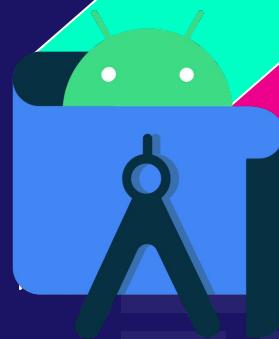


#5

Android Studio



Android Studio

" Set the value in onCreate, setValue in onClick and also in this third place in case they change the value there and also there and.. help.."

INNEHÅLLSFÖRTECKNING

01

Översikt

03

ViewModel

02

Setup &
LifeCycles

04

Uppgifter
&
Övningar

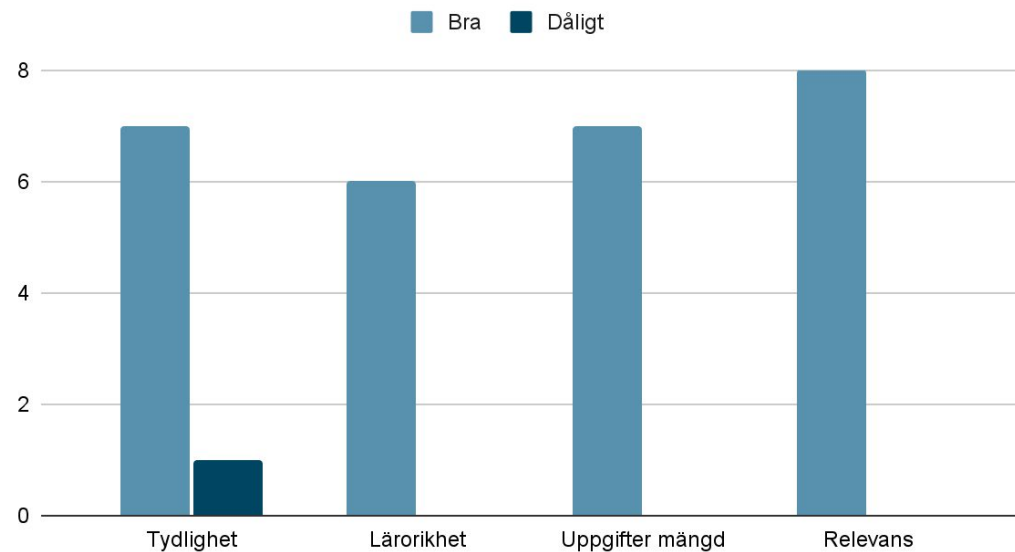
Poll Svar



Uppgifter POLL

Vad tyckte ni?

Resultat



Schema



Schema

Schemat har förändrat lite - vi har hunnit med mer på kortare tid!

Mer tid för eget arbete och nya koncept/teknologier!

| Vecka | Tema | Dag | Datum | Tid | På plats |
|-------|---|----------|-------|-------|----------|
| Vecka | Innehåll | Veckodag | Datum | Längd | Sal |
| V 6 | ✓ Kotlin - Intro | Torsdag | 9/2 | 9-16 | Sal 3 |
| V 7 | ✓ Android Studio - Intro | Tisdag | 14/2 | 9-16 | Sal 2 |
| | ✓ Navigation & Intent | Torsdag | 16/2 | 9-16 | Sal 6 |
| | ✓ Självstudier - Lab 1 intro | Fredag | 17/2 | 9-16 | Distans |
| V 8 | ✓ Switch Components, Viewbinding, Classes & Objects | Måndag | 20/2 | 9-16 | Sal 2 |
| | Lifecycles & ViewModel | Onsdag | 22/2 | 9-16 | Sal 4 |
| | Självstudier | Fredag | 24/2 | 9-16 | Distans |
| V 9 | Google Play | Måndag | 27/2 | 9-16 | Sal 4 |
| | ViewModel #2 | Tisdag | 28/2 | 9-16 | Sal 4 |
| | Självstudier - Lab 2 intro | Torsdag | 2/3 | 9-16 | Distans |
| V 10 | RecyclerView | Måndag | 6/3 | 9-16 | Sal 5 |
| | Navigation Controller | Onsdag | 8/3 | 9-16 | Sal 4 |
| V 11 | SQLite - Projektarbete intro | Tisdag | 14/3 | 9-16 | Sal 4 |
| | Självstudier | Onsdag | 15/3 | 9-16 | Sal 5 |



01

ÖVERSIKT

ViewModel... LifeCycles?



