

Zagadnienie specjalnościowe nr 14

Cykl życia aktywności oraz podstawowe elementy architektoniczne w złożonej aplikacji mobilnej dla systemu Android

Opracował: Karol Wyskocki 208488

1. Aktywność

Aktywność jest to podstawowy element składowy aplikacji androidowych. Aktywność reprezentuje pojedynczy ekran aplikacji. W aktywnościach brak jest typowego entry-pointu w postaci metody `main()`. Reakcje na zdarzenia związane z systemem i działaniem użytkownika są oparte o tzw. `callbacki`.

Przeważnie aktywności podzielone są na dwa elementy: plik z kodem aplikacji napisanym w języku `java/kotlin` (tutaj znajduje się logika aplikacji) oraz pliku `xml` – pliku `layout`. Opisuje on wygląd aplikacji.

Cykl życia przykładowej aktywności przedstawia Rysunek 1. Poniżej zostały krótko opisane `callbacki` pokazane na schemacie.

OnCreate -> logika która wykonuje się tylko raz na starcie aktywności. Inicjalizacja elementów interfejsu itd.

OnStart -> Jest to moment, w którym aktywność staje się widoczna dla użytkownika

OnResume -> Aplikacja dostaje „Focus” i sobie działa

OnPause -> informuje, że aplikacja straciła „Focus”. Dzieje się tak, gdy wyłączymy aplikację, pojawi się inne okienko na pierwszym planie czy mamy tryb multiokienkowy

OnStop -> uruchamia się, gdy aktywność nie jest już widoczna. Np. została przykryta inną aktywnością. Tutaj możemy wyłączać niepotrzebne rzeczy.

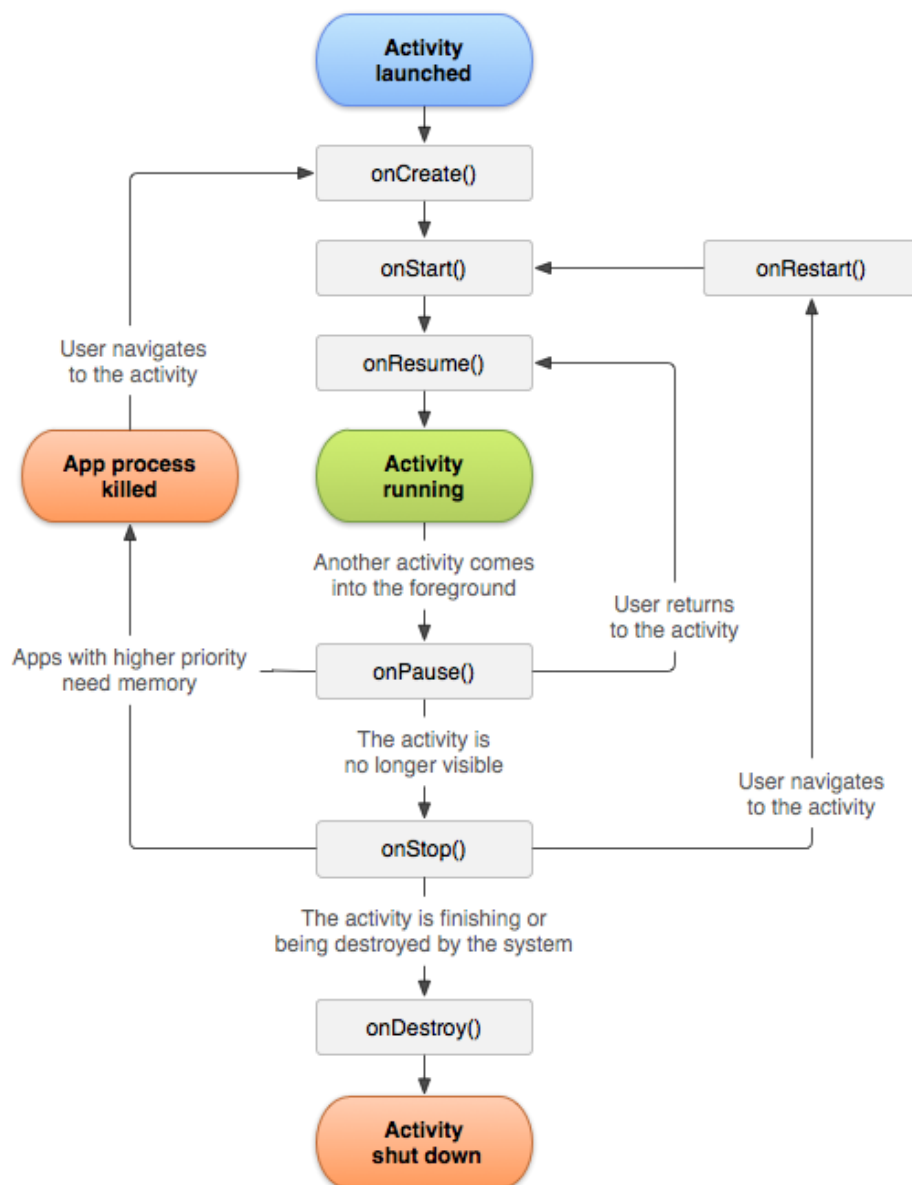
OnDestroy -> wywołane przed zniszczeniem aktywności. Służy do zwalniania zasobów, których nie zwolniliśmy wcześniej.

