Entreprenöriell programmering

TDP028

LINKÖPINGS UNIVERSITET

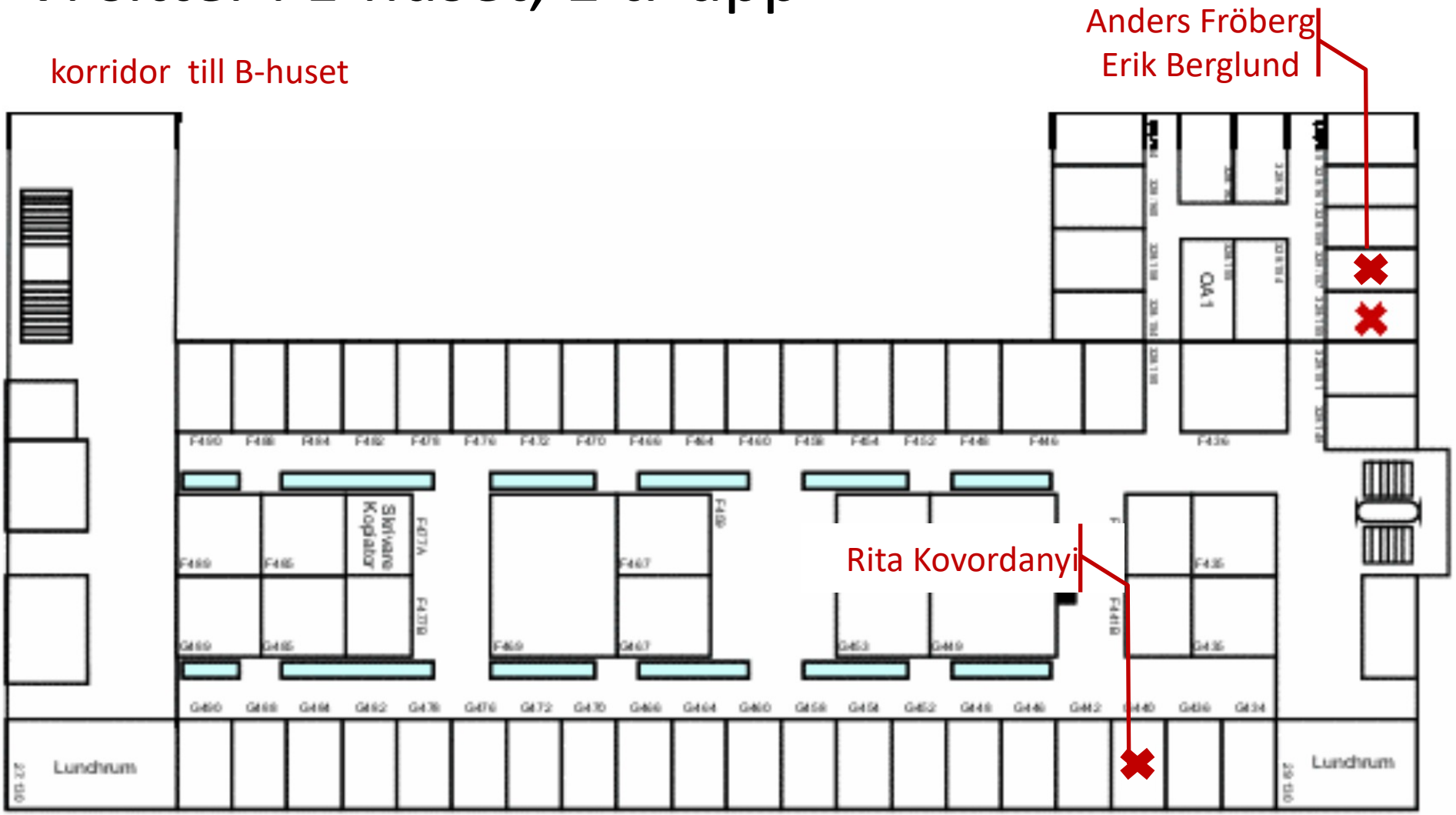
Översikt

- Kursupplägg
  - Projekt
  - Examination
- Öppna upp sin telefon
- Android introduktion

Kurspersonal

- **Rita Kovordanyi**, kursansvarig, examinator – Android
- **Erik Berglund** – entreprenöriella aspekter
- **Anders Fröberg** – Android, systemadministration, tekniska frågor
- **Liam Andersson** – handledare (liaan514@student.liu.se)
- **Madeleine Häger Dahlqvist**– kursadministratör

Vi sitter i E-huset, 1 tr upp



Bygga en häftig Android app

Vi kör Java för Android

https://codinginflow.com/java-or-kotlin-android-beginner

Kursmål

- Eget större projekt
  - Definiera och planera egen produkt
  - Använda ramverk och tekniker relevanta för entreprenöriella projekt
- Bli medveten om hur positionering och marknadsföring påverkar produktutveckling
- Utnyttja vetenskapliga rön i produktutveckling
- Öva på muntlig presentation - screencasts

Projektet

- Genomförs individuellt
  - Stöd i föreläsningar, bl.a.
    - Android grund och ‘best practices’
    - Konkurrensanalys
    - Litteratursökning
  - Seminarier
  - Resurstillfällen / handledning

Kurslitteratur

On-line tutorials, best practices, dokumentation, etc.

Bra startpunkt:

https://developer.android.com/training/basics/firstapp

Ändringar i kursen vi införde förra året

Åtgärd 1 & 2

Ni ville ha mer stöd för projektplanering

1. Vi kommer att ha milstolpar som stöd på vägen
   – Frivilligt att klara milstolparna, men poänggivande i projektet
2. Föreläsningarna kommer nu inte för fort, utan i takt med milstolparna

Åtgärd 3

Ensamt att arbeta på egen hand med projektet

3. Nu kommer ni kunna sitta tillsammans i samma lokal och koda

- Dessutom finns handledningstillfällen med lärare

Åtgärd 4

4. Vissa av poänggivande projektkraven är populära, andra väljs mer sällan
   - Vi har möblerat om bland kraven
   - Bättre beskrivning av kraven, lättare att förstå vad som ska göras
   - Större urval, samt större frihet att välja

Deluppgifter i projektet

Huvudsakliga uppgifter

- Definiera och bygg en större Android-app på 10 hp
- Versionshantera och skriv relevanta commit-kommentarer minst en gång i veckan under hela kursens gång
- Välj betygsnivå, och samla projektpoäng

Mindre uppgifter

- App-beskrivning
- Konkurrensanalys-övning
- Freemium-seminarium
- Tekniskt PM
- Två screen-casts i slutet av kursen

App-beskrivning

- Kort intro till appen, sammanhanget
- Tänkt användargrupp
- Syfte: Vad appen är tänkt att göra
- Kort beskrivning av användarens tänkta väg genom appen
  - Olika sätt att ta sig runt i appen

’Lean’ (bantad) utveckling

- Build-measure-learn feedback loop
  - Snabbt få ut en minimal app för att se kundernas reaktion → förbättra
- Minimum viable product (MVP)
  - T.ex. Facebook startade som studentkatalog
- Läs vidare
  - https://myva360.com/blog/examples-of-minimum-viable-products
  - http://theleanstartup.com/principles

Konkurrensanalys-övning

- Välj existerande app
- Analysera
  - Vad **måste** en *ny* app ha för features, för att kunna konkurrera?
  - Vad **kan** en *ny* app ha för att ”sticka ut” / profilera sig mot den existerande appen?
- En bra ny app skulle då ha minsta + det där extra
  - Underlag för detaljerad planering av egna appen

Freemium-seminarium

- Läs och diskutera artiklar:
  - Monetizing an infinite runner
  - Why do players buy in-game content? An empirical study on concrete purchase motivations
  - A Study of Crucial Factors for In-App Purchase of Game Software

Tekniskt PM

- Litteratursökning och referenshantering
  - T.ex. hur man kan använda analytics i lean utveckling
  - T.ex. hur modularisera kod i Android
    - Model View ViewModel (MVVM)?
  - Andra aspekter av kodkvalitet
    - Underhållbarhet?