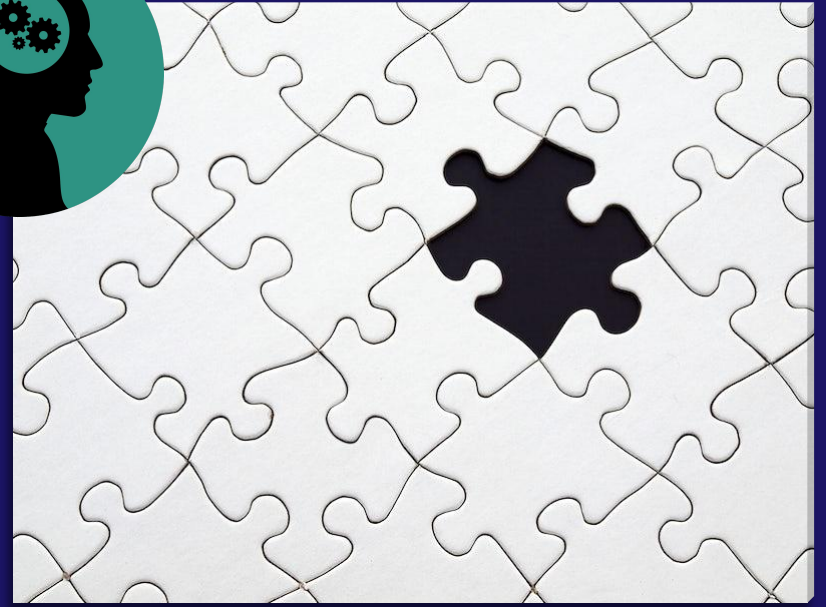
The problems

Livscyklar förändras konstant, varför arbetar man med ViewModels?

- Livscykel
- ViewModel
- Arv
- Skärm Roterig?



Frågor?



