TÀI LIỆU

HƯỚNG DẪN LẬP TRÌNH CƠ BẢN VỚI ANDROID

Hướng dẫn lập trình cơ bản với Android

Part 1

Bài này sẽ hướng dẫn cặn kẽ mọi người cách sử dụng Android trong Eclipse, hi vọng giúp những người mới chập chững bước vào lập trình Android có thêm thông tin và biết cách khai thác IDE của mình hiệu quả hơn.

I.Hướng dẫn cài đặt Android với Eclipse:
Thực chất anh Giáp đã viết một bài có nội dung tương tự, nhưng do Goolge thay đổi phương thức tải Android SDK nên mình quyết định viết lại, gộp luôn vào topic hướng dẫn sử dụng Eclipse.

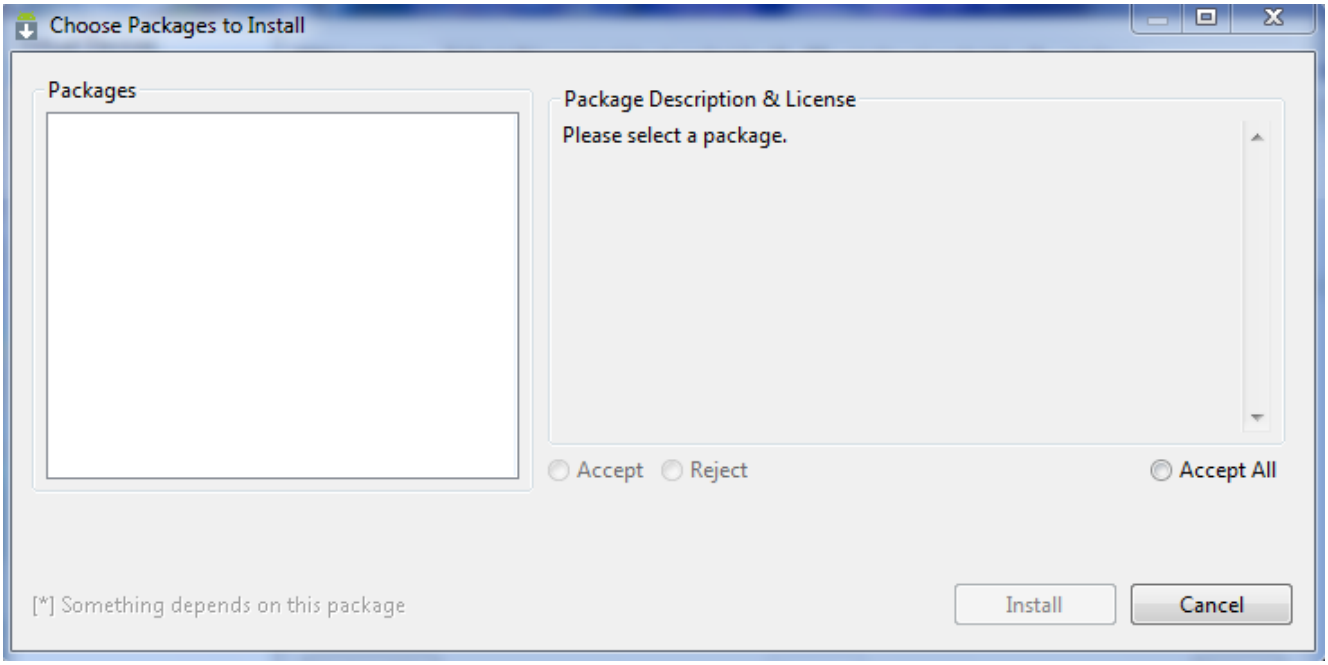
1.Download Android SDK:

Android SDK thực chất là tập hợp các công cụ và thư viện để phát triển các ứng dụng trên nền tảng hệ điều hành Android.

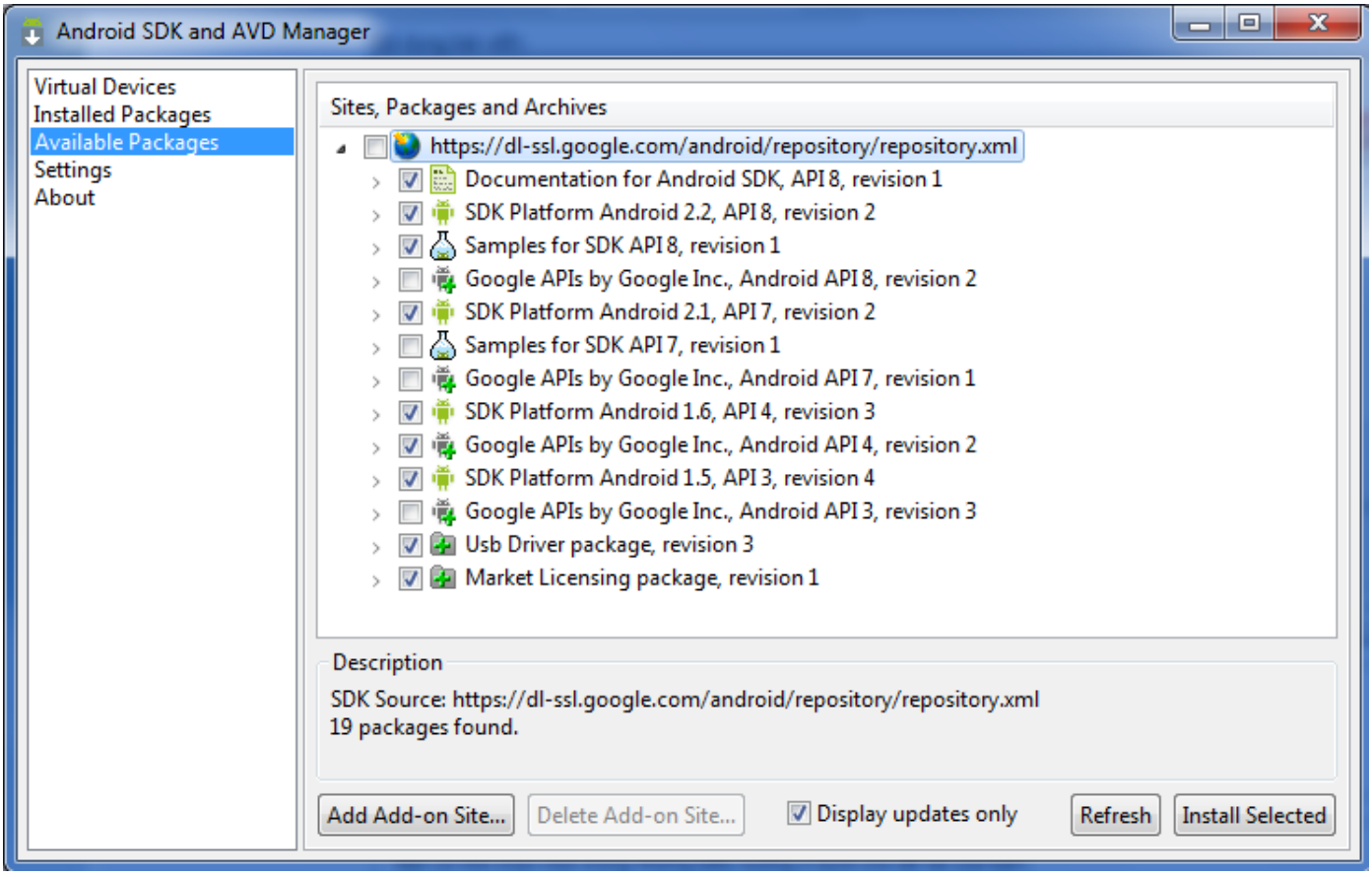
B1: Vào trang http://developer.android.com/sdk/index.html để tải Android SDK Starter. Tùy thuộc vào hệ điều hành mà bạn chọn bản Mac, Linux hay Window. Ở đây mình chọn tải bản cho Window.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r06-windows.zip	23293160 bytes	7c7fcec3c6b5c7c3df6ae654b27effb5
Mac OS X (intel)	android-sdk_r06-mac_86.zip	19108077 bytes	c92abf66a82c7a3f2b8493ebe025dd22
Linux (i386)	android-sdk_r06-linux_86.tgz	16971139 bytes	848371e4bf068dbb582b709f4e56d903

