UNIWERSYTET RADOMSKI
im. Kazimierza Pułaskiego

PROGRAMOWANIE APLIKACJI MOBILNYCH

Cykl życia aplikacji

dr Artur Bartoszewski

Cykl życia aktywności

Cyklem życia aplikacji nazywamy stany obiektu reprezentującego aplikację przez które przechodzi on od momentu uruchomienia, aż do usunięcia z pamięci operacyjnej.

Cykl życia aplikacji określa jak może się ona zachowywać i do jakich zasobów ma dostęp.

4 podstawowe stany aktywności

1. Aktywny *(ang. active*) – aktywność jest widoczna na pełnym ekranie i jest to jedyna aplikacja jaka jest obecnie otwarta
2. Zapauzowany (*ang. paused*) – użytkownik uruchomił inną aplikację, jednak nie zasłania ona całego ekranu (zapauzowana aplikacja jest nadal częściowo widoczna)
3. Zatrzymany (*ang. stopped*) – użytkownik zminimalizował aplikację lub otworzył inną aplikację, która zajmuje cały ekran.
4. Zniszczony (*ang. destroyed*) – system Android dynamicznie zarządza pamięcią. System może zniszczyć (usunąć z pamięci) aktywność, która jest w stanie zatrzymanym lub zapauzowanym (bardzo rzadko) w celu uzyskania dodatkowej pamięci.

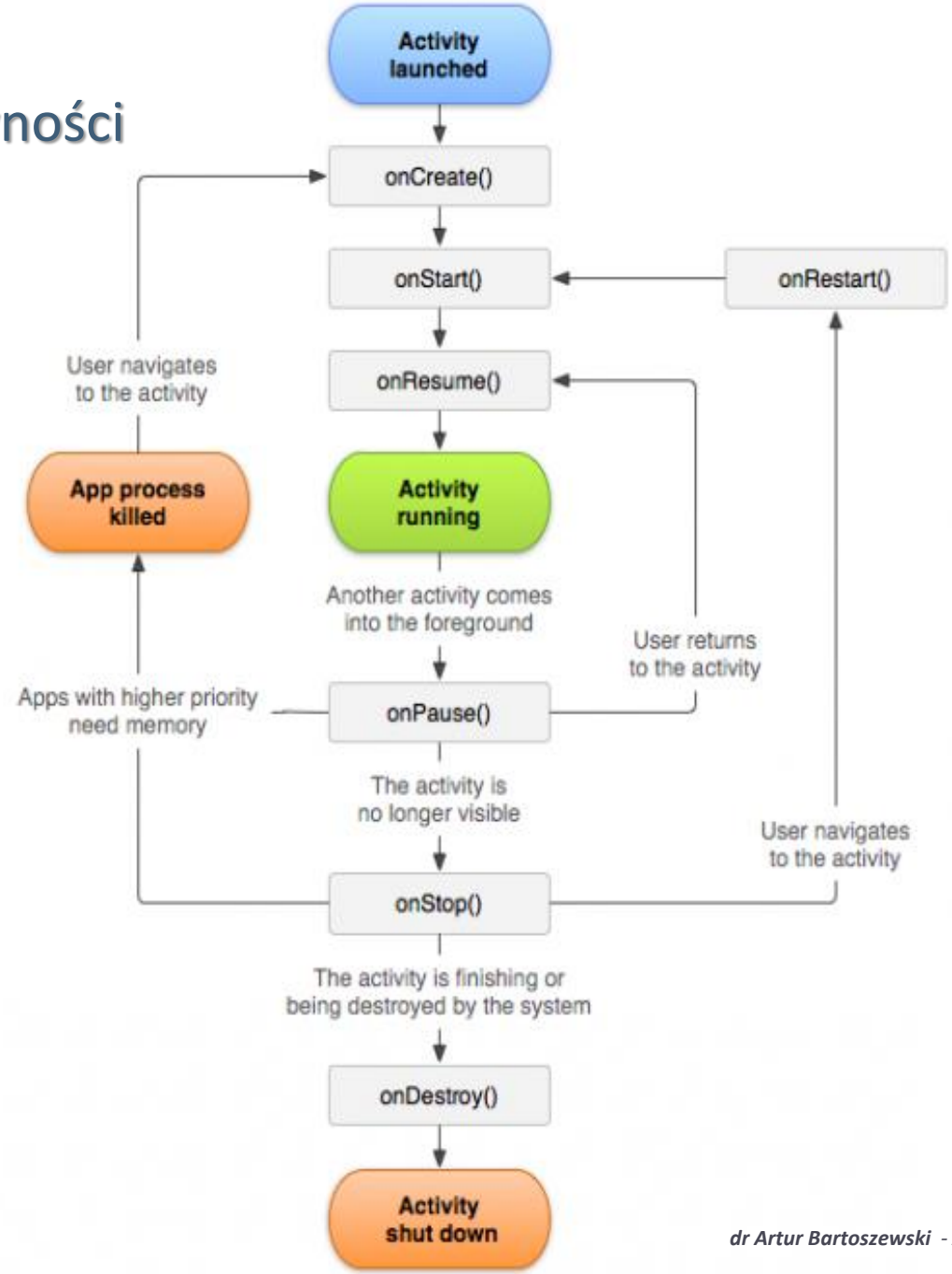
4 stany aktywności

Gdy użytkownik korzysta z aplikacji przechodzi ona przez różne stany cyklu życia. Klasa *Activity* udostępnia podstawowy zestaw siedmiu wywołań zwrotnych:

