# **LỜI CẢM ƠN**

Để có thể hoàn thành tốt bài báo cáo thực tập cơ sở em xin gửi lời cảm ơn đến tất cả giảng viên trường Đại học Công nghệ thông tin & Truyền thông đã trang bị cho em những kiến thực thực sự có ích trong quá trình em học tập. Đặc biệt em xin gửi lời cảm ơn chân thành đến tất cả giảng viên khoa Công nghệ thông tin đã rất nhiệt tình giảng dạy, tạo môi trường học tập hứng thú trong thời gian em học vừa qua.

Em xin gửi lời cảm ơn sâu sắc đến ThS.Hoàng Thị Cành đã rất nhiệt tình, hướng dẫn, chỉ bảo và truyền đạt kinh nghiệm cho em trong thời gian em thực hiện thực tập.

Trong quá trình em làm bài báo cáo thực tập cơ sở không tránh khỏi những thiếu sót, em mong quý thầy cô góp ý để bài báo cáo của em được hoàn thiện tốt hơn.

Em xin chân thành cảm ơn.

|  |  |
| --- | --- |
|  | **Sinh viên**  **Nguyễn Mạnh Huỳnh** |

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

# LỜI NÓI ĐẦU

**Đề tài**

Nghiên cứu hệ điều hành Android và xây dựng ứng dụng Freaking Math.

**Lý do chọn đề tài**

Học sinh hiện nay đang rất phụ thuộc vào máy tính bỏ túi. Vì vậy em muốn xây dựng một trò chơi để tăng khả năng tính nhẩm giúp tăng hiệu quả học tập và không quá phụ thuộc vào máy tính bỏ túi.

**Yêu cầu tổng quát**

Để xây dựng một ứng dụng như đã nêu thì có một số yêu cầu tổng quát sau:

* Giao diện: đơn giản, dễ dàng sử dụng.
* Câu hỏi: demo ứng dụng với những phép toán cộng trong phạm vi từ 0 đến 20
* Độ khó: độ khó sẽ tăng dần khi trả lời đúng

Trong báo cáo này, em xin phép chỉ trình bày quá trình tìm hiều, xây dựng ứng dụng Freaking Math trên hệ điều hành Android.

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Ngôn ngữ Java

### Java là gì?

Java là một ngôn ngữ lập trình được Sun Microsystems giới thiệu vào những năm cuối của thập niên 90. Từ đó nó đã trở thành một công cụ lập trình viên chuyên nghiệp, java được xây dựng trên nền tảng của C và C++ do vậy nó được sử dụng các cú pháp của C và các đặc trưng của C++.

Vào năm 1991, một nhóm các kỹ sư của Sun Microsystems có ý định thiết kế một ngôn ngữ lập trình để điều khiển các thiết bị điện tử như tivi, máy giặt, lò nướng… Mặc dù C và C++ có khả năng làm việc này nhưng trình biên dich lại phụ thuộc vào từng loại CPU. Trình biên dịch thường phải tốn nhiều thời gan để xây dựng nên rất đắt, vì vậy để mỗi loại CPU có một trình biên dịch riêng là rất tốn kém. Do đó nhu cầu thực tế đòi hỏi một ngôn ngữ chạy nhanh, gọn, hiểu quả và độc lập thiết bị tức là có thể chạy trên nhiều loại CPU khác nhau, dưới các môi trường khác nhau. “Oak” đã ra đời vào năm 1995 được đổi tên thành Java. Mặc dù mục tiêu ban đầu không phải cho Internet nhưng do đặc trưng không phụ thuộc thiết bị nên Java đã trở thành ngôn ngữ lập trình cho Internet. Java là ngôn ngữ lập trình được phát triển từ ngôn ngữ lập trình C/C++. Ngôn ngữ lập trình Java có một số đặc trưng tiêu biểu: đơn giản, hướng đối tượng, độc lập phần cứng và hệ điều hành, mạnh mẽ, bảo mật, phân tán, đa luồng và linh động.

Java là ngôn ngữ lập trình hướng đối tượng, mọi chương trình viết trên Java đều phải được xây dựng trên các đối tượng. Nếu trong C/C++ ta có thể tạo ra các hàm (chương trình con không gắn với đối tượng nào) thì trong Java ta chỉ có thể tạo ra các phương thức (chương trình con gắn liền với một lớp cụ thể).

Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu:

* Kiểu dữ liệu phải khai báo tường minh.
* Java không sử dụng con trỏ và các phép toán con trỏ.
* Java kiểm tra tất cả các truy nhập đến mảng, chuỗi thực thi để đảm bảo rằng các truy nhập đó không ra ngoài giới hạn kích thước.
* Trong chương trình Java, lập trình viên không phải bận tâm đến việc cấp phát bộ nhớ. Quá trình cấp phát, giải phóng được thực hiện tự động.
* Cơ chế bẫy lỗi của Java giúp đơn giản hóa quá trình xử lý lỗi và hồi phục sau lỗi.

“Tại sao lại lấy tên là Java? - Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là "Oak" (có nghĩa là "Cây sồi" - 1991), nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới -  và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java.”

### Ứng dụng thực tế của Java

Theo thông báo của Orecle hiện nay có khoảng 3 tỷ thiết bị chạy Java và con số này đang tăng dần theo từng phút vậy chúng ta có thể thấy được Java ở đâu? Trọng thực tế có rất nhiều nơi **Java được ứng dụng**, bắt đầu từ trang web thương mại điện tử đến các ứng dụng Android, từ ứng dụng khoa học đến các ứng dụng tài chính như hệ thống giao dịch điện tử, từ các trò chơi như Minecraft đến các ứng dụng máy tính để bàn như Eclipse, Netbeans và IntelliJ, từ một thư viện mã nguồn mở cho các ứng dụng J2ME, v.v. Hãy đi vào chi tiết nhưng noi mà Java được ứng dụng.

**Java làm ứng dụng Android.** Dễ thấy nhất là Java được ứng dụng trong viết các ứng dụng trên hệ điều hành Android. Theo thông kế của Tập đoàn Dữ liệu Quốc Tế (IDC) số lượng điện thoại HDH Android trên toàn thế giới là 85,1% (2017) và dự đoán nắm 2021 là 85,3% và hầu như các ứng dụng trên các điện thoại này đều được việt bằng Java. Chỉ cần có một chiếc Smatphone Android là có thể thấy được “một thế giới Java” ngay trong lòng bàn tay. Vài năm trở lạiđây Android đã cung cấp nhiều sự thúc đẩy cần thiết và ngày nay nhiều lập trình viên Java là nhà phát triển Ứng dụng Android. Có thể Android ứng dụng JVM theo cách khác nhau và cách đóng gói khác nhau, nhưng mã vẫn được viết bằng Java. Có một sự cạnh tranh khi Google hỗ trợ Kotlin như [ngôn ngữ First class trên Android](https://niithanoi.edu.vn/kotlin-vs-java-cuoc-chien-phat-trien-ung-dung-android-vi-dai.html). Tuy nhiên, lập trình ứng dụng Android với Java vẫn cực kỳ phổ biến.

**Java làm hệ thống giao dịch trong ngành Dịch vụ tài chính.** Java có vị trí rất lớn trong Dịch vụ tài chính. Rất nhiều ngân hàng đầu tư toàn cầu như Goldman Sachs, Citigroup, Barclays, Standard charted và các ngân hàng khác sử dụng Java để viết hệ thống giao dịch điện tử văn phòng trước và sau, hệ thống thanh toán và xác nhận, dự án xử lý dữ liệu và một số dự án khác. Java chủ yếu được sử dụng để viết các ứng dụng phía máy chủ, nhận dữ liệu từ một máy chủ (upstream), xử lý và gửi cho nó sang quá trình khác (downstream). Java Swing cũng rất phổ biến để tạo client GUI cho các nhà giao dịch. Tuy nhiên, hiện tại C # đang nhanh chóng chiếm lĩnh thị phần này.

**Java làm ứng dụng web.** Java chiếm vị trí rất lớn trong các ứng dụng web và website thương mại điện tử. Có rất nhiều RESTfull Service được tạo bằng Spring MVC, Struts 2.0 và các Framework tương tự. Ngay cả các ứng dụng web đơn giản dựa trên Servlet, JSP và Struts cũng khá phổ biến trong các dự án khác nhau của chính phủ. Nhiều cơ quan chính phủ, y tế, bảo hiểm, giáo dục, quốc phòng và một số bộ phận khác có ứng dụng web của họ được xây dựng bằng ngôn ngữ Java. Các Website làm bằng Java:

1. <https://www.linkedin.com/>
2. <https://www.aliexpress.com/>
3. <https://www.ebay.com/>

v.v.

**Java làm phần mềm phát triển.** Nhiều phần mềm và công cụ phát triển hữu ích được viết và phát triển bằng Java, ví dụ: Eclipse, InetelliJ Idea và Netbans IDE. Có thể nói rằng, chúng cũng là những ứng dụng Desktop được sử dụng nhiều nhất được viết bằng Java. Mặc dù đã có lúc Swing rất phổ biến để viết các ứng dụng lớn, chủ yếu trong lĩnh vực dịch vụ tài chính và ngân hàng đầu tư. Ngày nay. Java FX đang trở nên phổ biến nhưng vẫn không phải là sự thay thế của Swing và C # gần như đã thay thế Swing trong lĩnh vực Tài chính.