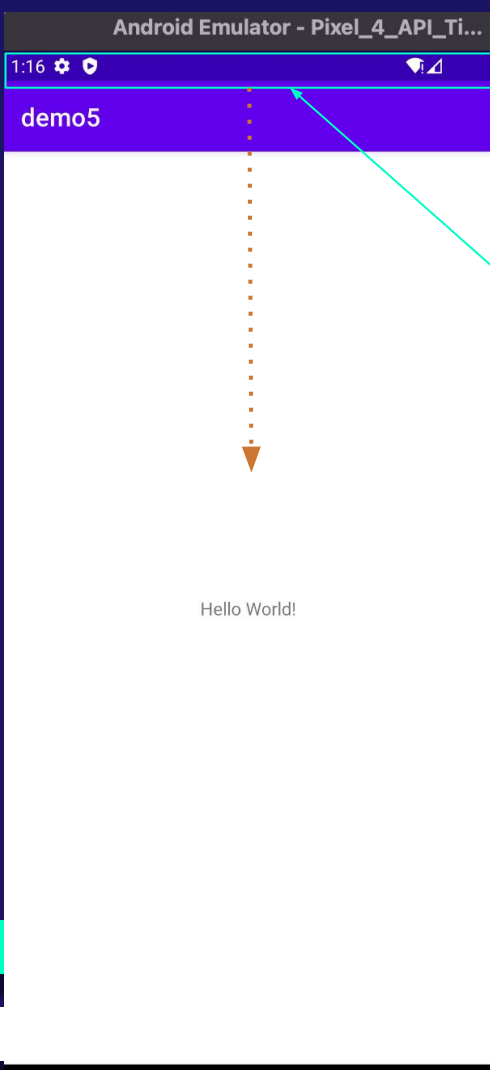


02

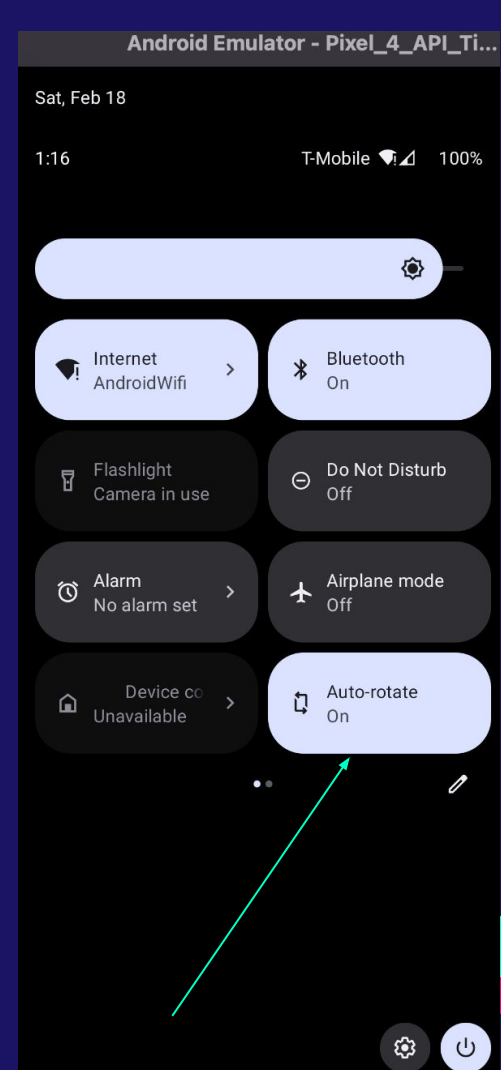
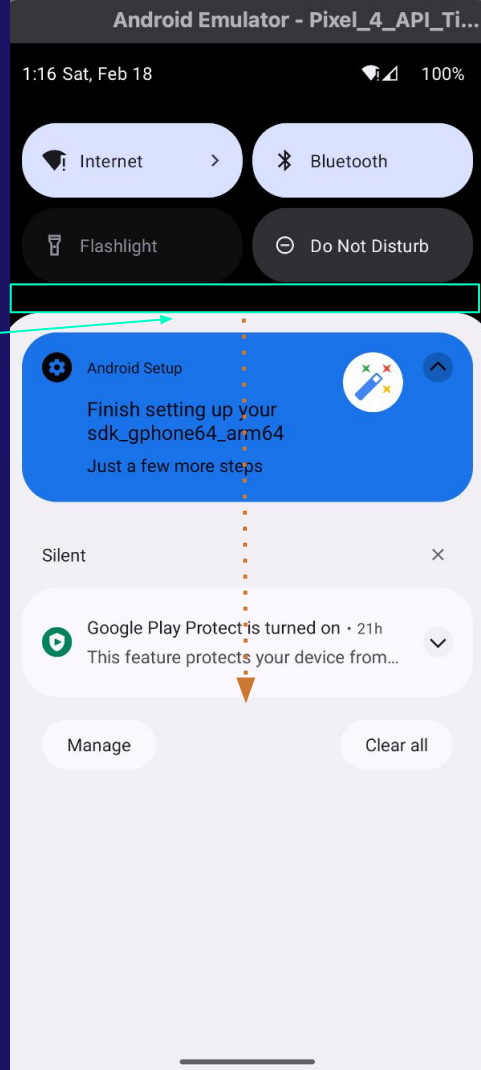
Setup & LifeCycles

Enable Auto Rotate





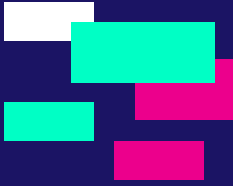
Drag och släpp
nedåt!





Auto Rotate

Detta är superviktigt för när vi börjar demonstrera!





Setup ViewBinding RECAP



Recap

✓ Gradle Scripts

 build.gradle (Project: demo5)

 build.gradle (Module :app)

```
buildFeatures {  
    viewBinding true  
}
```


Recap

```
private lateinit var binding: ActivityMainBinding // Binding

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    // Binding Boilerplate
    binding = ActivityMainBinding.inflate(layoutInflater)
    val view = binding.root
    setContentView(view)

    // ID
    val textViewCounter = binding.textViewCounter
```

Setup Counter Class



Counter Class

```
com.example.demo5
Counter
3  class Counter(var value: Int) {
4
5      fun add() {
6          value++
7      }
8
9  }
```

Konstruktör tar in ett 'heltal'

Klassen har en metod som ökar värdet med +1

Counter Class

▼ com.example.demo5

Counter

MainActivity

```
class MainActivity : AppCompatActivity() {  
  
    private val myCounter = Counter(value: 0)  
    private lateinit var binding: ActivityMainBinding
```

Instansiera en ny klass!

