1. Thread

* Thread là một tiến trình đơn vị xử lý của máy tính có thể thực hiện một công việc riêng biệt.
* Multi-thread là khái niệm cho nhiều tiến trình chạy đồng thời. Một ứng dụng Java ngoài luồng chính có thể có các luồng khác thực thi đồng thời làm ứng dụng chạy nhanh và hiệu quả hơn.
* Ưu điểm của đa luồng : ứng dụng có thể được tách

+ Mỗi luồng có thể dùng chung và chia sẻ nguồn tài nguyên trong quá trình chạy, nhưng có thể thực hiện một cách độc lập

+  Luồng chính chạy giao diện người dùng,  Các luồng phụ nhiệm gửi đến luồng chính.

* Nhược điểm của đa luồng: Càng nhiều luồng thì càng xử lý phức tạp
* Có 2 cách để tạo một luồng mới: extends Thread và implements Runnable
* Trong android một luồng phụ không thể thay đổi giao diện liên quan đến luồng chính

1. AsyncTask

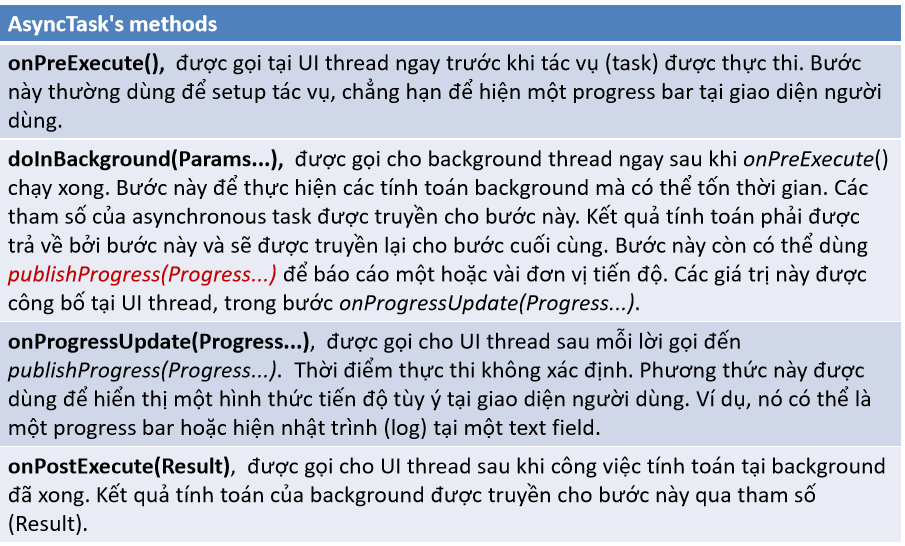
* Một chương trình chạy trên Android có thể sẽ có cấu trúc phức tạp. Yêu cầu kết nối đến Server, CSDL, tải file … Nếu chúng ta xử lý các công việc đó trên Main Thread sẽ làm ứng dụng có vẻ chạy chậm hay treo vì chúng làm gián đoạn việc cập nhật, xử lý trên GUI. Có nhiều cách để giải quyết vấn đề này: sử dụng Service, Thread hay đơn giản hơn là dùng Async Task (Asynchronous Task).
* Async Task cho phép thực hiện các hoạt động background và gửi kết quả cho UI thread mà không phải thao tác với thread và handler
* Có 3 đối số Generic Type

+ *Param*: Là giá trị ((biến) được truyền vào khi gọi thực thi tiến trình và nó sẽ được truyền vào doInBackground

+ Progress: Là giá trị (biến) dùng để update giao diện diện lúc tiến trình thực thi, biến này sẽ được truyền vào hàm onProgressUpdate.

+ Result: Là biến dùng để lưu trữ kết quả trả về sau khi tiến trình thực hiện xong. Những đối số nào không sử dụng trong quá trình thực thi tiến trình thì ta thay bằng Void.</params,>

* **Các methods của Async Task**



1. Handler

* Khi một tiến trình được tạo cho một ứng dụng, main thread của nó được dành riêng để chạy một message queue, queue này quản lý các đối tượng bậc cao của ứng dụng (activity, intent receiver, v.v..) và các cửa sổ mà chúng tạo ra.
* Ta có thể tạo các thead phụ, chúng tương tác với thread chính của ứng dụng qua một Handler.
* Khi ta tạo một Handler mới, nó được gắn với message queue của thread tạo ra nó – từ đó trở đi, nó sẽ gửi các message và các runnable tới message queue đó và thực thi chúng khi chúng ra khỏi message queue.
* Hai ứng dụng chính của Handler

+ xếp lịch cho các message và runnable cần được thực thi vào thời điểm nào đó

+ xếp hàng một action cần thực hiện tại một thread khác