Slutpresentation: screencasts

- Ca. 5 min långa
  - Användningsdemo
  - Kodgenomgång
    - Tekniska lösningar du vill lyfta fram
    - T.ex. de betygsgrundande features, API:er du har implementerat

Examination

Grundkrav

- Commit:a kontinuerligt under projektet, minst en commit i veckan
  - Commit messages som hjälper oss förstå vad som gjordes
- Kan commit:a dokument, ritade skisser och annat i repot (inte bara kod)

Samla poäng

- För olika betyg krävs olika antal samlade poäng (se mer info på hemsidan)
- Du väljer att jobba mot en viss betygsnivå
- Poängkategorier
  - Projektstyrning
  - Entrepenöriella aspekter
  - Teknisk kvalitet

Projektstyrning

Klara av milstolpar för att få projektpoäng

- **Fredag denna vecka**
  - Appbeskrivning och konkurrensanalys inlämnade i repo. Tagga med **Milstolpe-1**

    > git add .

    > git commit –m ”….” → *ger hash a029ac*

    > git push origin main

    > git tag –a Milstolpe-1 *a029ac* –m “Klar med …”

    > git push origin main --tag

Blir datumstämplat

Sex milstolpar kan ge 3 projektpoäng

En milstolpe varje vecka

1. Appbeskrivning och konkurrensanalys (0.5 p)
2. Enkel app med textruta och knapp (0.5 p)
3. Två skärmar med navigering emellan (0.5 p)
4. Visa faktiskt innehåll på skärmarna, dvs. enkel backend inkopplad (0.5 p)
5. Veckoplanering av resterande projektarbete (0.5 p)
6. Halvtids-screencast (~2 min): dema prototyp (0.5 p)

Halvtids-screencast

- Ca. 2 min lång
  - Användningsdemo
  - Visa funktioner
  - Hur man tar sig runt i appen

Deadline för slutinlämning

Början av januari (datum och tid enligt TimeEdit)

I readme i repo

- En kort app-beskrivning
- Viktigt att ange betygsambition
  - Vilken betygsnivå du har siktat på
  - Poäng för milstolpar du har samlat
  - Lista av poänggivande API:er i appen, med kort beskrivning av varje
  - Lista av poänggivande tekniska features, med kort beskrivning

Praktiska detaljer

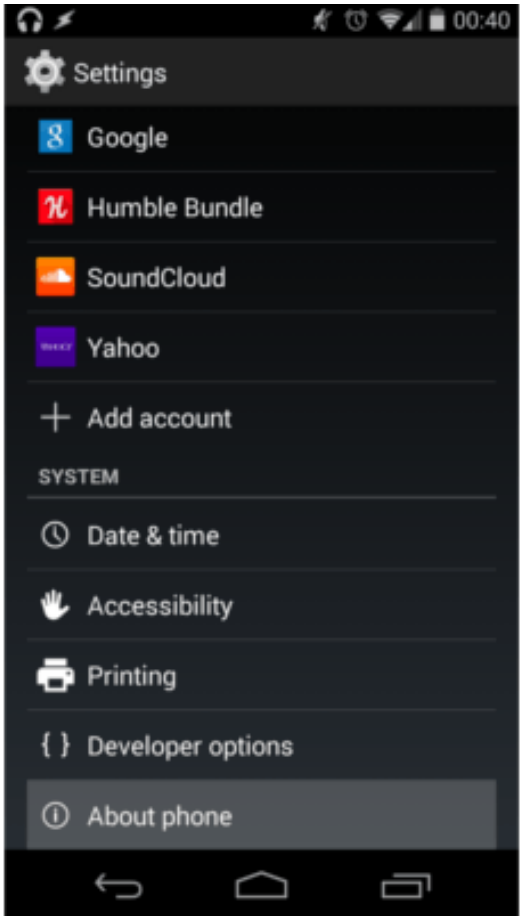
Installera miljön Android Studio

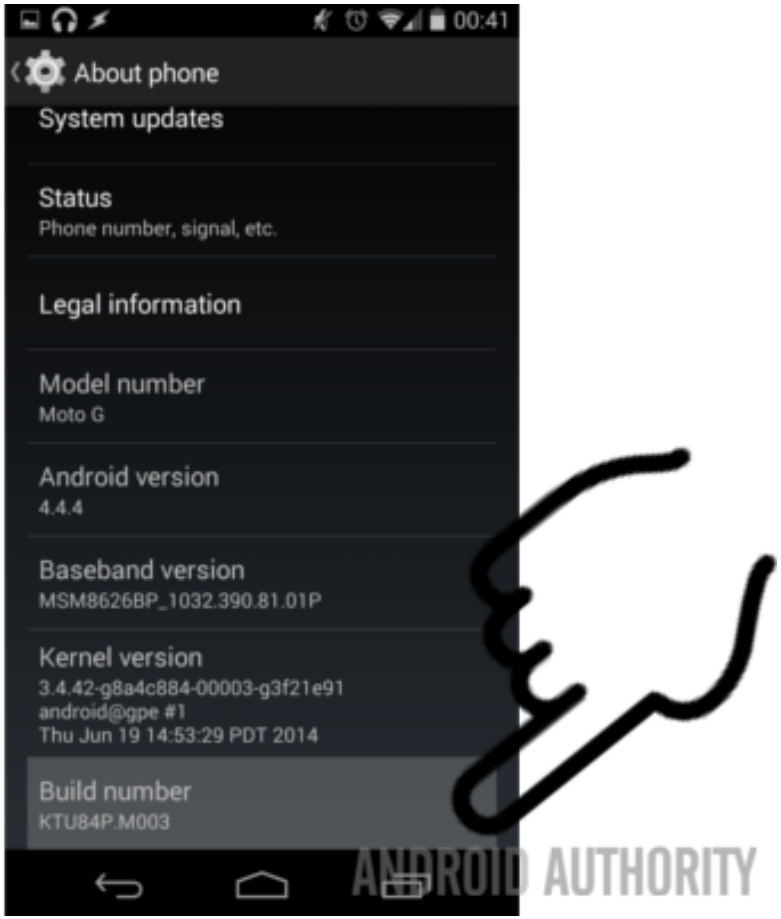
- Guide hur man installerar Android Studio: https://developer.android.com/studio/install.html

Hur köra appen?