**Java làm ứng dụng J2ME.** Mặc dù sự ra đời của iOS và Android gần như đã giết chết thị trường J2ME, nhưng vẫn có một thị trường của cấp thấp sử dụng J2ME. Đã có một thời khi hầu hết tất cả các trò chơi, ứng dụng có sẵn trong Android được viết bằng MIDP và CLDC, đó là một phần của nền tảng J2ME. Mặc dù ứng dụng J2ME trên điện thoại di động hầu như không còn. Nhưng J2ME vẫn phổ biến trên các sản phẩm như Blu-ray, Cards, Set top box, v.v ... Một trong những lý do khiến **WhatsApp** trở nên phổ biến là vì nó cũng có sẵn trong J2ME cho tất cả những chiếc điện thoại Nokia vẫn còn khá lớn.

**Java làm lập trình nhúng.** Java cũng có thị phần rất lớn trong các sản phẩm cần đến lập trình nhúng. Nó cho thấy nền tảng này có khả năng như thế nào, bạn chỉ cần 130 KB để có thể sử dụng công nghệ Java (trên thẻ thông minh hoặc cảm biến). Ban đầu Java được thiết kế cho các thiết bị nhúng. Trên thực tế, đây là một phần của chiến dịch "viết một lần, chạy ở bất cứ đâu" của Java.

**Java trong công nghệ Big Data.** Hadoop và các công nghệ dữ liệu lớn khác cũng đang sử dụng Java theo cách này hay cách khác. Ví dụ: HBase và Accumulo dựa trên Java (nguồn mở) cũng như ElasticSearch. Bởi vì Java không thống trị lĩnh vực này, chúng có những công nghệ phù hợp hơn như MongoDB được viết bằng C ++. Java có tiềm năng để có được sự chia sẻ lớn trên lĩnh vực đang phát triển này nếu Hadoop hoặc ElasticSearch phát triển lớn hơn.

### Những đặc điểm cơ bản của Java

Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là **"Write Once, Run Anywhere"** (*Viết một lần, chạy mọi nơi*), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, chúng ta viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS... Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:

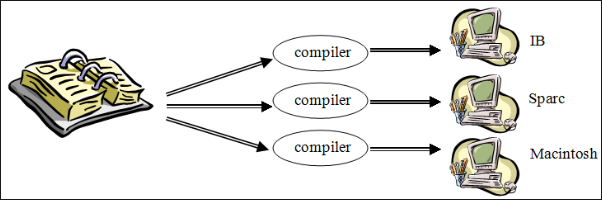
1. Đơn giản

Đặc điểm đầu tiên cũng như mục đích cơ bản của Java là tính đơn giản. Những người thiết kế mong muốn phát triển một ngôn ngữ dễ học và quen thuộc với đa số người lập trình. Do vậy Java loại bỏ các đặc trưng phức tạp của C và C++ như:

* Loại bỏ thao tác con trỏ, thao tác định nghĩa chồng toán tử.
* Không cho phép đa kế thừa mà sử dụng các giao diện.
* Một điều chứng minh cho thấy Java khá đơn giản là kích thước của bộ biên dịch cơ bản và lớp hỗ trợ là vô cùng nhỏ.

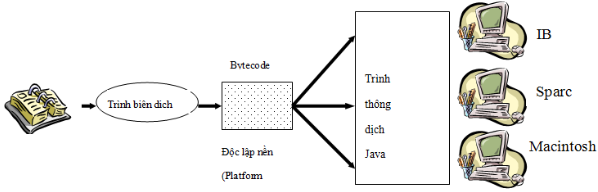
1. Độc lập phần cứng và hệ điều hành

Đối với các ngôn ngữ lập trình truyền thống như C/C++, phương pháp biên dịch được thực hiện như sau:

*Hình 2.1: Các biên dịch truyền thống*

Với mỗi nền phần cứng khác nhau, có một trình biên dịch khác nhau để biên dịch mã nguồn chương trình cho phù hợp với nền phần cứng ấy. Do vậy, khi chạy trên một nền phần cứng khác nhau bắt buộc phải biên dịch lại mã nguồn.

Đối với các dạng chương trình viết bằng Java, trình biên dịch Java sẽ biên dịch mã nguồn thành dạng bytecode. Sau đó, khi chạy chương trình trên các nền phần cứng khác nhau, máy ảo Java dùng trình thông dịch thông dịch Java để chuyển mã bytecode thành dạng chạy được trên các nền phần cứng tương ứng. Do vậy, khi thay đổi nền phần cứng, không phải biên dịch lại mã nguồn Java.

*Hình 2.2: Dịch chương trình Java*

1. Phân tán

Java được thiết kế hỗ trợ các ứng dụng phân tán bằng các lớp mạng (java.net). Hơn nữa Java hỗ trợ nhiều nền chạy khác nhau nên chúng được sử dụng rộng rãi như là công cụ phát triển trên Internet – nơi sử dụng nhiều nền khác nhau.

1. Bảo mật

Java được viết cho các ứng dụng chạy trên các môi trường phân tán. Do đó Java đã được thiết kế sao cho có khả năng sáng tạo được một hệ thống không virut, không bị phá rối. Kỹ thuật xác minh của Java dựa vào cách mã hóa theo khóa công khai.

1. Mạnh mẽ

Java là ngôn ngữ yêu cầu chặt chẽ về kiểu dữ liệu

* Kiểu dữ liệu phải khai báo tường minh.
* Java không sử dụng con trỏ và các phép toán con trỏ.
* Java kiểm tra tất cả các truy cập đến mảng, chuỗi khi thực thi để đảm bảo rằng các truy cập đó không ra ngoài giới hạn kích thước.
* Quá trình cấp phát giải phóng bộ nhớ được tự động.
* Cơ chế bẫy lỗi của Java giúp đơn giản hóa quá trình xử lý lỗi và hồi phục sau lỗi.

1. Thông dịch

Java là một ngôn ngữ thông dịch. Các ứng dụng của java có thể dễ dàng chạy trên các máy tính với các nền phần cứng khác nhau như Intel, Sun Mycrosystems… chỉ cần đi kèm với các ứng dụng dó là bộ thông dịch, bộ gỡ rỗi và nhất là bộ thư viện kiểm soát mã thực thi (Runtime Library). Khi đó các lập trình viên sẽ được tự do không còn lệ thuộc vào nội dung giới hạn của ngôn ngữ máy, mà chỉ chú ý tới các thư viện module dùng chung phục vụ cho ứng dụng.

1. Lập trình hướng đối tượng

Java là một ngôn ngữ lập trình hướng đối tượng. Ngôn ngữ lập trình hướng đối tượng có khả năng sau:

* Mô phỏng thế giới thực một cách tự nhiên bởi các đối tượng và mối quan hệ giữa chúng, thuận tiện cho việc thiết kế hệ thống phức tạp.
* Thừa kế mã có sẵn một cách dễ dàng, giúp tiết kiệm công sức và nâng cao năng suất của người lập trình, dễ bảo trì, dễ nâng cấp và mở rộng.

**Tính trường tượng hóa.** Con người đã đơn giản hóa các vẫn đề phức tạp thông qua sự trừu tượng hóa. Các tốt nhất để nắm vững ký thuật trừu tượng là dùng hệ thống phân cấp. Điều này cho phép bạn phân lớp các thành phần có ý nghĩa của cả hệ thống phức tạp, chia nhỏ chúng thành những phần đơn giản có thể quản lý được. Tất cả các ngôn ngữ lập trình hướng đối tượng đều có các cơ chế cho phép bạn triển khai các mô hình hướng đối tượng đó là tính đóng gói, kế thừa và đa hình.

**Tính đóng gói.** Đây là cơ chế dùng một vỏ bọc kết hợp phần dữ liệu và các thao tác trên dữ liệu đó (phần mã lệnh) thành một thể thống nhất, tạo nên sự an toàn, tránh việc sử dụng không đúng thiết kế, bảo vệ cho mã lệnh và dữ liệu chống việc truy xuất từ những đoạn mã lệnh bên ngoài. Trong Java tính đóng gói thể hiện qua khái niệm lớp (Class). Các đặc trưng của lớp bao gồm có hai phần chính: thuộc tính (Attribute) và hành vi (Behavior).

**Tính kế thừa.** Tính kế thừa là khả năng xây dựng các lớp mới từ các lớp đã có. Khi lớp đóng gói một số dữ liệu và phương thức, lớp mới sẽ kế thừa mọi cấu trúc dữ liệu và các phương thức của lớp mà nó kế thừa. Ngoài ra nó có thể bổ sung các dữ liệu và các phương thức của riêng mình.

**Tính đa hình.** Tính đa hình trong Java được hiểu là trong từng trường hợp, hoàn cảnh khác nhau thì đối tượng có hình thái khác nhau tùy thuộc vào từng ngữ cảnh. Đối tượng có tính đa hình được xem như một đối tượng đặc biệt vì có lúc đối tượng này trở thành một đối tượng khác và cũng có lúc đối tượng này trở thành một đối tượng khác nữa (tùy vào từng hoàn cảnh). Sự "nhập vai" vào các đối tượng khác nhau này giúp cho đối tượng đa hình ban đầu có thể thực hiện những hành động khác nhau của từng đối tượng cụ thể.

