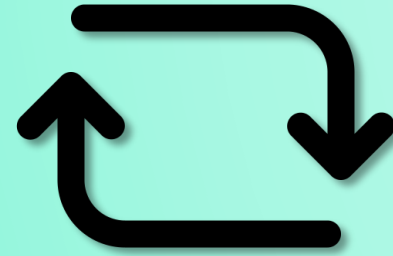


Laboratorul #3 PPA

Saptamanile: VIII – IX

Interval Orar: 17-19

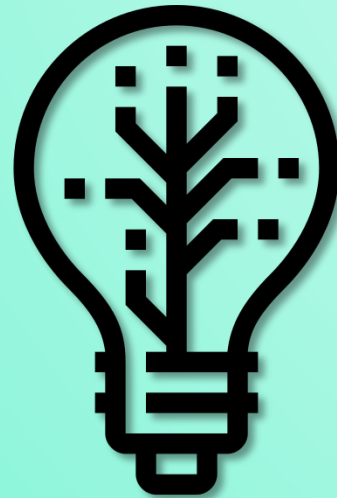
Recapitulare



- Activitatea – componenta **cruciala** a unei aplicatii Android.
- Activitatea – echivalentul “Main” din Java.
- Intentia – **mod de comunicare** intre doua Activitati/Fragmente (Intentia Explicita)

Componentele unei aplicatii

1. ~~Activitatea~~
2. Fragmentul
3. Serviciul
4. ~~Intentia~~

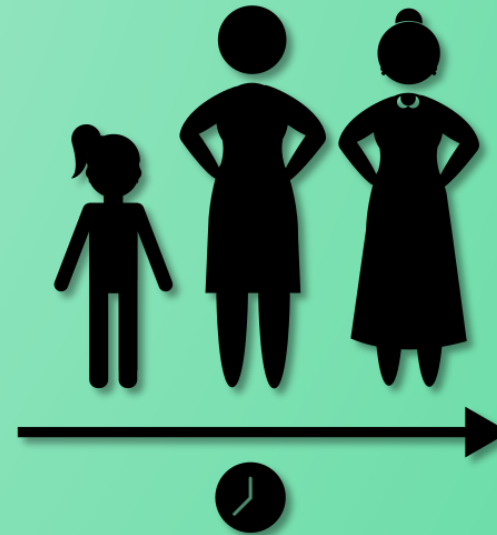
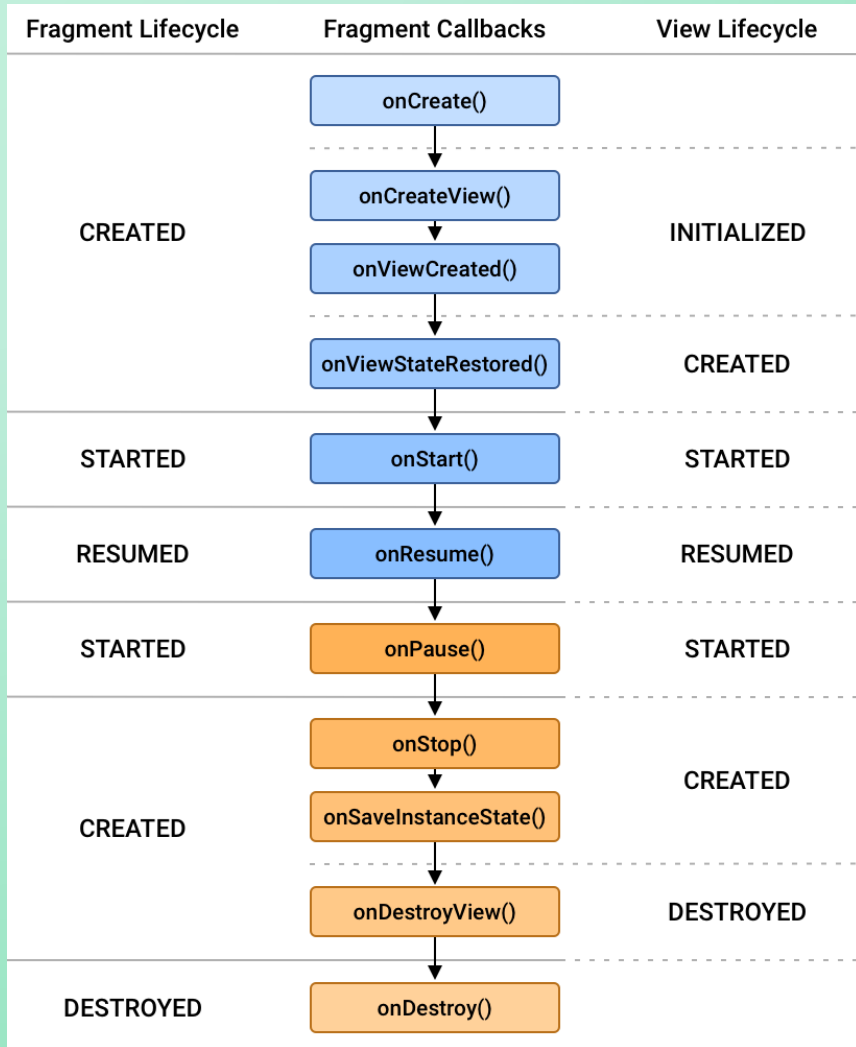