2. Intencje

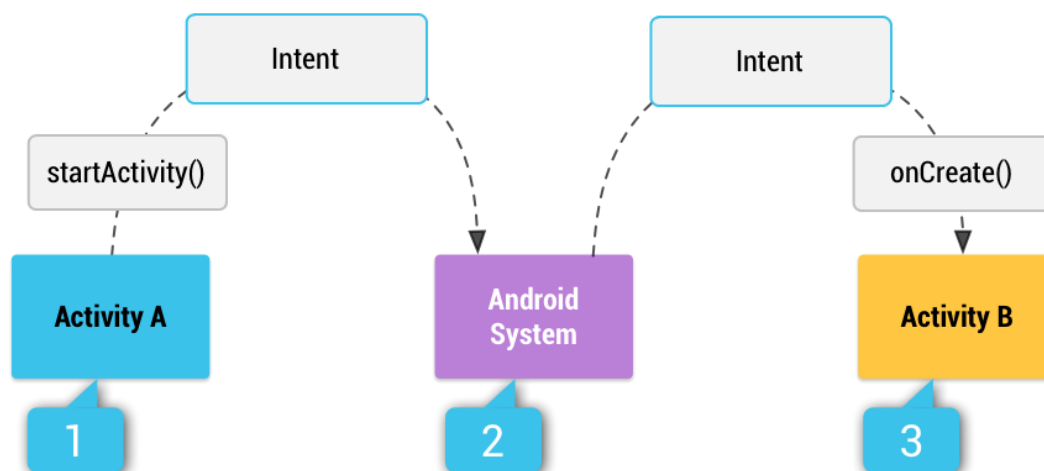
Intencje w systemie android odpowiadają za uruchamianie aktywności. W pewnym uproszczeniu używanie intencji wygląda tak jak przedstawia to Rysunek 2. Aktywność tworzy obiekt intencji, w którym zawiera informacje, co ma zostać uruchomione, a następnie obiekt ten jest przekazywany do metody `startActivity` – gdy chcemy tylko uruchomić inną aktywność, lub `startActivityForResult` – gdy chcemy uruchomić jakąś aktywność i uzyskać z niej informację zwrotną.

Istnieją dwa rodzaje intencji w systemie android:

- Intencje domniemane (ang. *implicit intents*) - określają akcję, jaka ma się wydarzyć w systemie, lecz nie jest podane, co ma daną akcję obsłużyć. Przykładem może być żądanie uruchomienia aparatu w telefonie, lecz nie jest zdefiniowane, która aplikacja aparatu ma zostać uruchomiona. O tym decyduje system operacyjny który dopasowuje żądanie i podejmuje stosowną akcję.
- Intencje konkretne (ang. *explicit intents*) – umożliwiają uruchomienie konkretnie podanej aktywności. Przydatne wówczas, gdy posiadamy aplikację z kilkoma aktywnościami między którymi chcemy nawigować.



Rysunek 1 Cykl życia aktywności



Rysunek 2 Tworzenie intencji. (1) aktywność A tworzy intencje i wysyła ją do systemu. (2) system dopasowuje intencję (3) aktywność B obsługująca dany rodzaj intencji jest uruchamiana przez system.

