

Cài đặt môi trường phát triển ứng dụng

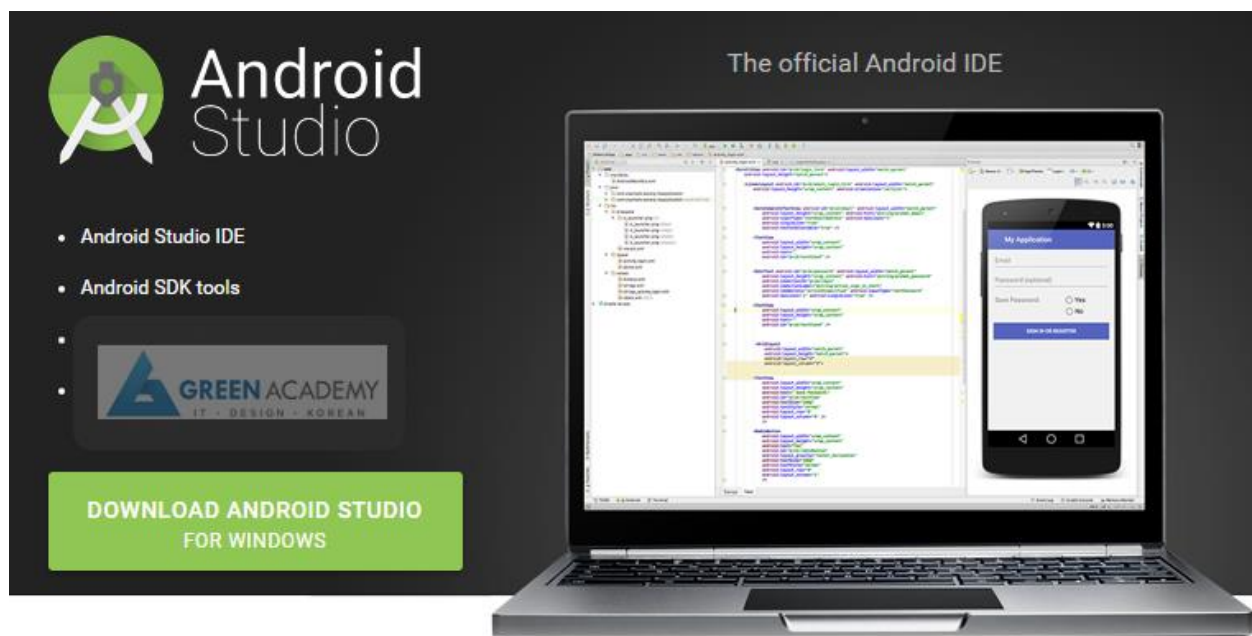
1 Lý thuyết

1.1 Tổng quan về môi trường phát triển ứng dụng

Android Studio là môi trường phát triển tích hợp (Integrated Development Environment - IDE) chính thức cho việc phát triển ứng dụng cho nền tảng Android. Nó được cung cấp miễn phí theo giấy phép Apache v2.0, có sẵn các phiên bản dành cho Windows, Mac OS X và Linux. Ngoài ra, ta cũng có thể dùng Eclipse để thay thế, tuy nhiên cần lưu ý là chỉ có Android Studio mới được hỗ trợ chính thức từ Google.

Android Software Development Kit (Android SDK) bao gồm một tập các công cụ hỗ trợ cho việc lập trình trên nền tảng Android như trình biên dịch, các thư viện, máy ảo, tài liệu, source code mẫu... Thông thường, Android SDK sẽ được tích hợp vào Android Studio hoặc Eclipse để hỗ trợ cho việc phát triển ứng dụng Android.

Để bắt đầu cho việc phát triển ứng dụng Android, ta có thể tải về bộ Android Studio Bundle mà bên trong nó bao gồm Android Studio, Android SDK và các thành phần khác tại trang chủ <http://developer.android.com>.



1.2 Android Studio Bundle

Yêu cầu hệ thống để cài đặt Android Studio cho từng hệ điều hành như sau:

	Windows	OS X	Linux
Hệ điều hành	Windows 10/8.1/8/7/Vista/2003/XP (32 or 64 bit)	Mac OS X 10.8.5 trở lên	Máy tính bàn GNOME hoặc KDE trên Ubuntu hoặc Fedora hoặc GNU/Linux Debian
RAM	2 GB RAM trở lên, khuyến cáo 4 GB RAM		
Dung lượng đĩa cứng	500 MB dung lượng đĩa cứng		
Khoảng trống cho Android SDK	Ít nhất 1 GB cho Android SDK, máy ảo, file tạm...		
JDK	Java Development Kit (JDK) 7 hoặc cao hơn		
Độ phân giải màn hình	1280x800 trở lên		

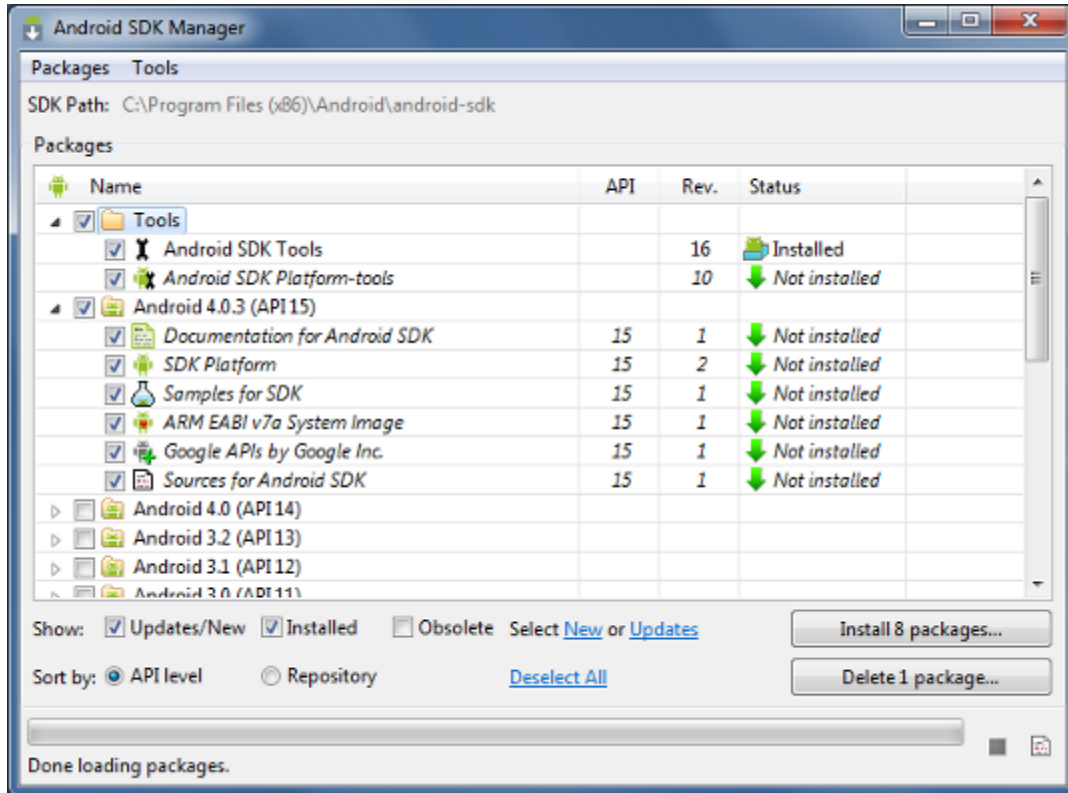
Trong đó, Java Development Kit (JDK) là một bộ hỗ trợ phát triển ứng dụng Java, có thể cài đặt trên nhiều nền tảng như Windows, Mac OS X, Linux... Xem thêm phụ lục để biết *cách cài đặt JDK cho Windows*. Ngoài ra, ta có thể tham khảo cách cài đặt JDK cụ thể cho từng nền tảng tại trang mô tả kỹ thuật của Oracle <http://docs.oracle.com>.

1.3 Android SDK

Mặc định, Android SDK không bao gồm mọi thứ ta cần để phát triển ứng dụng mà chỉ có một số thành phần ban đầu, thường ở các phiên bản mới nhất...