Để thể hiện tính đa hình, chúng ta cần đảm bảo hai điều kiện sau:

* Các lớp phải có quan hệ kế thừa với một lớp cha nào đó.
* Phương thức đa hình phải được ghi đè (override) ở lớp con. Tính đa hình chỉ được thể hiện ghi đã ghi đè lên phương thức của lớp cha.

1. Đối tượng, lớp

Khi định nghĩa một lớp, thuộc tính mà nó chứa được thể hiện bằng biến (member variable) và hành vi được thể hiện bởi hàm (method).

1. Cấu trúc một chương trình Java

Package packagename;

Class classname() {

//Khai báo biến

//Phương thức {

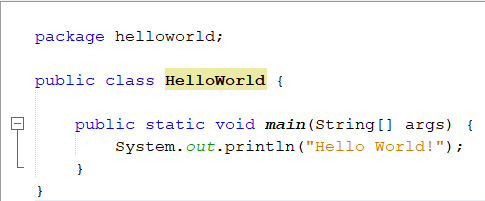
//Các câu lệnh thực hiện trong phương thức.

}

//Hàm main

}

Ví dụ:



*Hình 2.3: Chương trình HelloWorld*

1. Từ khóa this

Từ khóa this có thể dùng bên trong bất cứ phương thức nào để tham chiếu đến đối tượng hiện hành, khi biến đối tượng trùng tên với biến cục bộ.

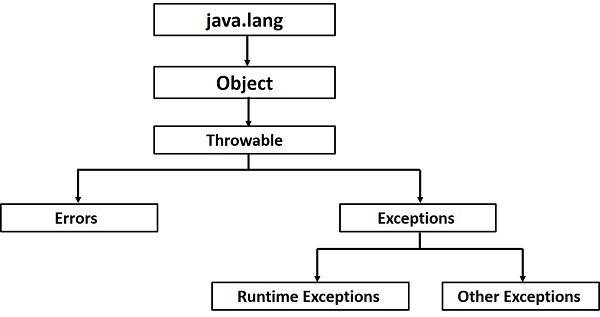
1. Khai báo kế thừa

Lớp được kế thừa gọi là lớp cha (SuperClass: là lớp cha trực tiếp).Lớp kế thừa gọi là lớp con (SubClass). Lớp con kế thừa các biến và các hàm định nghĩa trong lớp cha phụ thuộc vào từ khóa khai báo trong lớp cha: class ClassName extends SuperClass.

1. Xử lý ngoại lệ trong Java

Một chương trình đều tiềm ẩn khả năng sinh ra lỗi. Có thể lỗi do chủ quan hay do lỗi khách quan. Ngôn ngữ Java cung cấp bộ máy xử lý ngoại lệ. Có 2 cách để xử lý ngoại lệ là bắt ngoại lệ hoặc ném ngoại lệ.

Cấu trúc phân cấp của các lớp xử lý ngoại lệ:



*Hình 2.4: Cấu trúc phân cấp xử lý ngoại lệ*

### Nền tảng Java (Java Platform)

**Platform** (Nền tảng) là một **nhóm các công nghệ** được sử dụng **làm cơ sở** cho các ứng dụng, quy trình hoặc công nghệ khác phát triển tiếp.

**Java Platform** (**nền tảng Java**) là một tập hợp các chương trình giúp phát triển và chạy các chương trình được viết bằng ngôn ngữ lập trình Java. Nó bao gồm 3 thành phần chính:

* Java Virtual Machine (JVM): là 1 máy ảo **Java**, được dùng để thực thi các chương trình Java. Nó cung cấp môi trường để code java có thể được thực thi. (để Java có thể đa nền tảng, thì với mỗi hệ điều hành sẽ có một JVM riêng và câu nói **"Write Once, Run Anywhere"** được suất phát từ đây.)
* Java Application Programming Interface (*Java API*): API là một giao diện lập trình ứng dụng. Trong Java, API là một bộ sưu tập các gói, class và giao diện đã được viết sẵn với các phương thức tương ứng. Tương tự như các API khác Java API cũng tạo ra sự thuận lợi cho việc tương tác giữa con người và máy tính.
* Java Development Kit (*JDK*) là một bộ phần mềm cung cấp môi trường phát triển ứng dụng viết bằng ngôn ngữ Java (gồm trình biên dịch, thông dịch, trợ giúp, soạn tài liệu... và các thư viện chuẩn.)

## Hệ điều hành Android

### Giới thiệu về hệ điều hành Android

**Android** là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di động có [màn hình cảm ứng](https://vi.wikipedia.org/wiki/M%C3%A0n_h%C3%ACnh_c%E1%BA%A3m_%E1%BB%A9ng) như [điện thoại thông minh](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_tho%E1%BA%A1i_th%C3%B4ng_minh), [máy tính bảng](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_b%E1%BA%A3ng), smart tivi, smart watch và nhiều thiết bị thông minh khác. Android được Google thiết kế với kiểu mã nguồn mở, điều này giúp cho các nhà phát triển thiết bị, các nhà mạng, lập trình viên có thể tiếp cận nhanh chóng, điều chỉnh tự do các sản phẩm của mình.

Với việc Android sử dụng mã nguồn mở cùng với giấy phép không có quá nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên được điều chỉnh và phân phối Android một cách tự do. Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới. Android chiếm 75% thị phần [điện thoại thông minh](https://www.dienmayxanh.com/tag/dien-thoai-smartphone) trên toàn thế giới vào thời điểm quý 3 năm 2012, với tổng cộng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày. Tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng trên Android, và số lượt tải ứng dụng từ Google Play (cửa hàng ứng dụng chính của Android) ước tính khoảng 25 tỷ lượt. Theo thông kế của Tập đoàn Dữ liệu Quốc Tế (IDC) số lượng Smartphone chạy HDH Android trên toàn thế giới là 85,1% vào năm 2017 và dự đoán vào năm 2021 con số này khoảng 85,3%. Hiện nay có thêm sự cạnh tranh của các mẫu Smartphong chạy HDH IOS của Apple nhưng thị phần của Android vẫn duy trì ở mức ổn định là 85% nên Smartphone Android vẫn là mẫu điện thoại “quốc dân” trên toàn thế giới.

### Lịch sử của hệ điều hành Android

Hệ điều hành Android ra đời từ cách đây 12 năm (kể từ ngày chiếc Smartphone Android đầu tiên có tên là HTC Dream (hay còn được gọi là T-Mobile G1) ra đời. Video đập hộp chiếc T-Mobile G1 <https://www.androidauthority.com/first-android-phone-t-mobile-g1-htc-dream-906362/> ). Hiện nay, nó đã trở thành hệ điều hành di động phổ biến nhất trên thế giới, đánh bại nhiều đối thủ như Symbian, BlackBerry, Palm OS, webOS và Windows Phone.

**Sự ra đời của Android.** Vào tháng 10/2003, trước khi thuật ngữ “Smartphone - điện thoại thông minh” phổ biến, trước khi Apple công bố iPhone đầu tiên và hệ điều hành iOS, thì công ty Android Inc được thành lập ở Palo Alto, California. Với bốn người sáng lập là Rich Miner, Nick Sears, Chris White và Andy Rubin. Vào thời điểm thành lập, ông Rubin nói rằng Android Inc sẽ phát triển “thiết bị di động thông minh hơn có thể biết được vị trí và sở thích của chủ sở hữu”. Rubin sau này đã tiết lộ trong một bài phát biểu vào năm 2013 tại Tokyo rằng hệ điều hành Android đã được ra mắt để cải thiện hệ điều hành của máy ảnh số. Rõ ràng trước ngày ra mắt T-Mobile G1 thì Androi chỉ được phát triển để hỗ trợ hệ điều hành trên máy ảnh số. Cùng với sự đi xuống của thị trường máy ảnh kỹ thuật số, Android Inc đã quyết định chuyển Android thành một hệ điều hành cốt lõi cho một thiết bị di động. Sự chuyển biến này như việc biến một anh phụ bếp vô danh thành bếp trưởng Joël Robuchon vậy. (Joël Robuchon – 1945 “đầu bếp thế kỷ” giữ 26 sao Michelin). Thành lập bởi các “Ông to mặt lớn” nhưng Android Inc vẫn chỉ hoạt động âm thầm và cậu chuyện chỉ thực sự bắt đầu khi Android Inc được mua lại bởi Google vào 17/8/2005. Với sự hộ thuẫn mạnh mẽ về tài chính và nguồn tài nguyên khổng lồ của Google thì Android Inc mới việt lên cậu chuyện của chính mình.

**Biểu tượng của Android.** Hình ảnh quen thuộc hiện nay của hệ điều hành Android giống như sự kết hợp của một con robot và con bọ màu xanh lá cây do Irina Blok tạo ra.  Theo Blok, thiết kế cuối được lấy cảm hứng từ việc nhìn vào biểu tượng nhà vệ sinh quen thuộc đại diện cho “đàn ông” và “phụ nữ”.

