

6.What is “android activity and activity life cycle” ?

අපි දන්නවා android වල සෑම තනි screen එකක් සඳහාම java file එකක් සහ xml file එකක් අවශ්‍ය බව.java file එකට අපි activity කියලා කියනවා.xml file එකට අපි layout කියලා කියනවා.කලින් අපි මෙම xml file එක ගැන කතා කරා.දැන් අපි java file එක,එනම් activity එක ගැන බලමු.

දැන් අපි බලමු activity එකක් සාදන ආකාරය.activity එකක් සෑදීමට නම් අපි class එකක් සෑදිය යුතුයි.තවද මෙම සාදන class එක Activity යන class එකේ sub class එකක් ලෙස සෑදිය යුතුයි.අපි හිතමු activity එකේ නම MainActivity කියලා එවිට එය code වී තිබිය යුත්තේ public class MainActivity extends Activity {} යනුවෙනි. මෙසේ activity එකක් වෙනුවෙන් සාදන class එක activity class එකෙන් inherit කර ගත යුත්තේ ඇයි? මෙය තේරුම් ගැනීමට නම් අපි activity life cycle එක දැනගෙන සිටිය යුතුයි.

activity life cycle

android activity එකකට අවස්ථා(states) 4ක් පවතී.ඒවා පහත පරිදි වේ.

- activity does not exist (Destroyed)
- Foreground (running)
- Background (Stoped)
- Paused

තවද activity එකක් සතුව මූලික methods 7ක් පවතී.එම methods වලින් කීපයක් යම් කිසි sequence එකකට අනුව call කල විට එක් state එකක සිට තවත් state එකක් වෙත activity එක ගෙන යා හැකියි.එම ප්‍රධාන methods 7 පහත පරිදි වේ.

- onCreate() ;
- onStart() ;
- onResume() ;
- onPause() ;
- onStop() ;
- onRestart() ;
- onDestroy() ;

අපි උදාහරණයක් ආශ්‍රයෙන් බලමු activity කීපයක් තියෙන app එකක් ගෙන යම් කිසි activity එකක ඉහත කියන ලද states මොනවාද කියලා.

app එක start කරීමට ප්‍රථම එනම් අපි එහි icon එක click කිරීමට ප්‍රථම එය සිටින්නේ activity does not exist අවස්ථාවේ වේ.

අපි icon එක click කල විට මුලින්ම call වන methods තුනක් ඇත.එම methods 3න call වී සිදු වුනාත් පසුවයි app එක running state එකට එන්නේ.ඒවාත් call වන්නේ යම් පිළිවෙලකටයි.එනම් මුලින්ම call වන method එක වන්නේ onCreate() කියන method එකයි.ඉන් පසුව onStart() හා onResume() යන methods දෙක පිළිවෙලින් call වී app එක start වේ.මෙහිදී මුලින්ම screen එකට එන්නේ app එකේ ප්‍රදාන activity එකයි (Manifest.xml එකේ කියලා තියෙනවා කොයි activity එකද මුලින්ම load වෙන්න ඕන කියලා app එක start කල විට)

උදාහරණයක් ලෙස අපි මෙම app එක යනු අපේ contacts පෙන්නන app එක ලෙස සිතමු.එවිට මුලින්ම එය click කල විට එන්නේ contacts list එකයි.එයයි එම app එකේ ප්‍රථමයෙන්ම load වන screen එක.එනම් ප්‍රථම activity එක.අපේ පහසුව තකා මෙම activity එක අපි A නමින් නම් කර ගමු.ඉන් පසු අපි මෙම contacts list එකෙන් යම් contact එකක් උඩ click කළායැයි සිතමු.

එවිට එම app එකේ අලුත් activity එකක් load වෙනවා screen එක මතට.එනම් අපි click කල contact එකට අදාළ විස්තර සහිත අලුත් activity එකක් එනවා screen එක මතට.මෙම අලුත් activity එකට අපි පහසුව තබා B ලෙස කියමු.මෙවිට A activity එක background එකට එහෙම නැත්නම් stoped state එකට පැමිණේ.එසේ යම් activity එකක් running අවස්ථාවේ සිට stoped අවස්ථාවට යාමටනම් methods දෙකකට call කල යුතුයි.ඉන් පසුවයි එයට running අවස්ථාවේ සිට stoped අවස්ථාවට යා හැක්කේ.එම methods දෙක නම් onPause() හා onStop() යන්න වේ.so නිකන්ම onStop() method එක call කරලා මදි running අවතාවේ සිට stoped අවස්ථාවට යම් activity එකකට යාම සඳහා.පිළිවෙලින් onPause() හා ඉන් පසුව onStop() යන methods දෙක call කල යුතුයි.