Android SDK chia các công cụ, thư viện, ... thành các gói (package) mà ta có thể tải về khi cần, kể cả tải mới cũng như cập nhật cái có sẵn trong SDK. Xem thêm phụ lục để biết thêm chi tiết về các phiên bản Android.

Để tải về các packages, ta dùng Android SDK Manager, một ứng dụng có sẵn trong Android SDK.



1.4 Giả lập thiết bị

Android SDK bao gồm một giả lập (emulator) cho thiết bị Android chạy trên máy tính. Giả lập này cho phép chúng ta chạy ứng dụng Android mà không cần một thiết bị vật lý thật sự. Tuy nhiên, nó vẫn còn những hạn chế về giả lập camera và các cảm biến trọng lực, gia tốc, tiệm cận...

AVD Manager là một ứng dụng cho phép tạo và quản lý các thiết bị ảo (Android Virtual Devices - AVDs). Để mở AVD Manager, trong Android Studio chọn Tools > Android > AVD Manager hoặc chọn icon  trên thanh công cụ.

Android Debug Bridge (ADB) là công cụ hỗ trợ giao tiếp với thiết bị (cả giả lập lẫn thật). Nhờ có ADB mà ta có thể cài đặt ứng dụng, debugging, truyền tải file giữa thiết bị và máy tính... Phần phụ lục về ADB có trình bày một số thao tác đối với ADB.

Cần lưu ý AVDs không phải là lựa chọn duy nhất cho việc giả lập thiết bị trên Android. Có rất nhiều giải pháp thay thế khác trên thị trường như Genymotion, Droid4X, BlueStack... Thiết bị thật có lẽ là giải pháp hữu hiệu nhất vì ta sẽ có trải nghiệm ứng dụng thực tế tương tự với người dùng.

2 Bài tập thực hành

2.1 Cài đặt Android Studio Bundle

1. **Thời lượng:** Tùy thuộc vào tình trạng, cấu hình của máy. Sinh viên thực hiện tại nhà.
2. **Mục tiêu:** Học viên biết cách cài đặt Android Studio Bundle.
3. **Mô tả:** Sau khi tải về bộ Android Studio Bundle và các thành phần có liên quan, ta bắt đầu tiến hành cài đặt.
4. **Các bước thực hiện:**

Bước 1: Kiểm tra lại yêu cầu hệ thống về phần mềm cũng như phần cứng. Nếu đã thỏa tất cả yêu cầu thì sang bước 2.

Bước 2: Cài đặt Android Studio Bundle, khuyến cáo nên giữ nguyên các thông tin mặc định khi như đường dẫn cài đặt, các thành phần kèm theo.


Bước 3: Khởi động Android Studio. Nếu có lỗi phát sinh về JDK, xem thêm cách cài đặt JDK ở phần phụ lục.

2.2 Tải các thành phần cho Android SDK

1. **Thời lượng:** Tùy thuộc vào tình trạng, cấu hình của máy và kết nối Internet.
2. **Mục tiêu:** Học viên biết cách cài đặt, cập nhật các thành phần cho Android SDK.
3. **Mô tả:** Sử dụng SDK Manager để cài đặt, cập nhật các thành phần cho Android SDK.
4. **Các bước thực hiện:**

Bước 1: Khởi động SDK Manager

Có một vài cách để khởi động SDK Manager như sau:

- Trong Android Studio, chọn SDK Manager  trên thanh công cụ.
- Nếu không dùng Android Studio:
 - o Windows: Mở file *SDK Manager.exe* tại thư mục chứa Android SDK.
 - o Mac/Linux: Mở terminal và trở đến thư mục *tools* bên trong nơi Android SDK được cài đặt, sau đó chạy *android sdk*.

Bước 2: Chọn các thành phần và tải về

2.3 Tạo và cấu hình thiết bị ảo AVD

1. **Thời lượng:** 20 phút.

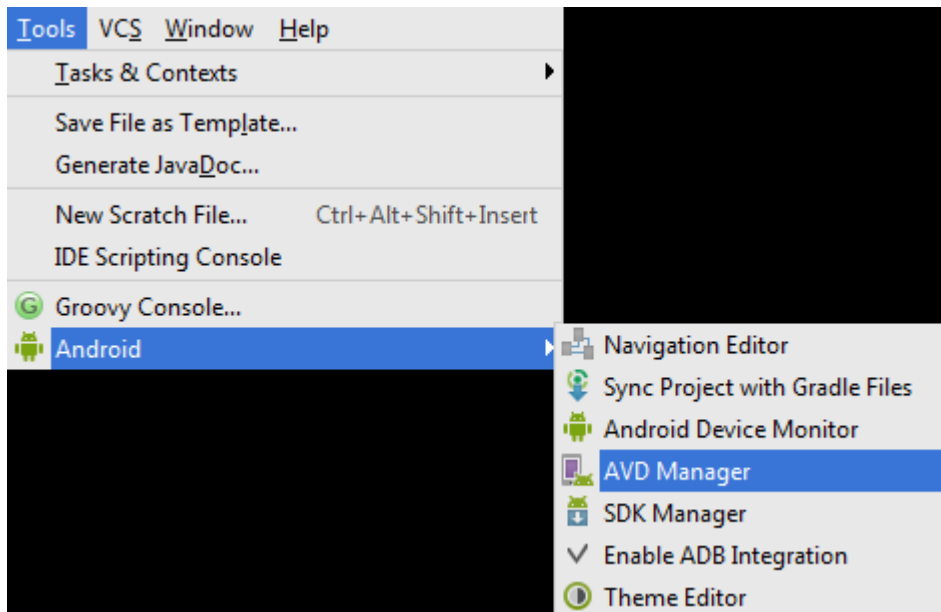
2. Mục tiêu:

- Học viên biết cách tạo thiết bị ảo AVD.
- Học viên biết cách cấu hình thiết bị ảo AVD.
- Học viên biết cách khởi động thiết bị ảo AVD.

3. Mô tả: Tạo thiết bị ảo và cấu hình bằng AVD Manager.

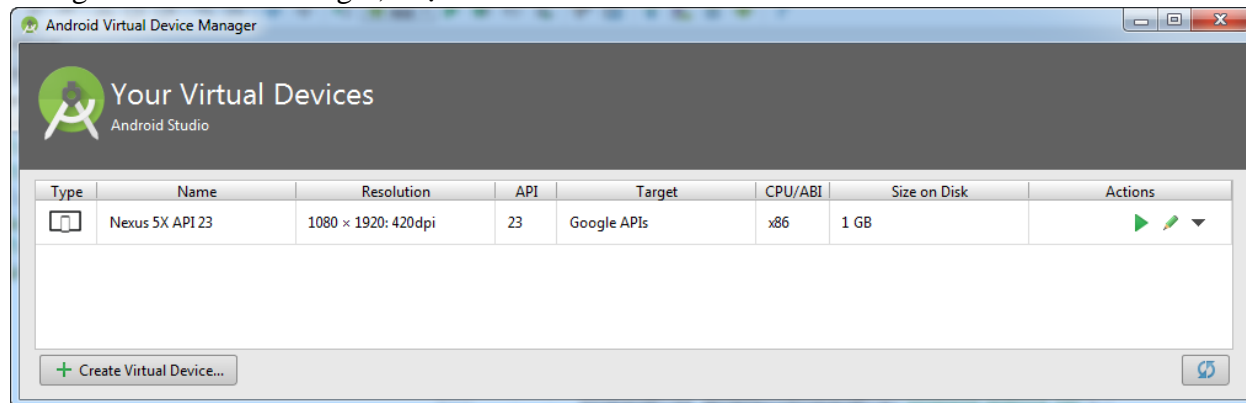
4. Các bước thực hiện:

Bước 1: Mở AVD Manager



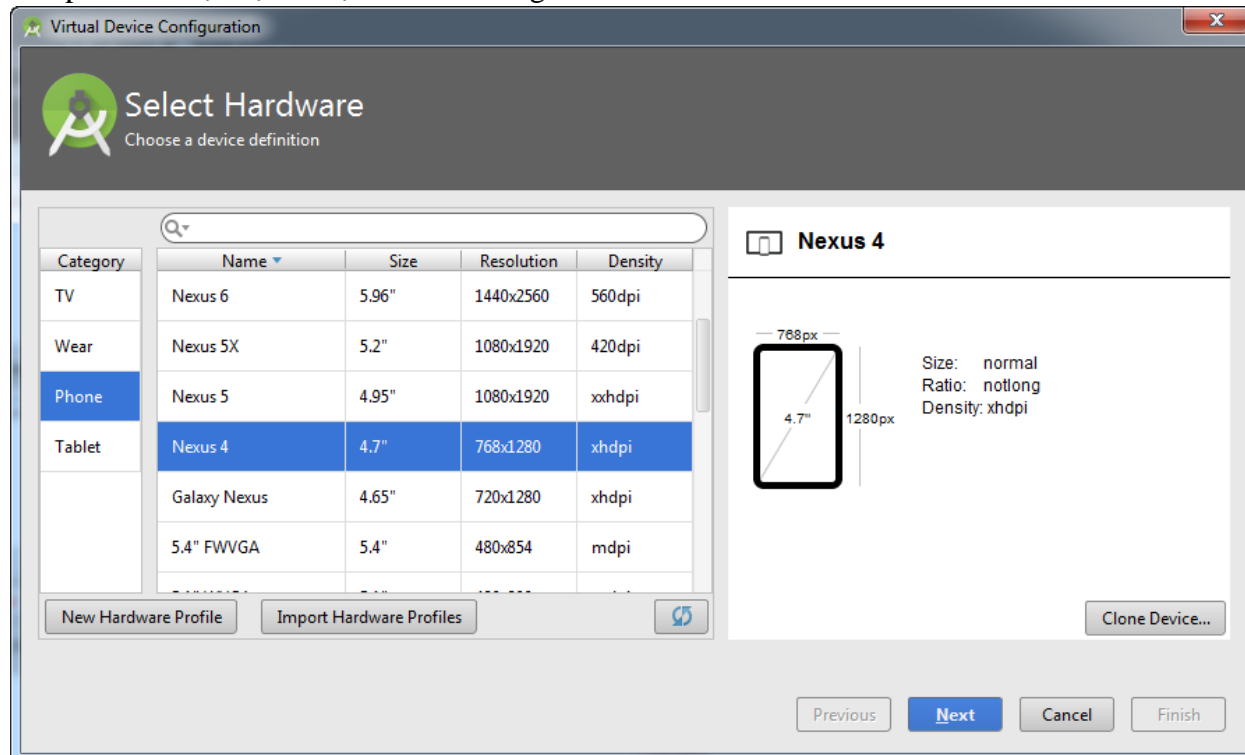
Bước 2: Tạo mới một AVD

- Trong cửa sổ AVD Manager, chọn Create Virtual Device...

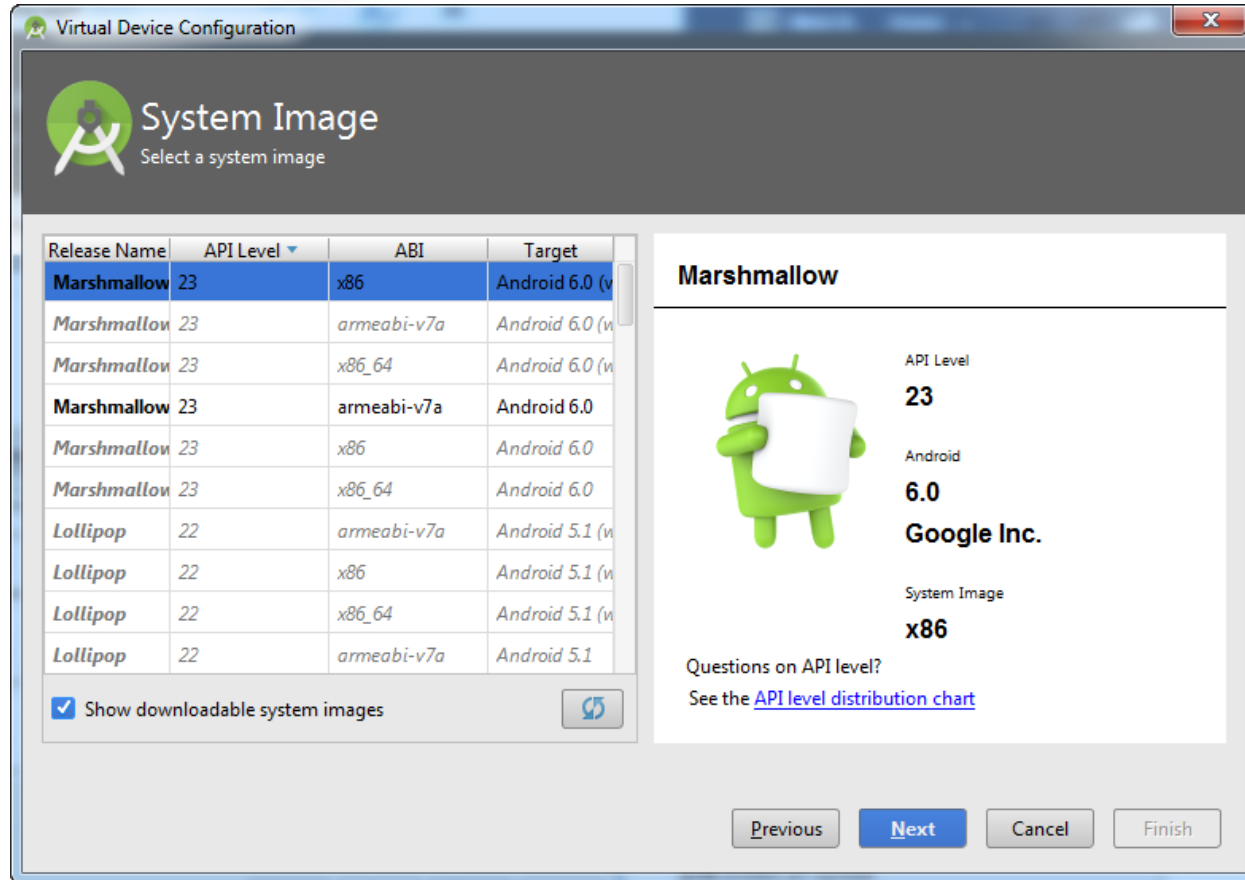


- Chọn loại thiết bị bất kỳ như TV, Phone, Tablet. Sau đó chọn thiết bị cụ thể. Nhấn Next.

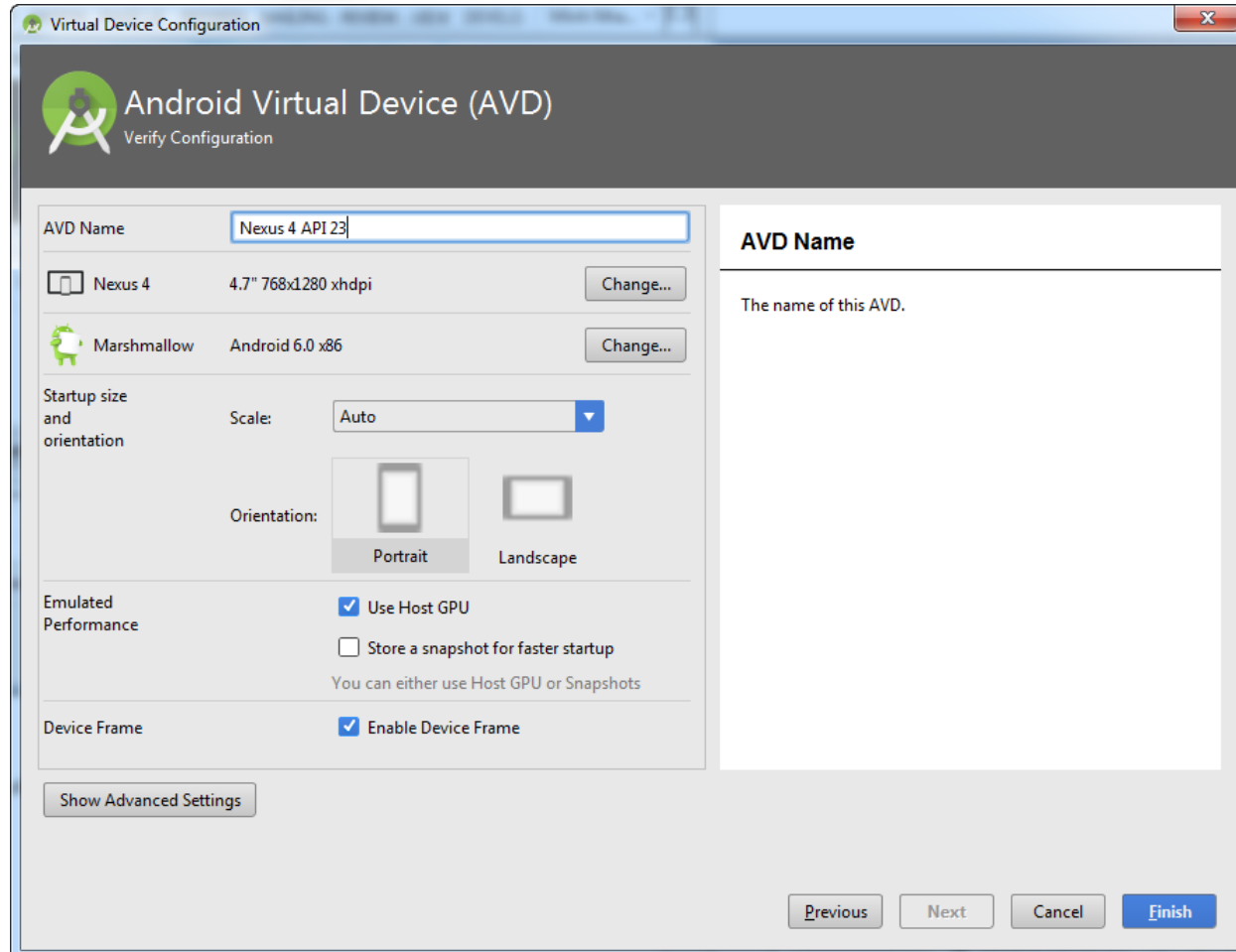
* Khuyến cáo chọn Nexus 4 hoặc Nexus 5X vì có độ lớn màn hình, độ phân giải vừa phải và được sự hỗ trợ nhiều từ Google.