Counter Class

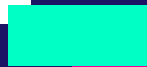
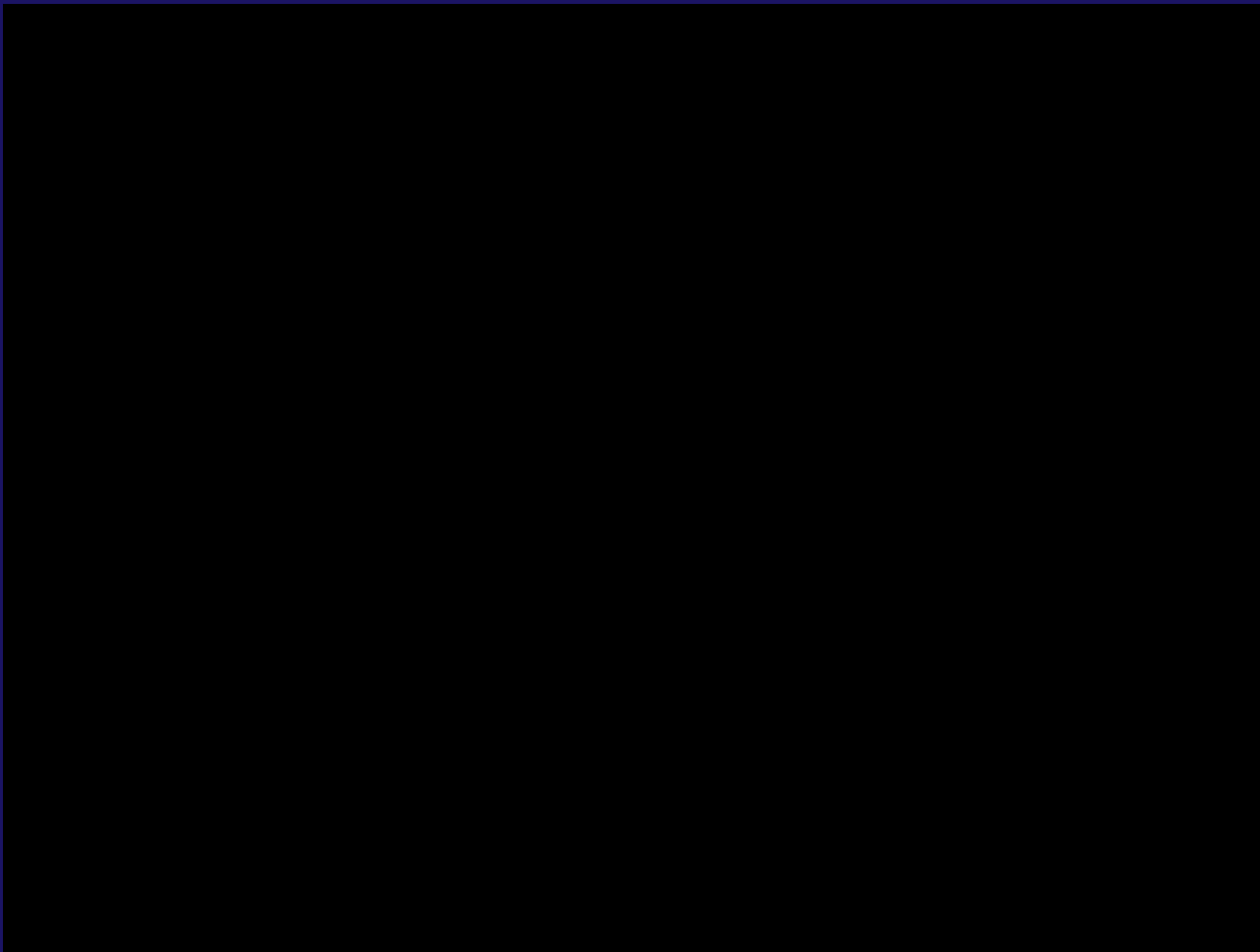
```
val textViewCounter = binding.textViewCounter

textViewCounter.setOnClickListener() { it: View!
    myCounter.add()
    textViewCounter.text = myCounter.value.toString()
}
```

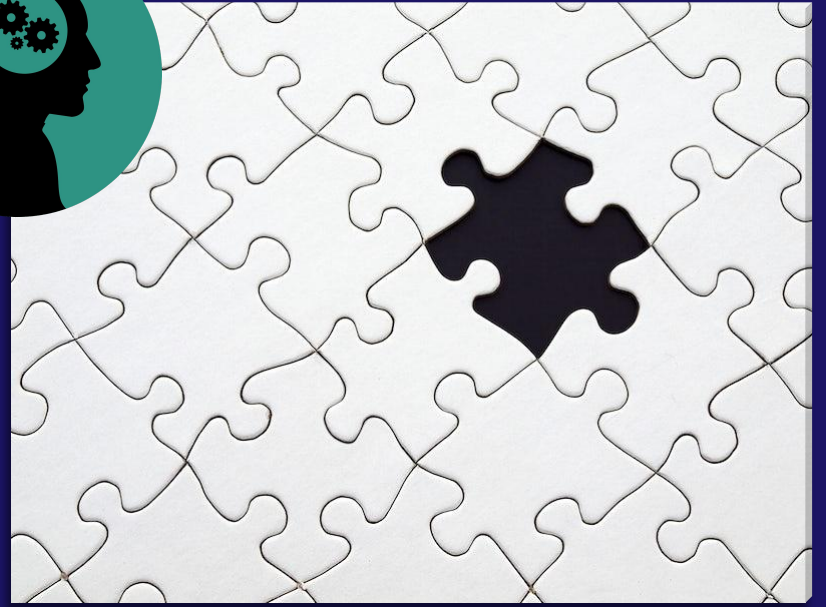
Skapa en onClickListener

Sätt textvy'ns värdet till objektets nuvarande men utökat +1 värde
(glöm inte göra om detta till en sträng)

Tryck på den
svarta rutan
för att spela
upp videon!



Frågor?