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

System wywołuje odpowiednie z nich, gdy proces wchodzi w nowy stan.

Cykl życia aktywności



Nazwa	Opis metody
onCreate()	Metoda wywoływana jest tylko raz w całym cyklu życia aktywności podczas jej uruchamiania. Powinna zawierać inicjalizowanie wszystkich widoków i zmiennych. Przekazywany jest do niej obiekt klasy Bundle zawierający zapisany stan aktywności z poprzedniego uruchomienia (jeżeli istnieje). Po niej wywoływana jest metoda onStart().
onRestart()	Wywoływana po przywróceniu aktywności na ekran - po onStop(). Po niej następuje metoda onStart()
onStart()	Wywoływana przed samym pojawieniem się aktywności na ekranie. Następuje po metodach onCreate() oraz onRestart(). Następną wywoływaną metodą jest onResume()
onResume()	Wywoływana, gdy aktywność jest już na ekranie. Może ją poprzedzać onStart() lub onPause(). W metodzie tej często umieszczane są animacje, a po niej następuje normalna praca aplikacji
onPause()	Wywoływana tuż przed uruchomieniem innej aktywności przez system. Zaleca się zapisywanie istotnych danych już w tej metodzie, gdyż system może zabić aktywność bez wywoływania metod onStop() i onDestroy(). Po tej metodzie wywołana zostanie onResume() lub onStop()
onStop()	Następuje po onPause(), gdy aplikacja nie jest już widoczna. Należy pamiętać, że metoda ta może w ogóle nie być wywołana gdy aktywność zostanie usunięta z pamięci już po onPause(). Po niej następuje onRestart() lub onDestroy()
onDestroy()	Wywoływana przed usunięciem aplikacji z pamięci. Nie można mieć pewności, że metoda ta zostanie w ogóle wywołana

Źródło: http://programowanie.opole.pl

Cykl życia aktywności

Aktywność aplikacji (od stworzenia do zamknięcia)