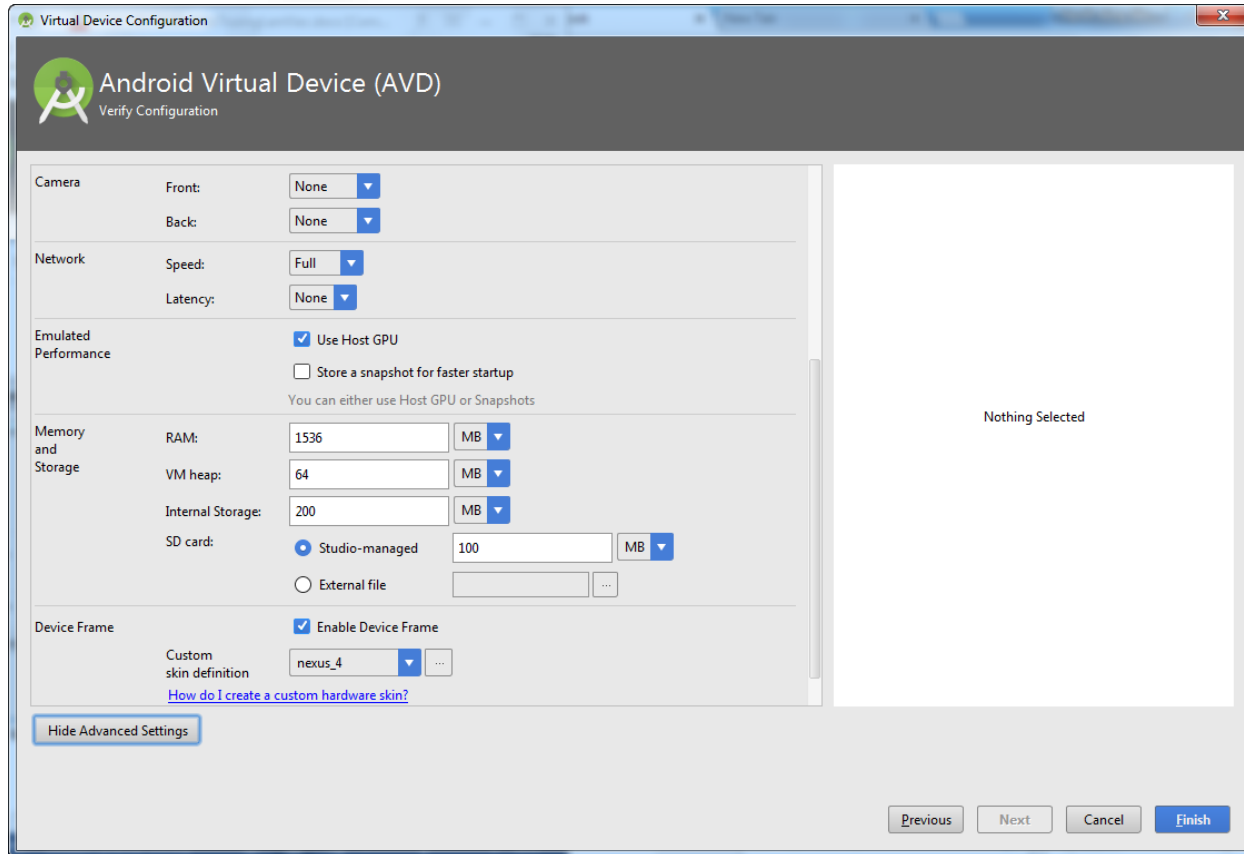
- Chọn cấu hình API level (phiên bản Android), CPU, ... mà ta mong muốn thiết bị chạy trên. Có thể tải về thêm những cấu hình chưa tồn tại trong Android SDK. Nhấn Next.



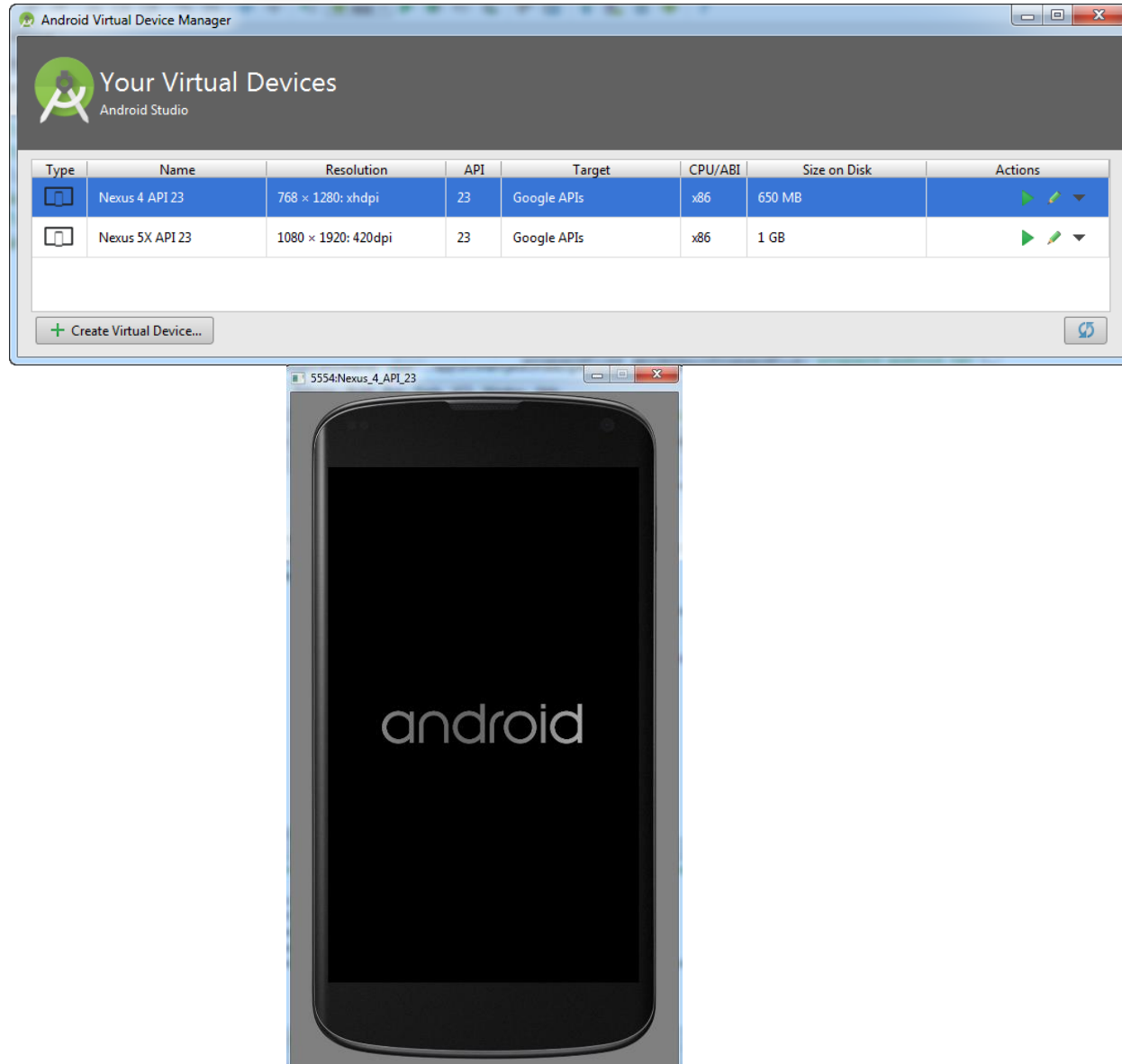
- Cấu hình các thuộc tính của thiết bị như tên, kích thước màn hình, API level, ...
 - o Emulated Performance: Thông thường chỉ được chọn một trong hai tùy chọn là dùng host GPU và dùng snapshot. Dùng host GPU sẽ giúp tăng hiệu suất của máy khi chạy, đặc biệt là về đồ họa. Dùng snapshot sẽ giúp tăng tốc độ khởi động của thiết bị ở những lần sau.



