**Contents**

[Android Studio: 1](#_Toc160430732)

[XML: 4](#_Toc160430733)

[Java: 5](#_Toc160430734)

[System: 8](#_Toc160430735)

Android Studio:

1. Cài Đặt?

* Vào Link
* <https://developer.android.com/studio>
* Click “Download Android Studio Giraffe” + cuộn xuống + Tick tôi đã đọc điều khoản + Click nút Download ở dưới + chờ nó tải + mở File vừa tải lên + tiến hành cài đặt mặc định
* Sau khi làm xong, trong thư mục “C:\Program Files” sẽ xuất hiện thư mục “Android”, chứa thư mục “Android Studio”, trong đây chứa thư mục “bin” chứa File “android64” và File “idea”, vào File “idea” dán dòng Code sau vào bất kì chỗ nào, để khỏi phải tải mấy cái không cần thiết

disable.android.first.run=true

* Mở File “android64” để chạy chương trình

1. Cấu Trúc Android Studio?

* Cửa sổ ban đầu gọi là cửa sổ tổng quan, liệt kê tất cả dự án bạn đã tạo
* Cửa sổ khi bạn vào 1 dự án gọi là cửa sổ dự án, chứa cấu trúc dự án
* 1 dự án thì bao gồm nhiều Module (ứng dụng)
* Khi Build dự án thì sẽ tạo ra các File APK ứng với các Module, điện thoại cài APK là chạy được Module

1. SDK (Software Development Kit) Manager?

* Để mở cửa sổ SDK Manager
* Tại cửa sổ tổng quan + vào Tab “Projects” + Click nút 3 chấm dọc góc phải trên + chọn “SDK Manager”
* Để cho Android Studio biết bạn chứa Android SDK ở thư mục nào
* Vào Tab “Languages & Frameworks” + chọn “Android SDK” + tại mục “Android SDK Location” + Click “Edit” + Click biểu tượng “Browse…” góc phải dưới + chọn thư mục chứa Android SDK + Click “OK” + Click “Next” 2 lần + nếu các gó Android SDK cần thiết chưa được tải thì nó sẽ tải + Click “Finish”
* Để tải vào hoặc gỡ ra các gói khỏi thư mục chứa Android SDK, tại mục “Android SDK” + vào Tab “SDK Platforms” hay “SDK Tools” tùy bạn muốn tải và gỡ cái gì + Tick “Show Package Details” để hiện thành phần trong các gói + Tick cái muốn tải và bỏ Tick cái đéo muốn + Click “Apply” + chờ nó cài và gỡ cài

1. Dự Án?

* Để tạo 1 dự án và đưa 1 Module mặc định
* Tại cửa sổ tổng quan + vào Tab “Projects” + Click “New Project” góc phải trên + chọn mẫu có sẵn, khuyên nên chọn “Empty Views Activity” để có Code Java Hello World + Click “Next” + tại mục “Name” điền tên Module, tên này là tiêu đề khi chạy Module trên điện thoại + tại mục “Package name” điền đường dẫn sẽ chứa File Java được chạy đầu tiên, đường dẫn này là con của thư mục “java”, ví dụ “com.foo.bar” + tại mục “Save location” điền đường dẫn đến thư mục lưu dự án + tại mục “Language” chọn “Java” + tại mục “Minimum SDK” chọn phiên bản Android với cơ chế càng thấp thì càng nhiều điện thoại dùng được ứng dụng của bạn, nhưng bù lại sẽ không sử dụng được các API đời mới + Click “Finish”, lưu ý tên thư mục Module mặc định là “app”
* Khi này cửa sổ tổng quan sẽ đóng và cửa sổ dự án sẽ mở
* Để trở về cửa sổ tổng quan
* Tại cửa sổ dự án + vào Tab “File” + chọn “Close Project”
* Để mở 1 dự án đã tạo
* Tại cửa sổ tổng quan + vào Tab “Projects” + Click vào dự án
* Mọi chỉnh sửa của bạn lên File sẽ được tự động lưu ngay lập tức