**Nhưng cái tên ngọt ngào.** Tên của các đời hệ điều hành Android được lấy theo tên các loại bánh ăn vặt. Ý tưởng được bắt nguồn từ giám đốc dự án tại Google, Ryan Gibson, nhưng lý do cụ thể anh chọn làm như vậy giờ vẫn chưa ai biết. Phải đến tận khi Android 4.4 KitKat ra mắt thì hãng mới đưa ra lời giải thích “chính thức” cho phong cách đặt tên hiệu phiên bản của mình rằng:“Vì những thiết bị chạy Android khiến cuộc sống của bạn trở nên ngọt ngào hơn, các phiên bản chạy Android sẽ được đặt tên theo một món tráng miệng ngọt”.

**Adroid 1.1** (Theo kỹ sư Android Jean-Baptiste Queru nói với trang Android Police năm 2012: Android 1.1, ra mắt tháng 2/2009, dù chưa có tên mã công khai nhưng nhiều báo cáo cho rằng nội bộ đội ngũ đã sử dụng code name “Petit four” – một loại bánh gato nhỏ của Pháp). Tháng 2/2009, bản nâng cấp đầu tiên của Android được trình làng, khoảng ba tháng sau khi G1 được bán ra. Phiên bản 1.1 không phải là một cuộc cách mạng gì to lớn bởi tính năng chính của nó là sửa một danh sách lỗi khá dài. Tuy nhiên, nó đã cho thấy khả năng nâng cấp thiết bị di động qua phương pháp Over-The-Air (tức tải về và cài đặt bản cập nhật ngay trên thiết bị, không cần kết nối với máy tính). Ở thời điểm đó, rất ít hệ điều hành di động có thể làm được việc này, hầu hết đều phải nhờ đến một phần mềm chuyên dùng nào đó trên PC. Trước đó, ở Mỹ có dòng máy Danger Hiptop (được biết nhiều hơn với tên Sidekick) đã có cập nhật dạng gần giống over the air theo từng giai đoạn, và chính Andy Rubin, người sáng lập công ty Android Inc. (sau đó Google mua lại) cũng chính là nhà đồng sáng lập hãng Danger.

**Android 1.5 Cupcake.** Tên mã công khai chính thức đầu tiên của Android đã không xuất hiện cho đến phiên bản 1.5 Cupcake vào tháng 4/2009. Nó đã bổ sung khá nhiều tính năng và cải tiến mới so với hai phiên bản công khai đầu tiên, như khả năng tải video lên YouTube, cách hiển thị màn hình của điện thoại tự động xoay sang đúng vị trí và hỗ trợ bàn phím của bên thứ ba. Một số điện thoại được phát hành với Cupcake bao gồm điện thoại Samsung Galaxy đầu tiên, cùng với HTC Hero.

**Android 1.6 Donut.** Google đã nhanh chóng tung ra Android 1.6 Donut trong tháng 9/2009. Một số tính năng mới hỗ trợ cho các nhà mạng sử dụng mạng CDMA. Điều này giúp đưa điện thoại Android được bán trên toàn thế giới. Các tính năng khác bao gồm việc giới thiệu Quick Search Box và nhanh chóng chuyển đổi giữa Máy ảnh, Máy quay và Thư viện để sắp xếp trải nghiệm chụp media. Donut cũng đã giới thiệu tiện ích Power Control để quản lý Wifi, Bluetooth, GPS. Một trong những chiếc điện thoại được bán với Donut được cài đặt là Dell Streak xấu số, màn hình 5 inch rất lớn và đã được mô tả tại thời điểm đó là “điện thoại thông minh / máy tính bảng”. Những ngày này, màn hình 5 inch được xem là có kích thước trung bình cho điện thoại thông minh.

**Android 2.0-2.1 Eclair.** Vào tháng 10/2009, khoảng một năm sau khi tung ra Android 1.0, Google đã phát hành phiên bản 2.0 của hệ điều hành với tên mã chính thức là Eclair. Phiên bản này là phiên bản đầu tiên bổ sung hỗ trợ Text-to-Speech và cũng giới thiệu các hình nền động, hỗ trợ nhiều tài khoản và điều hướng của Google Maps trong số nhiều tính năng và cải tiến mới khác. Motorola Droid là điện thoại đầu tiên có Android 2.0 cài đặt sẵn. Đây cũng là là điện thoại Android đầu tiên được Verzion Wireless bán. Mặc dù Google vẫn có thể sử dụng Android làm tên cho hệ điều hành của mình, thuật ngữ “Droid” đã được Lucasfilm đăng ký nhãn hiệu. Motorola đã phải xin phép, và trả một khoản tiền cho Lucasfilm, để sử dụng Droid làm tên cho điện thoại của mình. Motorola tiếp tục sử dụng thương hiệu Droid cho nhiều điện thoại của hãng vào cuối năm 2016.

**Android 2.2 Froyo.** Ra mắt vào tháng 5/2010, Android 2.2 Froyo (yogurt đông lạnh) đã chính thức ra mắt. Điện thoại thông minh có cài đặt Froyo có thể tận dụng một số tính năng mới, bao gồm chức năng điểm phát sóng di động Wi-Fi, đẩy thông báo qua dịch vụ C2DM của Android Cloud, hỗ trợ flash và nhiều hơn thế nữa. Điện thoại thông minh đầu tiên mang thương hiệu Nexus của Google, Nexus One, được tung ra với Android 2.1 cài sẵn vào đầu năm 2010 và nhanh chóng nhận được thông tin cập nhật Froyo vào cuối năm đó. Điều này đánh dấu một cách tiếp cận mới cho Google.

**Android 2.3 Gingerbread.** Android 2.3 Gingerbread, được tung ra vào tháng 9/2010, hiện là phiên bản cũ nhất của hệ điều hành mà Google vẫn đang liệt kê trong trang cập nhật phiên bản nền tảng hàng tháng. Tính đến ngày 13/9/2017, Google chỉ ra rằng chỉ 0,6 phần trăm của tất cả các thiết bị Android hiện đang chạy một số phiên bản của Gingerbread.

Hệ điều hành nhận được giao diện người dùng dưới tên Gingerbread và thêm hỗ trợ chức năng giao tiếp NFC. Điện thoại đầu tiên để thêm cả phần cứng Gingerbread và NFC là Nexus S, được đồng phát triển bởi Google và Samsung. Gingerbread cũng đặt nền móng cho selfie, bằng cách hỗ trợ nhiều camera và hỗ trợ trò chuyện video trong Google Talk.

**Android 3.0 Honeycomb.** Honeycomb được Google phát hành chỉ để cài đặt trên máy tính bảng và các thiết bị di động khác có màn hình lớn hơn so với điện thoại thông minh hiện tại. Honeycomb lần đầu tiên được giới thiệu vào tháng 2/2011, cùng với máy tính bảng Motolola Xoom đầu tiên và bao gồm các tính năng như giao diện người dùng được thiết kế lại dành riêng cho màn hình lớn cùng với thanh thông báo được đặt ở phía dưới màn hình của máy tính bảng.

**Android 4.0 Ice Cream Sandwich.** Được phát hành vào tháng 10/2011, phiên bản Ice Cream Sandwich của Android đã mang một số tính năng mới cho người dùng. Nó kết hợp nhiều tính năng của phiên bản Honeycomb dành cho máy tính bảng với Gingerbread theo định hướng smartphone. Nó cũng bao gồm một “khay yêu thích” trên màn hình chủ, cùng với sự hỗ trợ đầu tiên để mở khóa điện thoại bằng cách chụp ảnh khuôn mặt của chủ sở hữu. Loại hỗ trợ đăng nhập sinh trắc học đã phát triển và cải thiện đáng kể kể từ đó.

**Android 4.1 Jelly Bean.** Kỷ nguyên Jelly Bean của Android bắt đầu vào tháng 6/2012 với việc phát hành Android 4.1. Một số tính năng mới bao gồm thông báo mới cho thấy nhiều nút nội dung cùng với sự hỗ trợ đầy đủ cho phiên bản trình duyệt web Chrome của Android. Google Now cũng đã xuất hiện trên Search và “Project Butter” được giới thiệu để tăng tốc độ hoạt hình và cải thiện khả năng phản ứng cảm ứng của Android. Màn hình, chụp ảnh HDR và Miracast cũng được hỗ trợ. Hiện tại, khoảng 6,9 phần trăm của tất cả các sản phẩm Android sử dụng Jelly Bean.

**Android 4.4 KitKat.** Tên của Android 4.4 là phiên bản duy nhất của hệ điều hành thực sự sử dụng tên nhãn hiệu cho một miếng kẹo cho đến hiện tại. KitKat không có nhiều các tính năng mới, nhưng nó đã có một điều mà thực sự đã giúp mở rộng thị trường Android tổng thể. Nó đã được tối ưu hóa để chạy trên điện thoại thông minh có ít nhất 512 MB RAM. Điều này cho phép các nhà sản xuất điện thoại để có được phiên bản mới nhất của Android và cài đặt nó trên điện thoại rẻ hơn nhiều. Điện thoại thông minh Nexus 5 của Google là điện thoại thông minh đầu tiên sử dụng Android 4.4. Mặc dù KitKat đã được phát hành cách đây gần bốn năm, vẫn còn rất nhiều thiết bị vẫn đang sử dụng nó. Các trang cập nhật trang cập nhật nền tảng hiện tại của Google hiển thị rằng 15,1 phần trăm của tất cả các thiết bị Android đang chạy một số phiên bản Android 4.4 KitKat.