- Việc cấu hình có thể được mở rộng bằng cách nhấn vào nút Show Advanced Settings. Ta có thể cấu hình thêm về camera, network, memory, keyboard... Nhấn Finish để kết thúc quá trình tạo và cấu hình thiết bị ảo.



- Khởi động thiết bị ảo vừa tạo bằng cách chọn và nhấn nút ▶



2.4 Chạy ứng dụng trên thiết bị ảo

1. Thời lượng: 15 phút

2. Mục tiêu:

- Biết cách tạo một project trong Android Studio
- Biết cách chạy ứng dụng trên thiết bị ảo

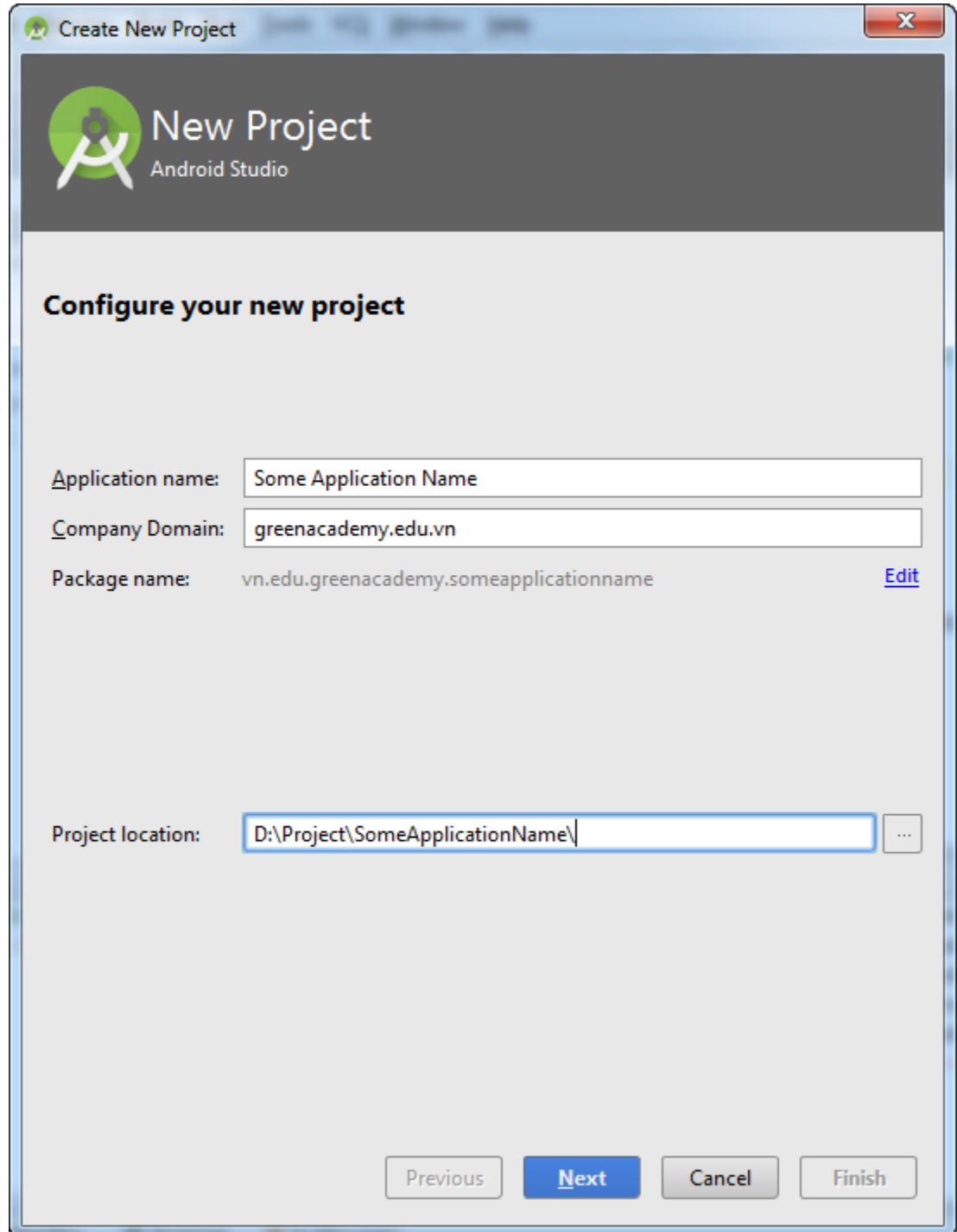
3. Mô tả: Học viên tạo một project trong Android Studio, với các thông tin của ứng dụng, các loại thiết bị mà ứng dụng hỗ trợ. Khi tạo project sẽ tự phát sinh một module kiểu Android

application với một Activity mặc định (kèm theo một layout tương ứng). Sau đó, học viên chạy thử ứng dụng trên thiết bị ảo đã được khởi động.

4. Các bước thực hiện:

Bước 1: Tạo Android Studio project

- Chọn menu File > New > New project...
- Nhập các thông tin
 - o Application name: Tên ứng dụng
 - o Company domain: Tên miền của công ty, cá nhân sở hữu ứng dụng
 - o Package name: Định danh của ứng dụng, thông thường là kết hợp của application name và company domain.
 - o Project location: Nơi chứa ứng dụng trên đĩa cứng máy tính



Create New Project

New Project
Android Studio

Configure your new project

Application name:

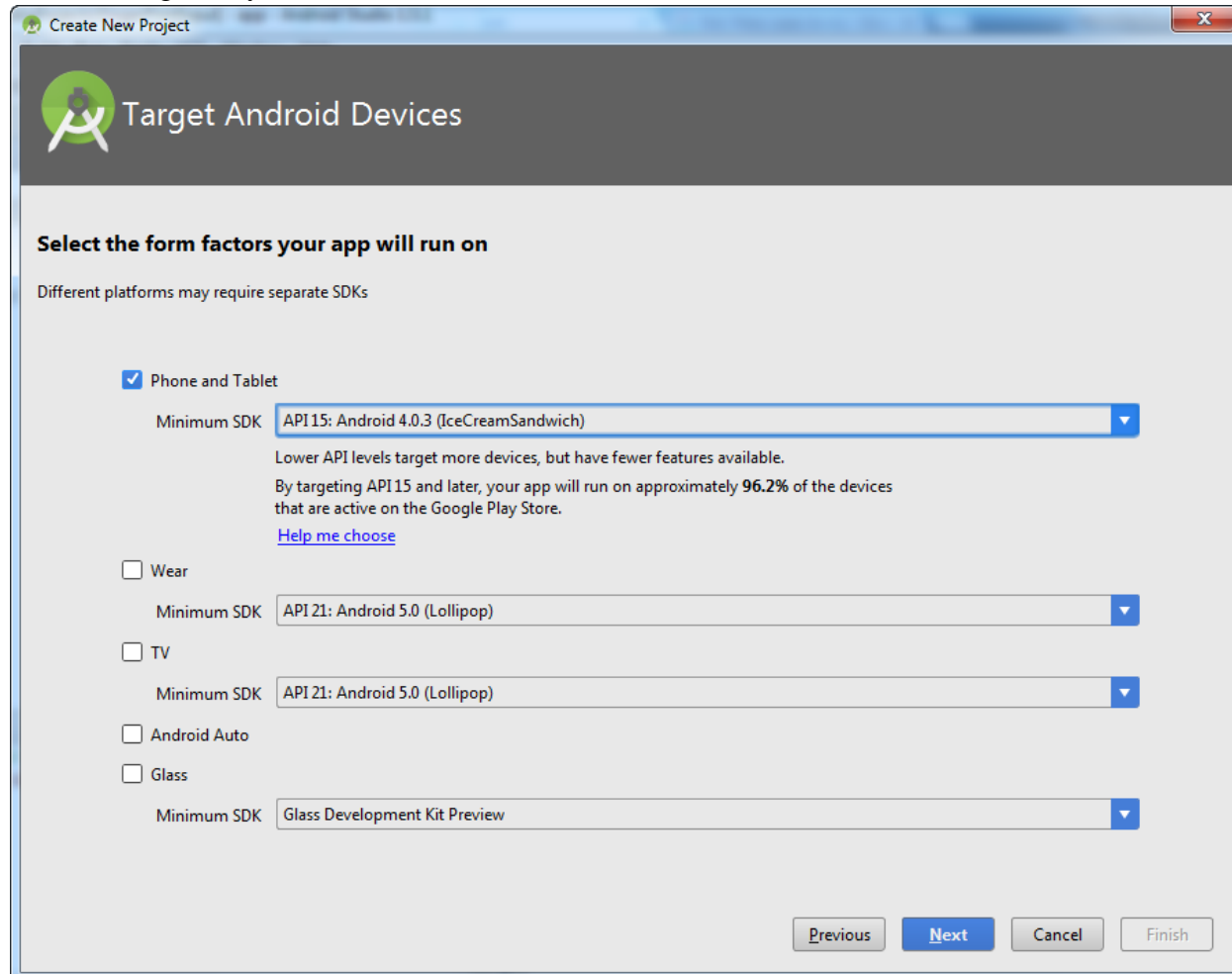
Company Domain:

Package name: vn.edu.greenacademy.someapplicationname [Edit](#)

Project location: ...

- Chọn form factors và minimum SDK tương ứng.
 - o Hiện tại ứng dụng Android có thể chạy trên nhiều loại thiết bị như phone, tablet, wear, TV, auto, glass...
 - o Minimum SDK là phiên bản hệ điều hành thấp nhất mà ứng dụng có thể chạy được. Thuộc tính này sẽ ảnh hưởng tới việc ứng dụng có tính tương

thích cao hay không. Ví dụ như trong hình, khi chọn giá trị API 15, ứng dụng có thể chạy trên 96.2% thiết bị trên thị trường (con số thống kê từ Google Play).



Create New Project

Target Android Devices

Select the form factors your app will run on

Different platforms may require separate SDKs

Phone and Tablet

Minimum SDK: API 15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available.
By targeting API 15 and later, your app will run on approximately **96.2%** of the devices that are active on the Google Play Store.
[Help me choose](#)

Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

TV

Minimum SDK: API 21: Android 5.0 (Lollipop)

