



# PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

## WYKŁAD 1

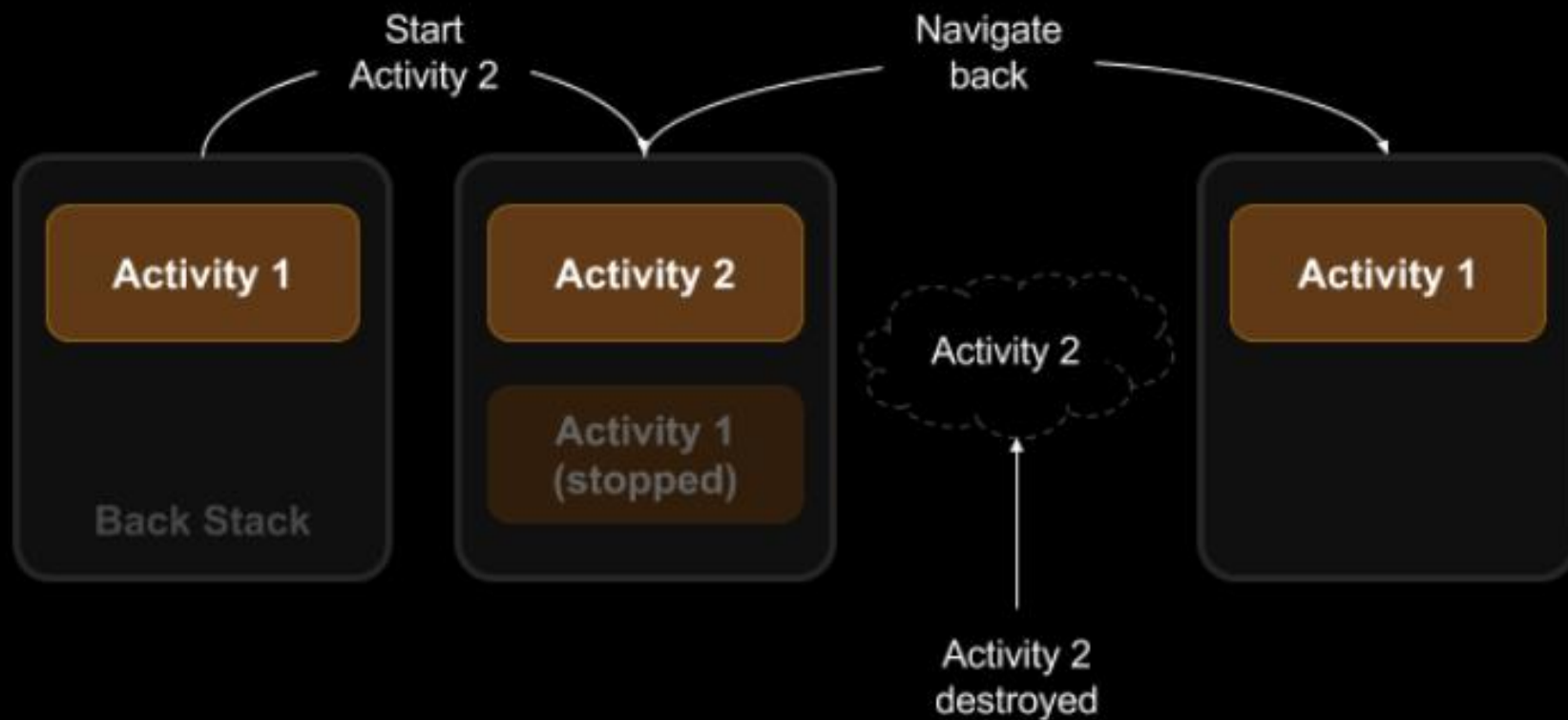
- ACTIVITY
- CYKL ŻYCIA AKTYWNOŚCI

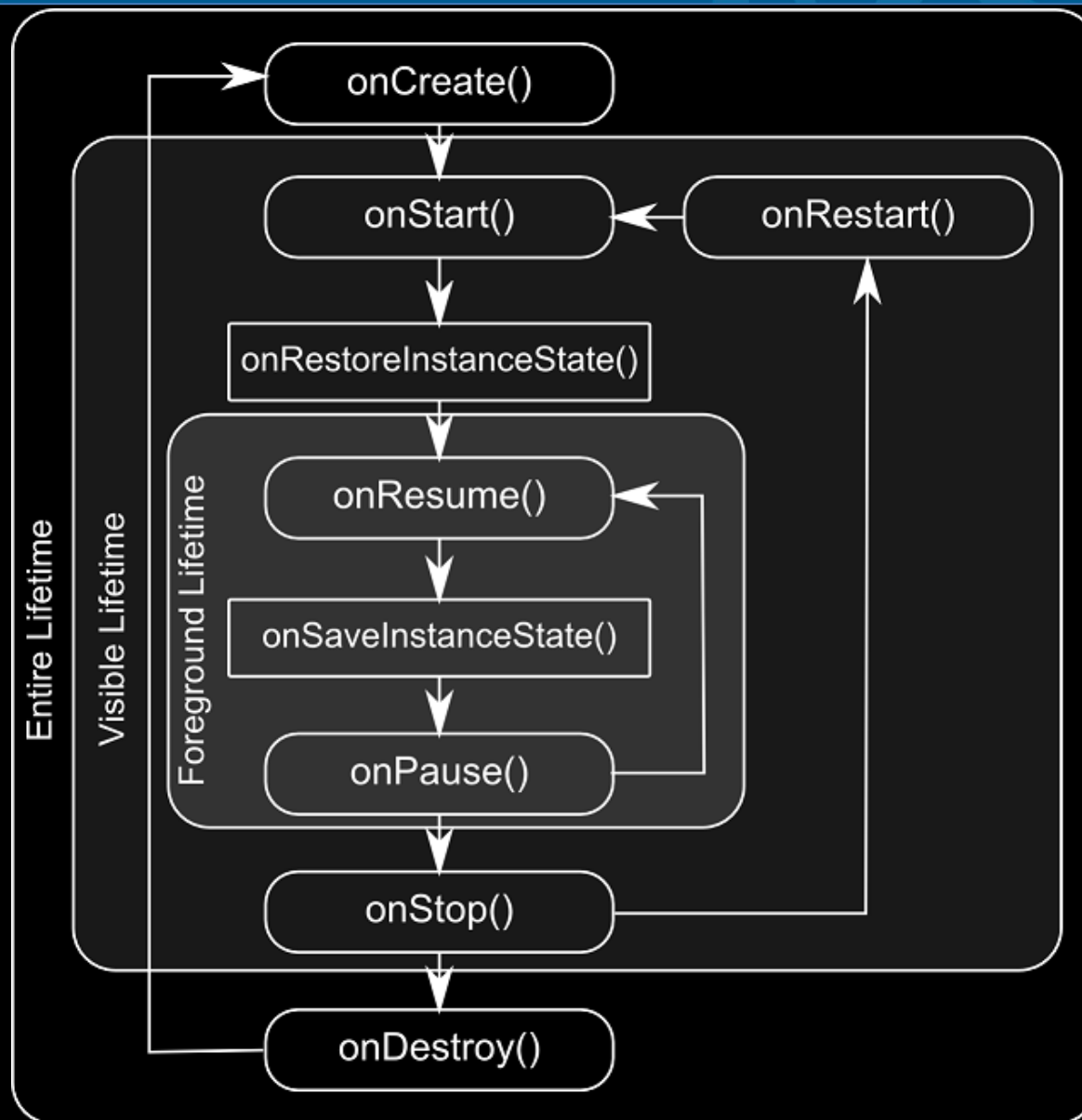
- Każda aplikacja Androida działa w swojej własnej instancji maszyny wirtualnej
- W każdym momencie kilka instancji maszyny wirtualnej może być aktywna (rzeczywista równoległość – nie *task switching*)
- Aplikacja Androida nie kontroluje całkowicie realizacji swojego cyklu życia
- OS może zakończyć każdy proces
  - Zasoby są krytycznie niskie
  - Duża liczba działających aplikacji
  - Aplikacja wymagająca bardzo dużych zasobów (energia, pamięć)

