**Bài tập 9: Thực hành về Toast Notification và Alert Dialog**

Vì trong tất cả các ứng dụng Tôi thấy rằng chúng ta hay sử dụng Toast và Alert Dialog để kiểm tra một điều gì đó, hay đơn giản chỉ là xuất thông báo. Vì vậy bài tập này Tôi sẽ hướng dẫn các bạn làm quen với Toast & Alert Dialog, sau đó các bạn sẽ được thực hành với các control cơ bản và nâng cao.

- Cả Toast và Alert Dialog khi hiển thị lên thì các tiến trình (hay các lệnh) khác  vẫn cứ tiếp tục làm việc.

**1) Toast:**

- Toast có thể được tạo và hiển thị trong Activity hoặc trong Servive.

- Không cho phép người sử dụng tương tác

- Khi hiển thị sau khoảng thời gian nào đó sẽ tự đóng lại

- Có 2 giá trị mặc định (ta nên sử dụng 2 giá trị này, không nên gõ con số cụ thể vào): hằng số **Toast.LENGTH\_SHORT** hiển thị trong  2 giây, **Toast.LENGTH\_LONG**hiển thị trong 3.5 giây.

Cách tạo Toast:

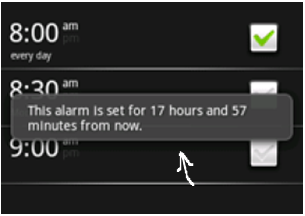
**Toast**toast=**Toast**.makeText(**YourActivity**.**this**, “Hiển thị gì thì ghi ở đây”,   **Toast.LENGTH\_SHORT**);

toast.show();

**- Khi nào bạn nên sử dụng Toast?**

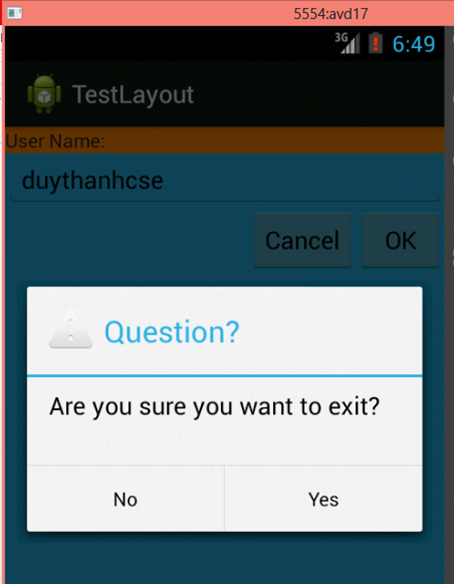
Theo Tôi thì tùy bạn, bạn có thể sử dụng trong trường hợp hiển thông báo trong các mục thiết lập thông số cấu hình, hay đơn giản chỉ là hiển thị lên để xem thông tin tạm thời nào đó (giống như để kiểm tra một vấn đề sảy ra chẳng hạn).

- Hình dưới đây cho bạn biết 1 Toast đang hiển thị:

[](http://duythanhcse.files.wordpress.com/2013/04/9_dialog_0.png)

**2) Alert Dialog:**

- Hiển thị và cho phép người dùng tương tác, ví dụ bạn nhìn hình Tôi chụp bên dưới, khi nhấn nút “Cancel”, chương trình sẽ hiển thị Alert Dialog hỏi xem có chắc chắn muốn xóa hay không? Bấm No thì không, bấm Yes thì tắt chương trình.

[](http://duythanhcse.files.wordpress.com/2013/04/9_dialog_1.png)- Cách tạo Alert Dialog:

**AlertDialog**.Builder b=**newAlertDialog**.Builder(**YourActivity**.**this**);

b.setTitle(“Question”);

b.setMessage(“Are you sure you want to exit?”);

b.setPositiveButton(“Yes”, **newDialogInterface**. OnClickListener() {

@Override

**publicvoid** onClick(DialogInterface dialog, **int** which)

{

finish();

}});

b.setNegativeButton(“No”, **new** DialogInterface.OnClickListener() {

@Override

**publicvoid** onClick(DialogInterface dialog, **int** which)

{

dialog.cancel();

}

});

b.create().show();

**Ý nghĩa của các hàm :**

- **setTitle** : thiết lập tiêu đề cho Dialog

- **setMessage**: Thiết lập nội dung cho Dialog

-**setIcon** : để thiết lập Icon

-**setPositiveButton**, **setNegativeButton** thiết lập hiển thị Nút chọn cho Dialog (ở đây bạn không quan tâm Nút chấp nhận hay Nút hủy nó thuộc PositiveButton hay NegativeButton vì đó là tùy thuộc bạn chọn. Chú ý là ở đối số thứ 2 của các hàm này sẽ là **DialogInterface**. OnClickListener chứ không phải **View.** OnClickListener

- **create**() để tạo Dialog

- **show**() để hiển thị Dialog.