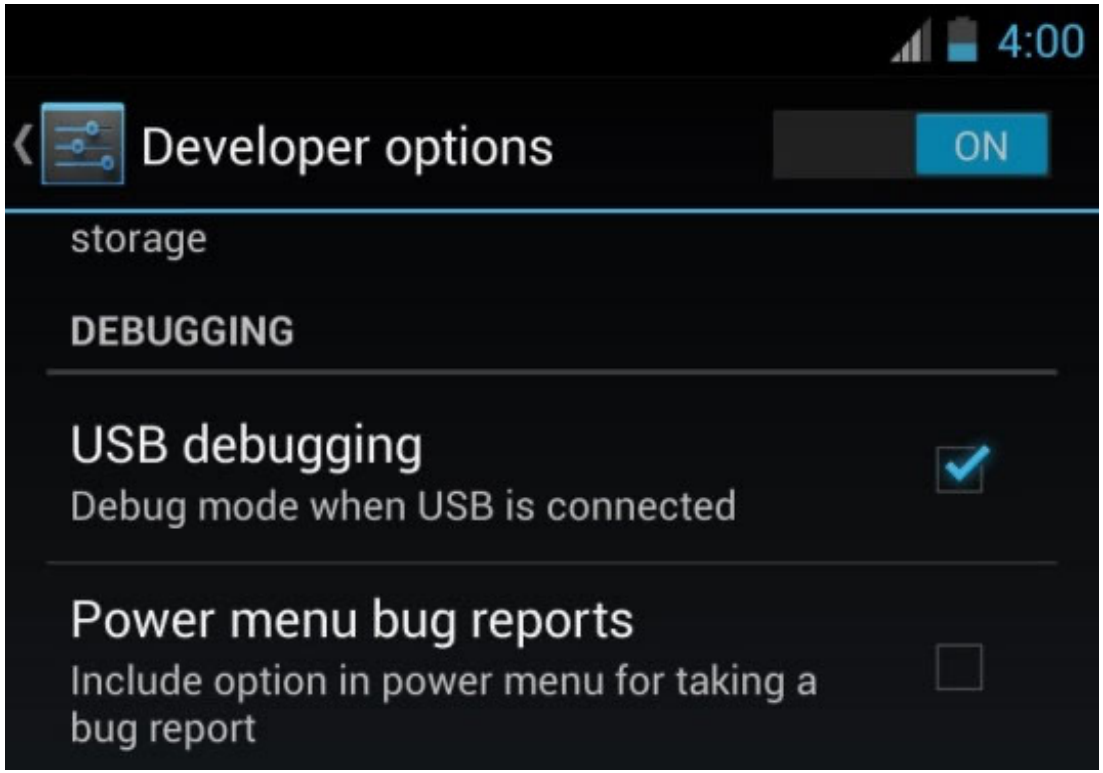
- Kör på egen telefon
- Eller emulator i Android Studio
  - Viss funktionalitet blir begränsad (t.ex. GPS)

