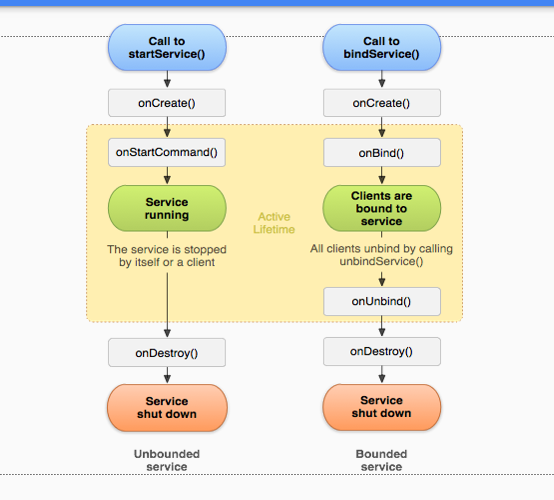
* Service có tác dụng xử lý các hành động sống lâu hoặc không cần tới giao diện người dùng
* Thực thi ngầm
  + VD: chạy nhạc, I/O, …
* Không ảnh hưởng bởi vòng đời của Activity
* 3 loại chính:
  + Foreground
    - Thực hiện các hành động người dùng có thể thấy, hiển thị qua notification
    - Chạy cả khi không có tương tác UI
    - VD: phát nhạc
  + Background
    - Thực hiện hành động người dùng không thấy (hiện trong phần process running tương tự như Task Manager)
    - VD: nén dữ liệu lưu trữ
  + Bound
    - Chỉ tồn tại khi liên kết với 1 thành phần ứng dụng
    - Cung cấp giao diện client-server
* Cài đặt Services:
  + Tạo lớp con hoặc sử dụng lớp có sẵn
  + Override một số phương thức:
    - onStartCommand
    - onBind
    - onCreate
    - onDestroy
  + Khai báo trong tệp manifest tag <application>
* Started Services là những services được khởi động bởi các thành phần khác qua pt startSerice
  + Vòng đời phụ thuộc thành phần khởi động
  + Sử dụng một trong hai lớp
    - Service: lớp cơ sở cho tất cả services
    - IntentService: lớp con của Service, xử lý lần lượt các yêu cầu khởi động, cài đặt phương thức onHandleIntent() để nhận yêu cầu
* Bound service:
  + Cài đặt phương thức onBind() để trả về một đối tượng IBinder - giao diện tương tác với service
  + Ứng dụng khác gọi bindService() để nhận giao diện tương tác và bắt đầu gọi các phương thức của service
  + Khi nhiều clients đồng thời bind tới một service, sử dụng phương thức unbindService() sau khi ngừng tương tác với service
* Gửi notification:
  + Toast Notification: xuất hiện trong 1 thời gian ngắn
  + Status Bar Notification: Người dùng có thể thực hiện hành động tương ứng background service
* Vòng đời của service:
  + Phụ thuộc loại service
    - Started Service: Tồn tại khi gọi stopSelf (từ chính service) hoạc stopService(từ thành phần khác)
    - Bound Service: Tạo khi thành phần khác gọi bindService(), đóng kết nối bằng unbindService(). Khi không có client nào bind đến, hệ thống sẽ hủy service
  + 2 vòng đời không hoàn toàn độc lập



* Lifecycle callbacks:
  + Theo dõi vòng đời service
    - Thời gian sống: onCreate -> onDestroy
    - Thời gian hoạt động: onStartCommand() or onBind -> onDestroy
* Inter-Process Communication (IPC):
  + Android cung cấp một cơ chế cho việc liên lạc giữa các tiến trình thông qua RPCs (Remote Procedure Calls)
  + Cho phép sử dụng API từ xa
  + Lời gọi hàm mang tính "local"
  + Tập trung định nghĩa và cài đặt giao diện lập trình RPCs
* AIDL (Android Interface Definition Language): Cú pháp Java
  + Phục vụ liên lạc giữa client và server sử dụng IPC
  + Cần chuyển các đối tượng về kiểu nguyên thủy cho phép hệ thống hiểu và vận chuyển
  + Cần thiết khi clients từ ứng dụng khác yêu cầu IPC và cần xử lý đa luồng
  + Lưu trong thư mục src/ với định dạng ".aidl" đối với cả ứng dụng cung cấp service và ứng dụng bind tới service đó
  + Khi build, một đối tượng IBinder được sinh ra trong thư mục gen/
  + Ứng dụng client gọi phương thức trong IBinder để thực hiện IPC