B2: Giải nén file zip bạn vừa tải về. Chạy SDK Setup.exe. Bạn có thể gặp thông báo lỗi Fetching https://dl-sl... Failed to fetch... Close thông báo này lại. Tiếp theo cửa sổ Choose Packages to Install xuất hiện. Nếu cửa sổ này trống rỗng -> Cancel.
-> Quay về cửa sổ Android SDK and AVD manager -> Chọn Setting, đánh dấu vào ô Force https://...
-> Chọn Available Packages



B3: Đánh dấu các Packages bạn muốn tải: Documents chính là phần Javadoc mô tả hoạt động của các phương thức và các lớp (phần này chắc chắn không thể thiếu rồi), Sample là các đoạn code mẫu, SDK Platform ứng với các phiên bản hệ điều hành (2.2 - API level 8, 2.1 - API level 7,...), và Google API để phát triển các phần mềm liên quan đến dịch vụ của Google (như Google Map nếu bạn muốn lập trình liên quan đến GPS).
Các bạn có thể tải hết nếu thích, còn muốn tối ưu thì có thể đánh dấu như mình (lưu ý USB drivers chỉ dành cho người sử dụng Windows và muốn phát triển ứng dụng test bằng điện thoại thật).
-> Install Selected
-> Install
-> Cửa sổ Install hiện ra
-> Ngồi chờ (>_<)



2.Tích hợp Android SDK vào Eclipse:

B1: Tải Eclipse nếu bạn chưa có. Mọi người có thể phân vân không biết tải bản nào cho phù hợp, nhưng theo ý kiến của mình thì có thể dùng 1 trong 2 bản sau: Eclipse for Java Developers, hoặc Eclipse for Java and Report Developers (mình dùng bản sau).

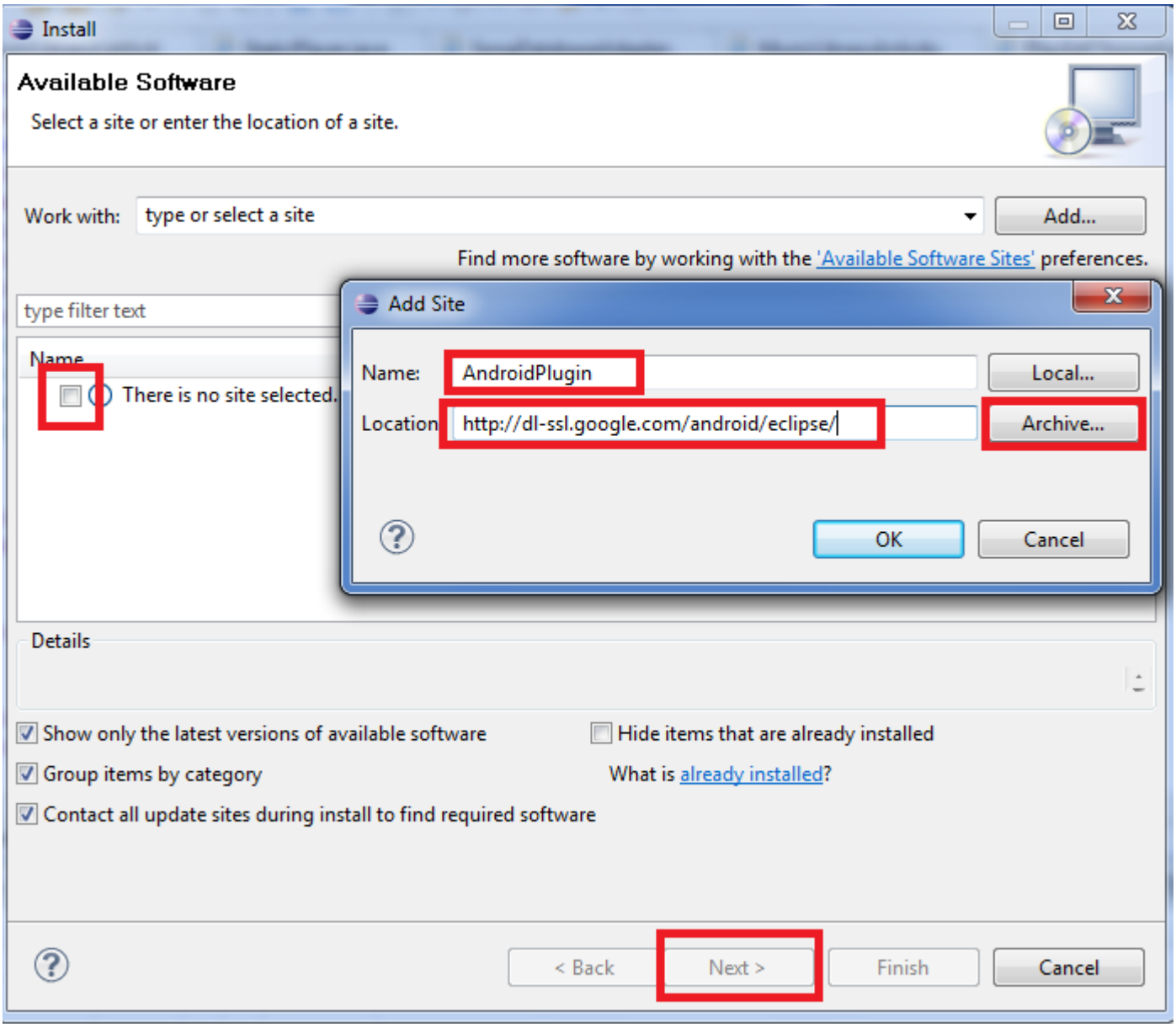
B2: Khởi chạy Eclipse, vào Help -> Install new softwares.
Chọn Add, gõ vào ô Name tên bạn muốn và Location gõ vào địa chỉ để tải về ADT:
HTML Code:

```
https://dl-ssl.google.com/android/eclipse/
```

hoặc
HTML Code:

