Android Training for Java Developers

2. Android Application Components

DMYTRO DMYTRENKO
dmytrenko.d@gmail.com
https://github.com/ddmytrenko

Application

`https://developer.android.com/reference/android/app/Application.html`

Base class for maintaining global application state. You can provide your own implementation by creating a subclass and specifying the fully-qualified name of this subclass as the `"android:name"` attribute in your `AndroidManifest.xml`'s `<application>` tag. The `Application` class, or your subclass of the `Application` class, is instantiated before any other class when the process for your application/package is created.

Intent

https://developer.android.com/reference/android/content/Intent.html

The instance of the `Intent` object represents the description/specification of the task you want Android system to perform.

Intents can be:

- * implicit
- * explicit (direct)

Activity

`https://developer.android.com/reference/android/app/Activity.html`

* activity is a single, focused thing that the user can do
* `Activity` class takes care of creating a window for you
* activities are often presented to the user as full-screen windows
* they can also be used in other ways: floating or embedded

BroadcastReceiver

`https://developer.android.com/reference/android/content/BroadcastReceiver.html`

A receiver can be registered to listen to system messages and intents. After that `BroadcastReceiver` gets notified by the Android system if the specified event occurs.

ContentProvider

https://developer.android.com/reference/android/content/ContentProvider.html

A provider defines a structured interface to application data. `ContentProvider` can be used for accessing data within one application, but can also be used to share data with other applications.

Service

`https://developer.android.com/reference/android/app/Service.html`

`Service` performs tasks without providing an user interface. It can communicate with other Android components and send notifications to the user. As the `Service` has no user interface, it is not bound to the lifecycle of an `Activity`.

REMEMBER:
`Service` runs on the main thread by default!

IntentService

`https://developer.android.com/reference/android/app/IntentService.html`

The `IntentService` is used to perform a certain task in the background. Once done, the instance of `IntentService` terminates itself automatically.

REMEMBER:
`IntentService` runs on the separate thread!