**Android 5.0 Lollipop.** Được ra mắt lần đầu vào mùa thu năm 2014, Android 5.0 Lollipop là một bước thay đổi lớn trong giao diện tổng thể của hệ điều hành. Đây là phiên bản đầu tiên của hệ điều hành sử dụng ngôn ngữ Material Design của Google, sử dụng các hiệu ứng ánh sáng và bóng tối tự do. UI cũng có một số thay đổi khác cho Lollipop, bao gồm một thanh điều hướng mới, thông báo phong phú cho màn hình khóa và nhiều hơn nữa. Bản cập nhật Android 5.1 đã có một số thay đổi. Các tính năng này bao gồm hỗ trợ chính thức cho các cuộc gọi thoại đôi, cuộc gọi thoại HD và bảo vệ thiết bị ngay cả sau khi khôi phục cài đặt gốc. Điện thoại thông minh Nexus 6 của Google cùng với máy tính bảng Nexus 9 là thiết bị đầu tiên được cài đặt sẵn Lollipop. Hiện tại, Android 5.0 Lollipop đang được cài đặt và sử dụng bởi khoảng 29% trong số tất cả các thiết bị Android đang hoạt động. Thực tế thú vị là Google đã sử dụng tên mã là “Lemon Meringue Pie” trong khi phát triển Android 5.0 trước khi quyết định sử dụng Kẹo Lollipop làm tên công khai của hệ điều hành.

**Android 6.0 Marshmallow.** Android 6.0 Marshmallow được phát hành vào mùa thu năm 2015. Nó bao gồm các tính năng như một ngăn kéo ứng dụng di chuyển theo chiều dọc, Google Now trên Tap, hỗ trợ mở khóa sinh trắc học, hỗ trợ USB Type-C, Android Pay, và nhiều hơn nữa.

**Android 7.0 Nougat.** Phiên bản 7.0 của hệ điều hành di động của Google được ra mắt vào mùa thu năm 2016. Trước khi Nougat được tiết lộ “Android N” đã được Google gọi nội bộ là “New York Cheesecake”. Một số tính năng mới của Nougat bao gồm các chức năng đa tác vụ tốt hơn cho số lượng ngày càng tăng của điện thoại thông minh có màn hình lớn, chẳng hạn như chế độ chia màn hình cùng với việc chuyển đổi nhanh giữa các ứng dụng.

Google cũng đã sử dụng phiên bản này để đẩy mạnh thị trường smartphone cao cấp. Các điện thoại thông minh có thương hiệu riêng của công ty, Pixel và Pixel XL, cùng với LG V20, là những người đầu tiên được phát hành với Nougat cài sẵn.

**Android 8.0 Oreo.** Vào tháng 3/2017, Google chính thức công bố và phát hành bản xem trước dành cho nhà phát triển đầu tiên dành cho Android O, còn được gọi là Android 8.0. Đây là lần thứ hai Google chọn một tên thương hiệu cho Android (Oreo thuộc sở hữu của Nabisco). Android Oreo bao gồm nhiều thay đổi trực quan trong menu Cài đặt, cùng với hỗ trợ chế độ ảnh-trong-ảnh, các kênh thông báo, API tự động điền mới để quản lý mật khẩu và điền dữ liệu tốt hơn.

**Android 9.0 Pie** (Android P), là phiên bản thứ 9 của hệ điều hành Android, được công bố lần đầu tiên vào tháng 3 năm 2018**.** Với các tính năng mới như:

* **Cử chỉ điều hướng**: Kéo nút home để chuyển qua lại giữa các ứng dụng, vuốt lên trên để mở App Drawer là hai trong số nhiều các cử chỉ tương tư thao tác vuốt trên [iPhone X](https://www.thegioididong.com/dtdd/iphone-x-64gb) giúp máy chuyển ứng dụng nhanh hơn, thao tác cảm ứng trên máy được tiện dụng hơn khá nhiều.
* **Thời gian sử dụng ứng dụng**: Một tính năng mới có tên là "dashboard" bao gồm một bảng tổng hợp cực kì trực quan các ứng dụng bạn sử dụng trong ngày, trong tuần, thời gian sử dụng từng ứng dụng khác nhau. Nhờ vào đó bạn có thể phân bổ thời gian hợp lý hơn, biết được mình sử dụng ứng dụng nào nhiều nhất và giảm bớt thời gian cho những ứng dụng không cần thiết.
* **Kết nối Bluetooth với 5 thiết bị khác nhau**: Trên phiên bản tiền nhiệm của mình là Android Oreo chỉ có thể kết nối với 2 thiết bị sử dụng Bluetooth cùng một lúc, thì trên phiên bản Android P máy có thể kết nối được với 5 thiết bị song song với nhau, hỗ trợ kết nối nhiều loa, tai nghe, chuột không dây hay nhiều thiết bị ngoại vi sử dụng Bluetooth dễ dàng hơn.
* **Pin "trâu" hơn**: Với tính năng mới có tên gọi "Adaptive Battery", pin trên Android P sẽ được tiết kiệm hơn rất nhiều nhờ vào máy đã có khả năng phân phối pin cho các ứng dụng bạn thực sự đang dùng. Với các ứng dụng chạy ngầm, máy sẽ hạn chế bớt các tác vụ này giúp tiết kiệm được nhiều pin hơn.
* **Nút xoay màn hình**: tính năng nhỏ nhưng cực kì tiện lợi, giờ đây khi bạn xem video, máy sẽ hiển thị thêm một nút bấm xoay màn hình nữa, giúp bạn xoay nhanh màn hình thành dạng ngang để xem video tốt hơn, thay vì cứ phải bấm tắt hoặc mở khoá xoay màn hình như trước. Cực kì tiện lợi.

**Android 10 (Android Q).** Không nhưng các đời anh chị đi trước Android 10 không còn được lấy nhưng cái tên ngọt ngào nữa mà Android 10 chỉ đơn giản là Android 10. Những phiên bản thử nghiệm đầu tiên của Android 10 được phát hành sớm từ tháng 3/2019, nửa năm sau đó, ngày 3/9/2019, phiên bản chính thức được phát hành. Với sự cạnh tranh ngày càng gay gắt của Apple, Google cũng phải thay đổi với một “rổ” các tính năng mới trên Android 10 cụ thể:

* Chế độ tối toàn diện: Với một phím tắt ngay trên Quick Setting, người dùng có thể nhanh chóng bật tắt chế độ tối. Hoặc có thể cài đặt bật tắt tự động theo khung giờ cố định một cách tiện lợi.
* **Thao tác điều hướng mới**: Thay vì 3 phím điều hướng đơn thuần như trước đây, Android 10 cung cấp tùy chọn ẩn thanh điều hướng, thay vào đó sử dụng thao tác vuốt để điều hướng.
* **Kích hoạt Google Assistant nhanh hơn:** Bên cạnh việc ra lệnh giọng nói “Hey, Google”. Android 10 còn cho phép gọi trợ lý ảo bằng cử chỉ vuốt vô cùng tiện lợi.
* Trả lời thông minh: Thay vì phải nhập tin nhắn thủ công, từ nay bạn chỉ cần click chọn, Android 10 sẽ gợi ý cho bạn những đoạn trả lời ngắn, dựa theo tin nhắn bạn nhận được.
* Giao diện được tùy biến sâu.
* **Tăng cường bảo mật và riêng tư.**
* **Cập nhập bảo mật trực tiếp từ Google Play.**
* **Kiểm soát sử dụng với Digital Wellbeing.**
* **Chế độ tập trung Focus Mode.**
* **Bật phụ đề tự động**
* **Linh hoạt hơn với màn hình gập.**
* **Nhiều chế độ chụp hình.**
* **Chia sẻ nhanh hơn.**
* **Hỗ trợ HDR 10 +:** Android 10 đã bổ sung gói hỗ trợ chính thức cho công nghệ HDR 10 +, nhờ vậy màu sắc màn hình sở hữu công nghệ này sẽ chân thực và sống động hơn bao giờ hết.

**Android 11** (Android R): Android 11 là bản nâng cấp lớn tiếp theo của hệ điều hành Android dự kiến sẽ được Google giới thiệu bản dùng thử vào tháng**5/2020** sắp tới. Hứa hẹn một “rổ” các cập nhật không kém gì Android 10.