1. Cấu Trúc Thư Mục Chứa Dự Án?

* Trong thư mục dự án chứa các thư mục Module cũng như các thư mục và File dùng để Build khác
* Trong mỗi thư mục Module sẽ chứa File “build.gradle.kts”, tại File này + ở mục “android” + chỉnh thuộc tính “compileSdk” về phiên bản Android SDK bạn sẽ dùng để biên dịch khi Build, cần đảm bảo bạn đã tải gói Android SDK phiên bản này, không là khi Build nó tự động tải + chỉnh thuộc tính “namespace” thành Package chứa File Java sẽ được chạy đầu tiên khi chạy Module, ví dụ “com.foo.bar” + ở mục “defaultConfig” + chỉnh thuộc tính “targetSdk” về phiên bản Android mà bạn mong muốn Module sẽ chạy trên ngon nhất + chỉnh thuộc tính “minSdk” về phiên bản Android tối thiểu cần để chạy Module + chỉnh thuộc tính “applicationId” thành 1 chuỗi độc nhất, nó mới là cái Play Store dùng để nhận diện Module của bạn khi bạn đưa lên
* Trong mỗi thư mục Module còn có thư mục “src”, rồi bên trong “main”, trong đây có “java” là thư mục chứa Code Java xử lí Logic, “res” là thư mục chứa XML để hiện thị giao diện và File “AndroidManifest.xml” chỉ định File Java sẽ được chạy đầu tiên, Icon ứng dụng, …, khi chạy Module, thì nó dựa vào thuộc tính “namespace” đã do bạn chỉ định trước đó để tìm thư mục lưu trữ File Java được chạy đầu tiên trong thư mục “java”, ví dụ với “namespace” là “com.foo.bar”, thì sẽ tìm trong thư mục “java/com/foo/bar”, để chỉ định tên File Java được chạy đầu tiên, vào File “AndroidManifest.xml” + rồi chỉnh thuộc tính “android:name” của thẻ “activity” thành “.<Tên File Java Không Có Phần Mở Rộng>”, ví dụ “.MainActivity”
* Để tạo 1 Module mới
* Vào Tab “File” + chọn “New” + chọn “New Module…” + tại mục “Application/Library name” điền tên Module + tại mục “Module name” điền tên thư mục Module + tại mục “Package name” điền đường dẫn sẽ chứa File Java được chạy đầu tiên trong thư mục “java”, ví dụ “com.foo.bar” + tại mục “Language” chọn “Java” + tại mục “Minimum SDK” chọn giá trị cho thuộc tính “minSdk” + Click “Next” + chọn “Empty Views Activity” cho có Code sẵn + Click “Next” + tại mục “Activity Name” + điền tên File Java + tại mục “Layout Name” điền tên File XML + Click “Finish”

1. Cấu Trúc File Java Được Chạy Đầu Tiên?

* Đây là cấu trúc mặc định

package <Đường Dẫn Đến File Này>;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class <Tên File> extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

}

* Khi File này chạy, hàm “onCreate” sẽ được chạy đầu tiên, phải kế thừa “AppCompatActivity” để có thể sử dụng các phương thức của Class cha này trong việc xử lí giao diện

1. Điện Thoại?

* Trước tiên vào cửa sổ dự án
* Để chạy 1 Module trên điện thoại
* Đầu tiên máy tính và điện thoại phải kết nối với cùng 1 Wifi + tại biểu tượng hình Robot “Select Run/Debug Configuration” + chọn tên thư mục Module muốn chạy + tại biểu tượng ngay bên phải + chọn “Pair Devices Using Wi-Fi” + lấy điện thoại rồi vào cài đặt + cuộn xuống cuối vào “Hệ thống” + cuộn xuống cuối chọn “Tùy chọn cho nhà phát triển” + cuộn xuống bật “Gỡ lỗi qua Wi-fi” + chọn “CHO PHÉP” + vào nó + chọn “Ghép nối thiết bị bằng mã QR” + đưa điện thoại vào hình QR trên máy tính để quét + chờ kết nối + Click “Close” cửa sổ QR trên máy tính + Click nút Run ngay bên phải nút hồi nãy + chờ nó Build Module này ra APK và cài con mẹ nó luôn trên điện thoại của bạn và chạy nó
* Để ngắt không chạy Module nữa thì vào đa nhiệm điện thoại rồi xóa nó hoặc trên máy tính, Click nút Stop hình vuông đỏ
* Nếu nút Run bị bôi xám, nhấn “Ctrl” + “Shift” + “O” để tái kích hoạt lại
* Khi đang chạy Module, có thể Click biểu tượng “Apply Changes and Restart Activity” để chạy lại Module khi có chỉnh sửa File
* Mỗi khi chạy Module, những gì được in ra cùng với tin nhắn hệ thống sẽ ở Tab “Logcat” bên dưới

