A CLOSER LOOK AT ANDROID ACTIVITIES

DHARANI.D
20MSS008

CONTENTS

- ✓ **ANDROID ACTIVITY**
- ✓ **ACTIVITY STATES**
- ✓ **ACTIVITY LIFETIMES**
- ✓ **ACTIVITY CLASSES**

ANDROID ACTIVITY

- ✓ Activity is nothing but a java class in Android which has some pre-defined functions which are triggered at different App states, which we can override to perform anything we want.
- ✓ Activity class provides us with empty functions allowing us to be the controller of everything.
- ✓ An Activity is responsible to create and destroy an App infinite number of times. So apart from controlling the app, Activity also controls creation, destruction and other states of the App's lifecycle.

ACTIVITY STATES

ACTIVE STATE:

- When an Activity is in active state, it means it is active and running.
- It is visible to the user and the user is able to interact with it.
- Android Runtime treats the Activity in this state with the highest priority and never tries to kill it.

PAUSED STATE:

- An activity being in this state means that the user can still see the Activity in the background such as behind a transparent window or a dialog box i.e it is partially visible.
- The user cannot interact with the Activity until he/she is done with the current view.

STOPPED STATE:

- When a new Activity is started on top of the current one or when a user hits the Home key, the activity is brought to Stopped state.
- The activity in this state is invisible, but it is not destroyed.

DESTROYED STATE:

- When a user hits a Back key or Android Runtime decides to reclaim the memory allocated to an Activity i.e in the paused or stopped state, It goes into the Destroyed state.
- The Activity is out of the memory and it is invisible to the user.

Method	What does it do?
onCreate()	Whenever an Activity starts running, the first method to get executed is onCreate(). This method is executed only once during the lifetime. If we have any instance variables in the Activity, the initialization of those variables can be done in this method. After onCreate() method, the onStart() method is executed.
onStart()	During the execution of onStart() method, the Activity is not yet rendered on screen but is about to become visible to the user. In this method, we can perform any operation related to UI components.
onResume()	When the Activity finally gets rendered on the screen, onResume() method is invoked. At this point, the Activity is in the active state and is interacting with the user.
onPause()	If the activity loses its focus and is only partially visible to the user, it enters the paused state. During this transition, the onPause() method is invoked. In the onPause() method, we may commit database transactions or perform light-weight processing before the Activity goes to the background.
onStop()	From the active state, if we hit the Home key, the Activity goes to the background and the Home Screen of the device is made visible. During this event, the Activity enters the stopped state. Both onPause() and onStop() methods are executed.
onDestroy()	When an activity is destroyed by a user or Android system, onDestroy() function is called.

ACTIVITY LIFETIMES:

FULL LIFETIME:

The entire lifetime of an activity occurs between the first call to onCreate() through to a single final call to onDestroy(). An activity will do all setup of "global" state in onCreate(), and release all remaining resources in onDestroy(). For example, if it has a thread running in the background to download data from the network, it may create that thread in onCreate() and then stop the thread in onDestroy().

VISIBLE LIFETIME:

The visible lifetime of an activity happens between a call to onStart() until a corresponding call to onStop(). During this time the user can see the activity on-screen, though it may not be in the foreground and interacting with the user. Between these two methods you can maintain resources that are needed to show the activity to the user.

ACTIVE LIFETIME:

The active lifetime of an activity happens between a call to onResume() until a corresponding call to onPause(). During this time the activity is in front of all other activities and interacting with the user. An activity can frequently go between the resumed and paused states -- for example when the device goes to sleep, when an activity result is delivered, when a new intent is delivered -- so the code in these methods should be fairly lightweight.

ANDROID ACTIVITY CLASSES:

MAP ACTIVITY:

Encapsulates the resource handling required to support a Map View widget within an Activity.

LIST ACTIVITY:

Wrapper class for Activities that feature a List View bound to a data source as the primary UI metaphor, and expose event handlers for list item selection.

EXPANDABLE LISTACTIVITY:

Similar to the List Activity but supports an Expandable List View.

Thank You