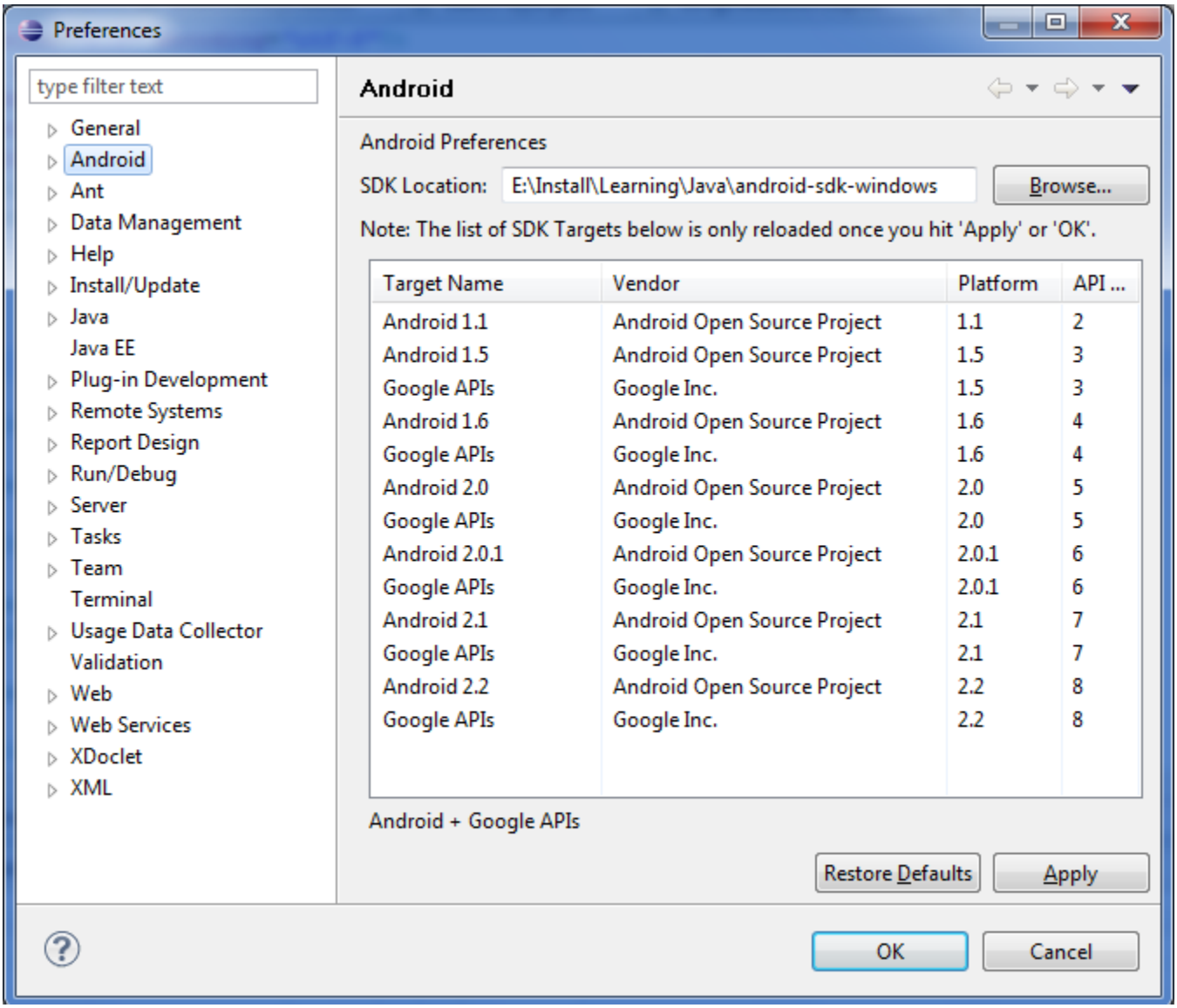
```
http://dl-ssl.google.com/android/eclipse/
```

nếu https không hoạt động. Ngoài ra bạn cũng có thể tải thẳng ADT về máy theo link http://dl.google.com/android/ADT-0.9.7.zip (bản mới nhất 0.9.7 ứng với Android 2.2), chọn Archive và browse tới file này (lưu ý không giải nén)
-> OK
-> Check vào phần dưới ô Name (sẽ hiện ra dòng Developer Tools).



B3: Next, next, Accept, next,...Finish (như Install mọi chương trình bình thường).

B4: Eclipse -> Windows -> Preferences -> Android
Nhấn nút Browse và chỉnh đường dẫn tới thư mục của Android SDK bạn tải lúc trước.
-> Apply
-> OK



Hướng dẫn lập trình cơ bản với Android

Part 2

3.Android Virtual Device (Emulator):