1. Chạy File Java Đơn Lẻ Được Không?

* Đéo

1. Build Tất Cả Module Ra APK?

* Vào Tab “Build” + chọn “Build Bundle (s) / APK (s)” + chọn “Build APK (s)” + chờ nó Build
* Khi này trong mỗi thư mục Module sẽ có thêm 1 thư mục “build”, vào thư mục này + vào “outputs” + vào “apk” + vào “debug” + ở đây File có tên

“<Tên Thư Mục Module Chứa Nó>-debug.apk” chính là File APK của Module tương ứng

1. Giả Lập Android Online Miễn Phí?

* Vào Link
* <https://appetize.io/>
* Click “Try Online Demo” + Click “Upload your app” góc phải trên + Click “Select a File” + chọn File APK muốn chạy thử + tại mục “WORK EMAIL” viết địa chỉ Email nhận Link, Link này sẽ dẫn đến 1 Tab trình duyệt khác đang chứa 1 chiếc điện thoại đã cài File APK của bạn và đang chạy nó + Click “Tap to Play” để mở điện thoại
* Bên dưới có các tùy chọn để chỉnh cấu hình điện thoại

XML:

1. Tác Dụng?

* Lưu trữ dữ liệu như JSON

1. Cấu Trúc?

* Dòng đầu tiên chỉ định phiên bản XML và cách mã hóa nó

<?xml version="1.0" encoding="UTF-8"?>

* Phần phía dưới là 1 cấu trúc cây, bao gồm chỉ 1 Node làm gốc, các Node tiếp theo bên trong làm con cháu, tên Node bất kì, cú pháp giống Tag trong HTML, có thể chỉ định thuộc tính bất kì, giá trị thuộc tính phải là String, bạn không thể chỉ định 2 thuộc tính có tên giống nhau cho 1 Node, nếu muốn làm điều này, phải đặt chúng trong 2 Namespace khác nhau, dùng dấu “:” như trong C++
* Ví dụ

<concac>

<conbuom>

<consau foo = "bar">

Abc123

</consau>

<consau bob:alice = "1" john:alice = "2">

Defgh12

</consau>

</conbuom>

<concho>

gaugau

</concho>

</concac>

* Ở đây “concac” làm Node gốc, “conbuom” và “concho” là thế hệ 1, …

1. Phần Mở Rộng File XML?

* “.xml”

1. XML Parser?

* Là trình đọc File XML và trích dữ liệu từ nó
* Để 1 chỉ định 1 đoạn mã trong File XML không phải là XML, mà coi nó là văn bản thông thường, ví dụ <concac> không còn có nghĩa là Node nữa, mà nó chỉ đơn thuần là chuỗi “<concac>”, ta có dùng cách này để viết chuỗi có chứa < và >

//<![CDATA[

<Văn Bản Thuần Túy>

//]]>

* Trong HTML thì cái này vô dụng, do đã có cách viết khác để biểu thị < và >

1. Đọc File XML Bằng Java?

* Đầu tiên, Import các thư viện sau

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.DocumentBuilder;

import org.w3c.dom.Document;

* Bước tiếp theo, tạo đối tượng Document ứng với File XML, từ đối tượng này ta có thể truy xuất ra các Node bên trong, phải đặt đoạn Code này vào khối Try Catch để xử lý mọi lỗi có thể xảy ra

try {

DocumentBuilderFactory <Factory> = DocumentBuilderFactory.newInstance();

DocumentBuilder <Builder> = <Factory>.newDocumentBuilder();

Document <Document> = <Builder>.parse(<Đường Dẫn Đến File XML Có Phần Mở Rộng>);

<Làm Gì Đó Với Document>

} catch (Exception <Error>) {

<Xử Lí Error>

}

Java:

1. Cài Đặt?

* Vào Link
* <https://www.oracle.com/java/technologies/downloads/>
* Vào Tab “Windows” + Click vào Link ứng với “x64 MSI Installer” + chờ nó tải + chạy File vừa tải để tiến hành cài đặt theo mặc định
* Sau khi làm xong, trong thư mục “C:\Program Files” sẽ xuất hiện thư mục “Java”, trong đây sẽ có thư mục có tên “jdk-<Phiên Bản>”, ví dụ “jdk-21”, tương ứng với phiên bản bạn tải, trong thư mục này có thư mục “bin” chứa các File lệnh để bạn chạy Code
* Thêm đường dẫn “C:\Program Files\Java\jdk-21\bin” vào biến hệ thống Path

1. Phần Mở Rộng File Java?

* “.java”

1. Cấu Trúc 1 File Java?

* 1 File toàn Class, nghĩa là bất kì Code nào cũng phải đặt trong 1 Class
* Chỉ có tối đa 1 Public Class, nếu có 1 Public Class, tên File Java phải trùng tên Public Class, còn nếu không có, thì đặt tên nào cũng được
* Nếu Class mà trùng tên với File Java chứa nó thì = Public Class nhưng không thể bị Import, hay nói cách khác là bị ẩn

1. Chạy 1 File Java?

* Bước 1, biên dịch ra các File Byte Code, mỗi Class trong File Java thành 1 File Byte Code, có phần mở rộng “.class”, nếu Class nằm trong Class thì tên File kiểu đường dẫn, dùng dấu “$” để ngăn cách, ví dụ Class “foo” chứa Class “bar” thì File Byte Code ứng với “bar” là “foo$bar.class”, nhập lệnh

javac <Đường Dẫn Đến File Java Có Phần Mở Rộng> -d

<Đường Dẫn Tới Thư Mục Chứa Các File Byte Code>

* Tên File Byte Code = tên Class
* Bước 2, chạy 1 File Byte Code nào đó, nhập lệnh

java -cp <Đường Dẫn Tới Thư Mục Chứa File Byte Code>

<Tên File Byte Code Không Phần Mở Rộng>

* Khi này, hàm có đúng cú pháp sau trong File Byte Code sẽ được chạy

public static void main(String[] <Tên Tham Số CMD>){

<Làm Gì Đó>

}

* Ví dụ trong File “foo$bar.class” có hàm trên, tức là trong File Java ban đầu ta đặt hàm trên vào Class bar, rồi đặt Class bar vào Class foo, thì khi chạy File Byte Code này, hàm này sẽ được chạy

1. Cú Pháp Java?

* Tương tự C++
* Khi ghi 1 số, được dùng dấu “\_” giữa các số, ví dụ 12\_34\_\_5\_678\_\_9.45\_7 = 123456789.457

1. Kiểu Dữ Liệu?

* Tương tự C++, chỉ khác là “bool” thành “boolean”, “char” kích thước 16 Bit, nghĩa là dùng bảng mã Unicode, “long” thì 64 Bit, “byte” = “char” trong C++, và không có kiểu dữ liệu không dấu hay kiểu dữ liệu kết hợp
* Khi ghi 1 số nguyên, thì kiểu mặc định của nó là “int”
* Để chỉ định kiểu của số đó là “long”, thêm “L” vào cuối số, ví dụ 12345L
* Khi ghi 1 số thập phân, thì kiểu mặc định của nó là “double”
* Để chỉ định kiểu của số đó là “float”, thêm “F” vào cuối, ví dụ 12.456F
* Đối với String, thì y chang như những kiểu khác
* Ví dụ

String foo = "bar";

1. Package Và Import?

* Ta chỉ có thể import 1 File Java khi nó là 1 Package
* Ta không thể chạy trực tiếp 1 Package
* Để chỉ định 1 File là 1 Package, đặt lệnh sau vào đầu File

package <Đường Dẫn Tới Thư Mục Chứa File Từ 1 Thư Mục Nào Đó>;

* Để import 1 Package cùng gốc

import <Đường Dẫn Tới Package Từ Gốc>;