***Start***

**Life as an Android Application:  
Active / Inactive  
Visible / Invisible**

***End***





## 1. Active/Running

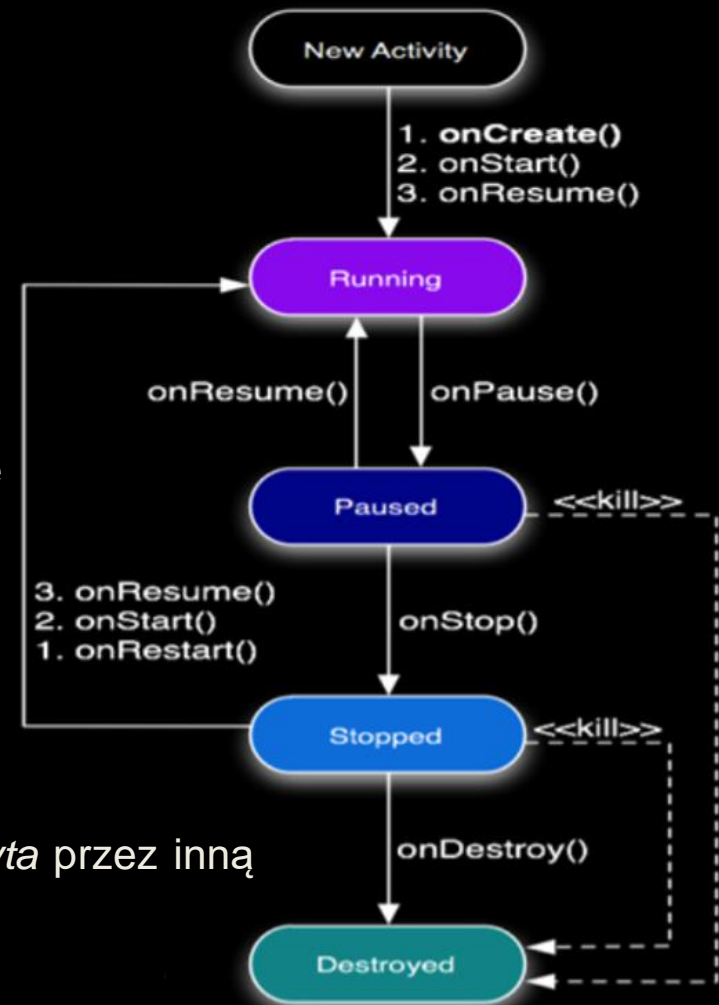
- Aplikacja jest na pierwszym planie
- Znajduje się na szczycie stosu