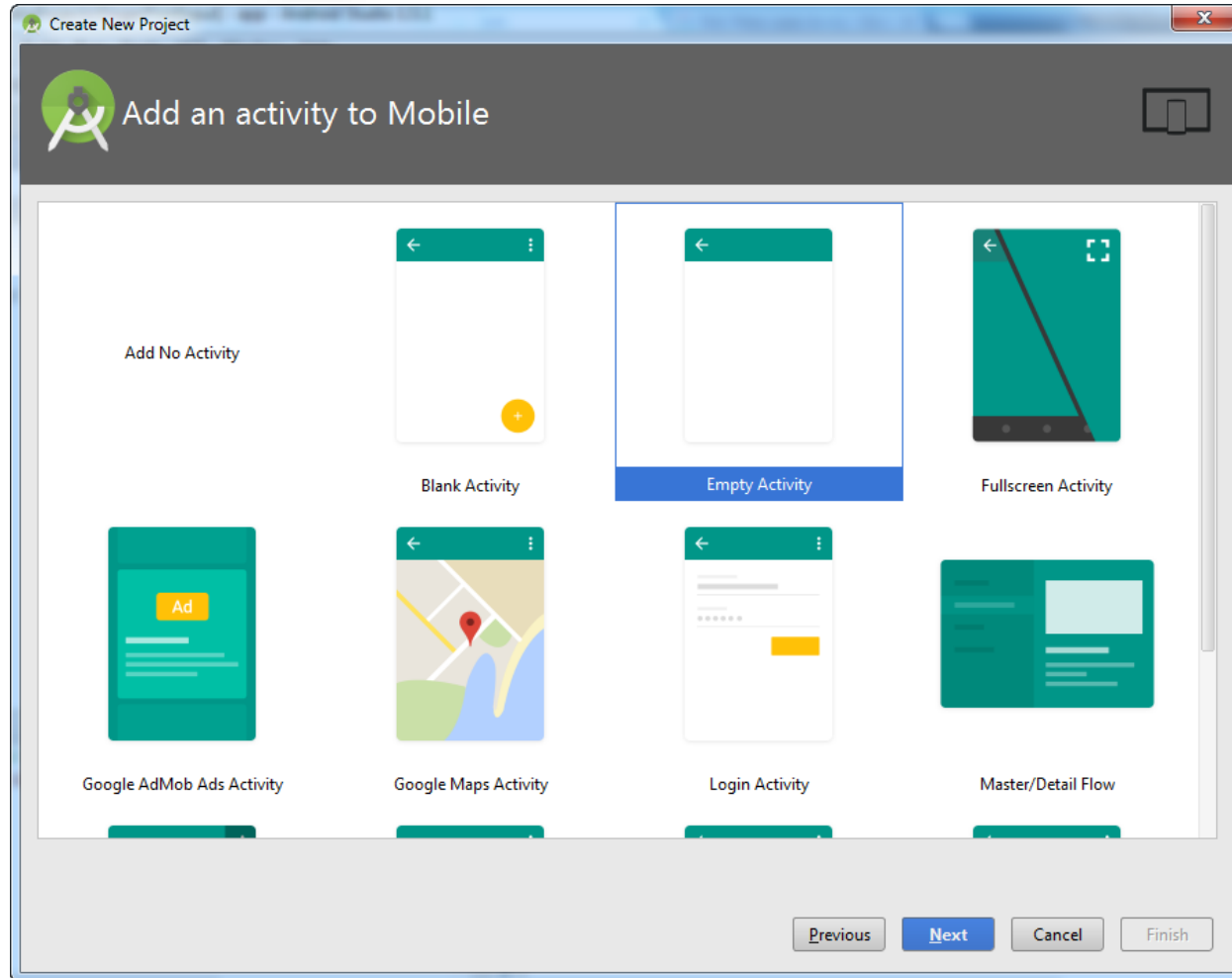
Android Auto

Glass

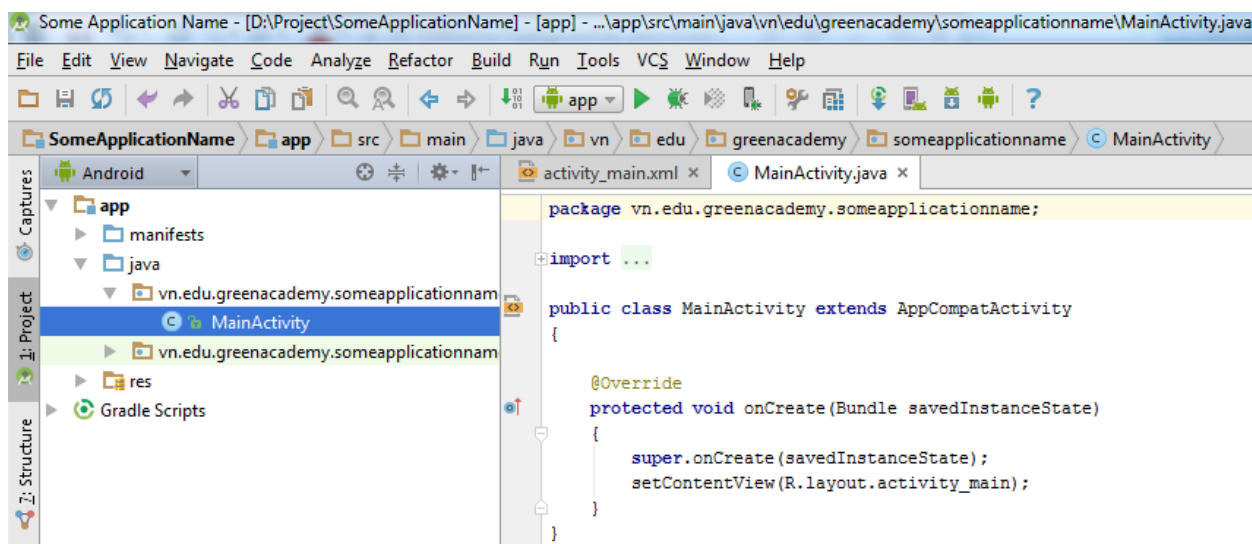
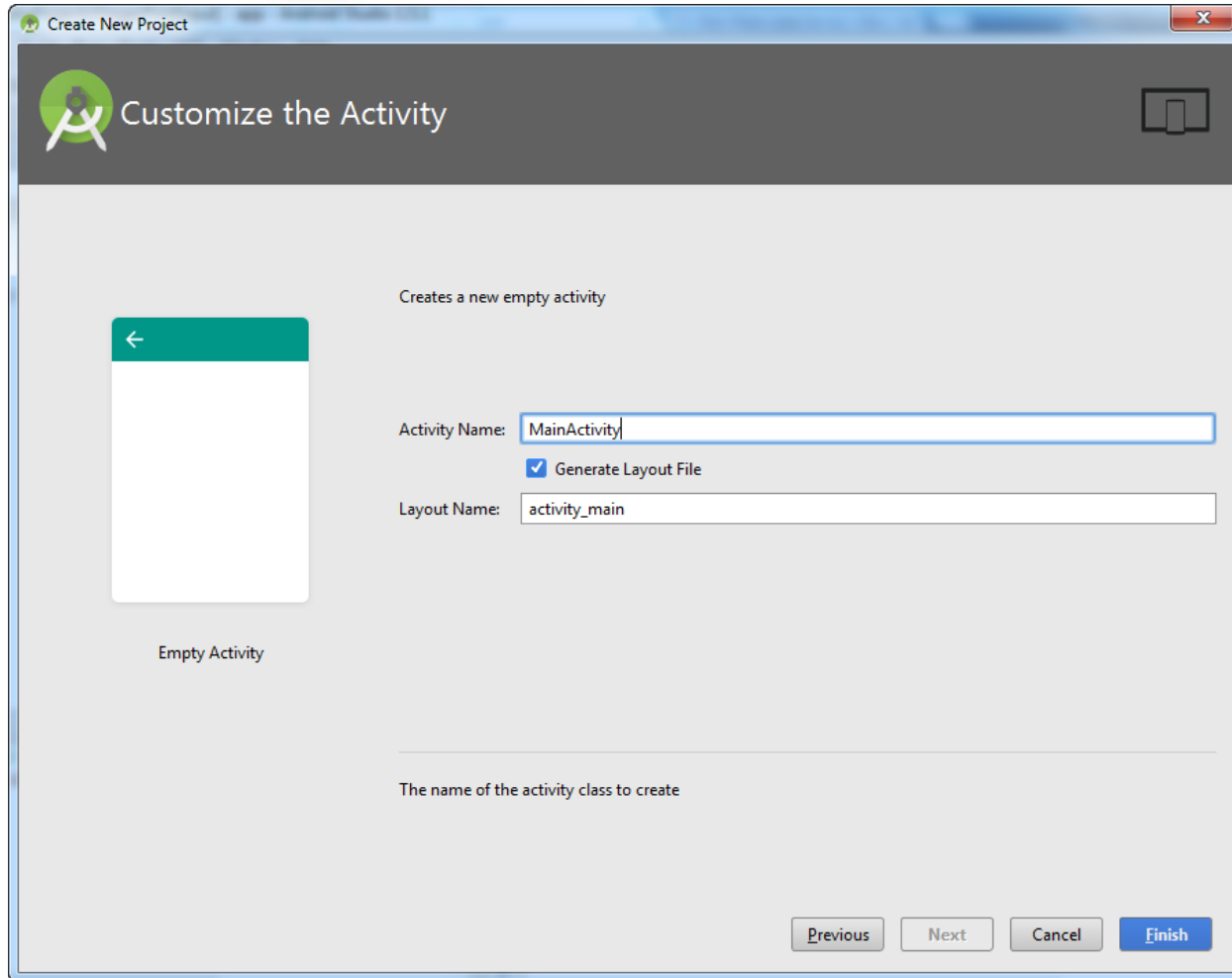
Minimum SDK: Glass Development Kit Preview

Previous Next Cancel Finish

- Tạo activity ban đầu. Ta chọn một trong các template có sẵn. Khuyến cáo chọn Empty Activity vì lý do kết quả tạo ra sẽ đơn giản.



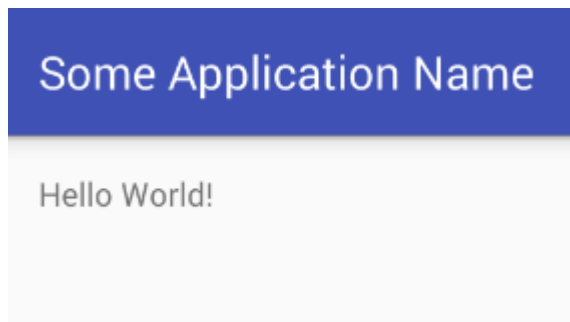
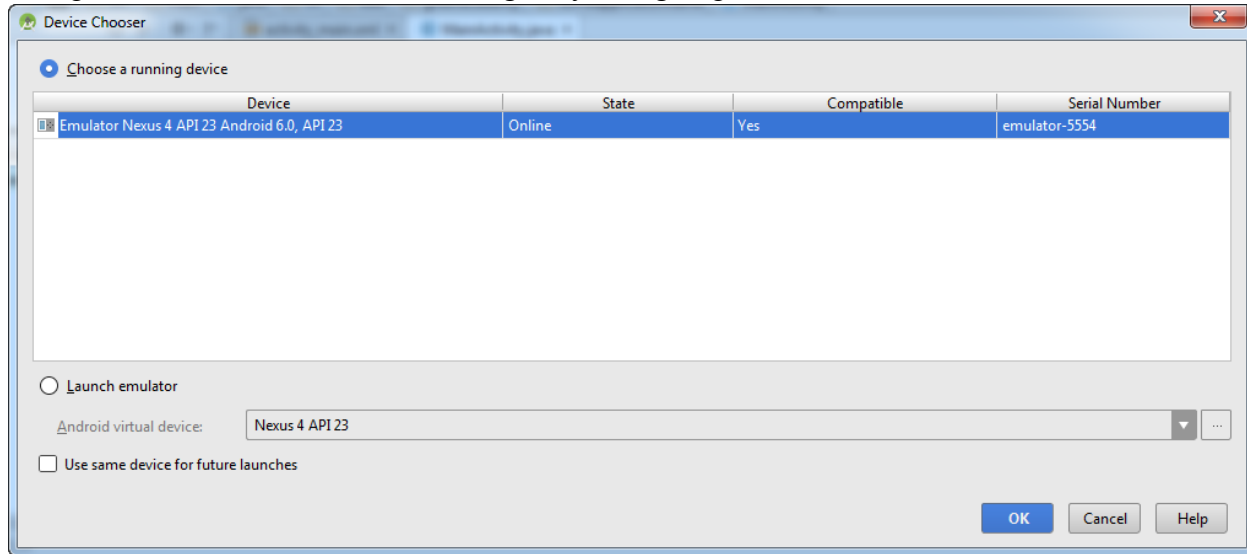
- Nhập các thông tin như tên Activity, tên layout kèm theo. Khuyến cáo giữ nguyên mặc định. Nhấn Finish để kết thúc việc tạo project.