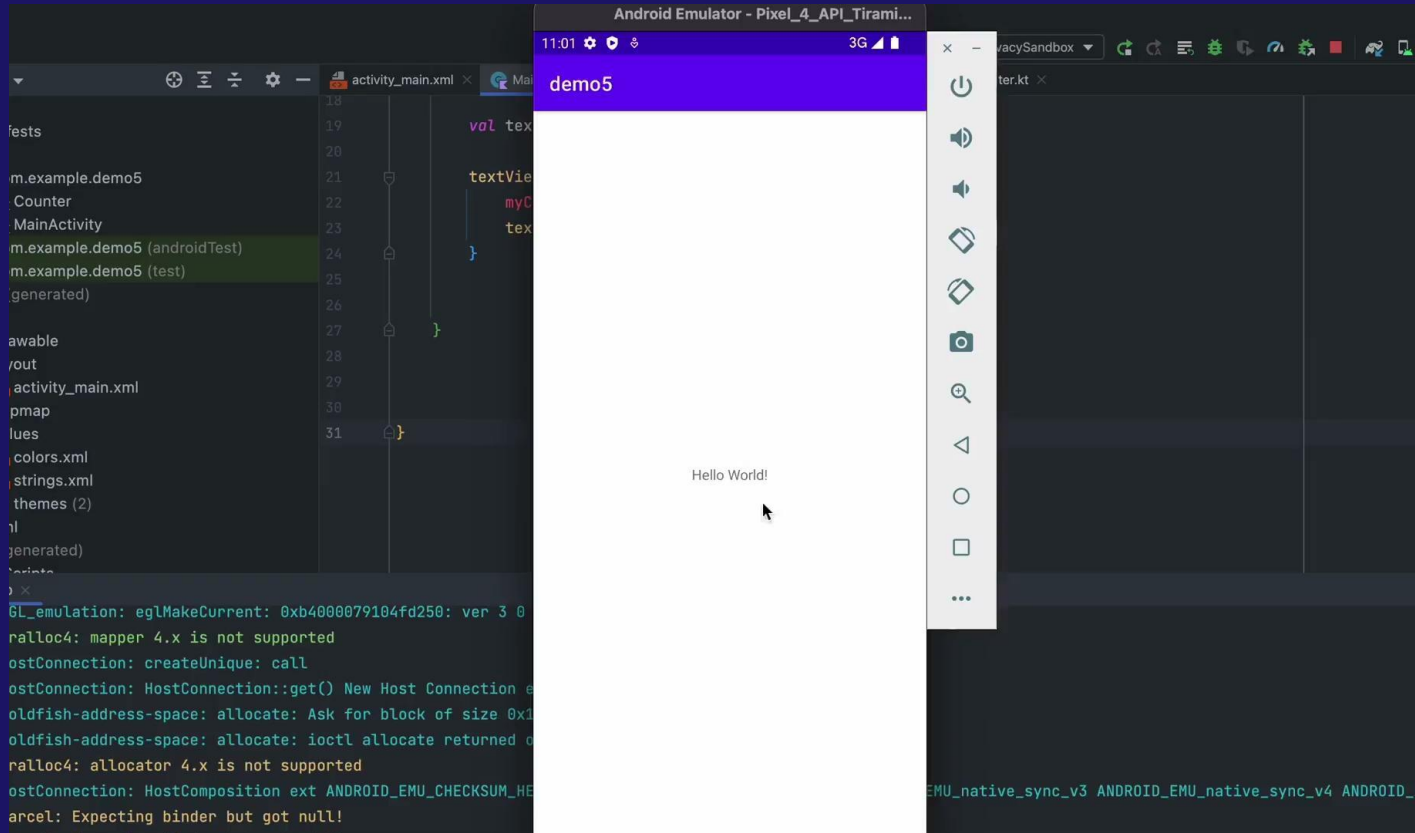


Introducera Problemet



Rotation

Klicka på videon!



LifeCycle





“As a user navigates through, out of, and back to your app, the [Activity](#) instances in your app transition through different states in their lifecycle”