Egen mobil eller platta





Felsökning via USB-koppling



Emulatorn

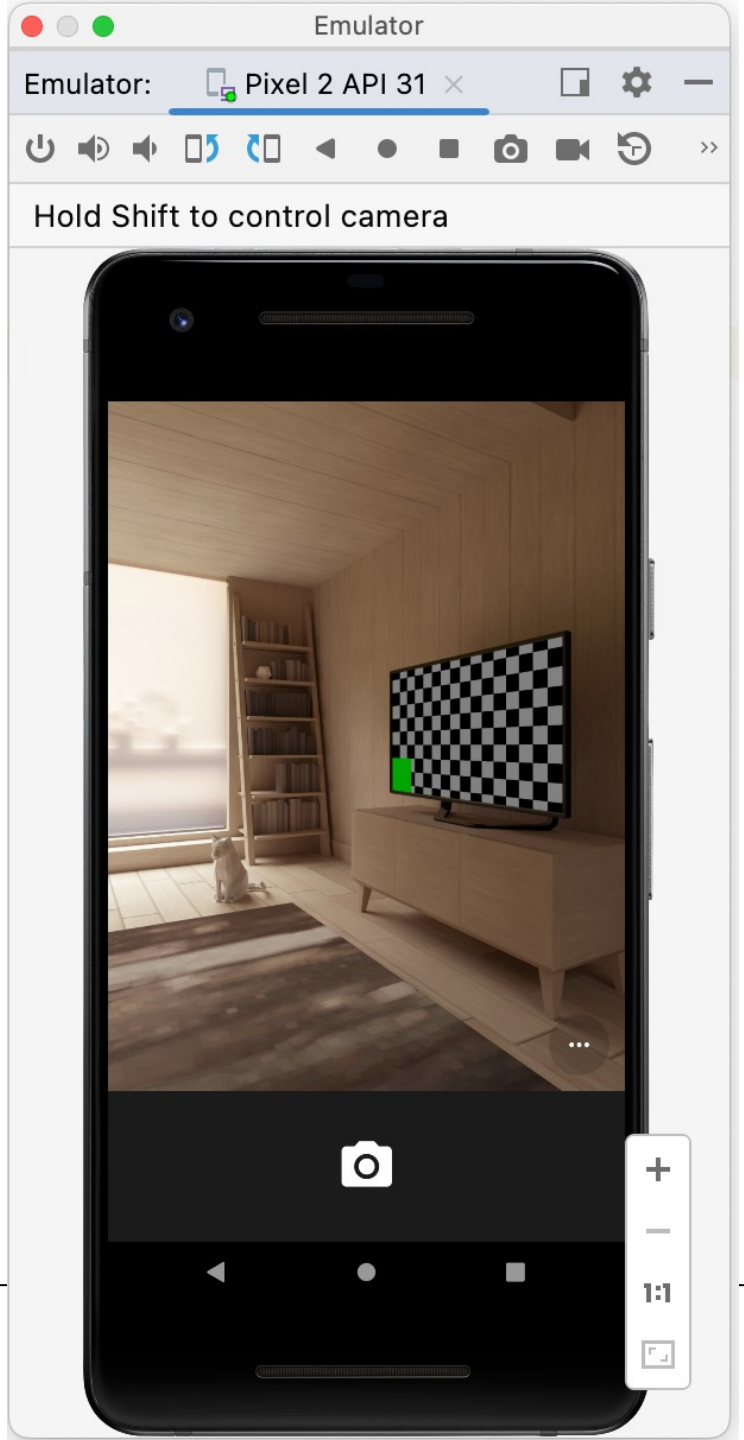
kontroller

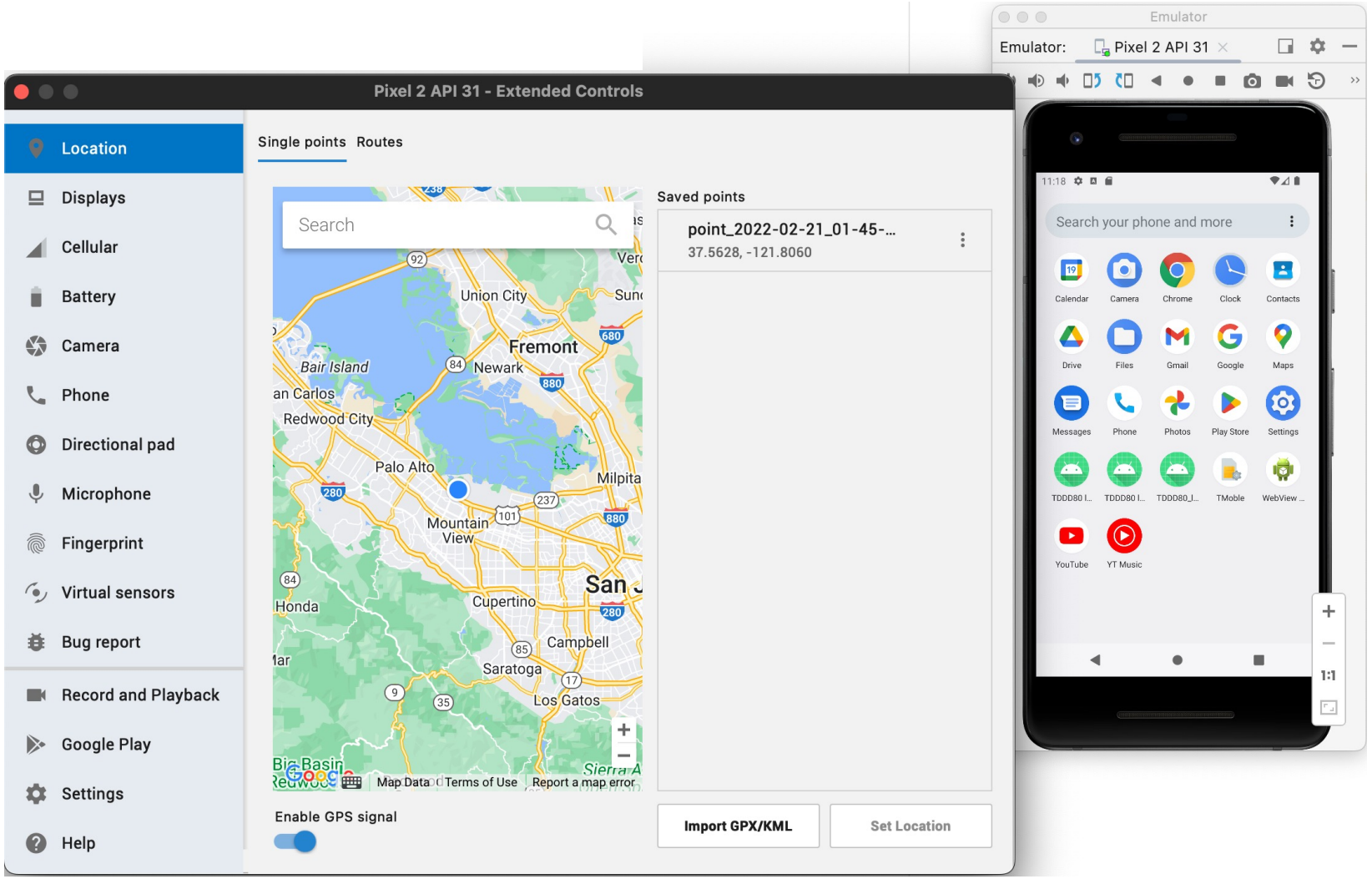
Internet

- Emulatorn delar datorns internetuppkoppling
- Kolla genom att t.ex. öppna emulatorns webläsare, och se om kontakt med internet
  - Annars: Inställningar på datorn för att dela internet

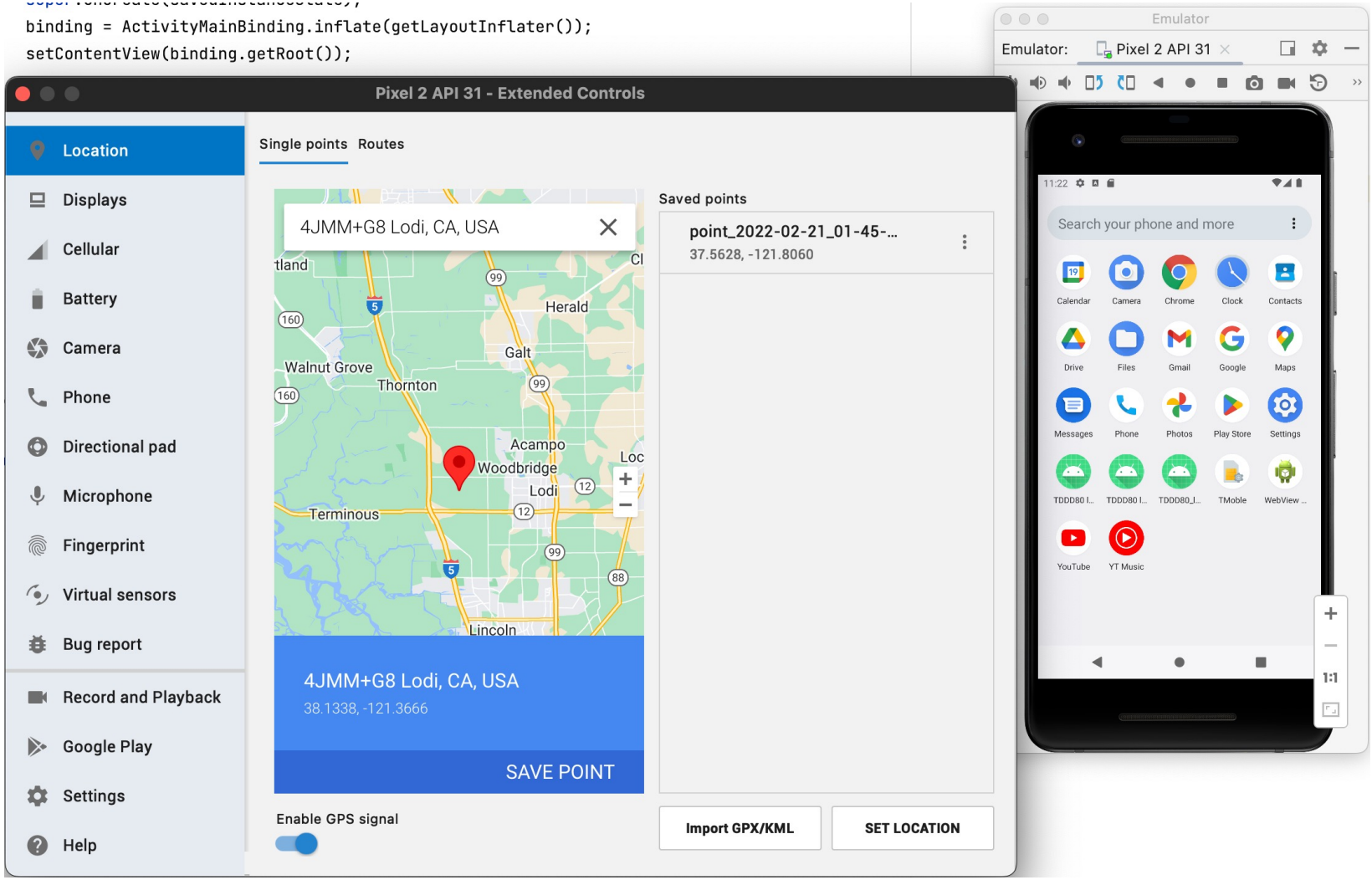
Virtuell kamera

Nås genom att klicka på appen Kamera




Pixel 2 API 31 - Extended Controls
Location
Displays
Cellular
Battery
Camera
Phone
Directional pad
Microphone
Fingerprint
Virtual sensors
Bug report
Record and Playback
Google Play
Settings
Help
Single points
Routes
Search
Union City
Fremont
Newark
Bair Island
Redwood City
Palo Alto
Milpita
Mountain View
Cupertino
Campbell
Saratoga
Los Gatos
Map Data
Terms of Use
Report a map error
Saved points
point_2022-02-21_01-45-...
37.5628, -121.8060
Enable GPS signal
Import GPX/KML
Set Location
Emulator
Emulator:
Pixel 2 API 31
Search your phone and more
Calendar
Camera
Chrome
Clock
Contacts
Drive
Files
Gmail
Google
Maps
Messages
Phone
Photos
Play Store
Settings
YouTube
YT Music

```
binding = ActivityMainBinding.inflate(getLayoutInflater());
setContentView(binding.getRoot());
```