“A” activity එක background එකට යද්දි “B” activity එක foreground එකට එහෙම නැත්නම් running අවස්ථාවට පැමිණේ.එය පැමිණෙන්නේද ඉහත මුලින් කියන ලද methods තුන call වුනාට පසුවය.එනම් අපි contacts list එකේ එම contact එක click කල විට ඉහත කියන ලද methods තුන පිළිවෙලට run වෙන්න කියලා කියන්න ඕන code එකේ හරි නම්.

android වලදී activities ටික stack එකක store වේ.එනම් stack data structure එකකයි activities store වන්නේ.එම නිසයි අපි phone එකේ back button එක ඔබපුහම කලින් activities වලට පිළිවෙලින් යන්න පුළුවන්.stack යනු රාක්කයක පොත් ගොඩක් වැනිය.අලුත් පොතක් නියන්නේ අපි තවත් පොතක් උඩින්ය.එවිට කලින් උඩින් තිබූ පොත එකක් යටට යනු ලබයි.බැරි වෙලාවත් යට නියෙන පොතක් ගැනීමට අවශ්‍ය නම් ඊට උඩින් නියෙන පොත් අයින් කරලයි ගැනීමට සිදු වන්නේ.activities ද මෙසේම වේ.අපේ උදාහරණයේ නම්,දැන් B activity එක උඩින් ඇත.A activity එක එයට පහලින් ඇත stack එකේ.අපි දැන් සිතමු මේ මොහොතේදී අපි phone එකේ back button එක press කළා කියලා.එවිට උඩින් නියෙන B activity එක destroy වී A activity එක උඩට එනවා.එනම් A activity එක stoped අවස්ථාවේ සිට running අවස්ථාවටත්,B activity එක running අවස්ථාවේ සිට destroyed අවස්ථාවටත් යනවා.දැන් බලමු යම් activity එකක් stoped (එහෙම නැත්නම් background) අවස්ථාවේ සිට running අවස්ථාවට ගෙන ඒමට call කල යුතු methods මොනවාද කියල හා app එකක් running අවස්ථාවේ සිට destroy කිරීමටත් call කල යුතු methods මොනවාද කියල.

A activity එක background එකේ සිට නැවත running අවස්ථාවට පැමිණියා.එසේ වීමට නම් අපි methods 3ක් call කල යුතුයි.ඒවා නම් onRestart(), onStart(), onResume() යන methods වේ.

B activity එක running අවස්ථාවේ සිට destroy වුනා.එසේ වීමට methods 3ක් call කල යුතුයි.ඒවා නම් onPause(), onStop(), onDestroy() යන ඒවා වේ.

