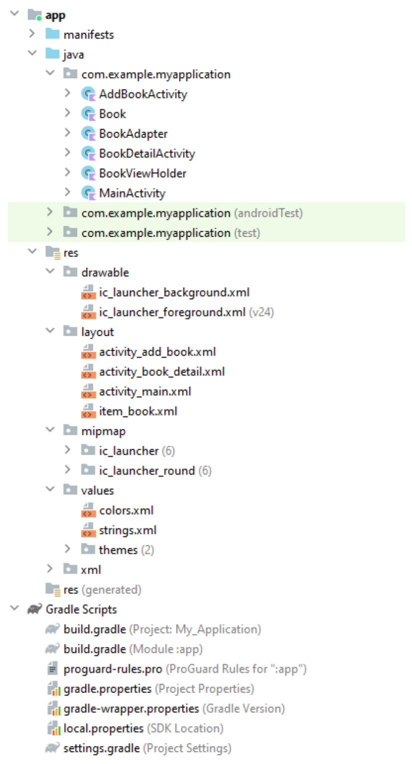
Mobile Application Development

Mobile Platforms and Application Development fundamentals

Agenda

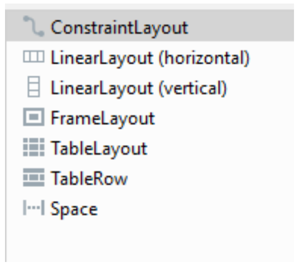
- Android Project Structure
- Activity Layouts
- Activity Kotlin
- Activity Lifecycle
- Android Operating System
- Intent

Android Project Structure



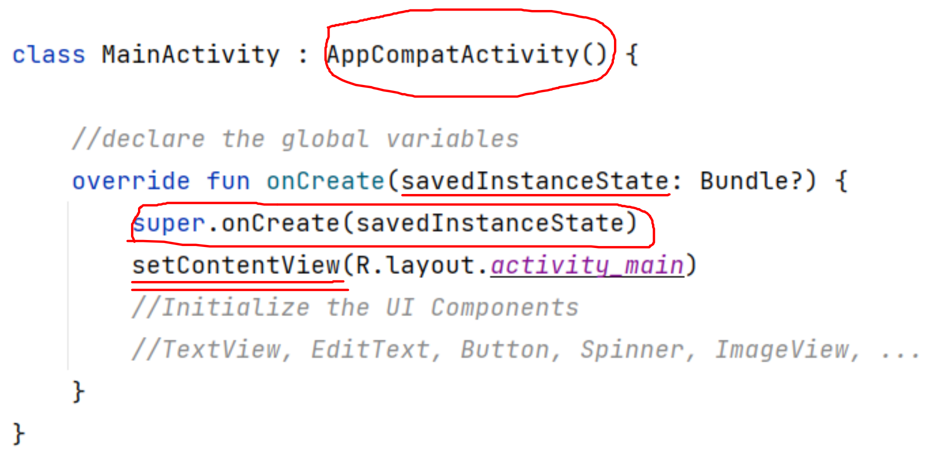
```
app
  manifests
  java
    com.example.myapplication
      AddBookActivity
      Book
      BookAdapter
      BookDetailActivity
      BookViewHolder
      MainActivity
    com.example.myapplication (androidTest)
    com.example.myapplication (test)
  res
    drawable
      ic_launcher_background.xml
      ic_launcher_foreground.xml (v24)
    layout
      activity_add_book.xml
      activity_book_detail.xml
      activity_main.xml
      item_book.xml
    mipmap
      ic_launcher (6)
      ic_launcher_round (6)
    values
      colors.xml
      strings.xml
      themes (2)
    xml
  res (generated)
Gradle Scripts
  build.gradle (Project: My_Application)
  build.gradle (Module :app)
  proguard-rules.pro (ProGuard Rules for ":app")
  gradle.properties (Project Properties)
  gradle-wrapper.properties (Gradle Version)
  local.properties (SDK Location)
  settings.gradle (Project Settings)
```

Activity Layouts



ConstraintLayout
LinearLayout (horizontal)
LinearLayout (vertical)
FrameLayout
TableLayout
TableRow
Space

