

Android Components

Activities

They dictate the UI and handle the user interaction to the smart phone screen.

- Một Activity đại diện cho một màn hình giao diện người dùng và nó dùng để tương tác thực hiện làm một cái gì đó như là hiển thị màn hình chụp ảnh, danh sách danh bạ, lịch, ghi chú, bản đồ,...
- Mỗi Activity hoạt động giống như là một cửa sổ (nơi sẽ hiển thị giao diện người dùng). Các cửa sổ này thường sẽ lấp đầy cả màn hình hoặc nhỏ hơn màn hình và nổi trên một cửa sổ khác.

```
# Java
public class MainActivity extends Activity {}
```

```
# Kotlin
class MainActivity : Activity {}
```

Services

Là đối tượng điều khiển và xử lý ngầm của một ứng dụng.

- Service là một thành phần chạy ở background để xử lý những công việc dài hạn như: phát một bài nhạc ở background trong khi người dùng đang sử dụng một ứng dụng khác, hoặc có thể là lấy dữ liệu thông qua mạng internet mà không tạm hoãn các hoạt động tương tác hiện tại của người dùng trên Activity.

```
# Java
public class MyService extends Service {}
```

```
# Kotlin
class MyService : Service {}
```

Broadcast Receivers

Là đối tượng quản lý giao tiếp qua lại giữa hệ điều hành Android và ứng dụng.

- Broadcast Receivers simply respond to broadcast messages from other applications or from the system. For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.
- A broadcast receiver is implemented as a subclass of BroadcastReceiver class and each message is broadcaster as an Intent object.

```
# Java
public class MyReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {}
}
```

```
# Kotlin
class MyReceiver : BroadcastReceiver {
    override fun onReceive(context: Context?, intent: Intent?) {}
}
```

Content Providers

They handle data and database management issues.

- A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the `ContentResolver` class. The data may be stored in the file system, the database or somewhere else entirely.
- A content provider is implemented as a subclass of `ContentProvider` class and must implement a standard set of APIs that enable other applications to perform transactions.

```
# Java
public class MyContentProvider extends ContentProvider {
    @Override
    public boolean onCreate() {}

    @Nullable
    @Override
    public Cursor query(...) {}

    @Nullable
    @Override
    public String getType(...) {}

    @Nullable
    @Override
    public Uri insert(...) {}

    @Override
    public int delete(...) {}

    @Override
    public int update(...) {}
}
```

```
# Kotlin
class MyContentProvider : ContentProvider() {
    override fun onCreate(): Boolean {}

    override fun query(...): Cursor {}

    override fun getType(...): String {}

    override fun insert(...): Uri {}

    override fun delete(...): Int {}

    override fun update(...): Int {}
}
```