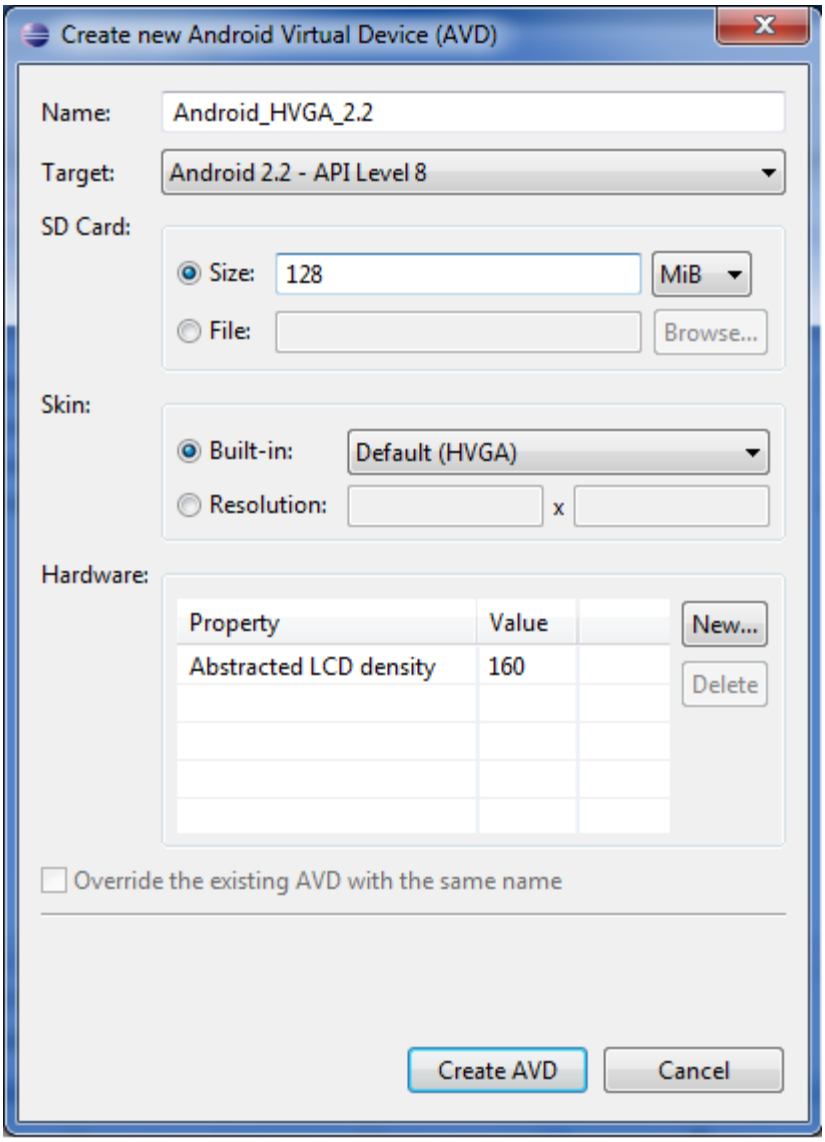
AVD là máy ảo cho developer phát triển ứng dụng. Để tạo 1 AVD bạn vào Windows -> Android SDK and AVD Manager -> Virtual Devices chọn New.
-> Cửa sổ Create new AVD hiện ra, bạn điền thông tin cho AVD bạn muốn:

Name: Tùy ý (nhưng chỉ được sử dụng các ký tự "a-z", "A-Z", ".-_", nghĩa là cả khoảng trắng cũng ko đc).

Target: Chọn phiên bản hệ điều hành bạn muốn (thường mình tạo một Android 1.6 và một Android 2.2 để test).

SD Card: gõ vào Size SD card ảo cho AVD, hoặc chỉnh tới file đã có sẵn. Nhiều AVD có thể dùng chung 1 Sdcard (chỉ cần tạo lần đầu, các lần sau chỉnh đường dẫn tới file đó).

Skin: có thể để Default (HVGA) hoặc chọn kích cỡ màn hình bạn muốn. Chỉ cần quan tâm tới 3 option: HVGA (phân giải 320-480 như G1, G2, i5700...), QVGA (240-320 như HTC Wildfire...), WVGA854 (480-854 như Milestone, NexusOne...) -> Create AVD.



II.Một số chức năng cần biết khi lập trình Android với Eclipse:

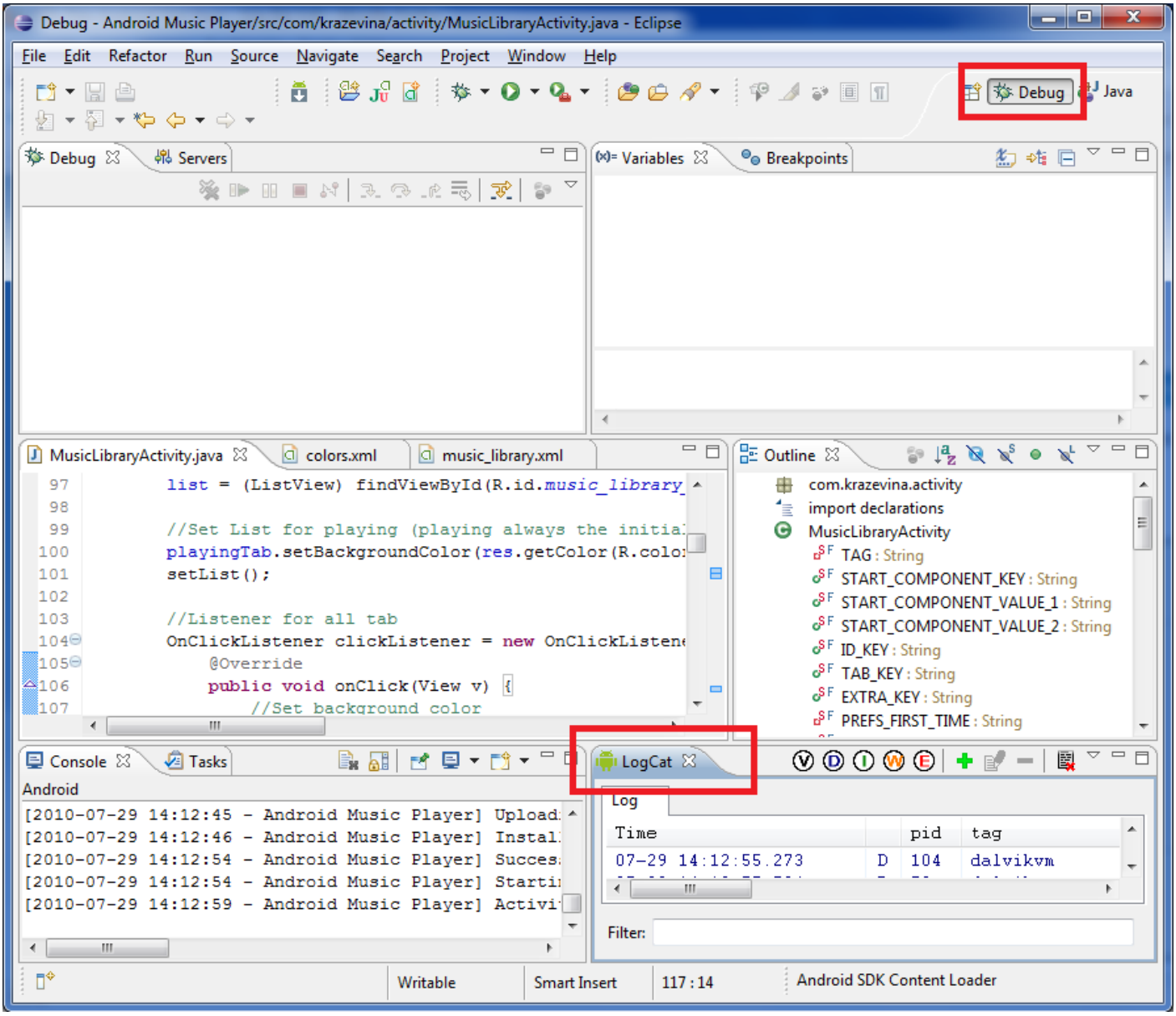
Những điều mình nêu ra dưới đây đều là cơ bản, có thể các pro đã biết hết rồi, nhưng như mình đã nói, bài này là dành cho newbie với hi vọng gia tăng kiến thức cơ bản cùng giảm thời gian tìm kiếm cho các bạn.