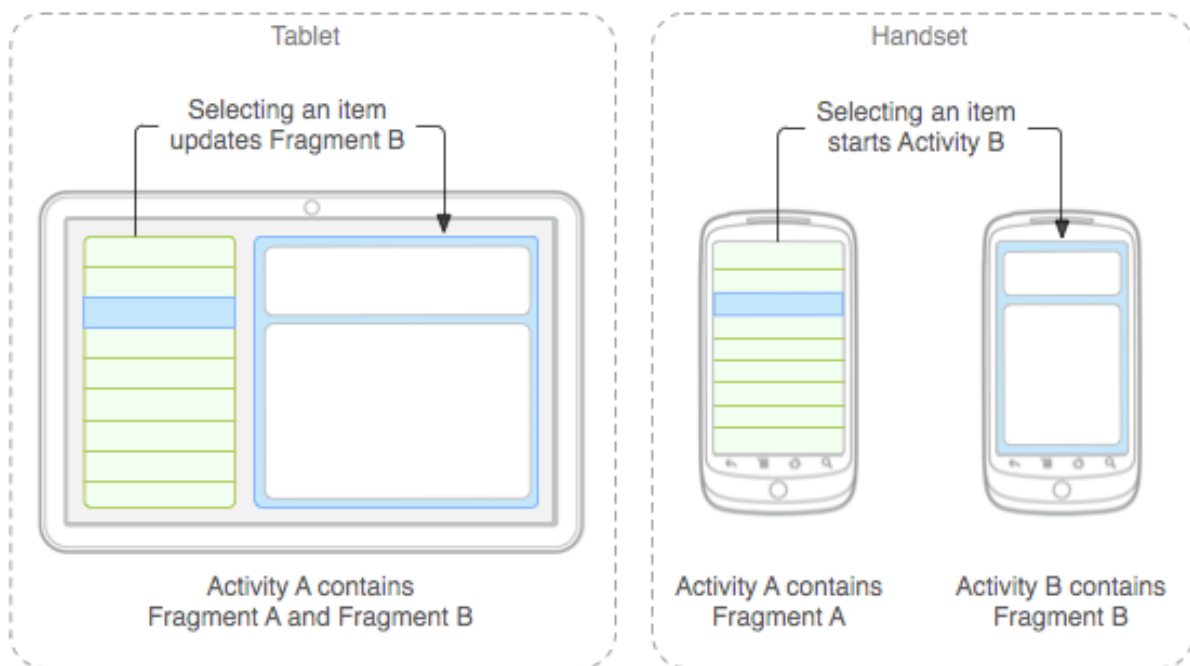
Poniżej przykład kodu odpowiedzialnego za uruchomienie dialera z wybranym numerem telefonu:

Listing 1 wywołanie aktywności dialera

```
public void dialPhoneNumber(String phoneNumber) {  
    Intent intent = new Intent(Intent.ACTION_DIAL);  
    intent.setData(Uri.parse("tel:" + phoneNumber));  
    if (intent.resolveActivity(getPackageManager()) != null) {  
        startActivity(intent);  
    }  
}
```

3. Fragmenty

Fragmenty są elementem aplikacji, które organizują interfejs graficzny w mniejsze, re-używalne komponenty. Dzięki temu łatwiej zarządzać aktywnością, która składa się z fragmentów. Istotnym jest, że fragmenty nie mogą występować samodzielnie. Zawsze są hostowane przez jakąś aktywność. Posiadają jednak własny cykl życia. Rysunek 3 przedstawia koncepcje fragmentów.