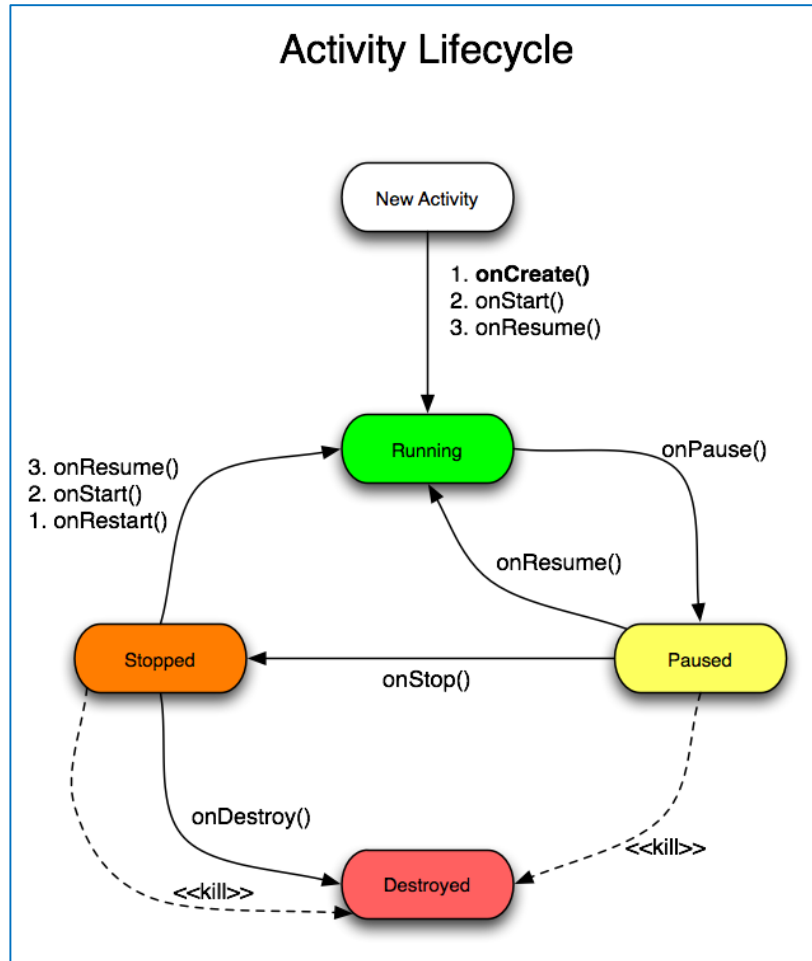
දැන් screen එකේ load වෙලා නියෙන්නේ “A” activity එකයි.එනම් contacts list එකයි.අපි හිතමු මෙම අවස්ථාවේදී call එකක් පැමිණියා,එහෙම නැත්නම් phone එකේ power button එක press කළා කියල.එවිට A activity එක paused අවස්ථාවට පත් වේ.

paused හා stoped අවස්ථා දෙක අතරේ වෙනස නම්,යම් activity එකක් pause වුනොත් ආපහු run කල විට, pause කරද්දී කර කර හිටපු තැන ඉඳන් කරගෙන ගිය හැකියි.නමුත් stop කර නැවත run කරද්දී මුල ඉඳන් කරන් යා යුතුයි එම activity එකේ අදාළ ක්‍රියාවලීන්.

යම් activity එකක් run අවස්ථාවේ සිට pause අවස්ථාවට පත් වීමට නම් එක method එකක් call කල යුතුයි.එම method එක නම් onPause() යන්නයි.අපි හිතමු call එක පැමිණ අවසන් කියලා.එහම නැත්නම් නැවතත් power button එක press කළා කියලා.එවිට activity එක නැවත running අවස්ථාවට පත් වේ.ඒ සඳහා call කල යුතු method එක නම් onResume() යන්නයි.

So මෙමගින් පැහැදිලි වන තවත් කරුණක් නම් යම් activity එකක් run අවතාවේ සිට stop අවස්ථාවට යාමේදී එය pause අවස්ථාව පසු කරගෙනයි යන්නේ.මොකද stop වන විට මුලින්ම call කරන method එක වන්නේ onPause() යන්නයි.ඉන් පසුවයි onStop() කියන method එකට call කරන්නේ.

ඉහත විස්තර කර ඇති activity life cycle එක හා ඊට අදාළ methods පිළිබඳව පැහැදිලි අවබෝදයක් පහත රූප සටහන අද්‍යයය කිරීමෙන් ලබා ගත හැක.



so දැන් මෙය අනුව අපිට තේරෙනවා එක activity එකක් සඳහා වුවද විවිද අවස්ථා තියෙනවා කියලා. යම් යම් දේවල් සිදු වන විට, අපි එම activity එක අදාළ state එකට පත් කල යුතුයි. එය ඉතාම අපහසු කටයුත්තක් වේ. අපි මෙම activity life cycle එක ගැන කතා කරන්න පටන් ගත්තේ, ඇයි අපි activity එකක් සඳහා හඳුනා class එක Activity කියන class එකේ sub class එකක් ලෙස implement කරන්නේ යන ප්‍රශ්නය අසමින්ය.

