

Android

Retrospektiv vecka 1

Context

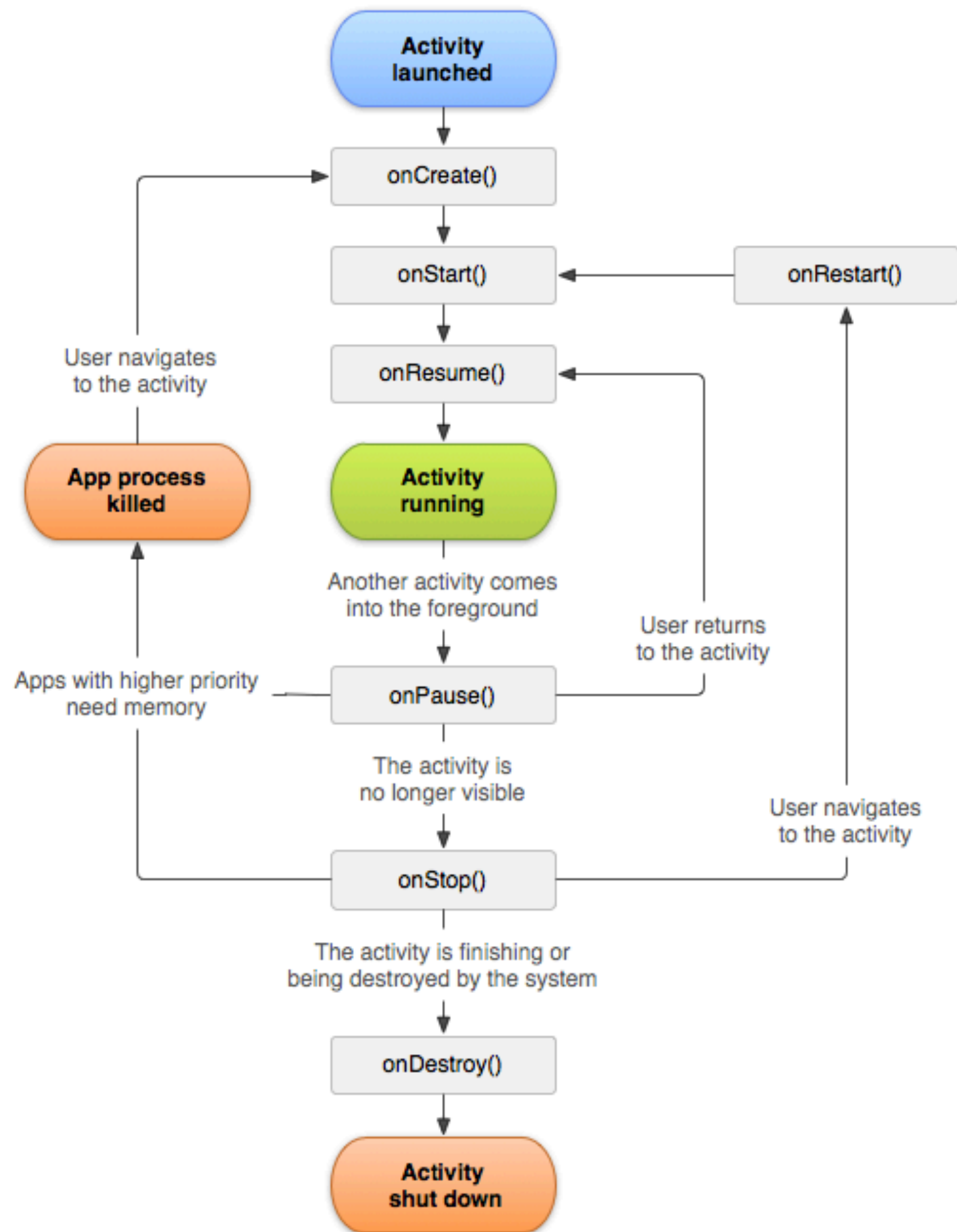
- Klass i Android som är ParentClass till vår Application, Activity och services.
- Vårt sätt att nå systeminformation och system-services
- Activity ärver från Context, därför kan vi använda “this” i vår aktivitet när ett objekt kräver ett Context.

Android manifest

- AndroidManifest.xml
- Namnger vårt Java-package name
- Beskriver de komponenter som finns i vår app, aktiviteter, services, broadcast receivers och content-providers.
- Deklarerar de permissions som vår app kräver
- Minimum Api-version för Android

Activity

- Generellt sett en skärm, kan finnas många i en app
- Huvudsyfte är att interagera med användaren
- Olika states
- Livscykel-callbacks



Explicit intent

- Vårt sätt att starta en ny aktivitet med ett visst namn
- Går att lägga till “extras” som man vill skicka med i intenten
- En aktivitet skickar iväg en intent, Android OS söker genom efter aktiviteter som kan hantera den, och om den hittar en aktivitet, så startas den
- Startas med hjälp utav `Context.startActivity()` eller `Context.startActivityForResult()`

Resultat från startad aktivitet

- Använd `startActivityForResult` istället för `startActivity`
- Sätt upp en lyssnare på `onActivityResult` för att få tillbaka resultat.
- Den aktivitet som vill returnera resultat använder sig av `setResult()`

Resources

- values/strings.xml
- menu/menu.xml
- drawable/drawable(.png|.9.png|.jpg|.gif|.xml)
- mipmap/ic_launcher.png
- layout/layout.xml
- color/color.xml
- Alla går att nås via R klassen, statisk klass som skapas vid build och innehåller alla resurser.

RecyclerView

- Använder sig av ViewHolderPattern
- Återanvändning av views
- Behöver en Adapter som har ansvaret att skapa nya views när de behövs och även binda om de som är skapade
- LayoutManager är det objekt som placerar våra vyer på rätt sätt. Finns LinearLayoutManager, GridLayoutManager och StaggeredGridLayoutManager

LayoutInflater och MenuInflater

- Tar en xml fil, och parsar den till Vyer.

Widgets

- TextView
- EditText, Glöm inte att sätta inputType
- Button

Logging och log-levels

- Log.log-level för att logga någonting, Log.d eller liknande
- Loglevels:
 - Error, Använd när ett error har skett, tillexempel i ett catch block
 - Warn, Använd när någonting shady har hänt, kanske har applikationen hamnat i ett konstigt state som kan leda till Error.
 - Info, Används för att rapportera success, tillexempel “Connection to database established”
 - Debug, Debug logg, tillexempel variabel värden eller kod-flow
 - Wtf (Enligt google, What a terrible failure...), När någonting helt hemskt har hänt. Ni vet, de där catch blocken med en kommentar i som säger: “// This should never happen”

Support library

- Implementera nya features på gamla versioner av Api:t. Eller falla tillbaka till liknande funktionalitet.
- Innehåller även klasser och bibliotek för att förenkla vår vardag som Android-utvecklare
- RecyclerView
- Dessa bibliotek och klasser är starkt rekommenderade att användas
- Xml attribut som inte stöds av en viss device ignoreras och är därför safe att använda. Betyder att det är viktigt att

Fortsättning Support

- Används när man vill lägga till en specifik feature för ett nyare Api.
- Kompatibilitet för relaterade biblioteksfunktioner.
- Generell Device kompatibilitet. AppCompatActivity istället för Activity. Detta för att ha liknande UI på alla versioner av Android och liknande features.
- Vid tillfällen där det inte finns en lösning för en viss version så har vi möjlighet att kontrollera version innan.

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {  
    ActionBar actionBar = getActionBar();  
    actionBar.setDisplayHomeAsUpEnabled(true);  
}
```