

Activity Stack -> A

Activity A

Visible:

onResume //press button to move to activity B

onPause

onStop

//Hide the activity

Activity B

Visible:

onCreate

onStart

onResume // press button to go back

Activity Stack -> A, B

onPause

onStop

onDestroy

//gone forever, it will be onCreate if we want to come back

Activity Stack -> A

Activity A

Visible:

onRestart-> onStart

onResume // press back button

onPause

onStop

onDestroy // new devices keep the recent apps so it won't be called unless cleared

Activity Stack -> NOTHING/Launcher

Activity Lifecycle: <https://developer.android.com/guide/components/activities/activity-lifecycle>

Fragment Lifecycle: <https://developer.android.com/guide/fragments/lifecycle>

– onAttach, **onCreateView**, **onViewCreated**, **onDestroyView**, onDetach

Launch Modes: <https://dev.to/mohitrajput987/launch-modes-of-android-activity-59eo>

ViewModel: <https://developer.android.com/topic/libraries/architecture/viewmodel>

- How does the view model stay alive throughout activity rotation?
- What makes view model lifecycle aware? How does Viewmodel keep track of activity's lifecycle?

Superclass in Kotlin:

All classes in Kotlin have a common superclass, Any, which is the default superclass for a class with no supertypes declared:

class Example // Implicitly inherits from Any

Superclass in Java:

All classes in Java have common superclass, Object.

Question:

- Intent
 - Implicit
 - Explicit
 - Pending //notifications etc

Major components

- Activity: responsible for UI and handling entry point/starting to the app (user interactions, visuals)
- Broadcast receiver: responsible for handling announcements, device level or app level (battery low, charging, new sms/call, location updated, bluetooth connection. . .)
- Service: similar to ACTIVITY (it has its own context), only difference is it cannot have the UI (music player, radio, downloading files, syncing contacts/files. . .) (by default runs on MAIN thread)
- Content Provider: responsible for sharing data between different applications (contacts list, music albums, images, videos, documents, sms, calls, . . .)