1.Debug cho ứng dụng Android:

Nhiều người chuyển từ J2SE hoặc J2ME sang Android sẽ ngạc nhiên vì câu lệnh debug kinh điển **System.out.println()** không còn in ra trên cửa sổ Console nữa.

Google đã thay thế nó bằng Logcat, một cửa sổ ghi lại toàn bộ hoạt động của hệ điều hành. Để mở Logcat, trước tiên các bạn chọn Window -> Open Perspective -> Debug. Nếu ko thấy option Debug thì chọn Other và tìm Debug trong cửa sổ mới hiện ra. Sau đó chọn tab Debug mới xuất hiện ở góc trên bên phải của Eclipse (xem hình). Theo kinh nghiệm của mình thì tốt nhất là Maximize Logcat ra luôn cho dễ quan sát.

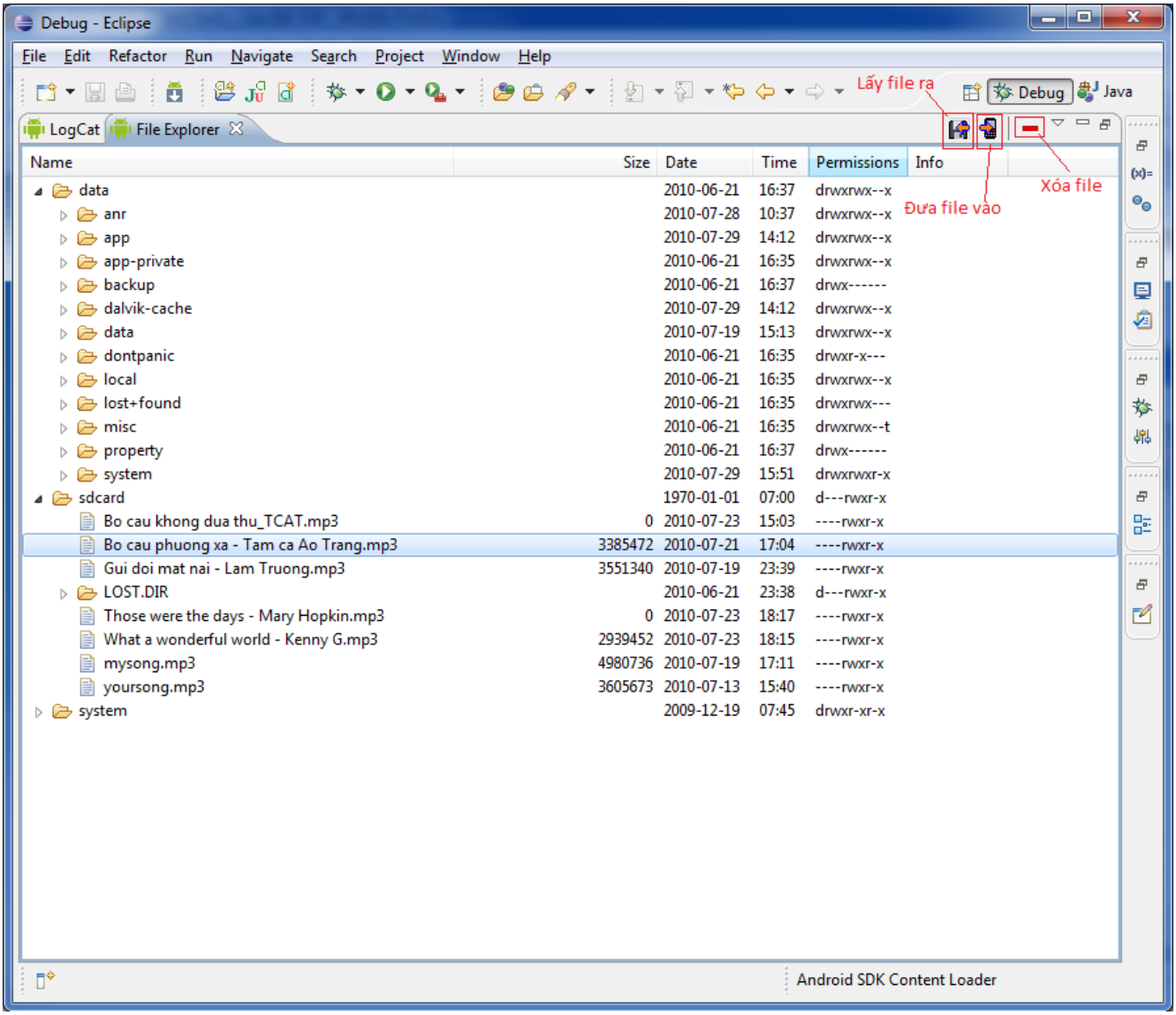
Cảm nhận ngày xưa khi mới sử dụng Logcat là rối và khó dùng. Nhưng càng về sau mình càng quen và thấy nó tiện hơn Console nhiều, vì Console chỉ đưa ra thông báo do chúng ta gọi, còn Logcat thì đưa cả luôn những thông báo của hệ điều hành, giúp chúng ta nắm được hệ điều hành đang làm gì, gọi đến cái gì, khởi chạy những gì...



2.File Explorer của Android:

File Explorer là một chức năng hữu ích Google đưa vào giúp chúng ta quản lý file trong sd card và cả file system data (chỉ quản lý được của emulator, không thể truy nhập system data của thiết bị thật). FE giúp bạn dễ dàng đưa file vào / lấy file ra trong sdcard ảo của emulator, xóa cơ sở dữ liệu của chương trình để khởi tạo lại (only emulator)...

Mở FE bằng cách vào Window -> Show View -> Others -> Android -> File Explorer. Mình để FE trong cửa sổ Debug cho tiện quản lý.