### Mô tả về hệ điều hành Android

**Giao diên.** Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm, kéo giãn và thu lại để xử lý các đối tượng trên màn hình. Sự phản ứng với tác động của người dùng diễn ra gần như ngay lập tức, nhằm tạo ra giao diện cảm ứng mượt mà, thường dùng tính năng rung của thiết bị để tạo [phản hồi rung](https://vi.wikipedia.org/w/index.php?title=Ph%E1%BA%A3n_h%E1%BB%93i_rung&action=edit&redlink=1) cho người dùng. Những thiết bị phần cứng bên trong như [gia tốc kế](https://vi.wikipedia.org/w/index.php?title=Gia_t%E1%BB%91c_k%E1%BA%BF&action=edit&redlink=1), [con quay hồi chuyển](https://vi.wikipedia.org/wiki/Con_quay_h%E1%BB%93i_chuy%E1%BB%83n) và [cảm biến khoảng cách](https://vi.wikipedia.org/w/index.php?title=C%E1%BA%A3m_bi%E1%BA%BFn_kho%E1%BA%A3ng_c%C3%A1ch&action=edit&redlink=1) được một số ứng dụng sử dụng để phản hồi một số hành động khác của người dùng, ví dụ như điều chỉnh màn hình từ chế độ hiển thị dọc sang chế độ hiển thị ngang tùy theo vị trí của thiết bị, hoặc cho phép người dùng lái xe đua bằng xoay thiết bị, giống như đang điều khiển [vô-lăng](https://vi.wikipedia.org/w/index.php?title=V%C3%B4-l%C4%83ng&action=edit&redlink=1).

**Ứng dụng.** Android có lượng ứng dụng của bên thứ ba ngày càng nhiều, được chọn lọc và đặt trên một cửa hàng ứng dụng như [Google Play](https://vi.wikipedia.org/wiki/Google_Play) hay [Amazon Appstore](https://vi.wikipedia.org/wiki/Amazon_Appstore) để người dùng lấy về, hoặc bằng cách tải xuống rồi cài đặt tập tin [APK](https://vi.wikipedia.org/w/index.php?title=APK&action=edit&redlink=1) từ trang web khác. Các ứng dụng trên Play Store cho phép người dùng duyệt, tải về và cập nhật các ứng dụng do Google và các nhà phát triển thứ ba phát hành. Play Store được cài đặt sẵn trên các thiết bị thỏa mãn điều kiện tương thích của Google. Ứng dụng sẽ tự động lọc ra một danh sách các ứng dụng tương thích với thiết bị của người dùng, và nhà phát triển có thể giới hạn ứng dụng của họ chỉ dành cho những nhà mạng cố định hoặc những quốc gia cố định vì lý do kinh doanh. Nếu người dùng mua một ứng dụng mà họ cảm thấy không thích, họ được hoàn trả tiền sau 15 phút kể từ lúc tải về, và một vài nhà mạng còn có khả năng mua giúp các ứng dụng trên Google Play, sau đó tính tiền vào trong hóa đơn sử dụng hàng tháng của người dung. Đến tháng 9 năm 2012, có hơn 675.000 ứng dụng dành cho Android, và số lượng ứng dụng tải về từ Play Store ước tính đạt 25 tỷ.

### Bảo mật và tính riêng tư của hệ điều hành Android

Các ứng dụng Android chạy trong một "[sandbox](https://vi.wikipedia.org/w/index.php?title=Sandbox&action=edit&redlink=1)", là một khu vực riêng rẽ với hệ thống và không được tiếp cận đến phần còn lại của tài nguyên hệ thống, trừ khi nó được người dùng trao quyền truy cập một cách công khai khi cài đặt. Trước khi cài đặt ứng dụng, [Cửa hàng Play](https://vi.wikipedia.org/wiki/Google_Play) sẽ hiển thị tất cả các quyền mà ứng dụng đòi hỏi: ví dụ như một trò chơi cần phải kích hoạt bộ rung hoặc lưu dữ liệu vào [thẻ nhớ SD](https://vi.wikipedia.org/wiki/Th%E1%BA%BB_nh%E1%BB%9B), nhưng nó không nên cần quyền đọc tin nhắn [SMS](https://vi.wikipedia.org/wiki/SMS) hoặc tiếp cận danh bạ điện thoại. Sau khi xem xét các quyền này, người dùng có thể chọn đồng ý hoặc từ chối chúng, ứng dụng chỉ được cài đặt khi người dùng đồng ý.

## Công cụ Android Studio

### Android Studio là gì?

**Android Studio:** là một công cụ để phát triển các ứng dụng chạy trên hệ điều hành Android. Android Studio lần đầu tiên được công bố tại hội nghị Google I/O vào năm 2013 và được phát hành cho công chúng vào năm 2014 sau nhiều phiên bản beta khác nhau.

Có nhiều công cụ để phát triển Android nhưng đến nay công cụ chính thức và mạnh mẽ nhất là Android Studio. Đây là IDE (Môi trường phát triển tích hợp) chính thức cho nền tảng Android, được phát triển bởi Google và được sử dụng để tạo phần lớn các ứng dụng mà bạn có thể sử dụng hàng ngày. **Android Studio**chạy trên Windows, Mac và Linux, nhằm thay thế cho **Eclipse Android Development Tool**(ADT) vốn được sử dụng làm IDE chính trong các năm trước đó.

Android Studio khiến việc tạo ứng dụng trở nên dễ dàng hơn đáng kể so với phần mềm không chuyên dụng. Chức năng của Android Studio là cung cấp giao diện để tạo các ứng dụng và xử lý phần lớn các công cụ quản lý file phức tạp đằng sau hậu trường. Ngôn ngữ lập trình được sử dụng ở đây là [Java](https://quantrimang.com/java-la-gi-tai-sao-nen-chon-java-143130) và được cài đặt riêng trên thiết bị của bạn. Android Studio rất đơn giản, bạn chỉ cần viết, chỉnh sửa và lưu các dự án của mình và các file trong dự án đó. Đồng thời, Android Studio sẽ cấp quyền truy cập vào Android SDK.

**Một số tính năng nổi bật:**

* Bộ công cụ build ứng dụng dựa trên Gradle (thay vì Maven).
* Chức năng dò và sửa lỗi nhanh, hướng Android.
* Công cụ chỉnh sửa màn hình dạng kéo thả tiện lợi.
* Các wizard tích hợp nhằm giúp lập trình viên tạo ứng dụng từ mẫu có sẵn.
* Tích hợp Google Cloud Platform, dễ dàng tích hợp với Google Cloud Messaging và App Engine của Google.

Và đi kèm với Android Studio là **Android SDK.** Nếu Android Studio là trình soạn thảo code (IDE) thì Android SDK là bộ tổng hợp các công cụ để build app, các bản mẫu máy ảo Android (sử dụng để test app) cần thiết để làm ra một ứng dụng Android hoàn chỉnh.

### Cấu trúc của một ứng dụng Android

Hệ điều hành Android là một ngăn xếp của các thành phần phần mềm mà có thể đại khái phân chia thành 5 khu vực và 4 lớp chính.



*Hình 2.5: Cấu trúc ứng dụng Android*

1. Lớp Linux Kernel trong Android

Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, … Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.

1. Libraries trong Android

Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.

1. Android Libraries

Phần này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn. Dưới đây là một số Android Library cốt lõi có sẵn cho lập trình viên Android:

* **android.app**: Cung cấp truy cập tới mô hình ứng dụng và nó là nền móng cho tất cả ứng dụng Android.
* **android.content**:Việc truy cập nội dung, các thông điệp giữa các ứng dụng và các thành phần ứng dụng trở nên dễ dàng hơn.
* **android.database**:Được sử dụng để truy cập dữ liệu được công bố bởi Provider và bao gồm các lớp quản lý cơ sở dữ liệu SQLite.
* **android.opengl**: Một Java Interface cho OpenGL ES 3D thông qua API.
* **android.os**: Cung cấp cho các ứng dụng sự truy cập tới các dịch vụ chuẩn của hệ điều hành như thông báo, dịch vụ hệ thống và giao tiếp nội tiến trình.
* **android.text**: Được sử dụng để phục hồi và thao tác text trên một thiết bị hiển thị.
* **android.view**: Các khối kiến trúc nền tảng của ứng dụng UI.
* **android.widget**: Một tập hợp các UI được xây dựng trước như button, label, list view, layout manager, radio button, …
* **android.webkit**: Một tập hợp các lớp cho phép khả năng để trình duyệt trên web được xây dựng bên trong các ứng dụng.

1. Android Runtime

Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là Dalvik Virtual Machine là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android.

Máy ảo Dalvik VM sử dụng các tính năng cốt lõi của Linux như quản lý bộ nhớ, đa luồng, mà thực chất là bên trong ngôn ngữ Java. Máy ảo Dalvik cho phép tất cả các ứng dụng Android chạy trong tiến trình riêng của nó.

Android Runtime cũng cung cấp bộ thư viện cốt lõi, cho phép các lập trình viên Android sử dụng để viết các ứng dụng Android.

1. Application Framework

Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

Application Framework bao gồm các dịch vụ chính sau:

* **Activity Manager**: Điều khiển các khía cạnh của vòng đời ứng dụng và Activity Stack.
* **Content Providers**: Cho phép các ứng dụng công bố và chia sẻ dữ liệu với các ứng dụng khác.
* **Resource Manager**: Cung cấp sự truy cập tới các resource được nhúng (không phải code) như chuỗi, thiết lập màu, UI layout.
* **Notifications Manager**: Cho phép các ứng dụng hiển thị thông báo tới người dùng.
* **View System**: Một tập hợp các view được sử dụng để tạo UI cho ứng dụng.