Activity Kotlin



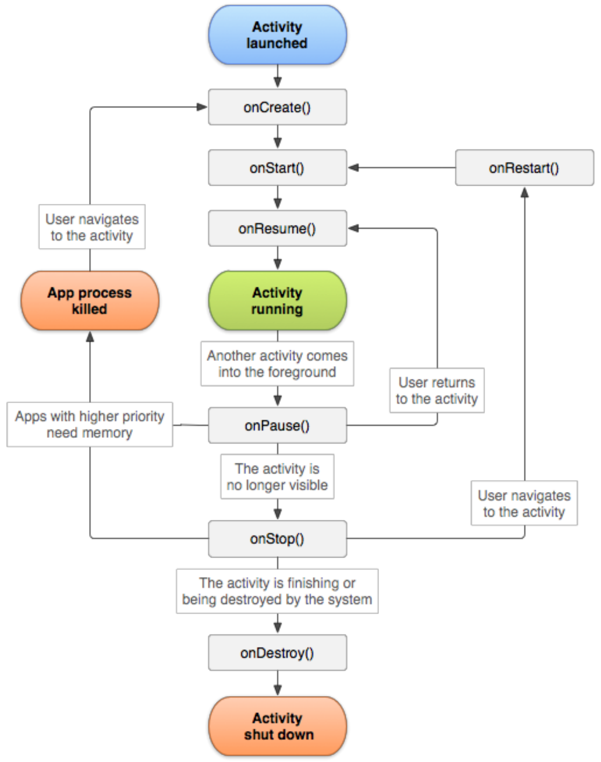
```kotlin
class MainActivity : AppCompatActivity() {

    //declare the global variables
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        //Initialize the UI Components
        //TextView, EditText, Button, Spinner, ImageView, ...
    }
}
```

Activity Lifecycle

- onCreate():
  - Called when the activity is first created.
  - This is where you typically perform one-time initializations, like setting up the UI with setContentView() or initializing member variables.
- onStart():
  - Called when the activity becomes visible to the user but isn't yet in the foreground.
  - The activity is now visible but not yet interactive.
- onResume():
  - Called just before the activity starts interacting with the user.
  - At this point, the activity is in the foreground and user can interact with it.

Activity Lifecycle

- onPause():
  - Called when the system is about to put the activity into the background or when another activity becomes semi-transparent on top of the current activity.
  - Typically, you'd use this to pause ongoing tasks or save state changes.
- onStop():
  - Called when the activity is no longer visible to the user.
  - You'd typically use this to release resources that aren't needed when the user isn't interacting with the activity.
- onRestart():
  - Called after onStop() and before onStart() when the activity is being brought back to the foreground.
  - This is where you'd typically undo changes or perform tasks you did in onStop().
- onDestroy():
  - Called just before the activity is destroyed, either because the system is temporarily destroying it to save resources or because it's finishing due to a call to finish().
  - This is your last chance to release all resources and save state.


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

Android Operating System

1. **Linux Kernel**:
   1. At the base of the Android OS is the Linux kernel. It's responsible for device drivers, resource access, power management, and process management. Android uses the Linux kernel but with a different set of system calls.
2. **Libraries**:
   1. These are a set of C/C++ libraries used by various components of the Android system.
   2. Some of the notable ones include the Surface Manager (for drawing on the screen), Media Framework (for audio and video playback), SQLite (for database operations), and WebKit (for web browsing functionality).
3. **Android Runtime**:
   1. This includes the Dalvik Virtual Machine (in older Android versions) or ART (Android Runtime) in newer versions. They allow apps to run as separate processes with their instance of the Dalvik VM.
   2. There's also a core set of libraries that provide Java functionality.
4. **Application Framework**:
   1. This layer provides higher-level services to applications in the form of Java classes. It's the framework that Android developers interact with when creating applications.
   2. Some of the essential parts of the application framework include the Activity Manager (managing application lifecycle), Content Providers (data exchange between apps), and the Telephony Manager (managing all voice calls).

Android Operating System

5.Applications:

1. This is where all the applications reside. Android comes with a set of core applications like the dialer, web browser, contact manager, etc.
2. All applications, including system and third-party apps, are built on the same application framework and have equal access to the core APIs.