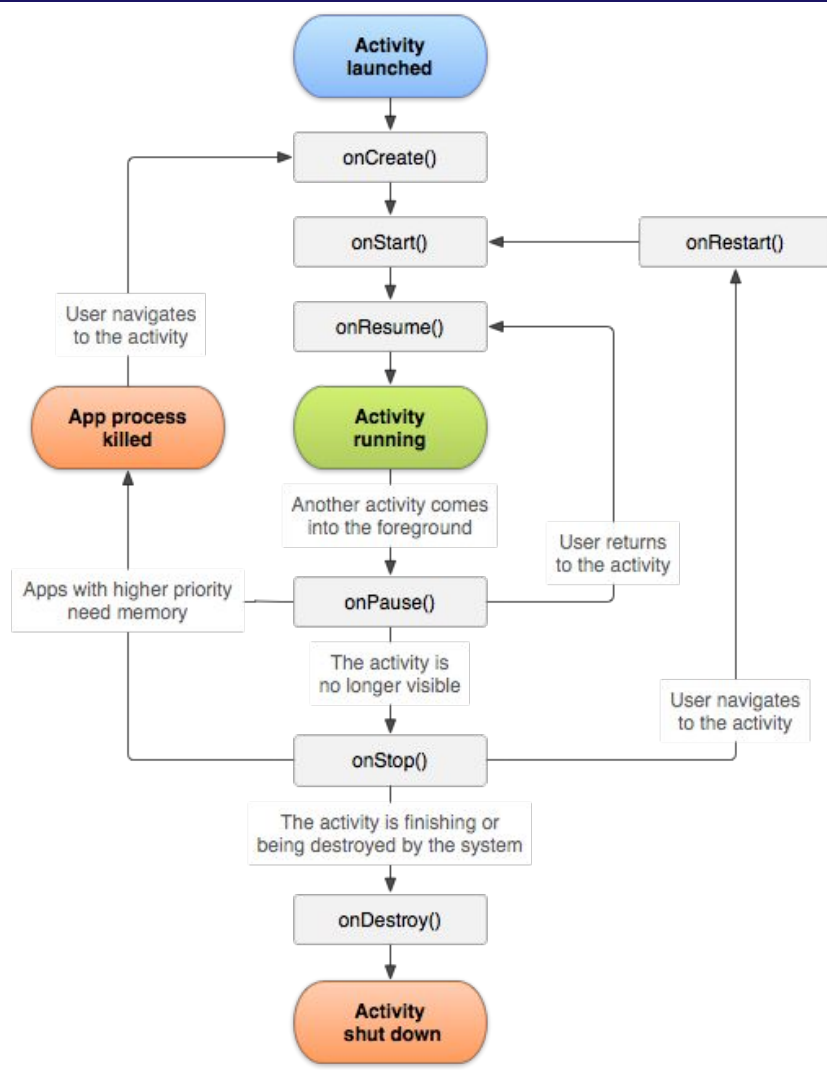
- developer.android.com

LifeCycle

For example, good implementation of the lifecycle callbacks can help ensure that your app avoids:

- Crashing if the user receives a phone call or switches to another app while using your app.
- Consuming valuable system resources when the user is not actively using it.
- Losing the user's progress if they leave your app and return to it at a later time.
- Crashing or losing the user's progress when the screen rotates between landscape and portrait orientation.

Source: [android developers](#)



Livscykeln är något vi diskuterar så fort vi pratar om vyer AKA 'activities'

Varje vit rektangel symboliserar en livscykel!

LifeCycle

Activity created

onCreate

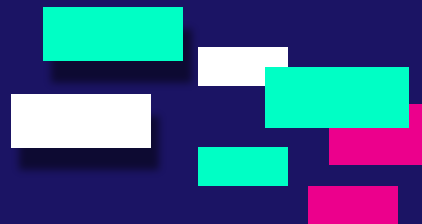
onStart

onResume

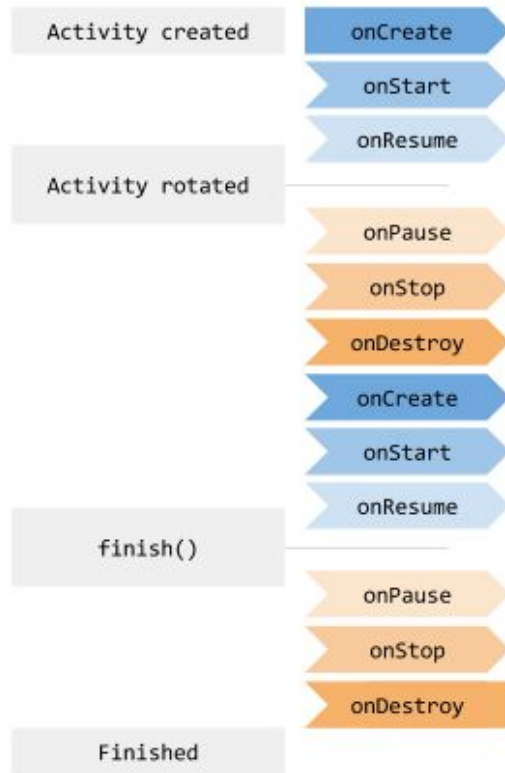
```
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
}
```

Så fort aktiviteten skapas:

```
onCreate( )  
onStart( )  
onResume( )
```



LifeCycle



Activity rotated!