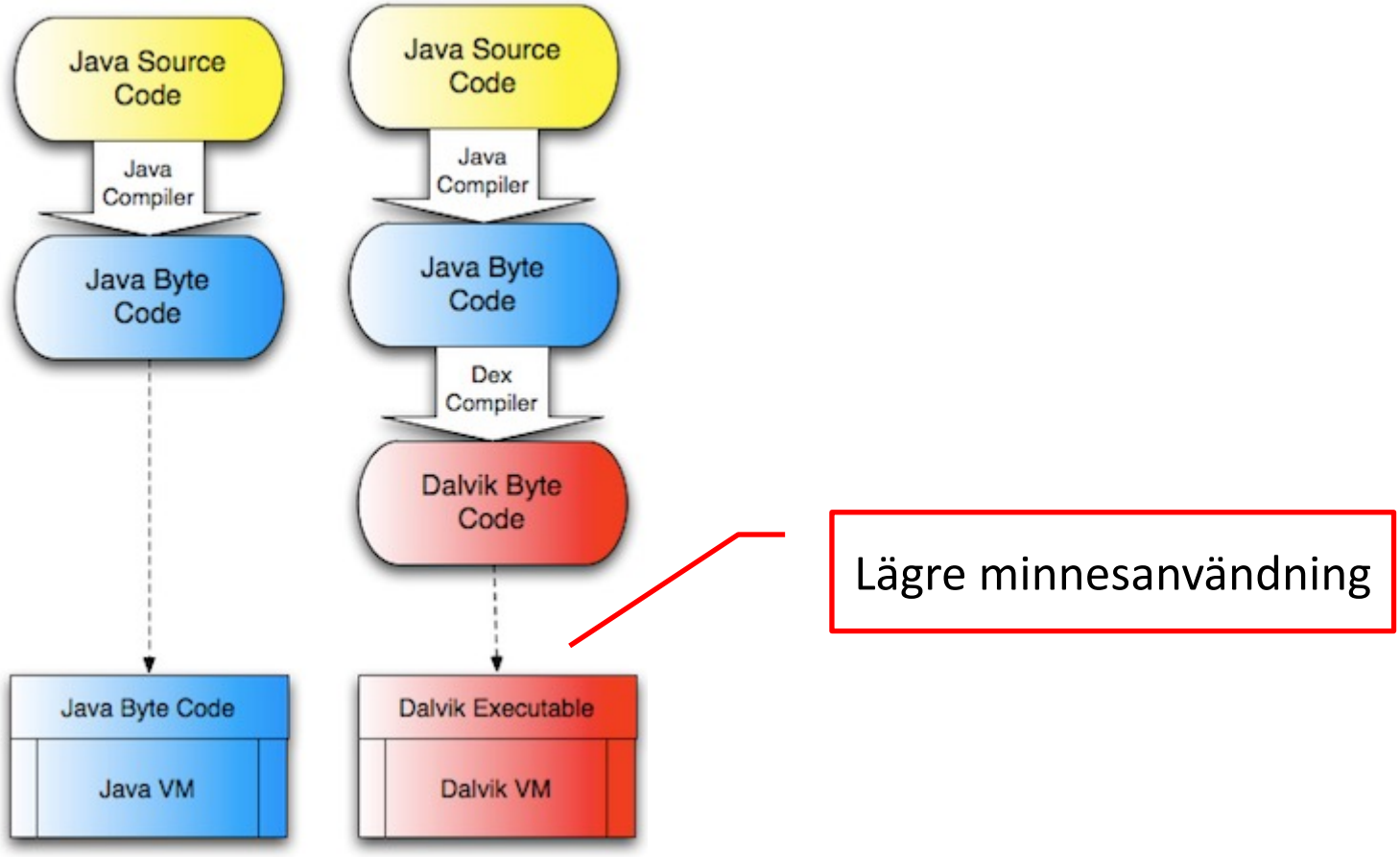


Android introduktion

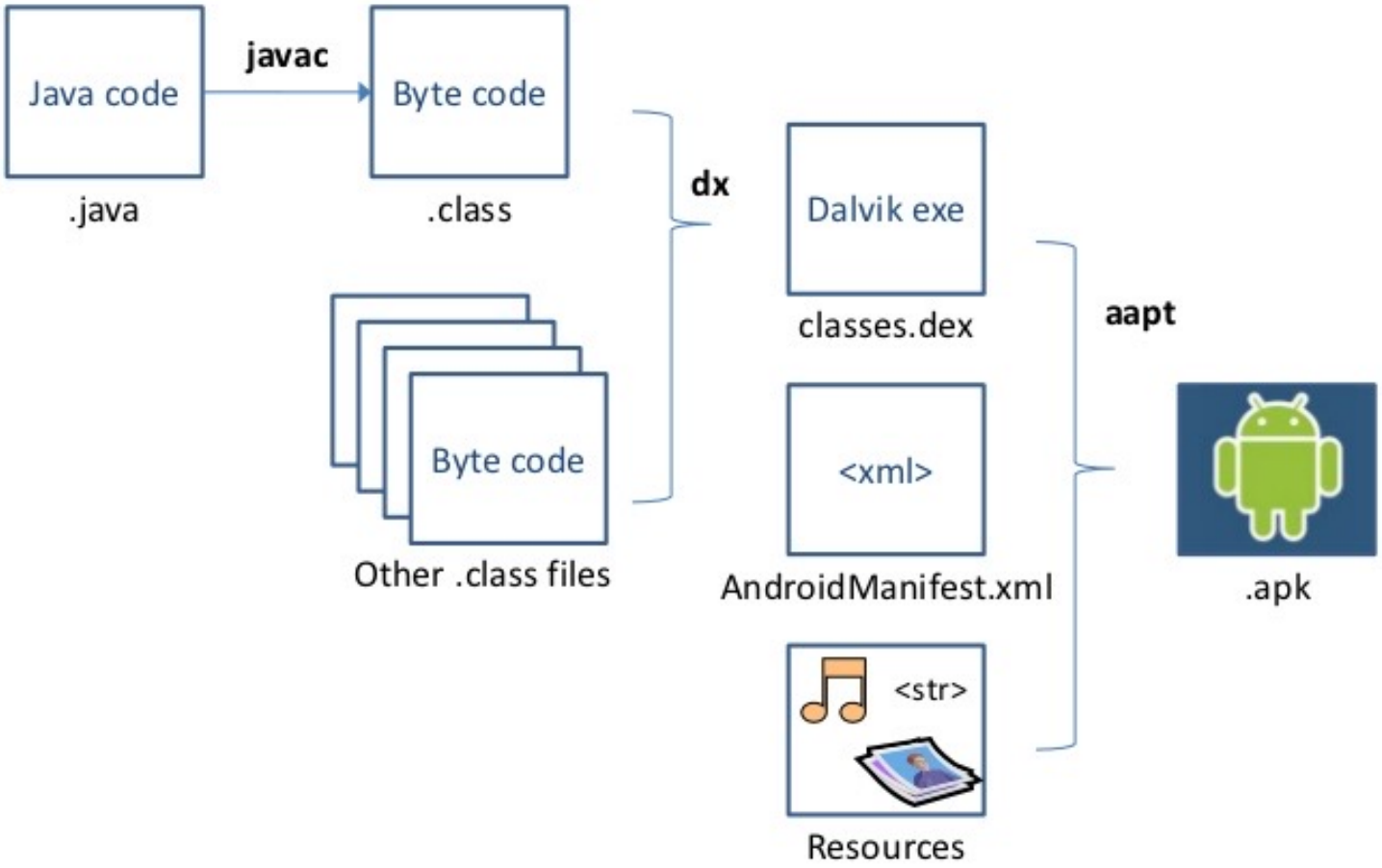
Vad som skiljer Android app från “vanlig” app

