtXY, txtXZ, txtZY;

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

        txtXY = (TextView) findViewById(R.id.txtXY_angle);
        txtXZ= (TextView) findViewById(R.id.txtXZ_angle);
        txtZY = (TextView) findViewById(R.id.txtZY_angle);

        mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    }

```

```

@Override
public void onSensorChanged(SensorEvent event) {
    synchronized (this) {
        switch (event.sensor.getType()) {
            case Sensor.TYPE_ACCELEROMETER:
                txtXY.setText(event.values[0] + " ");
                txtXZ.setText(event.values[1] + " ");
                txtZY.setText(event.values[2] + " ");
                break;
        }
    }
}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {}

protected void onResume()
{
    super.onResume();

    mSensorManager.registerListener(
        this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER),
        300000);
}

@Override
protected void onDestroy()
{
    super.onDestroy();
    mSensorManager.unregisterListener(
        this,
        mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER));
}
}

```

Sensors

XY angle: -10.199299

XZ angle: 0.65242

ZY angle: 1.0833763

Stående landscape. Tilsvarende på hovedet giver modsat fortegn på øverste.

Sensors

XY angle: -0.021547815

XZ angle: 10.061633

ZY angle: 0.44651863

Stående portrait. Tilsvarende på hovedet giver modsat fortegn på midterste.

Sensors

XY angle: -0.06584055

XZ angle: 0.15083471

ZY angle: 9.654618

Liggende på ryggen. Med skærmen nedad giver modsat fortegn på nederste.

Sensors

XY angle: -5.8598084

XZ angle: -5.5725045

ZY angle: 5.649119

Stående skråt på det ene hjørne. Der er 8 forskellige fortegn alt efter hvilket hjørne, der er nedad samt hældning frem eller bagover

Tråde

21. Netværk

Der må ikke udføres netværksoperationer i main-tråden i et Android program.

Dette kan løses på følgende måde.

```
Thread thread = new Thread(new Runnable()
{
    @Override
    public void run()
    {
        try {
            //Netværkstilgang