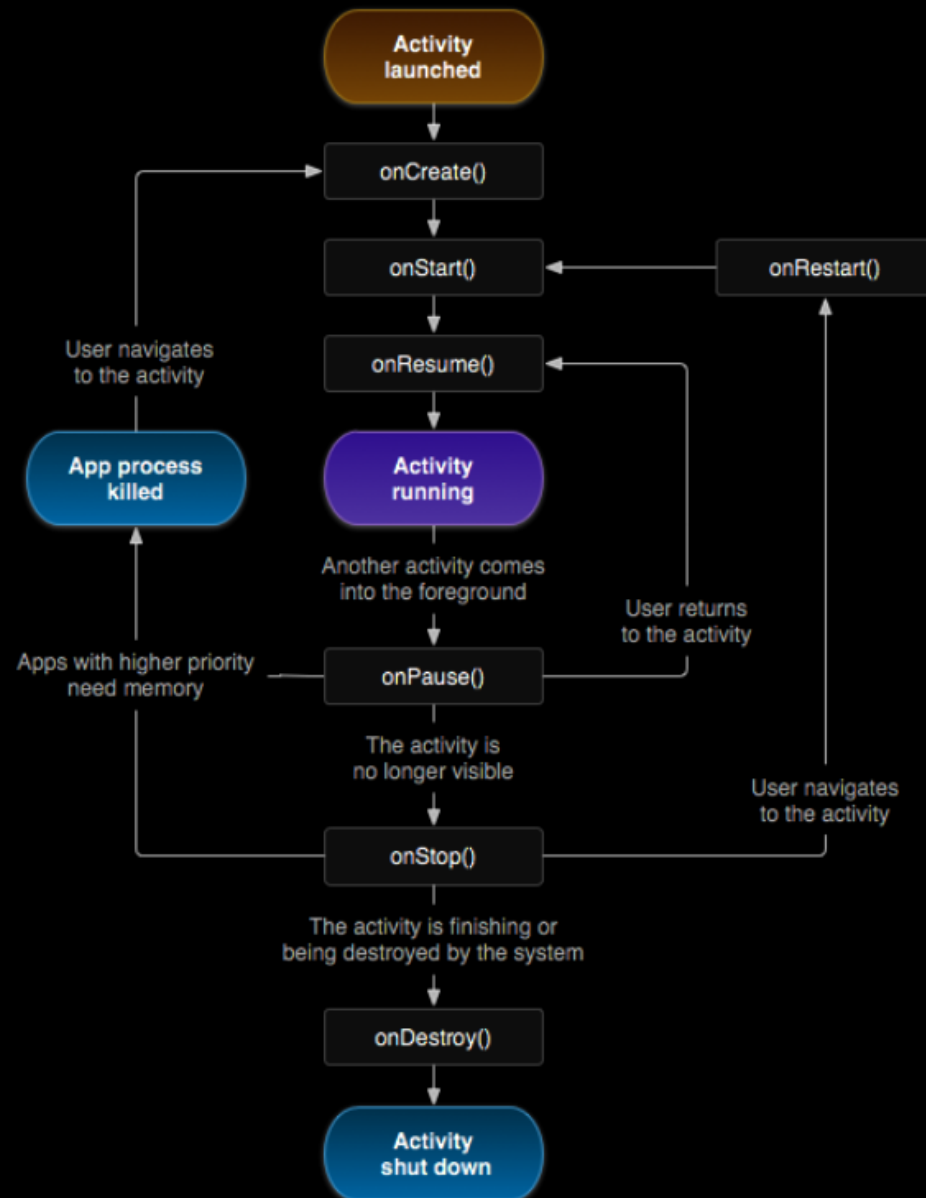
## 2. Paused

- Ta aktywność dalej pozostaje widoczna dla użytkownika
- Aktywność pozostająca w tym stanie zachowuje swój stan informacji
- Może zostać zabita przez OS
- Kontynuuje *update* UI itp

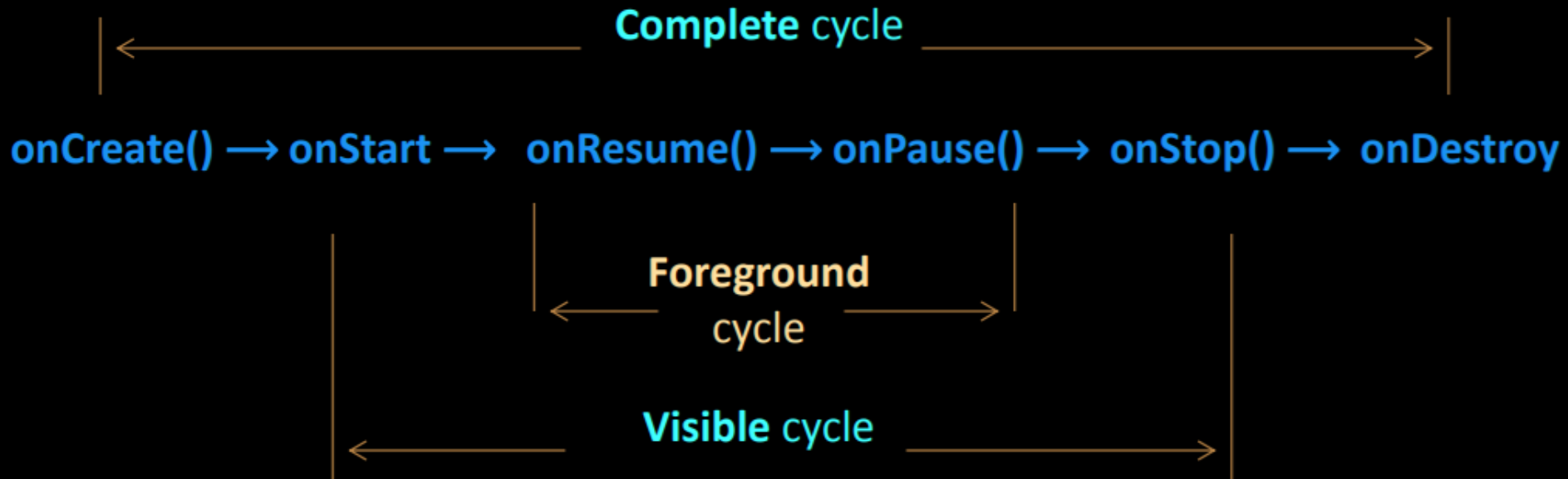
## 3. Stopped

- Aktywność zatrzymana jest całkowicie *przykryta* przez inną aktywność
- Nie jest widoczna dla użytkownika
- *Update* UI itp zostaje zatrzymany





# Cykl Życia – Foreground



- Aplikacja nie musi implementować wszystkich metod przejścia
- Są jednak metody wymagane, oraz wysoko rekomendowane
- Wymagane:
  - `onCreate()` – musi zostać zaimplementowana przez każdą aktywność – początkowe ustawienia
  - Ta metoda jest wywoływana tylko raz podczas jednego cyklu
- Rekomendowane
  - `onPause()` – powinna zostać zaimplementowana gdy aplikacja posiada dane które chcemy zachować



- Pierwsza wywoływana metoda gdy aktywność jest uruchamiana
- Większość kodu znajduje się w tej metodzie
- Typowo używana do zainicjowania struktur danych aplikacji, połączenia elementów UI, zdefiniowania zachowania elementów itp.
- Może otrzymać obiekt *Bundle* zawierający poprzedni stan aktywności
- Metoda poprzedzająca onStart()