- Responsivt
  - Appar får ett par sekunder på sig att reagera på användarinput
- Resurssnålt
  - Appar i bakgrunden dödas när minnet behövs för annat
    - Så fort app:en hamnar i bakgrunden är det fritt fram för Android OS att döda den
      - Jfr. Quit på en desktop, som klickas av användaren
    - Viktigt att spara undan information i appen

Vanlig java jämfört med Android



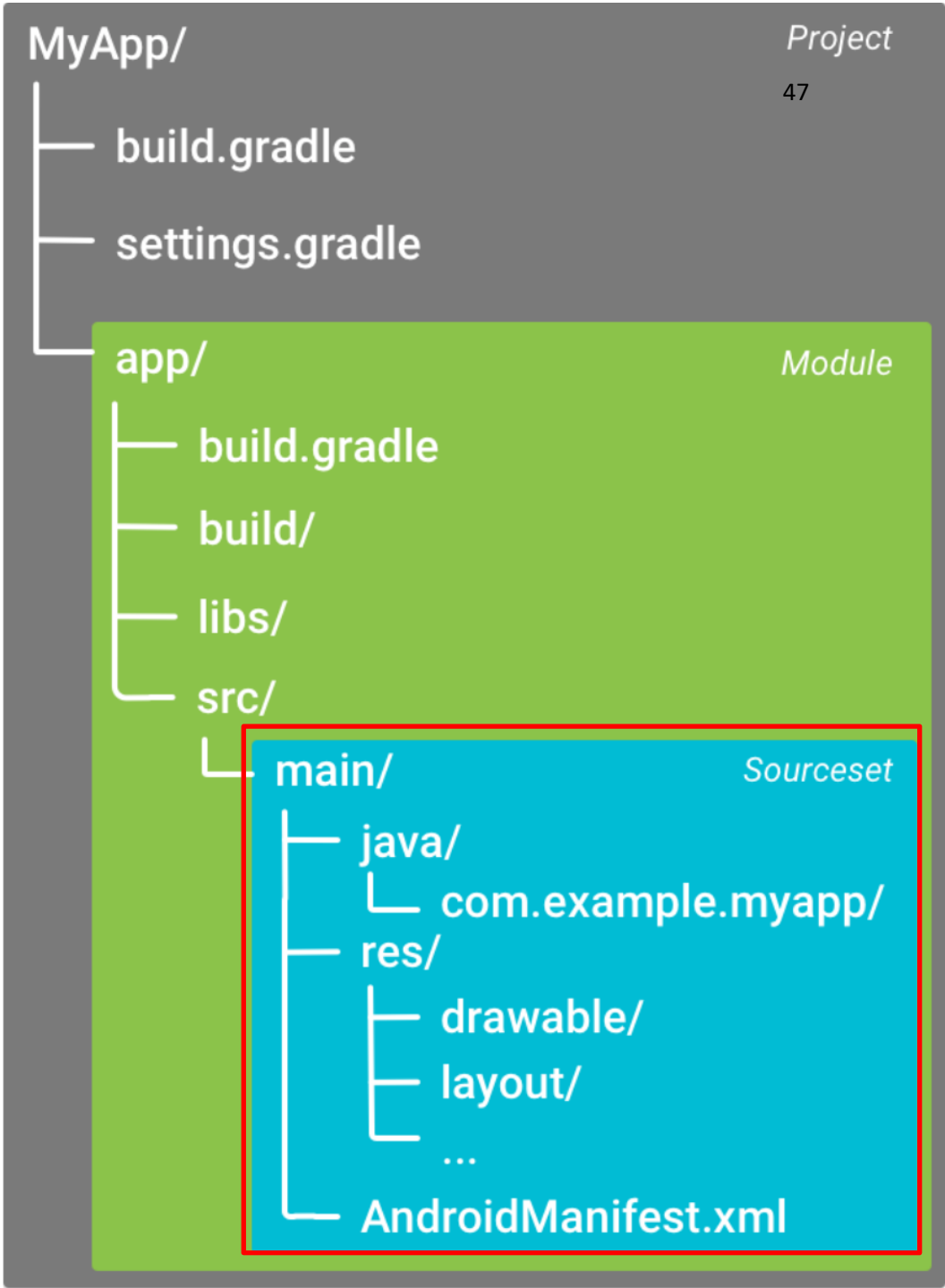
Android app


Java code
javac
Byte code
.java
.class
dx
Dalvik exe
classes.dex
aapt
Byte code
Other .class files
<xml>
AndroidManifest.xml
.apk
<str>
Resources

Kompilering

- Sköts av Gradle
- Olika inställningsfiler
  - Talar om vilka moduler som ska kompileras
  - Vilka bibliotek man vill ha med
  - Vilken klass som är startklassen, dvs. main

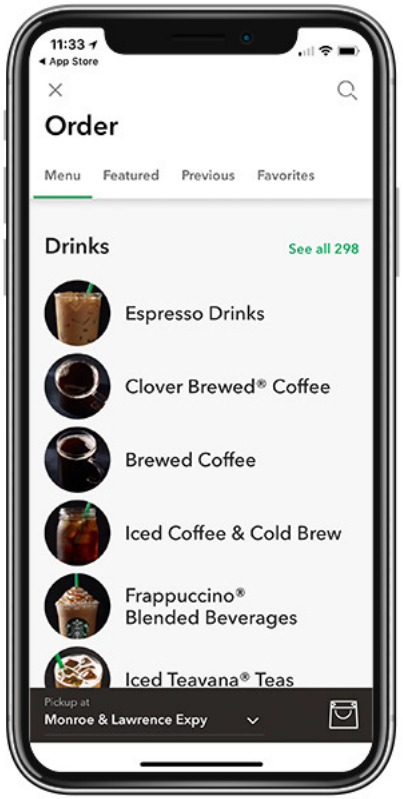
Projektstruktur

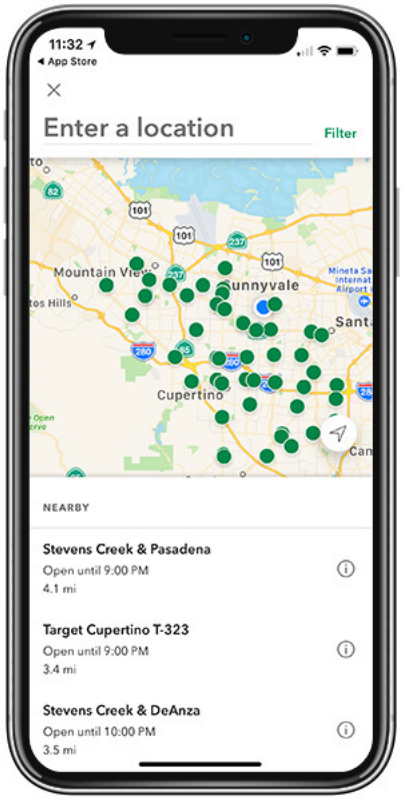


App - Activity

En app består av en eller flera Activities

Activity 1:
Välj kaffe





Activity 2:
Var hämta

Activity

- En avgränsad mängd funktionalitet som motsvarar en uppgift för användaren
  - T.ex. lista alla mail i inkorgen
  - T.ex. skriva mail