එයට පිළිතුරු තමා අපි එසේ inherit කර ගත්තේ නැත්නම් Activity class එකෙන්, ඉහත විස්තර කරන ලද සියලු දේවල් අපිට manage කිරීම සඳහා code ලිවීමට සිදු වෙනවා. එසේ Activity class එකේ sub class එකක් ලෙස activity එක සඳහා හඳුනා class එක සෑදීමෙන්, ඉහත දේවල් ගැන වැඩිය නොයිතා අපිට code කරන්න යාමට හැකි වෙනවා.

එම නිසා activity එකක් නිර්මාණය කිරීමේ මුල්ම step එක තමා Activity class එකේ child class එකක් ලෙස class එකක් නිපදවීම. e.g : `public class MainActivity extends Activity {}`.

දැන් අපි class එකක් හැදුව. එහෙනම් ඊළඟ step එක තමා method එකක් සෑදීම. අපි දන්නවා c, java වැනි language වල මුලින්ම ක්‍රියාත්මක වන method එක තමා main() method එක. මෙලසම සෑම activity එකක් සඳහාමත් මුලින්ම ක්‍රියාත්මක වන method එකක් ඇත. කලින්ත් එය ගැන සඳහන් කළා. එම method එක තමා `onCreate()` යන method එක. මෙම method එක අපි implement කල යුතුයි සාදා ගත් class එක තුල.

so දෙවන step එක වන්නේ `onCreate` method එක සෑදීමයි. මෙයට සාමන්යයෙන් protected access modifier එක තමා අපි දෙන්නේ. e.g : `protected void onCreate() {}`

ඊළග step එක තමා අපි මෙම method එක තුළ කිවිය යුතුයි layout folder එක තුළ screen එකේ display විය යුතු දේවල් මොනවාද හා ඒවා කෙසේ display විය යුතුද කියලා xml file එකක code කරලා තියෙනවා, එය අරගෙන display කරන්න කියලා. layout folder එක තියෙන්නේ res folder එක තුළයි. එනම් layout folder එකේ තිබෙන xml file එක resource එකක් වේ. එසේ නම් අපිට එම කරුණ භාවිතා කර එය access කළ හැක.

R.layout.activity_main ලෙස අපිට එම xml file එක access කළ හැක (අපි කලින් දැක්කා layout folder එක තුළ xml file දෙකක් තියෙනවා කියලා. එනම් activity_main හා content_main ලෙස. මෙයින් content_main යන xml file එකෙයි තියෙන්නේ ui components හා ui groups ගැන විස්තර. නමුත් අපි java code එකේදී call කළ යුත්තේ මෙයට නෙමේ අනික් file එකටයි. එවිට මෙම content_main.xml file එකේ තියෙන ආකාරයට තමා screen එක display වන්නේ).

R.layout.activity_main මගින් කරන්නේ අදාළ xml file එක තියෙන location එක access කරන එක පමණි. අපි කියන්න ඕන එම location එකේ තියෙන file එකේ කියලා තියෙන විදියට එම ui components ගත්ත හා ඒ අනුව සකසන්න කියලා. ඒ සඳහා අපිට setContentView() යන method එක භාවිතා කළ හැක. so අපිට onCreate කියන method එක තුළ පහත පරිදි code කළ යුතුයි xml file එකේ තියෙන විදියට screen එක සැකසෙන්න කියලා.

`setContentView(R.layout.activity_main);` යන ආකාරයටයි.

so activity එකක් සෑදීම සඳහා steps ටික සාරාංශයක් වශයෙන් පහත පරිදි වේ.

1. Activity class එකේ child class එකක් වන පරිදි public class එකක් සාදන්න.
2. එය තුළ protected void ලක්ෂණ වලින් යුත් onCreate() method එක සාදන්න.
3. එම method එක තුළට setContentView(R.layout.activity_main); මගින් xml layout file එක screen එකේ set කරන්නේ.

උදා: android studio වලින්,

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

මෙහි AppCompatActivity යනුද Activity class එකේ child class එකකි. so මෙහිදී එම child class එකෙන් inherit කළා යනු අවසානයේ Activity class එකෙන් inherit කළා වගේම තමා විශේෂත්වය වන්නේ මෙම class එකේ implement කරලා තියෙනවනම් තව අලුත් methods එහෙම, ඒවත් දැන් මෙම MainActivity යන class එක මගින් inherit කරගන්න එකයි.