Debug - Eclipse
File Edit Refactor Run Navigate Search Project Window Help
Lấy file ra
Debug
Java
LogCat
File Explorer
Đưa file vào
Xóa file
Name Size Date Time Permissions Info
data 2010-06-21 16:37 drwxrwx--x
anr 2010-07-28 10:37 drwxrwx--x
app 2010-07-29 14:12 drwxrwx--x
app-private 2010-06-21 16:35 drwxrwx--x
backup 2010-06-21 16:37 drwx------
dalvik-cache 2010-07-29 14:12 drwxrwx--x
data 2010-07-19 15:13 drwxrwx--x
dontpanic 2010-06-21 16:35 drwxr-x---
local 2010-06-21 16:35 drwxrwx--x
lost+found 2010-06-21 16:35 drwxrwx---
misc 2010-06-21 16:35 drwxrwx--t
property 2010-06-21 16:37 drwx------
system 2010-07-29 15:51 drwxrwxr-x
sdcard 1970-01-01 07:00 d---rwxr-x
Bo cau khong dua thu_TCAT.mp3 0 2010-07-23 15:03 ----rwxr-x
Bo cau phuong xa - Tam ca Ao Trang.mp3 3385472 2010-07-21 17:04 ----rwxr-x
Gui doi mat nai - Lam Truong.mp3 3551340 2010-07-19 23:39 ----rwxr-x
LOST.DIR 2010-06-21 23:38 d---rwxr-x
Those were the days - Mary Hopkin.mp3 0 2010-07-23 18:17 ----rwxr-x
What a wonderful world - Kenny G.mp3 2939452 2010-07-23 18:15 ----rwxr-x
mysong.mp3 4980736 2010-07-19 17:11 ----rwxr-x
yoursong.mp3 3605673 2010-07-13 15:40 ----rwxr-x
system 2009-12-19 07:45 drwxr-xr-x
Android SDK Content Loader

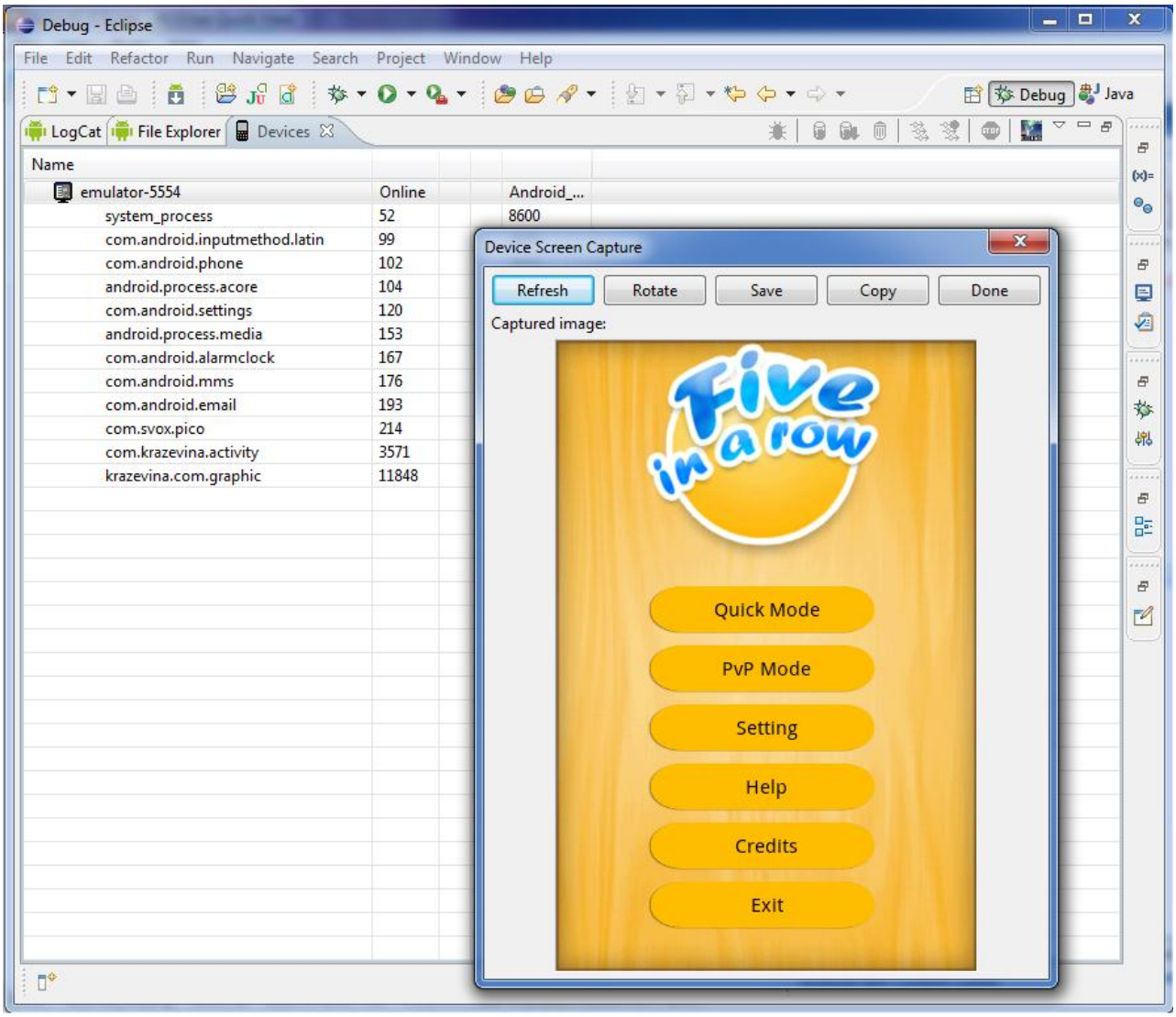
Hướng dẫn lập trình cơ bản với Android

Part 3

Device của Android:

Device cũng là một chức năng hữu ích nữa trong Android giúp bạn quản lý thiết bị ảo cũng như thật của mình. Mở Device bằng cách vào Window -> Show View -> Device hoặc vào Window -> Show View -> Others -> Android -> Device.

Chức năng mình thường sử dụng nhất của device là Screen Capture, cực kỳ tiện để lấy ảnh minh họa làm thuyết trình hoặc giới thiệu trên Google Market.



III.Một số thủ thuật trong Android và Eclipse:

1.Hiển thị tiếng Việt:

Window -> Preferences -> General -> Workspace.

Trong Workspace phần Text File Encoding chọn Other -> UTF-8.

2.Tự động sổ code:

Tổ hợp phím kinh điển Ctrl + Space (điều kỳ lạ là khá nhiều người không biết).

3.Comment:

Ctrl + "/": tự động thêm cụm "//" vào đầu dòng (ko tiện lắm).
Ctrl + Shift + "/": tự động thêm "/* */" vào cụm được bôi đen.
Ctrl + Shift + "\": tự động bỏ "/* */" vào cụm được bôi đen.