- Samlas i en klass som ärver från Activity

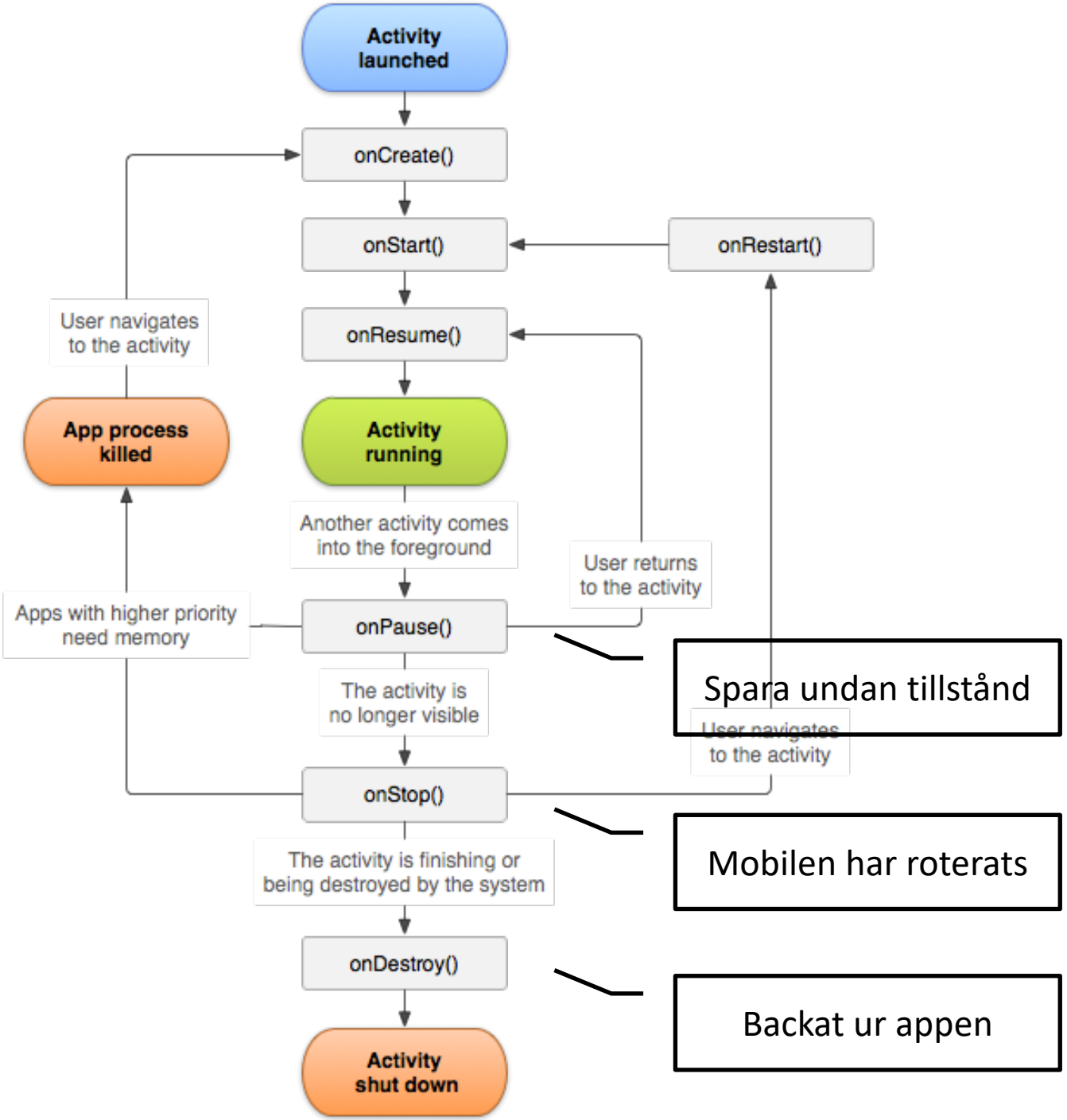
Ingen konstruktor, utan onCreate()

```
public class MainActivity extends Activity {

    private EditText newText;
    …

    @Override
    protected void onCreate(…) {
        …
        setContentView(R.layout.activity_main);
```

metod

Activity skapas och dödas av Android

- Stoppas om
  - T.ex. telefonen roteras
- Dödas om
  - T.ex. appen hamnat i bakgrunden (och ont om minne)
- Spara undan tillstånd i god tid. Två sätt:
  - Ge dina komponenter Id, och låt systemet spara dessa
  - Sköt det i onSaveInstanceState()


Activity launched
onCreate()
onStart()
onRestart()
onResume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
Spara undan tillstånd
The activity is no longer visible
User navigates to the activity
onStop()
Mobilen har roterats
The activity is finishing or being destroyed by the system
onDestroy()
Backat ur appen
Activity shut down

Två viktigaste life cycle metoderna

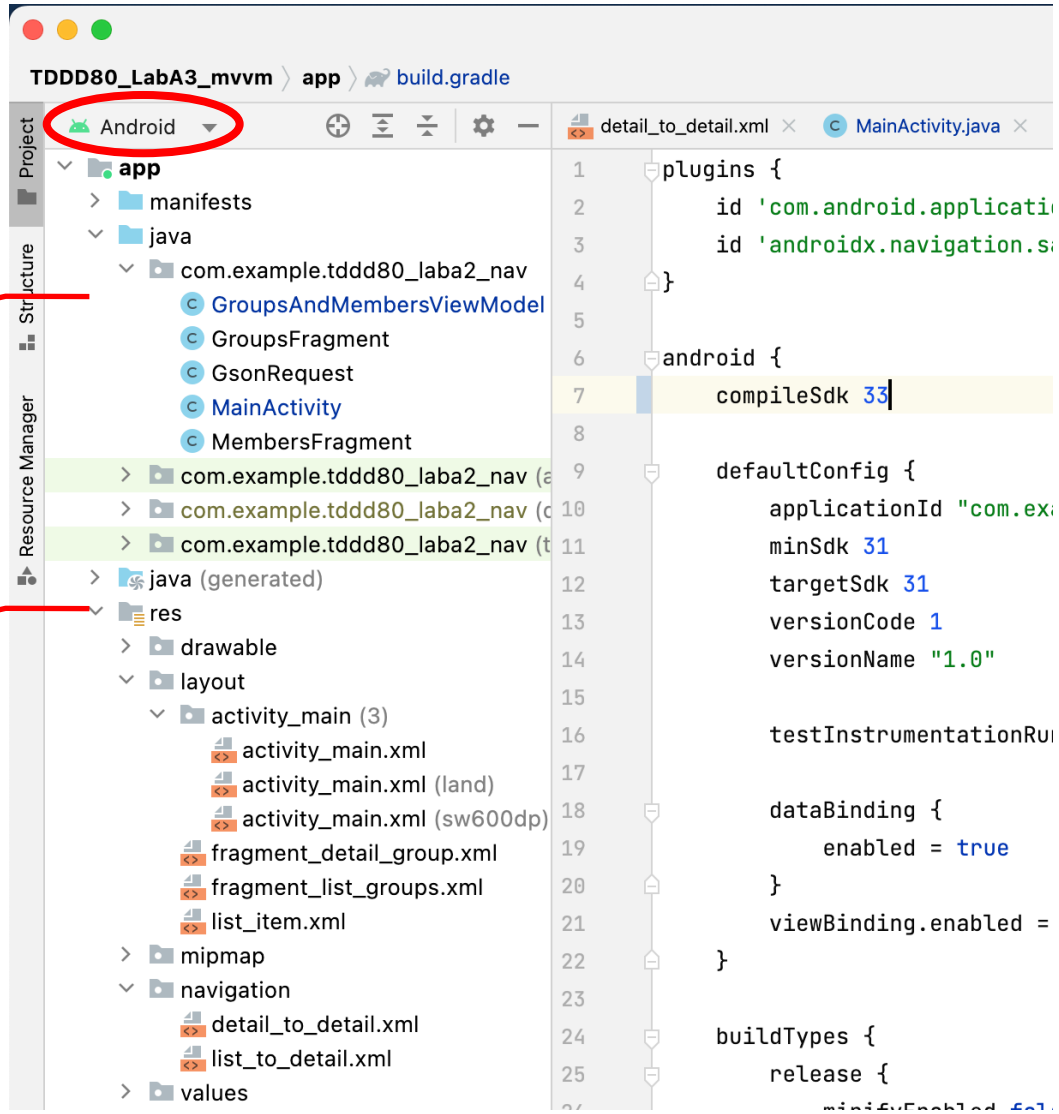
```
public class MainActivity extends Activity {

    @Override
    protected void onCreate(…) {
        …  // koppla till xml, hitta textfält, etc.

    @Override
    protected void onPause(…) {
        …  // spara undan tillstånd
```