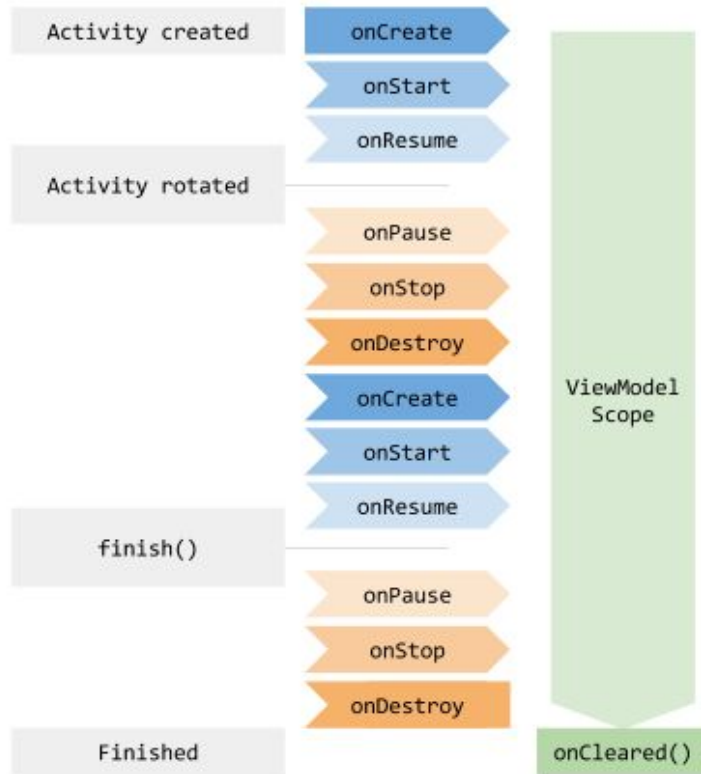
```
onPause()  
onStop()  
onCreate()  
onStart()  
onResume()
```

Activiteten återskapas!



LifeCycle

ViewModel 'överlever' och man kan t.o.m säga att den ligger 'utanför' livscykeln på detta sätt!





Problem

Vad är en livscykel?

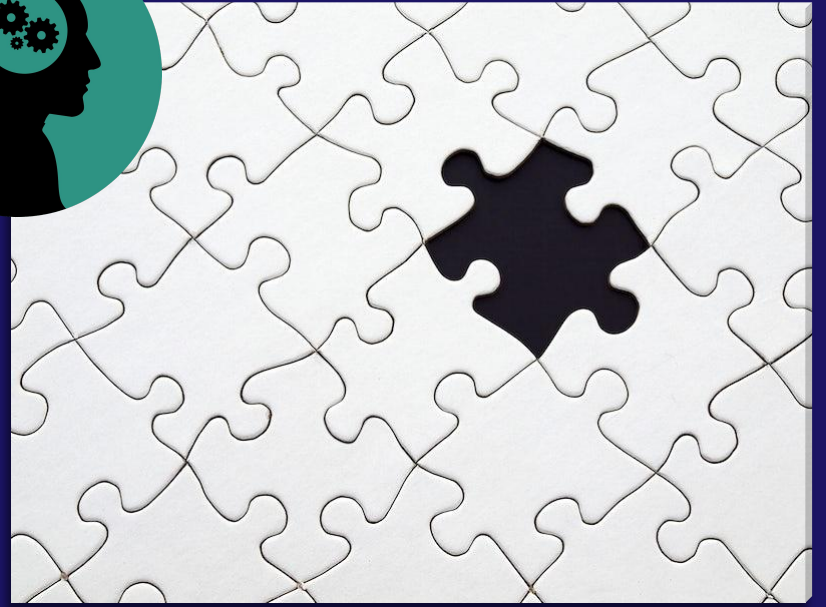


Solution

Aktiviteter skapas, pausas, återupptas, stoppas och förstörs. Detta är individuella delar AKA livscykler



Frågor?





03

ViewModel

Step by Step



ViewModel

Extenda din klass med 'ViewModel'

```
class Counter : ViewModel() {  
  
    var value: Int = 0  
  
    fun add() {  
        value++  
    }  
  
}
```

NOTERA: Jag har tagit bort min konstruktor ()

ViewModel

Finslipa kod inom Main!

```
8    class MainActivity : AppCompatActivity() {  
9  
10     // Initialization  
11     → private lateinit var counterViewModel: Counter    // ViewModel  
12     private lateinit var binding: ActivityMainBinding  // ViewBinding  
13
```

Skriver koden snyggare!



ViewModel

Skapa koppling med 'ViewModelProvider'

```
// ViewModel SETUP
counterViewModel = ViewModelProvider(owner: this)[Counter().javaClass]
textViewCounter.text = counterViewModel.value.toString()

// On click
textViewCounter.setOnClickListener() { it: View!
    counterViewModel.add()
    textViewCounter.text = counterViewModel.value.toString()
}
```



Problem

Varför ViewModels?