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
});
thread.start();
```

Java

Access Levels				
Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
private	Y	N	N	N

Android Studio og Java

I Android Studio under menu Files vælg Settings.

Under Build - Build Tools - Gradle -> Gradle JDK: Her ses hvilken Java version der bruges. Det lader til Android selv har installeret en Java 11 under Android mapperne.

Det hele ligger i C:\Program Files\Android\Android Studio1\jre\bin

Det ses også at den kan se JAVA_HOME som er sat i miljøvariablerne, som henviser til en særskilt installation af Java.

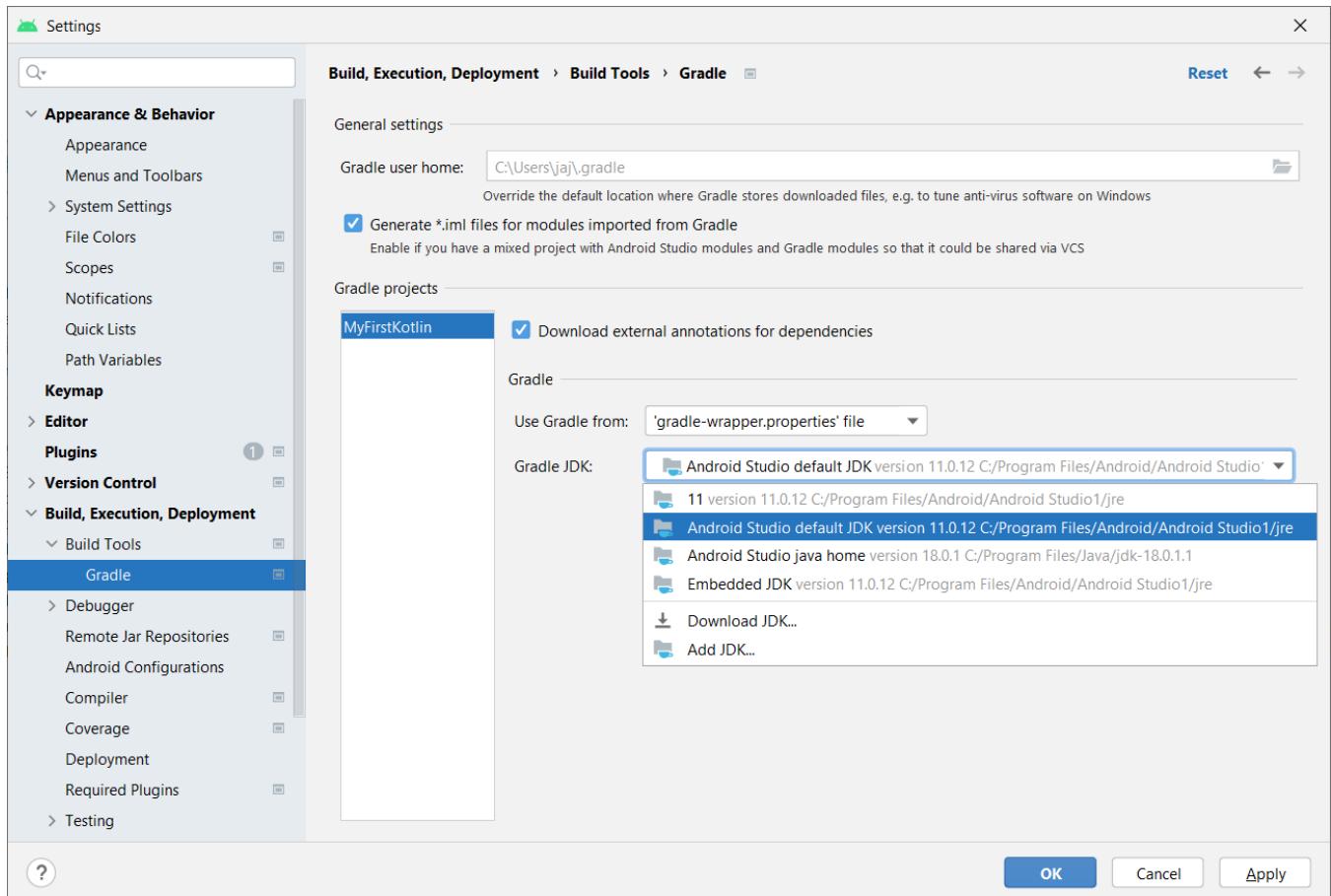
I skrivende stund har jeg ikke en Path indgang til min særskilte installation af Java

Når jeg i en CMD skriver >java -version fås 18.0.1.1

Det viser sig at der er en indgang i path: Path=C:\Program Files\Common Files\Oracle\Java\javapath;

Her ligger java.exe og javac.exe plus to mere.

Mappen javapath har i stifinderen en blå pil, hvilket vel betyder at det er en genvej, men egenskaber viser ikke til hvad.



Noter

RadioGroup

Kan have android:orientation="horizontal"

Følgende sættes på hver radiobutton for at de får lige meget plads.

Android:layoutweight=1

EditText

Android:gravity sætter om teksten skal skrives i højre side, når det er tal er skal indtastes.

Java versus C#

- Java er ældre end C# og da de to sprog minder meget om hinanden, må man jo have efterlignet Java da man udviklede C#.
- Java har package i modsætning til C# namespace. Package er mere fysisk, idet det angiver et filmappe hierarki.
- Java simple datatyper, som f.eks. int, decimal, boolean og char, er virkelig simple. Det er ikke som i C# structs, men bare værdier. Der er ikke andre ressourcer, som f.eks. metoderne parse() og toString(). Derfor findes der wrapper-klasser, så som Integer, både med metoder i og muligheden for at oprette et objekt af reference-type med en værdi. Simple datatyper er altså value-typer og klasser er reference-typer.
- Java kan ikke override operatorer, hvilket er muligt i C#. Dog er operatoren (+) overrided, sådan at den både kan bruges til at lægge to tal sammen og til at sammensætte to strings.
I C# er (==) overrided i klassen String, sådan at hvis man sammenligner to strings, som jo er referencetyper, undersøges der ikke om det er samme reference, men om de har samme indhold.
I Java vil (==) på strings undersøge om de to objekter har samme reference, altså om det er det samme objekt referencer peger på. I stedet må man bruge metoden str1.equals(str2), lige som på alle andre objekter.
- Java har ikke, som C#, der både har klassen String og keyword string, der er en alias til klassen String, keyword string men kun klassen String. Derfor skal det altid skrives med stort S.
Det samme gælder Object, som også i C# har keywordet object.
- Java fik først i version 1.5/Java 5, enum. Til gengæld er enum i Java mere avanceret end enum er i C#. I Java er enum en klasse, som kan have andre attributter end enum-værdien. Når der refereres til en enum-definition, er det et objekt af enum-klassen. Enum-klassen indeholder også metoder.
- Java har ikke, i modsætning til C#, delegates. Når der skal oprettes eventhåndlere i Java, tilføjes et objekt af en klasse der implementerer et krævet interface, og derfor har den eller de ønskede metoder, der skal kaldes ved bestemte events.

- En klasse arver en anden klasse ved at angive keyword extends, og implementerer et interface ved at angive keyword implements.
- Foreach-løkke angives i Java som: for(type element : samling-af-elementer-af-type).
- Java har mulighed for at oprette indre klasser og anonyme indre klasser. C# har også fået anonyme klasser, men de er meget primitive, kun med attributter med værdier.

Flytte et projekt i Android Studio

Jeg oplevede efter at have kopieret et projekt et andet sted hen at jeg ændrede i de originale filer og udførte de lokale, som ikke var ændret.

Søgning i Stackoverflow:

Copy the project directory from source to destination machine. Then follow the steps.

1. Open Android Studio
2. Go to File -> Open
3. Browse to the project location.
4. Select build.gradle and open.

Android Studio will automatically make the necessary changes for the target machine and set up the project.

NOTE : It usually confuses the first timers, as after copying project files into target machine, when you try to open the project, it considers it as a normal directory, not as android project.

4

I moved one project from PC to another with the same version of Android Studio (now for me is 2.3.3) and I have failed to launch in emulator because the path of build test was not found. However the solution is simply:

Build->Clean and Rebuild.

I finally find the solution. this is simple:

- 1- Go to your project folder, then open gradle>wrapper>gradle-wrapper.properties with note pad.
- 2- In last line, change your distribution URL to
distributionUrl=https\://services.gradle.org/distributions/gradle-2.10-all.zip(or each version of gradle you have.)
- 3- then go to Android studio and in project build.gradle and change your class path to latest version of gradle you have. classpath 'com.android.tools.build:gradle:2.1.2'
- 4- Go to app gradle.build and set buildToolsVersion "23.0.3"

Noter

Kigge på Regner med spinner og menu med array i strings.xml, custom_item.xml til ArrayAdapter, kan der bruges noget med flere widgets?

EditText med forslag til valg, hvilket gør det det bliver en dropdown. En elev vist Daniel brugte det i personClientApi.

To fingre gør den grafiske cirkel mindre og større.

Dialogs

Toast

Menu

Views

Fejl

Fejl

Jeg lavede et nyt Android projekt, med Basic Activity, som fejlede med duplicate class.

Løsning: Tilføj nedenstående i Build.gradle modul