Layout - beteende

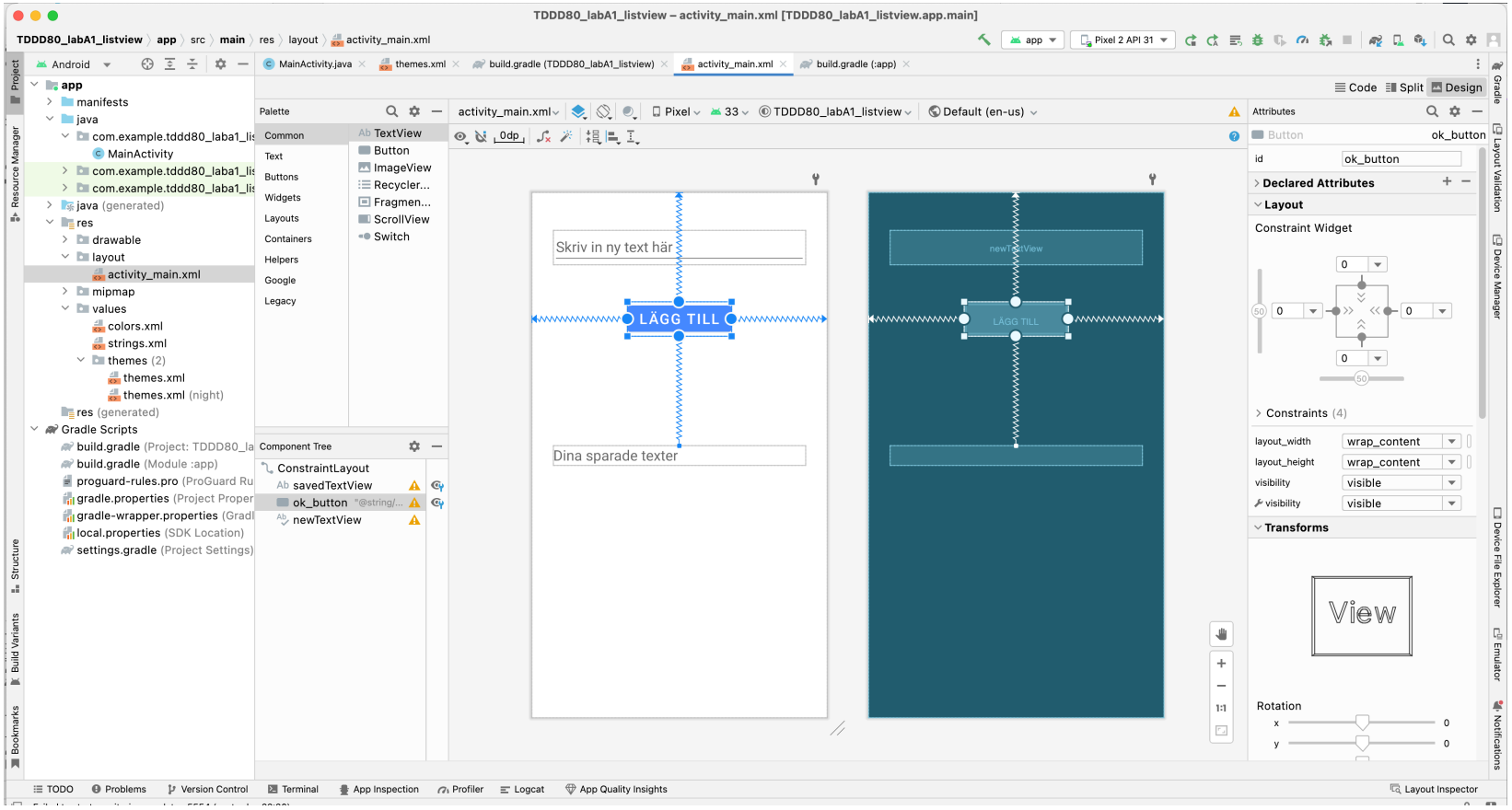
Projektstruktur

Kod

Resurser



Resurser > layout-mapp > activity_main.xml



Constraint layout tutorial

- https://www.youtube.com/watch?v=XamMbnzI5vE

Layout - beteende

- Ytan, dvs. användargränssnittet (UI) på appen: t.ex. textfält, knappar
  - Beskrivs i separat ”textfil” (XML)
- Dynamiska beteendet
  - Klickhantering, etc. sköts i Java-kod

XML vs. kod

- XML
  - Statisk placering och utseende på komponenter (dvs. kan inte ta bort under körning)
    - Tomma behållare (t.ex. FrameLayout) kan sedan fyllas dynamiskt när kod körs
- Java kod
  - Dynamiskt tillägg/borttagning av element
  - Fyllning av GUI-komponenter med innehåll

I koden anges vilken XML som ska användas

```
public class MainActivity extends Activity {

    private EditText newText;
    …

    @Override
    protected void onCreate(…) {
        …
        setContentView(R.layout.activity_main);
```

var dekl

metod

Android UI layout i XML

```
<LinearLayout xmlns:android="http://schemas.android.com/..."
   android:orientation=”vertical”

   <EditText
      android:layout_width="wrap_content”
      android:layout_height="wrap_content”
      android:id="@+id/inputText”
      android:inputType="textAutoCorrect”
      android:hint="@string/inputTextHint" />

</LinearLayout>
```

Hitta textfält i XML-filen

```
public class MainActivity extends Activity {

    private EditText newText;
    …

    @Override
    protected void onCreate(…) {
        …
        setContentView(R.layout.activity_main);
        newText = (EditText) findViewById(R.id.inputText);
```

Klickhantering

Fånga upp klick (enkelt)

- I XML:
  - <Button
    android:layout_width="wrap_content”
    android:layout_height="wrap_content”
    android:text="@string/button_send”
    android:onClick="sendMessage" />

- I Activity-kod:

  public void sendMessage(View view) {…}

Fånga upp klick (rekommenderat)

- Skapa lyssnare i Activity-koden:

```
Button btn = (Button) findViewById(R.id.mybutton);
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sendMessage(view);
    }
});

public void sendMessage(View view) {…}
```

Fördelar med denna lösning

- Anonym lyssnare
  - Kan ha olika lyssnare kopplade till olika knappar inom samma klass
- Skickar anropet direkt vidare till en lokal metod
  - Kan testa (anropa) metoden utan att behöva klicka på knappen varje gång
  - Underlättar automatiska tester

Avslutande tips: Android Codelab

- Ta gärna en titt på Codelab Hello-world:

- https://codelabs.developers.google.com/codelabs/android-training-hello-world/index.html?index=..%2F..index#0

Tack! Frågor?

www.liu.se

LINKÖPINGS UNIVERSITET