4.Override:

Đôi lúc bạn muốn Override phương thức của một lớp có sẵn, ko lẽ chúng ta sẽ đi tìm tên phương thức trên mạng và gõ đúng lại như thế? Nhiều bạn đã gặp lỗi khi định viết lại phương thức này và debug mãi ko ra (vì nó ko phải là lỗi):
Mã:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

}
```

đơn giản vì gõ thiếu chữ s.

Eclipse cung cấp phương tiện giúp ta ghi đè phương thức của lớp cha với Source -> Override/Implement Method, nhưng cực kỳ nhiều người không biết.

Hướng dẫn lập trình cơ bản với Android - Bài 1

Yêu cầu kiến thức cho lập trình Android:

Để lập trình android, mình nghĩ mọi người chỉ cần kiến thức java căn bản là hoàn toàn ok. Căn bản ở đây có nghĩa là hiểu được thế nào là class, package, biết ý nghĩa của các từ khóa như public, private, protected,... thành thạo các lệnh cơ bản như if, for(), switch(), while(), ... biết sd các lệnh như Integer.parseInt() hay String.valueOf()... Nên có thêm kiến thức về gói java.util vì đây là gói hỗ trợ nhiều lớp rất mạnh được sử dụng trên mọi nền, ngoài ra các gói như java.io, java.net... cũng được recommended

Các kiến thức về các gói lập trình cho desktop như java.awt, java.swing hoàn toàn không cần thiết (bản thân mình cũng chưa sd cái này bao giờ, nhảy vào học java là học J2ME luôn), hay các gói của J2ME cũng vậy Lập trình Android tuy cũng là lập trình di động, nhưng các điện thoại sử dụng hđh Android có cấu hình rất mạnh (Nexus One có VXL lên tới 1Ghz), vì vậy 2 nền tảng Android và J2ME cũng rất khác nhau. Android có những gói riêng hỗ trợ lập trình cho nó và không yêu cầu khắt khe về việc tối ưu code như J2ME. Thật đáng tiếc vì J2ME mình học ko ứng dụng được mấy vào lập trình Android (tuy nhiên 1 số kỹ thuật cơ bản cho lập trình game 2D như Sprite, double buffering, Tile... thì vẫn ko hề phí phạm chút nào)

Cài đặt Android để lập trình:

Để lập trình Android thì mỗi bộ SDK của Google là không đủ, bạn còn cần tích hợp nó vào một IDE như Eclipse. Anh Giáp đã có 2 bài hướng dẫn rất chi tiết về cài đặt Android trong Eclipse cũng như Netbeans, nhưng theo mình mọi người nên sử dụng Eclipse hơn vì nó có nhiều tính năng hỗ trợ lập trình Google, còn Netbeans thì plugin cho Android vẫn chưa hoàn thiện

Eclipse

Netbeans

Tiện thể mình nói luôn, mình học Android theo 2 cuốn *Professional Android Application Development* và *Unlocking Android*. Cả 2 cuốn đều dành cho beginner nhưng cuốn đầu code nhiều, giải thích ít, cuốn thứ 2 giải thích rõ ràng hơn. Nếu

có ai có ý định tham khảo thì nên đọc cuốn UA trước để hiểu rõ hơn Android, sử dụng cuốn PAAD trong việc tham khảo các đoạn code cho lập trình.

Understanding Android Application:

Việc hiểu được các thành phần (component) tạo nên một ứng dụng Android là rất cần thiết cho việc lập trình. Các thành phần này được chia làm 6 loại bao gồm:

1.Activity: hiểu một cách đơn giản thì Activity là nền của 1 ứng dụng. Khi khởi động 1 ứng dụng Android nào đó thì bao giờ cũng có 1 main Activity được gọi, hiển thị màn hình giao diện của ứng dụng cho phép người dùng tương tác.

2.Service: thành phần chạy ẩn trong Android. Service sử dụng để update dữ liệu, đưa ra các cảnh báo (Notification) và không bao giờ hiển thị cho người dùng thấy.

3.Content Provider: kho dữ liệu chia sẻ. Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.

4.Intent: nền tảng để truyền tải các thông báo. Intent được sử dụng để gửi các thông báo đi nhằm khởi tạo 1 Activity hay Service để thực hiện công việc bạn mong muốn. VD: khi mở 1 trang web, bạn gửi 1 intent đi để tạo 1 activity mới hiển thị trang web đó.

5.Broadcast Receiver: thành phần thu nhận các Intent bên ngoài gửi tới. VD: bạn viết 1 chương trình thay thế cho phần gọi điện mặc định của Android, khi đó bạn cần 1 BR để nhận biết các Intent là các cuộc gọi tới.

6.Notification: đưa ra các cảnh báo mà không làm cho các Activity phải ngừng hoạt động.

Activity, Service, Broadcast Receiver và Content Provider mới là những thành phần chính cấu thành nên ứng dụng Android, bắt buộc phải khai báo trong AndroidManifest (tham khảo bài 2 có giới thiệu đầy đủ về file này).

Understanding Android Application Life Cycle:

Android có cơ chế quản lý các process theo chế độ ưu tiên. Các process có priority thấp sẽ bị Android giải phóng mà không hề cảnh báo nhằm đảm bảo tài nguyên.