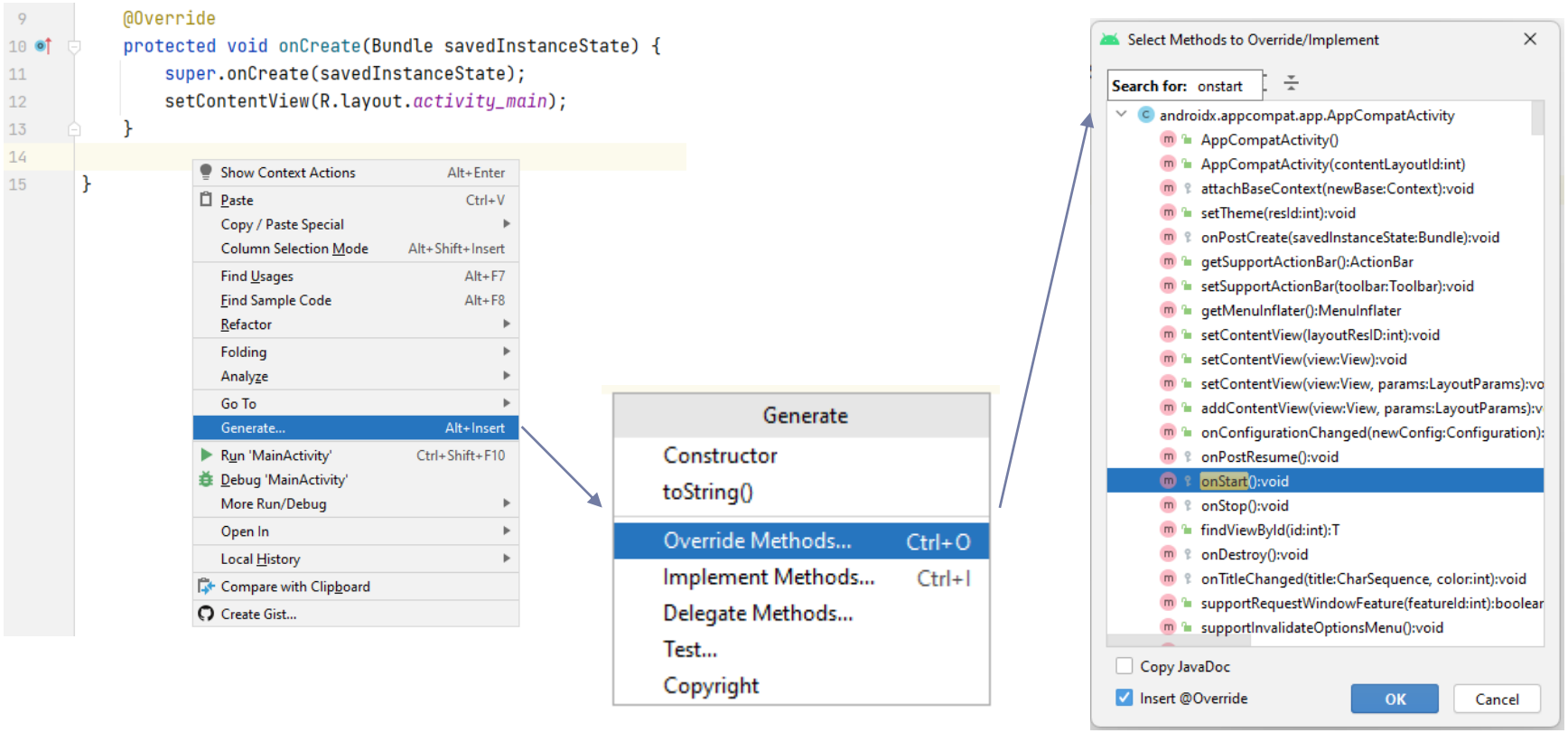
1. Aplikacja zostaje uruchomiona, czyli zostaje utworzona instancja obiektu Aktywność (system tworzy obiekt aktywności poprzez uruchomienie jego konstruktora).
2. Po uruchomieniu obiektu aktywności zostaje wywołana metoda onCreate(), w której ciele należy umieścić kod który ma się wykonać wraz ze startem aplikacji.
3. Następnym cyklem jest działanie aktywności, w trakcie kiedy jest włączona na ekranie i użytkownik może prowadzić z nią interakcję (jest to główny stan aplikacji).
4. Metoda onDestroy() w aktywności jest wywoływana tuż przed zamknięciem aplikacji (pozwala ona na zwolnienie używanych przez nią zasobów).
5. Po wykonaniu metody onDestroy() aktywność przestaje istnieć, a aplikacja zostaje zamknięta.

Cykl życia aktywności

Metody onStop(), onPause() i onResume()

1. Metoda onStop() jest uruchamiana, podczas gdy aplikacja jest zamykana.
2. Metoda onPause() istnieją trzy momenty jej wywołania:
   - jest wywoływana przed metodą onStop() w przypadku zamykania aplikacji.
   - jest wywoływana, gdy użytkownik przejdzie do innej aktywności
   - jest wywołania w momencie, w którym obracamy ekran smartfona.
3. Metoda onResume() jest wywoływana tak jak metoda onPause() w momencie, gdy obracamy ekran smartfona, lub przy powrocie do działania po wcześniejszym wywołaniu metody onPause()

Cykl życia aktywności



Klasa Activity po której dziedziczy aplikacja posiada komplet obsługi metod zdarzeń cyklu życia dlatego należy nadpisać te z nich których które chcemy samodzielnie obsłużyć

Ponieważ metod na liście jest bardzo dużo najlepiej skorzystać z wyszukiwania

Cykl życia aktywności

```
@Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //Aktywność jest tworzona
    }
    @Override
    protected void onStart() {
        super.onStart();
        //Aktywność jest przed ukazaniem się na ekranie
    }
    @Override
    protected void onResume() {
        super.onResume();
        //Aktywność jest już na ekranie
    }
    @Override
    protected void onPause() {
        super.onPause();
        //Aktywność jest przed przysłonięciem przez inną Aktywność
    }
    @Override
    protected void onStop() {
        super.onStop();
        //Aktywność nie jest już widoczna
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        //Aktywność jest przed usunięciem z pamięci
    }
```

Cykl życia aktywności

W systemie Android istnieje wiele scenariuszy, którym odpowiadają określone sekwencje przejść pomiędzy stanami aplikacji.

Przykładowo, gdy nastąpi **zmiana orientacji ekranu z pionowej na poziomą**, aktywność jest niszczona i ponownie odtwarzana. Zamykana jest aktywność związana z pionowym układem ekranu i wywoływane są dla niej zdarzenia:

```
onPause() -> onStop() -> onDestroy()
```

Następnie tworzona jest nowa aktywność (związana z poziomym układem ekranu) dla której wywołane są zdarzenia:

```
onCreate() -> onStart() -> onResume()
```

Efektem tej sekwencji zdarzeń może być reset stanu wszystkich komponentów widocznych na ekranie i utrata danych zapisanych w komponentach częściowo już wypełnionego formularza.