1. Applications

Chúng ta sẽ thấy tất cả các ứng dụng Android ở lớp trên cùng. Ứng dụng sẽ được cài đặt vào lớp này. Ví dụ của những ứng dụng này là Contacts Books, Browser, Games, …

**2.3.3 Các Component cơ bản**

Các Component là các khối kiến trúc nền tảng tạo nên một ứng dụng Android. Những Component này được ghép bởi AndroidManifest.xml, miêu tả mỗi thành phần và cách chúng tương tác với nhau.

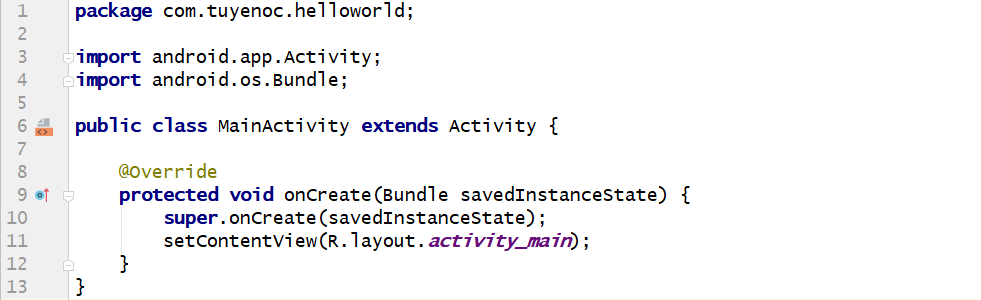
Có 4 Component chính có thể được sử dụng bên trong một ứng dụng Android:

* **Activity**: Chúng điều khiển UI và xử lý tương tác người dùng trên màn hình smartphone.
* **Services**: Chúng xử lý các tiến trình background được gắn kết với một ứng dụng.
* **Broadcast** **Receivers**: Chúng xử lý giao tiếp giữa Android OS và các ứng dụng.
* **Content** **Providers**: Chúng xử lý dữ liệu và quản lý cơ sở dữ liệu.

1. Activity trong Android

Một Activity biểu diễn một màn hình với một giao diện UI. Nói đơn giản, Activity thực hiện các hành động trên màn hình. Ví dụ, một ứng dụng email có thể thực hiện một activity mà hiển thị một danh sách các email mới, activity khác để soạn thảo một email, và activity khác để đọc email. Nếu một ứng dụng có nhiều hơn một activity, thì một trong số chúng sẽ được đánh dấu như là activity mà sẽ được hiển thị khi ứng dụng chạy.

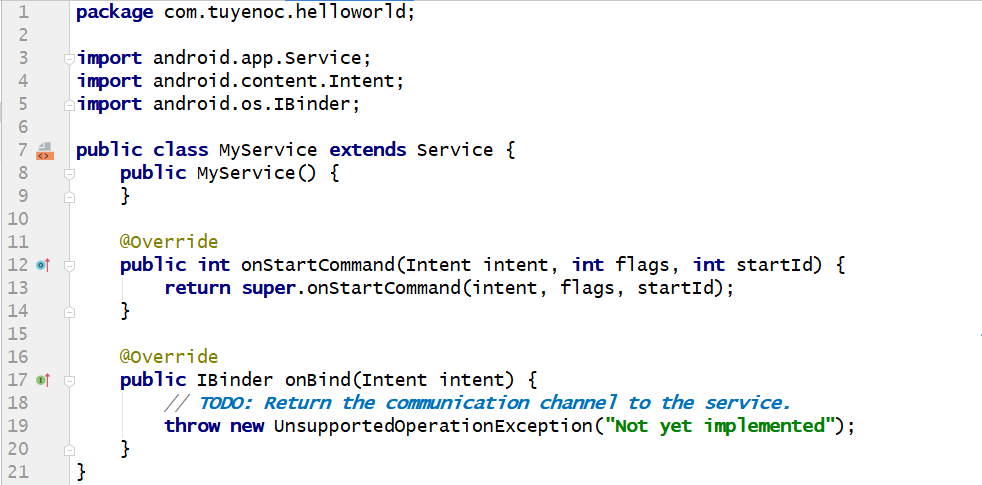
Một activity được triển khai như là một lớp con của lớp Activity, như sau:

*Hình 2.6: Lớp MainActivity kế thừa lớp Activity*

1. Services trong Android

Một service là một Component mà chạy trong Background để thực hiện các hoạt động. Ví dụ, một service có thể chơi nhạc nền trong khi người dùng là trong một ứng dụng khác, hoặc nó có thể lấy dữ liệu qua mạng mà không cần chặn tương tác người dùng với một activity.

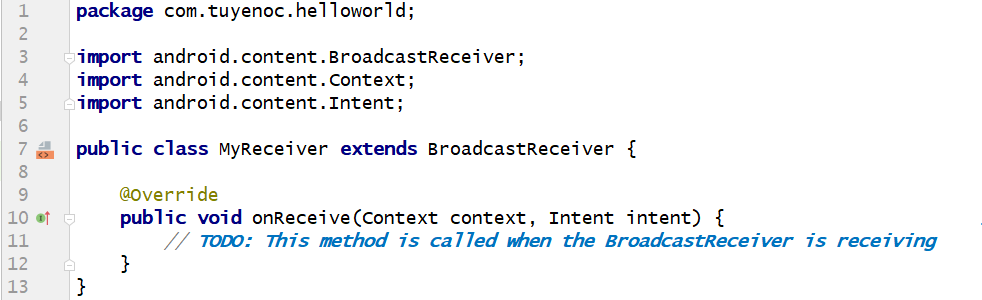
Một service được triển khai như là một lớp con của lớp Service, như sau:

*Hình 2.7: Triển khai một Service*

1. Broadcast Receivers trong Android

Về cơ bản, Broadcast Receivers phản hồi các thông điệp từ ứng dụng khác hoặc từ hệ thống. Ví dụ, các ứng dụng cũng có thể khởi tạo các broadcast để báo cho các ứng dụng khác biết rằng có một số dữ liệu đã được tải về thiết bị và nó là có sẵn cho các ứng dụng này để sử dụng chúng. Vì thế, Broadcast Receiver sẽ thông dịch thông tin này và sẽ khởi tạo hành động thích hợp.

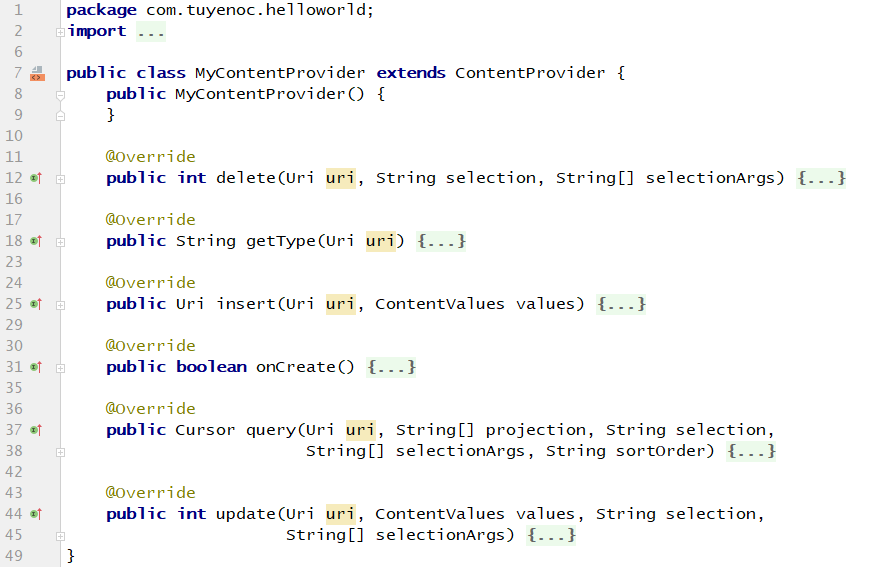
Một Broadcast Receiver được triển khai như là một lớp con của BroadcastReceiver và mỗi thông điệp được truyền như là một đối tượng Intent.

*Hình 2.8: Triển khai một BroadcastReceiver*

1. Content Providers trong Android

Thành phần Content Provider cung cấp dữ liệu từ một ứng dụng tới các ứng dụng khác tùy theo yêu cầu. Những yêu cầu này được xử lý bởi các phương thức của lớp ContentResolver. Dữ liệu có thể được lưu trữ trong hệ thống file, cơ sở dữ liệu hoặc bất cứ đâu.