1.Foreground process: là process của ứng dụng hiện thời đang được người dùng tương tác.

2.Visible process: là process của ứng dụng mà activity đang hiển thị đối với người

dùng (onPaused() của activity được gọi).
3.Service process: là Service đang running.
4.Background process: là process của ứng dụng mà các activity của nó ko hiển thị với người dùng (onStoped() của activity được gọi).
5.Empty process: process không có bất cứ 1 thành phần nào active.
Theo chế độ ưu tiên thì khi cần tài nguyên, Android sẽ tự động kill process, trước tiên là các empty process.

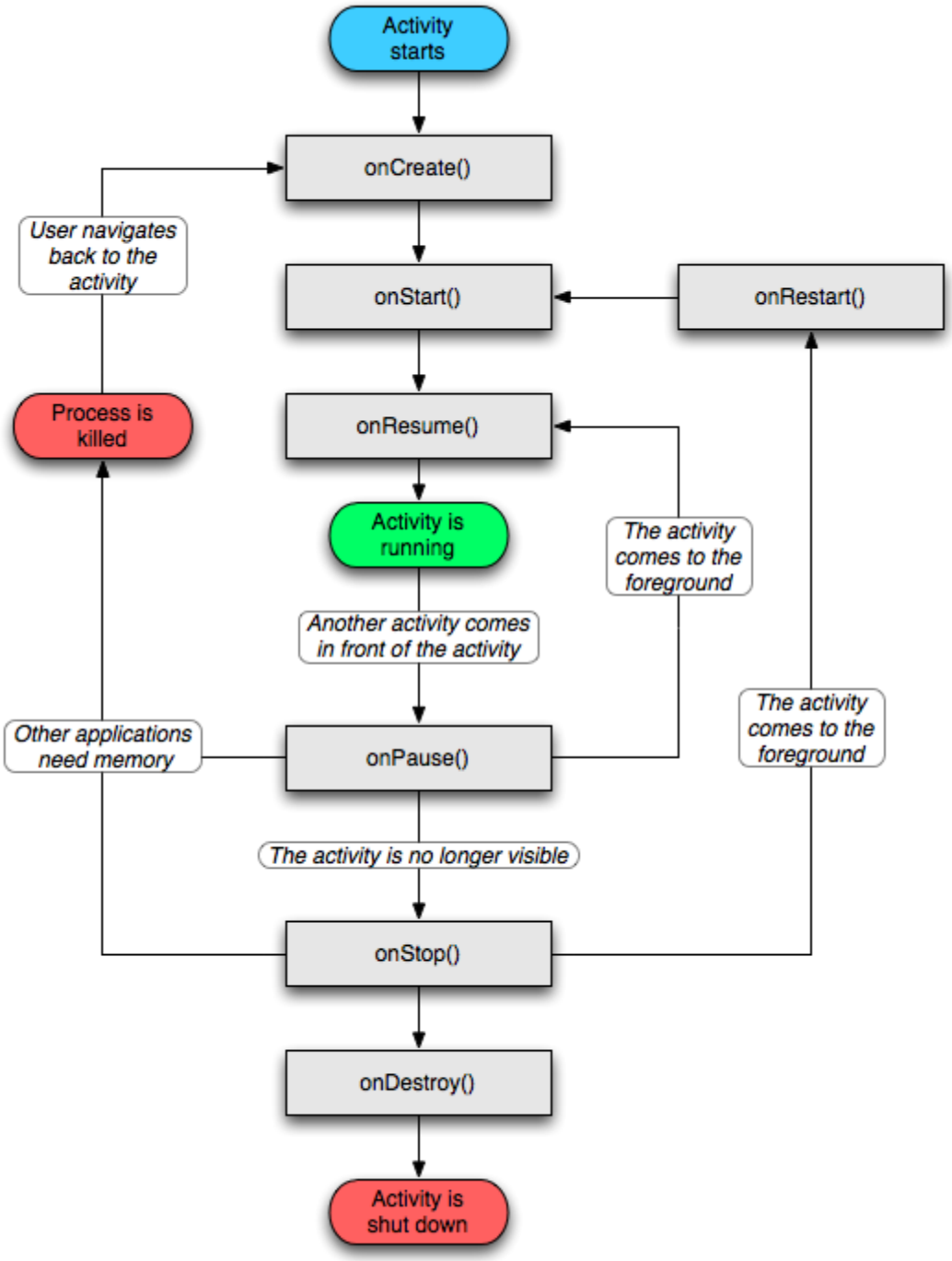
Hướng dẫn lập trình cơ bản với Android

Part 4

Android Activity Life Cycle:
Như mình đã giới thiệu ở trên , Actitvity là thành phần quan trọng nhất và đóng vai trò chính trong xây dựng ứng dụng Android. Hệ điều hành Android quản lý Activity theo dạng stack: khi một Activity mới được khởi tạo, nó sẽ được xếp lên đầu của stack và trở thành ***running activity***, các Activity trước đó sẽ bị tạm dừng và chỉ hoạt động trở lại khi Activity mới được giải phóng.

Activity bao gồm 4 state:
- ***active (running):*** Activity đang hiển thị trên màn hình (foreground).
- ***paused:*** Activity vẫn hiển thị (visible) nhưng không thể tương tác (lost focus). VD: một activity mới xuất hiện hiển thị giao diện đè lên trên activity cũ, nhưng giao diện này nhỏ hơn giao diện của activity cũ, do đó ta vẫn thấy được 1 phần giao diện của activity cũ nhưng lại không thể tương tác với nó.
- ***stop:*** Activity bị thay thế hoàn toàn bởi Activity mới sẽ tiến đến trạng thái ***stop***
- ***killed:*** Khi hệ thống bị thiếu bộ nhớ, nó sẽ giải phóng các tiến trình theo nguyên tắc ưu tiên. Các Activity ở trạng thái ***stop*** hoặc ***paused*** cũng có thể bị giải phóng và khi nó được hiển thị lại thì các Activity này phải khởi động lại hoàn toàn và phục hồi lại trạng thái trước đó.

Biểu đồ miêu tả Activity state



Vòng đời của Activity:
- ***Entire lifetime:*** Từ phương thức **onCreate( )** cho tới **onDestroy( )**
- ***Visible liftetime:*** Từ phương thức **onStart( )** cho tới **onStop( )**
- ***Foreground lifetime:*** Từ phương thức **onResume( )** cho tới **onPause( )**

Khi xây dựng Actitvity cho ứng dụng cần phải viết lại phương thức **onCreate( )** để

thực hiện quá trình khởi tạo. Các phương thức khác có cần viết lại hay không tùy vào yêu cầu lập trình.

XML trong Android:
Không giống như lập trình java thông thường, lập trình android ngoài các lớp được viết trong *.java còn sử dụng XML để thiết kế giao diện cho ứng dụng. Tất nhiên bạn hoàn toàn có thể thiết kế 1 giao diện như ý muốn mà không cần tới bất cứ 1 dòng XML nào, nhưng sd XML sẽ đơn giản công việc đi rất nhiều. Đồng thời sd XML sẽ giúp việc chỉnh sửa ứng dụng sau này trở nên dễ dàng.

Về nguyên tắc, khi lập trình ứng dụng ta thiết kế giao diện bằng XML và cài đặt các xử lý khi tương tác với giao diện trong code.

1 số thành phần cơ bản trong Android:
1.Các layout:
Layout được dùng để quản lý các thành phần giao diện khác theo 1 trật tự nhất định.
- FrameLayout: Layout đơn giản nhất, thêm các thành phần con vào góc trên bên trái của màn hình.
- LinearLayout: thêm các thành phần con theo 1 chiều nhất định (ngang hoặc dọc). Đây là layout được sử dụng nhiều nhất.
- RelativeLayout: thêm các thành phần con dựa trên mối quan hệ với các thành phần khác hoặc với biên của layout.
- TableLayout: thêm các thành phần con dựa trên 1 lưới các ô ngang và dọc.
- AbsoluteLayout: thêm các thành phần con dựa theo tọa độ x, y.

Layout được sử dụng nhằm mục đích thiết kế giao diện cho nhiều độ phân giải. Thường khi lập trình nên kết hợp nhiều layout với nhau để tạo ra giao diện bạn mong muốn.

2.XML unit:
Để hiểu được các thành phần cơ bản của XML cũng như việc sử dụng XML kết hợp với code, ta sẽ đi xây dựng thử một chương trình đơn giản.

Yêu cầu: Xây dựng 1 ứng dụng cho phép gõ 1 nội dung vào rồi hiển thị ra nội dung đó ở bên dưới.

B1: Khởi tạo 1 project (ở đây sử dụng Eclipse để minh họa).
Vào thẻ File -> New -> Android Project. Nếu bạn mới lập trình Android lần đầu thì

có lẽ dòng Android Project sẽ không hiện ra, khi đó xuống phía cuối chọn Other rồi vào Android -> Android Project.

B2: Điền thông tin cho project