Fragmentul

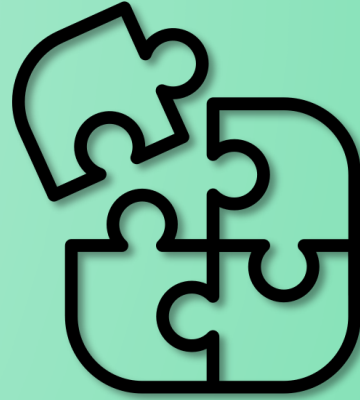


- Parte optionala lansata de o Activitate.
- **Modular si Reutilizabil**
- **Are propriul ciclu de viata**, separat de activitatea din care e lansat.
- **Daca activitatea este terminata, fragmentul nu poate supravietui singur!**

Ciclul de viata al unui Fragment

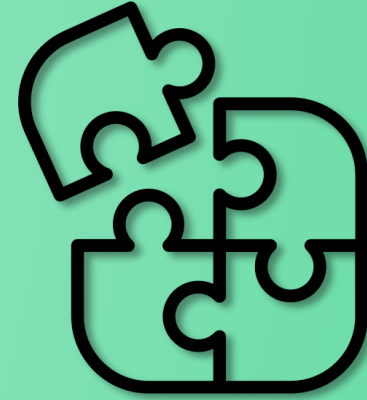


Fragmentul



- Cand Fragmentul este instantiat starea initiala a View-ului este **INITIALIZED**.
- Pentru a se putea trece prin restul ciclului de viata se va adauga un **FragmentManager**.

FragmentManager

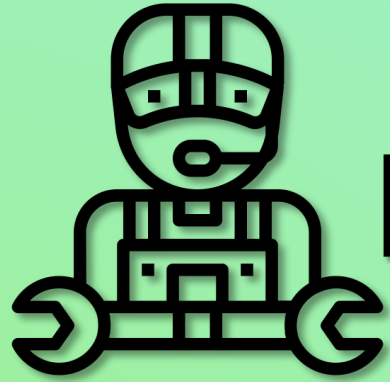


- Este ca un tutore pentru obiectul de tip Fragment.
- **Gestioneaza ciclul de viata** al Fragmentului
- Responsabil pentru **atasarea/detasarea** Fragmentului de Activitate.

Serviciul



- Componenta care ruleaza in **fundal**
- Nu are o **interfata grafica**
- Se pot implementa **executii** la intervale de **timp regulate**
- Thread **separat** fata de **activitate/fragment**



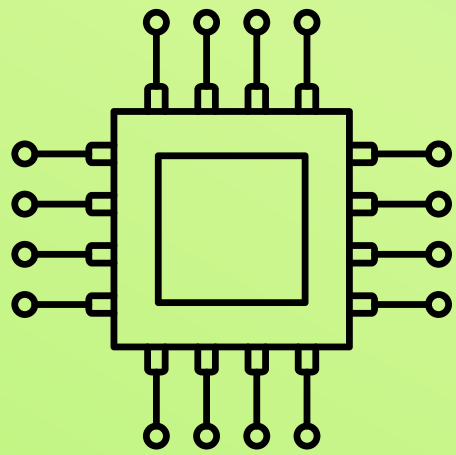
Foreground Service

- Se gaseste in “**prim-planul**” aplicatiei
- **Trebuie** sa fie vizibil utilizatorului (de obicei prin bara de stare/status bar)
- Notificarea de functionare din bara de stare **NU** va disparea decat daca **serviciul** este **inchis**.
- Ex: Spotify cand vrem sa schimbam melodia din bara de notificari a telefonului



Background Service

- Se gaseste in “fundalul” aplicatiei
- **Poate sa nu fie vizibil pentru utilizator**
- Chiar daca se numesc “Background”/“Foreground” amandoua ruleaza in **fundal** pe un **Thread/Fir separat**.
- Ex: Alarma de la telefon va suna la ora stabilita chiar daca utilizatorul a inchis aplicatia.



Introducere in Multithreading Java

- Multithreading – ruleaza pe mai multe fire/thread-uri de executie
- Poti face mai multe lucruri in acelasi timp
(Ex: merg si vorbesc in acelasi timp)
- In Java putem lucra multithreaded prin implementarea interfetei Runnable.



Aplicatie

- Sa se creeze o aplicatia avand urmatoarele:
 - 0 activitate cu doua fragmente.
 - Primul fragment va avea:
 - Un Serviciu care ne trimite un Toast la fiecare 10 secunde cu mesajul "Laborator 3"
 - Un Buton care sa ne trimita spre al doilea Fragment.
 - Un Buton care sa ne porneasca Serviciul.

Aplicatie



- Al doilea Fragment va avea:
 - Un buton care va opri Serviciul din primul Fragment.
 - Un buton care ne va intoarce la primul Fragment.
 - Un element de tip TextView in care sa afisam starea Serviciului ("Oprit"/"Pornit").



Fin.
The End.
Sfarsit.