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    // The activity is being created.  
}
```

- Wywoływana w momencie zmiany aktywności
- Używana do zapisania danych i zatrzymania pracy
- Metoda poprzedzająca
  - onResume() – gdy aktywność wraca na pierwszy plan
  - onStop() – gdy aktywność przestaje być widoczna dla użytkownika
- Spauzowana aktywność może zostać zabita przez system

@Override

```
protected void onPause() {  
    super.onPause();  
    // Another activity is taking focus  
    // this activity is about to be "paused"  
}
```

- OS może zabić apke gdy zasoby są niezbędne do operacji o wyższym znaczeniu
- Gdy aktywność osiągnie metody onPause(), onStop(), onDestroy(), może zostać zabita
- onPause() jest jedynym stanem gwarantującym ukończenie przed zabiciem

```
@Override
```

```
protected void onDestroy() {
```

```
    super.onDestroy();
```

```
    // The activity is about to be destroyed.
```

```
}
```

Zmiana konfiguracji wymaga przeładowania layoutu oraz innych zasobów gdy:

- Następuje zmiana orientacji urządzenia
- Zostaje zmieniony język
- Użytkownik wejdzie w tryb multi-window

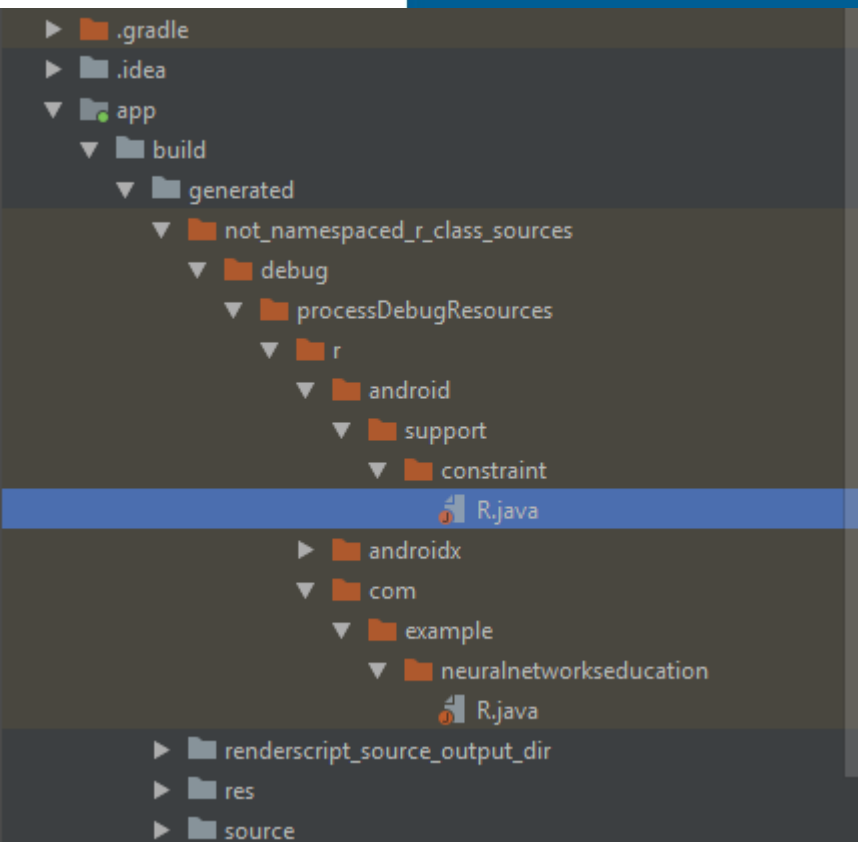
Przy zmianie konfiguracji Android:

1. Wyłącza aktywność – `onPause()`, `onStop()`, `onDestroy()`
2. Startuje nową instancję aktywności – `onCreate()`, `onStart()`, `onResume()`



Uniwersytet  
Wrocławski

# Klasa R



the build folder are generated and should not be edited.

```
3269 }
3270 public static final class id {
3271     public static final int @+id/menu_tryAtHome=0x7f0d0483;
3272     public static final int DetailFilter=0x7f0d0222;
3273     public static final int LinearLayout1=0x7f0d016a;
3274     public static final int Logout=0x7f0d0215;
3275     public static final int about_dashboard=0x7f0d0257;
3276     public static final int about_text=0x7f0d006e;
3277     public static final int about_us=0x7f0d0213;
3278     public static final int ac_home=0x7f0d043e;
```