Bước 2: Khởi động thiết bị ảo

Bước 3: Chạy ứng dụng trên thiết bị ảo

- Chọn Run > Run ‘...’, hoặc nhấn Shift + F10, hoặc nhấn vào nút ▶ trên thanh công cụ.
- Trong cửa sổ hiện ra, chọn thiết bị đang chạy tương ứng.



2.5 Chạy ứng dụng trên thiết bị thật

Làm tương tự như bài tập chạy ứng dụng trên thiết bị ảo, thay vì khởi động thiết bị ảo thì kết nối thiết bị thật vào máy tính qua cổng USB. Lưu ý là máy tính phải được driver của thiết bị và thiết bị đã bật cài đặt cho phép debugging thông qua USB với chế độ developer.

3 Phụ lục

3.1 Cài đặt JDK

Java Development Kit (JDK) là một bộ hỗ trợ phát triển ứng dụng Java, có thể cài đặt trên nhiều nền tảng như Windows, Mac OS X, Linux... Để cài đặt JDK cho máy chạy Windows, ta có thể thực hiện theo các bước sau:

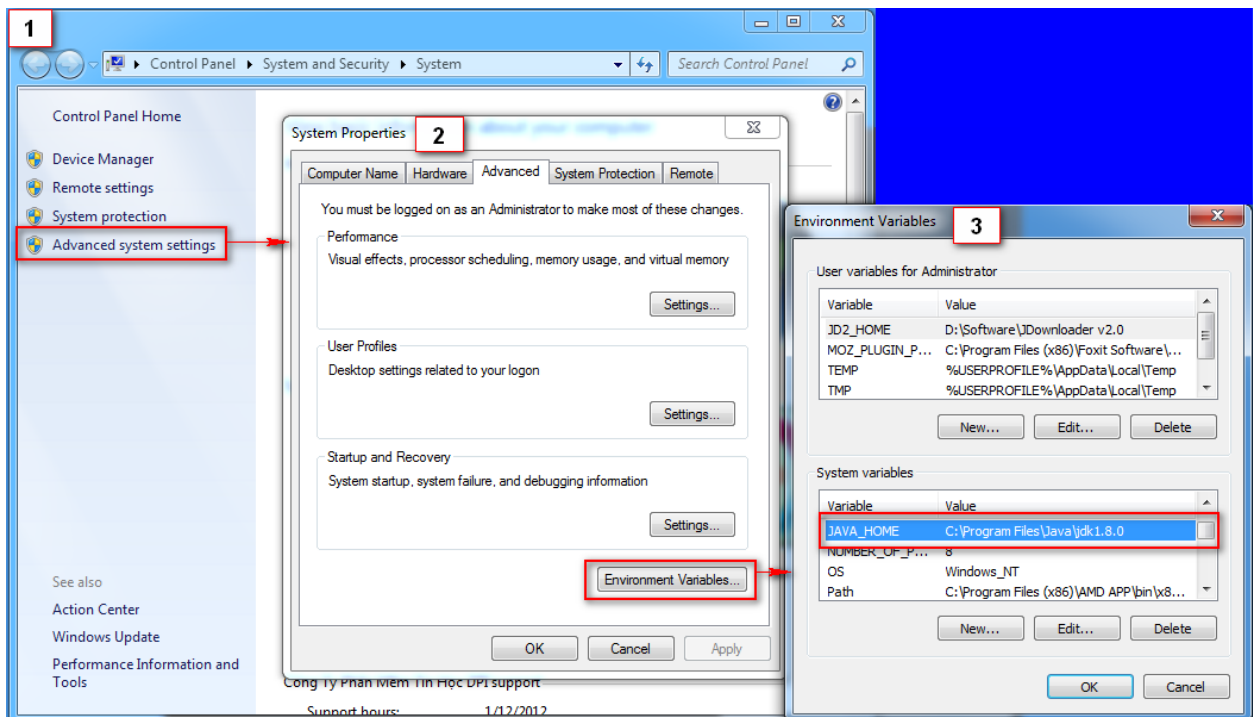
Bước 1: Tải JDK tại trang chủ <http://www.oracle.com/>

Bước 2: Cài đặt JDK

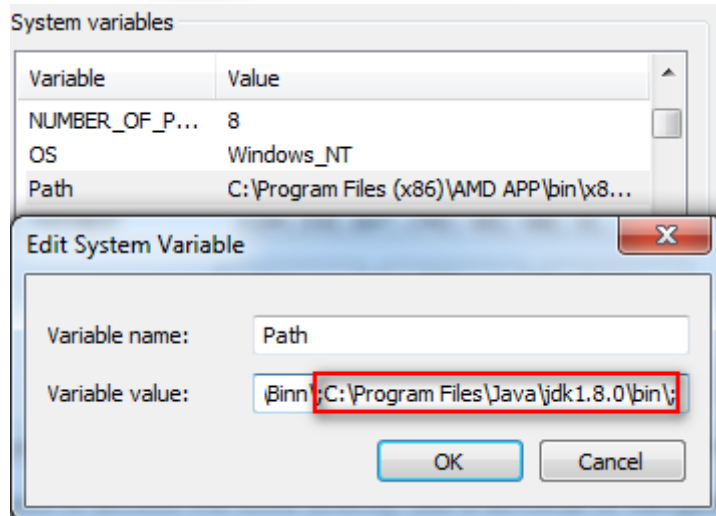
Khuyến cáo nên giữ nguyên các thông tin mặc định khi cài đặt như đường dẫn chứa thư mục JDK, các thành phần kèm theo.

Bước 3: Cài đặt các biến môi trường (Environment variable) JAVA_HOME và Path

- Chọn Start menu > Computer > System Properties > Advanced System Properties
- Mở Advanced tab > Environment Variables
- Chỉnh sửa giá trị của biến JAVA_HOME thành *đường-dẫn-tới-JDK*, ví dụ như *C:\Program Files\Java\jdk1.8.0*



- Chỉnh sửa giá trị của biến Path, thêm *;đường-dẫn-tới-thư-mục-bin-của-JDK;*, ví dụ như *;C:\Program Files\Java\jdk1.8.0\bin;*



Bước 4: Kiểm tra lại xem JDK đã cài đặt thành công hay chưa

Dùng câu lệnh `java -version` và `javac -version` để kiểm tra, nếu có kết quả tương tự hình bên dưới tức là JDK đã cài đặt thành công.

```
C:\Users\Administrator>java -version
java version "1.8.0_25"
Java(TM) SE Runtime Environment (build 1.8.0_25-b18)
Java HotSpot(TM) Client VM (build 25.25-b02, mixed mode, sharing)

C:\Users\Administrator>javac -version
javac 1.8.0
```

Đối với các nền tảng như Mac OS X, Linux, để cài đặt ta cũng thực hiện tương tự các bước như trên. Tuy nhiên, mỗi nền tảng đều có một đặc thù riêng, ta có thể tham khảo cách cài đặt JDK cụ thể cho từng nền tảng tại trang mô tả kỹ thuật của Oracle <http://docs.oracle.com>

3.2 Các phiên bản Android

Lịch sử phát triển của Android được bắt đầu từ phiên bản alpha năm 2007. Sau đó, phiên bản thương mại đầu tiên được ra đời - Android 1.0 - vào 09/2008. Kể từ đó, Android tiếp tục được phát triển bởi Google và Open Handset Alliance (OHA) và cho ra đời các phiên bản mới.

Hầu hết các phiên bản Android đều có một codename và một API level. Với góc độ người lập trình, thông thường ta sẽ quan tâm đến API level nhiều hơn. Dưới đây là danh sách các phiên bản Android cập nhật cho đến thời điểm 01/2016.

Code name	Phiên bản	API level
-----------	-----------	-----------

Code name	Phiên bản	API level
Marshmallow	6.0	API level 23
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4

Code name	Phiên bản	API level
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no code name)	1.1	API level 2
(no code name)	1.0	API level 1

Nguồn: <https://source.android.com/source/build-numbers.html>