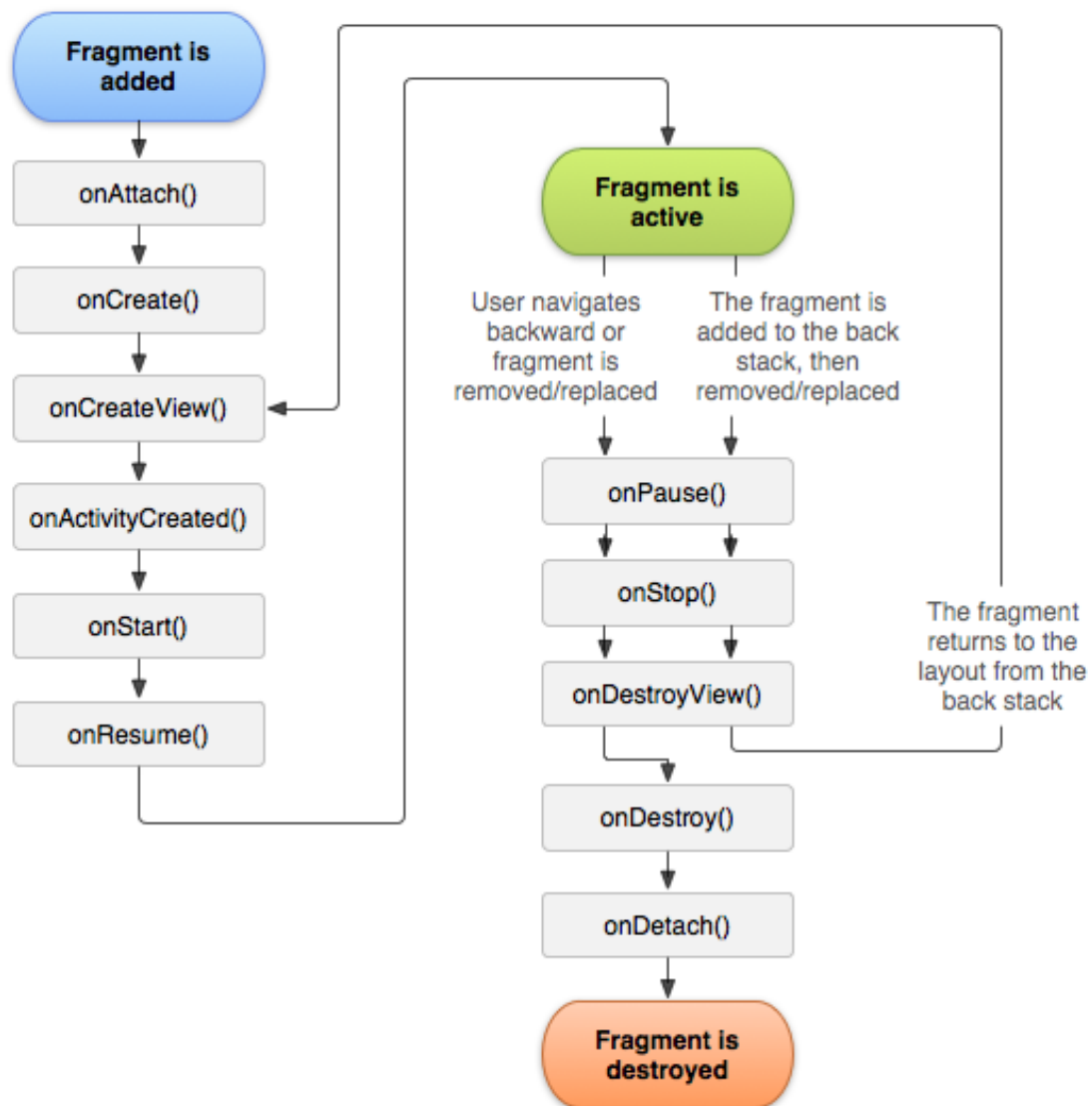


Rysunek 3 koncepcja re-używalnych fragmentów. Te same fragmenty mogą być użyte w różny sposób w zależności od konfiguracji urządzenia

Rysunek 4 przedstawia z kolei cykl życia pojedynczego fragmentu.

4. Serwisy

Serwisy wykazują pewne podobieństwa do aktywności w systemie android. Nie posiadają one jednak interfejsu graficznego, przez co nie potrzebują plików xml. Serwisy często są wykorzystywane do wykonywania zadań, do których interfejs graficzny jest niepotrzebny – np. zarządzanie połączeniem BT. Serwisy „przeżywają” np. zmianę orientacji ekranu, gdyż nie są niszczone w celu przerysowania interfejsu graficznego jak ma to miejsce w przypadku aktywności. Serwisy posiadają swój własny cykl życia. Istotnym jest, że serwisy nie działają w osobnym wątku, lecz współdzielą główny wątek z resztą aplikacji.



Rysunek 4 Cykl życia fragmentu

5. Broadcasts receiver

Broadcast receiver jest elementem systemu android, który umożliwia powiadamianie aplikacji o różnych zdarzeniach występujących w systemie. Rozwiązanie to oparte jest o wzorec publikacja-subskrypcja. Aplikacja subskrybuje dany kanał, po czym, w razie wystąpienia jakiegoś zdarzenia, otrzymuje odpowiednią informację. Przykładem zdarzeń mogą być: zmiana sieci, rozpoczęcie ładowania, mała ilość miejsca w pamięci itd. Element aplikacji obsługujący broadcasts receiver, podobnie jak serwisy, nie posiada interfejsu użytkownika.

6. Linki

Pod poniższym adresem można znaleźć informacje na tematy opisane w niniejszym opracowaniu:

http://marek.piasecki.staff.iar.pwr.wroc.pl/dydaktyka/pam/L02_AND/readings/Android-applicationFundamentals.pdf