* Ví dụ minh họa, ta có cấu trúc đường dẫn sau

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| foo |  |  |  |  |
|  | bar.java |  |  |  |
|  | far.java |  |  |  |
|  | boo |  |  |  |
|  |  | barl |  |  |
|  |  |  | fool |  |
|  |  |  |  | dark.java  package boo.barl.fool; |
|  |  |  | lord.java  pakage boo.barl; |  |
|  |  | haiz.java  package boo; |  |  |
|  | bob |  |  |  |
|  |  | burh |  |  |
|  |  |  | lmao.java  package bob.burh; |  |

* Từ File “bar.java”, ta có thể Import File “lord.java” bằng lệnh

import boo.barl.lord;

* Bây giờ, nếu biên dịch File “bar.java” sang Byte Code, cả File “bar.java” và “lord.java” đều sẽ được biên dịch sang “bar.class” và “lord.class”, nghĩa là chỉ có File Byte Code của Public Class là được tạo ra, nhưng nếu trong các Public Class có gọi đến các Class thường thì File Byte Code của các Class thường cũng được tạo ra, File “bar.class” sẽ phụ thuộc vào “lord.class”, nếu “lord.class” bị xóa hoặc dời đi đâu khác vị trí ban đầu là không chạy được “bar.class”, tương tự với các File Byte Code phụ thuộc vào nhau khác
* Từ trong File “dark.java”, ta vẫn có thể Import “haiz.java”, thì chúng cùng gốc “boo”

import boo.haiz;

* Để ý thấy ta đang Import các Public Class, còn các Class thường, ta không thể Import được, mà chỉ có thể dùng nội bộ trong cùng 1 File
* Với ví dụ trên, giả sử trong File “haiz.java”, Public Class “haiz” chứa hàm “haha”, thì sau khi Import, ta có thể gọi hàm này bằng cách gọi hàm thông thường

haiz.haha();

* Lưu ý nếu chỉnh sửa 1 Package thì phải xóa File Byte Code của Package đó trước khi gọi nó, vì Java sẽ lấy File Byte Code sẵn, xóa đi để cho nó biên dịch lại
* Ngoại lệ, nếu 2 File Java cùng chung 1 thư mục, thì chúng có thể sử dụng Class có tên trùng với tên File chứa nó của nhau trực tiếp, tức là gọi thẳng tên Class đó luôn, không cần biết có Public hay không

1. Subclass?

* Là Class kế thừa từ Class khác

class <Con> extends <Cha> {

<Các Thuộc Tính Và Phương Thức>

}

1. Modifier?

* Ở đây ta sẽ xét trong 1 File Java, Class, phương thức, và thuộc tính sẽ có 2 trạng thái là ẩn hoặc hiển thị, ẩn thì sẽ không truy cập được, hiển thị thì truy cập được, trạng thái ẩn hay hiển thị sẽ phụ thuộc vào vị trí ta đang đứng
* Mặc định chỉ hiển thị nếu thằng truy cập ở cùng File
* Public sẽ hiển thị bất kì chỗ nào kể cả ở File khác
* Private chỉ hiển thị nếu thằng truy cập ở trong cùng 1 Class cấp cao nhất
* Protected = mặc định nhưng bạn có thể truy cập nó trong Subclass kế thừa từ Class chứa thằng Protected ở File khác
* Class, phương thức, thuộc tính mà thuộc kiểu Static thì nó sẽ được dùng chung, nghĩa là khi tạo đối tượng mới thì sẽ không tốn thêm bộ nhớ, khi đó nếu 1 đối tượng thay đổi thì tất cả đối tượng khác cũng thay đổi, phương thức Static không thể bị ghi đè
* Class, phương thức, thuộc tính mà thuộc kiểu Final thì sẽ không thể bị kế thừa, ghi đè, và thay đổi giá trị

1. Override (Ghi Đè) Và Overload (Nạp Chồng)?

* Chỉ có thể ghi đè 1 phương thức khi nó là phương thức của Class cha mà Class hiện tại đang kế thừa, ghi đè tức là thay thế hoàn toàn, đặt lệnh sau ngay trước khi ghi đè

@Override

* Nạp chồng là việc ta dùng 1 tên hàm để thực hiện các nhiệm vụ khác nhau tùy vào tham số được Pass vào
* Ví dụ

int foo(){

return 1;

}

int foo(int a){

return a;

}

System:

1. In Cái Gì Đó Ra Rồi Xuống Dòng?

System.out.println(<Giá Trị>);