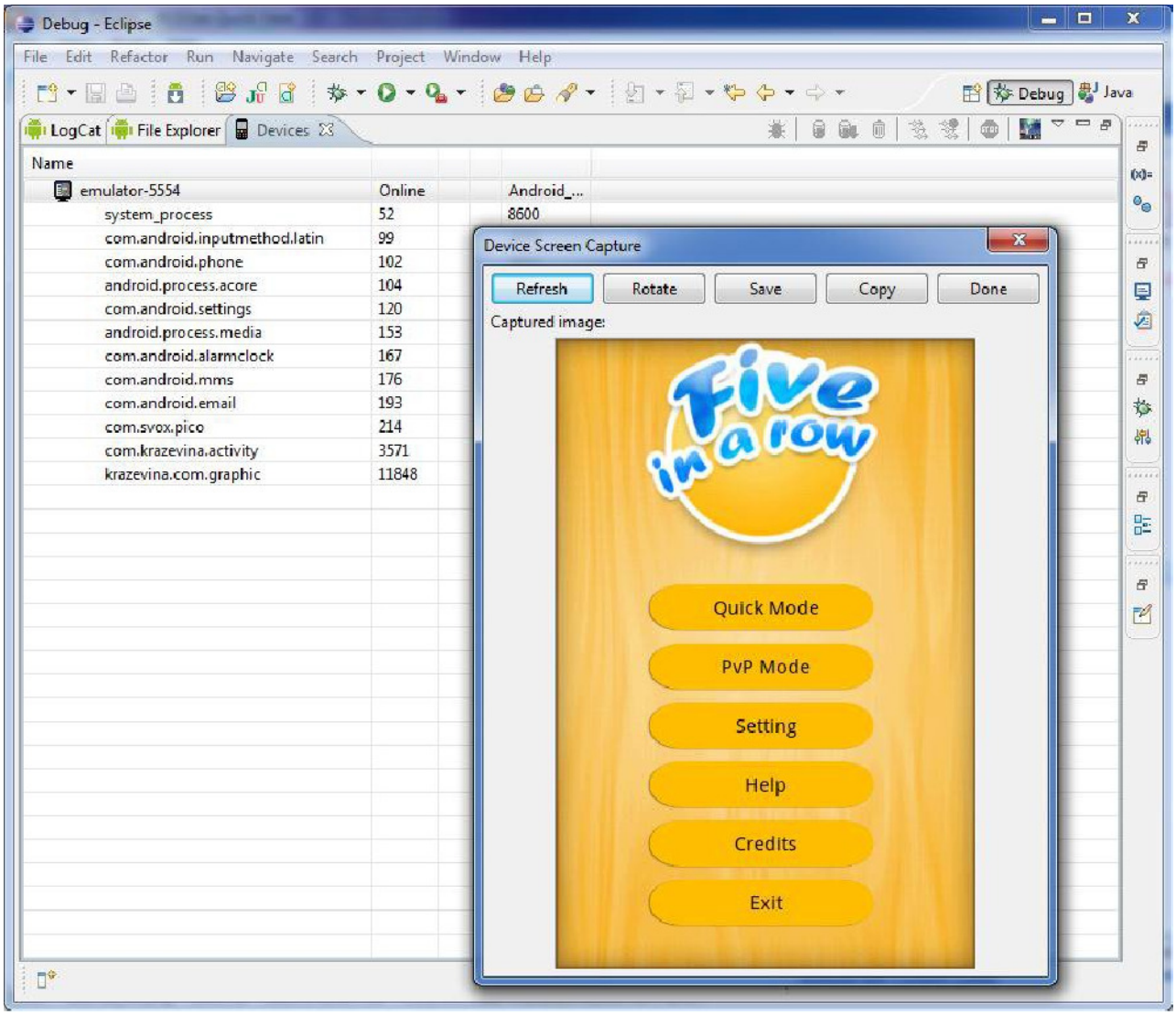
Hướng dẫn lập trình cơ bản với Android

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.