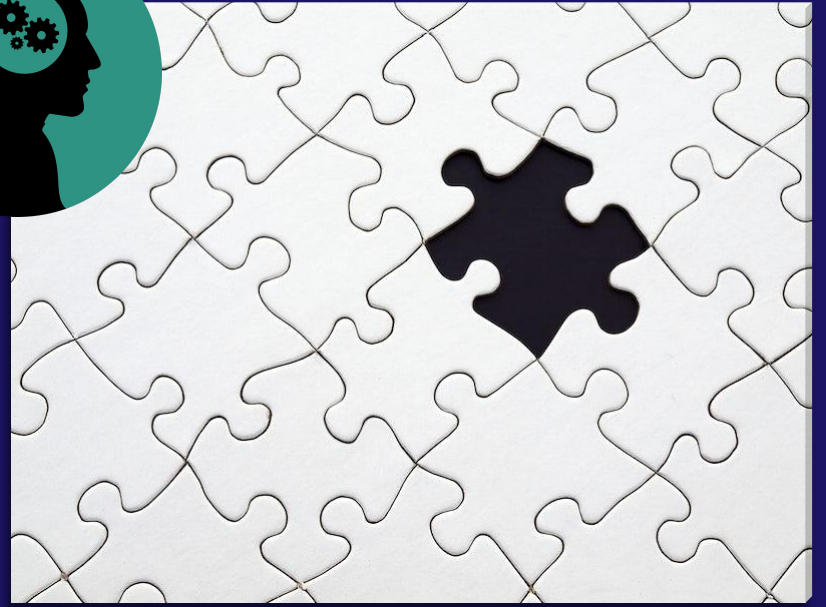
Solution

När vi roterar telefonen så försvinner
vår data då denna aktivitet
återskapas...

ViewModel överlever livscykeln!



Frågor?





04

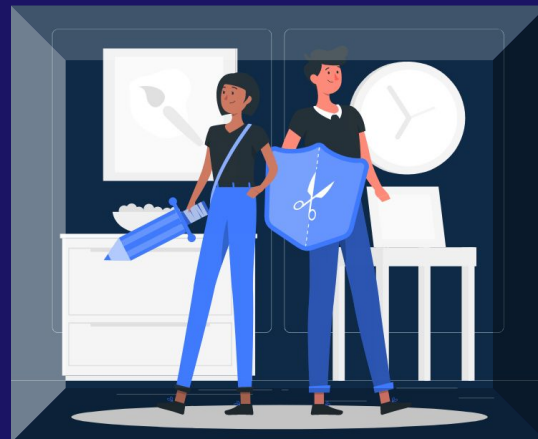
Uppgifter
&
Eget Arbete

Välkommen till första uppgiften!

Uppgifterna är till för att testa dina färdigheter och kunskaper för att både öva och repetera på det vi har arbetat med under föreläsningarna.

Dessa är **INTE** obligatoriska.
Men är starkt rekommenderat att arbeta med.

Uppgifter



MINNS DU?

```
// ViewModel, vad är det? Hur tillämpas  
den?
```

```
// Förklara: Aktivitet & livscyklar
```

```

1          // -Uppgift #1- //
2
3      /* INSTRUCTIONS
4
5          Skapa ett nytt projekt!
6
7          Döp projektet till: Lektion_5_uppgifter
8
9          Skapa en 'Counter' klass!
10         Denna klass skall kunna räkna varje gång
11         användaren trycker på en 'knapp'
12
13         Skapa även en textvy som visar upp talet!
14
15     */
16
17     // HINT & Examples
18     hint(" Back to basics! Är det något du glömt?
19     Titta på tidigare lektioner och kom sedan
20     tillbaka!
21
22     Glöm inte att göra snabba google sökningar om ni
23     inte kommer ihåg hur man skriver ett kodstycke!
24 ")

```

Uppgift #1

Kom igång enkelt med uppgift #1

```

1          // -Uppgift #2- //
2
3      /* INSTRUCTIONS
4
5          Nu skall din 'Counter' klass ärva från
6          klassen 'ViewModel'.
7
8          counterViewModel =
9      ViewModelProvider(this)[Counter().javaClass]
10
11          Glöm inte senare att visa upp och
12          sätta texten så fort aktiviteten startas.
13
14      */
15
16      // HINT & Examples
17      hint(" Kodstycket som ni skall copy/paste:a är
18      till för kopplingen.
19
20      Lägg till detta kod stycke inom MainActivity.
21
22      Slide #38 ")
23

```

Uppgift #2

Koppling är viktigt, utan den kommer detta ej fungera!

KAN DU?

TOUGH NUT

```
// Nu har vi diskuterat ViewModels...  
Vi kommer oundvikligt behöva prata om  
'context' längre fram inom Android  
utveckling.
```

```
// Ta reda på vad ett 'context' är...
```

THANKS!

Do you have any questions?
kristoffer.johansson@sti.se

sti.learning.nu/

CREDITS: This presentation template was created by
Slidesgo, including icons by Flaticon, and
infographics & images by Freepik.

You can also contact me VIA Teams (quicker response)
Du kan också kontakta mig VIA Teams (Snabbare svar)