6.Hardware Abstraction Layer (HAL):

1. This is a layer of libraries that offers a standard interface to Android's hardware functionalities, making it easier for manufacturers to provide device-specific implementations. For instance, regardless of the hardware specifics, Android sees the same camera interface, allowing developers to write camera applications that work across a multitude of devices.

7.System Apps and User Apps:

1. System Apps are those that come pre-installed with the device's OS. They are present in the system partition and usually have more privileges than regular apps.
2. User Apps are those installed by users from the Play Store or other sources. They run in individual user accounts and are isolated from each other for security.

Intent

1.Explicit Intents:

1. These specify the component to start by name (i.e., the fully-qualified class name).
2. They're typically used for communicating within your own app. For instance, you might start a new activity within your application using an explicit intent.

2.Implicit Intents:

1. These do not name a specific component. Instead, they declare a general action to perform, which allows a component from another app to handle it.
2. For example, if you want to show a web page, you can use an implicit intent to request that another app (like a browser) show it.

Main features and uses of intents

- Starting Activities:
  - You can use intents to start a new activity by passing it to startActivity(). This could be an activity within your app (using explicit intent) or an activity in another app (using implicit intent).
- Starting Services:
  - Intents can also be used to start services using the startService() method.
- Delivering Broadcasts:
  - Intents can broadcast system-wide messages using methods like sendBroadcast(), sendOrderedBroadcast(), and sendStickyBroadcast().
- Transferring Data:
  - Intents can carry data payloads of various types within the Bundle that's attached to them. This feature is useful for passing data between components.

Main features and uses of intents

- Action, Data, and Category:
  - The "action" specifies the general action to be performed (e.g., ACTION_VIEW, ACTION_EDIT).
  - The "data" specifies the data on which the action should be performed (like a URI).
  - The "category" provides an additional way to categorize the intent (e.g., CATEGORY_LAUNCHER, CATEGORY_BROWSABLE).
- Requesting Results:
  - Intents can also be used to request a result back after an action is performed. For example, you can start an activity that takes a photo and returns the photo to the caller. This is achieved using startActivityForResult().
- Component, Flags, and Extras:
  - "Component" is used in explicit intents to specify the target component by its name.
  - "Flags" can modify the behavior of the intent, like how the new activity is launched.
  - "Extras" are key-value pairs stored in a Bundle that carry additional information.

Case study-based questions

- With the rise of remote work in 2021, XYZ Corp. is launching a video conferencing mobile app. Features include screen sharing, meeting scheduling, and real-time chat. Select the incorrect statements about this application from the list below. (select One)
  - a. The application will allow multiple participants in a single meeting.
  - b. The application doesn't require access to the device's camera and microphone.
  - c. Users will be able to schedule future meetings within the app.
  - d. The application can function as a task management tool.
  - e. All the above statements are incorrect.

Case study-based questions

- As audiobooks gain traction, BookBuddy is set to release an app for audiobook enthusiasts. Features include playback speed control, bookmarking, and offline listening. Select the incorrect statements about this application from the list below.
- Select one:
  - a. Users can print physical copies of the books from the app.
  - b. The application will allow users to control the speed of the narration.
  - c. The application will provide recommendations based on listening history.
  - d. The application supports text-to-speech functionality for all ebooks.
  - e. All the above statements are incorrect.

Case study-based questions

- The National Museum Department is launching a mobile app to provide virtual tours of historical sites. Users can view 3D models of artifacts, listen to audio guides, and read about the history of each site. Select the unsuitable feature for this application from the following list.
- Select one or more:
  - a. Activities
  - b. Services
  - c. Broadcast Receivers
  - d. WebSockets
  - e. All the mentioned features are suitable for this application

Case study-based questions

- The Ministry of Transport is unveiling a mobile app for public transportation schedules. Users can check bus and train schedules, book tickets, and receive real-time location updates. Select the unsuitable feature for this application from the following list.
- Select one or more:
  - a. SharedPreferences
  - b. SQLite Database
  - c. Toast
  - d. Content Providers
  - e. All the mentioned features are suitable for this application