```
// Tilføjet i dependencies fordi der kom fejlen:  
// Problem duplicate class androidx.lifecycle.viewmodel found in modules  
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.4.0'
```

Fejl

I made the mistake of upgrading eclipse today and now can't get my a new Android project to get going.

I get the message Proguard.cfg (the file can't be found).

Where is this thing I can't seem to find it? Is it possible to get rid of it I don't need Obfuscation in this project...

5 Answers

If you really don't need Proguard to obfuscate your release builds, you can remove the following line from the default.properties file in your project root folder:

proguard.config=proguard.cfg

If you want a proguard.cfg template, you can create a new Android project from scratch with Eclipse's project wizard, then copy proguard.cfg from the new project over to the previous one.

- 3 If you are working from the command line, rather than Eclipse, run `android update project` on the project to create the `proguard.cfg` file. – [CommonsWare Dec 14 '10 at 23:02](#)

Thanks Julio; It never created the default properties file either, so I switched back to version 1.5 the proguard file appeared. Looks like some kind of bug version 2.3 upgrade...I will try as u suggest. –

Resource is out of sync with the filesystem

I develop with Eclipse (and Java) and regularly get the message "resource is out of sync with the filesystem". right-click | "Refresh" will always clear this.

Why can Eclipse not Refresh automatically when it finds this condition? Are there cases where you want the resource to be out of sync?.

If there are such conditions and they don't apply to my work, is there a way of getting Eclipse to Refresh automatically when it encounters this state?. (I appreciate that it should refresh as little as it needs to in normal development to increase performance for human developers.)

Answer

You can enable this in Preferences - General - Workspace - Refresh Automatically

The only reason I can think why this isn't enabled by default is performance related.

For example, refreshing source folders automatically might trigger a build of the workspace. Perhaps some people want more control over this.

There is also an [article on the Eclipse site](#) regarding auto refresh.

Basically, there is no external trigger that notifies Eclipse of files changed outside the workspace. Rather a background thread is used by Eclipse to monitor file changes that can possibly lead to performance issues with large workspaces.

Android sdk main.out.xml parsing error

21 down vote

It is not your fault. When you try to compile a android project from its **xml resource** , eclipse considers it as executing xml file and so you get some output from it, in the same name of the xml with **.out.xml** extension.

To avoid this, you have to **right click the project** and select **run as -> Android Application** for the first time and then follow the below steps,

Window -> Preferences -> Run/Debug -> Launching -> Launch Operation -> Always Launch the previously launched application

Once you provide this in the **window->preferences**, from the next time when you press "**run**" or "**ctrl+f11**" from even a xml file, your project will get executed.

EDIT: You must also delete *res/layout/*.out.xml*, the file that Eclipse created and you do not need it.

R cannot be resolved

Den foreslog selv at upgrade Project/Gradle Plugin