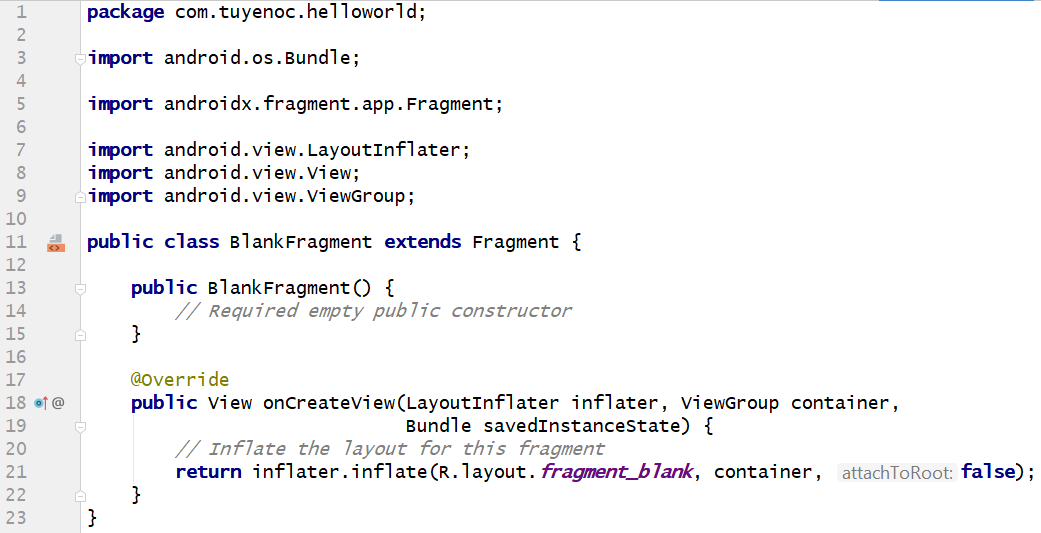
Một Content Provider được triển khai như là một lớp con của lớp ContentProvider và phải triển khai một APIs chuẩn mà cho phép các ứng dụng khác thực hiện các transaction.

*Hình 2.9: Triển khai một Content Providers*

1. Một số Component khác trong Android

Một số Component khác được sử dụng trong xây dựng các thực thể, tính logic của chúng và kết nối chúng với nhau. Chúng bao gồm:

* Fragment: Biểu diễn một phần của giao diện UI trong một Activity.



*Hình 2.10 Fragment trong Android*

* View: Các phần tử giao diện UI mà được vẽ trên màn hình bao gồm button, list, form, …



*Hình 2.11: Một số View trong Android*

* Layout: Cấu trúc thứ bậc của view mà điều khiển định dạng màn hình và bề mặt của các view



*Hình 2.12: Một số Layout trong Android*

* Intent: Các thành phần kết nối các thông báo với nhau
* Resource:Các phần tử ngoại vi, như các chuỗi, hằng, và hình ảnh có thể vẽ.
* Manifest: Chính là Configuration file cho ứng dụng.



*Hình 2.13 File Android Manifest*

### UI Layout trong Android

Khối kiến trúc nền tảng cho giao diện UI là một đối tượng View được tạo từ lớp View và chiếm một khu vực hình chữ nhật trên màn hình và chịu trách nhiệm để vẽ và xử lý sự kiện. View là lớp cơ sở cho Widget, mà được sử dụng để tạo các thành phần UI có tính tương tác như button, các trường text, …

ViewGroup là một lớp con của View và cung cấp Container vô hình mà giữ các View khác hoặc các ViewGroup khác và định nghĩa các thuộc tính Layout của chúng.

Tại tầng thứ ba, chúng ta có nhiều Layout khác nhau mà là lớp phụ của lớp ViewGroup và một Layout đặc trưng định nghĩa cấu trúc nhìn thấy cho một giao diện UI trong Android và có thể được tạo hoặc tại runtime bởi sử dụng các đối tượng View/ViewGroup hoặc bạn có thể khai báo Layout của bạn bởi sử dụng XML file đơn giản là main\_Layoutxml, được đặt trong thư mục res/layout của Project.

1. Các loại Layout trong Android

Android cung cấp một số Layout đểsử dụng trong hầu hết các ứng dụng Android giúp cung cấp cái nhìn và cảm nhận khác nhau:

* LinearLayout: LinearLayout là một view group mà căn chỉnh các view con theo một hướng nào đó: chiều dọc hay chiều ngang.
* RelativeLayout: RelativeLayout là một view group mà hiển thị các view con trong các vị trí cân xứng với nhau.
* TableLayout: TableLayout là một view mà nhóm tất cả các view vào trong các hàng và các cột.
* AbsoluteLayout: AbsoluteLayout cho phép bạn xác định vị trí chính xác của các view con.
* FrameLayout: FrameLayout là một placeholder trên màn hình mà bạn có thể sử dụng để hiển thị một view đơn.
* ListView: ListView là một view group mà hiển thị một danh sách item có thể scroll.
* GridView: GridView là một view group mà hiển thị các item trong một lưới hai chiều có thể scroll.

1. Các thuộc tính của Layout trong Android

Mỗi Layout có một tập hợp các thuộc tính mà định nghĩa các thuộc tính mang tính thị giác cho Layout Có một số thuộc tính chúng cho tất cả các Layout và cũng có một số thuộc tính riêng cho một Layout cụ thể. Dưới đây là một số thuộc tính chung và sẽ được áp dụng cho tất cả Layout:

* android:id: Đây là ID mà nhận diện duy nhất View
* android:layout\_width: Đây là độ rộng của Layout
* android:layout\_height: Đây là chiều cao của Layout
* android:layout\_marginTop: Đây là không gian phụ (extra space) trên cạnh trên của Layout
* android:layout\_marginBottom: Đây là extra space trên cạnh dưới của Layout.
* android:layout\_marginLeft: Đây là extra space trên cạnh trái của Layout.
* android:layout\_marginRight: Đây là extra space trên cạnh phải Layout.
* android:layout\_gravity: Xác định cách các view con được đặt tại đâu.
* android:layout\_weight: Xác định có bao nhiêu extra space trong Layout nên được cấp phát tới View đó.
* android:layout\_x: Xác định tọa độ x của Layout.
* android:layout\_y: Xác định tọa độ y của Layout.
* android:paddingLeft: Đây là left padding được điền cho Layout.
* android:paddingRight: Đây là right padding được điền cho Layout.
* android:paddingTop: Đây là top padding được điền cho Layout.
* android:paddingBottom: Đây là bottom padding được điền cho Layout.

1. View ID trong Android

Một đối tượng View có thể có một ID duy nhất được gán tới nó, mà sẽ nhận diện View đó một cách duy nhất bên trong cấu trúc cây. Cú pháp cho một ID, bên trong một thẻ XML là:

Description: Capture

Biểu tượng @ tại phần đầu của chuỗi chỉ rằng XML Parser nên phân tích cú pháp và mở rộng phần còn lại của chuỗi ID và nhận diện nó như là một ID resource.

Biểu tượng + nghĩa rằng đây là một tên resource mới mà phải được tạo và được thêm tới các Resource của chúng ta. Để tạo một sự thể hiện của đối tượng View và bắt nó từ Layout, sử dụng cú pháp sau:

Description: Capture

### UI Control trong Android.

Input Control là các thành phần có tính tương tác trong giao diện UI của ứng dụng. Android cung cấp nhiều control đa dạng để bạn có thể sử dụng trong UI, như button, text field, seek bar, checkbox, zoom button, toggle button, …

Một View là một đối tượng mà vẽ cái gì đó trên màn hình mà người dùng có thể tương tác với. Một ViewGroup là một đối tượng mà giữ các đối tượng View (và ViewGroup) khác để định nghĩa Layout của giao diện UI.

Định nghĩa Layout trong một XML file mà cung cấp một cấu trúc con người có thể đọc cho Layout đó, tương tự như HTML.

1. **Các UI Control trong Android**

Android cung cấp một số UI Controlcho phépxây dựng đồ họa cho giao diện UI cho ứng dụng:

|  |  |
| --- | --- |
| UI Control | Miêu tả |
| [TextView](https://vietjack.com/android/textview_trong_android.jsp) | Control này được sử dụng để hiển thị text tới người dùng |
| [EditText](https://vietjack.com/android/edittext_trong_android.jsp) | EditText là một lớp con được định nghĩa trước của TextView mà bao gồm các khả năng chỉnh sửa đa dạng |
| [AutoCompleteTextView](https://vietjack.com/android/auto_complete_view_trong_android.jsp) | AutoCompleteTextView là một view tương tự như EditText, ngoại trừ rằng nó hiển thị một danh sách các đề nghị tự động trong khi người dùng soạn text |
| [Button](https://vietjack.com/android/button_trong_android.jsp) | Một nút có thể được nhấn, hoặc click bởi người dùng để thực hiện một hành động |
| [ImageButton](https://vietjack.com/android/imagebutton_trong_android.jsp) | Là một AbsoluteLayout cho bạn khả năng xác định vị trí chính xác của các view con |
| [CheckBox](https://vietjack.com/android/checkbox_trong_android.jsp) | On/Off có thể được chuyển đổi bởi người dùng. Bạn nên sử dụng nó khi biểu diễn cho người dùng với một nhóm các tùy chọn có thể chọn mà không loại trừ lẫn nhau |
| [ToggleButton](https://vietjack.com/android/togglebutton_trong_android.jsp) | Hiển thị trạng thái checked/unchecked giống một nút on/off với một light indicator |
| [ProgressBar](https://vietjack.com/android/progressbar_trong_android.jsp) | ProgressBar view cung cấp một phản hồi có thể nhìn thấy về một số tác vụ, như khi chúng ta thực hiện tác vụ ra ngoài trong background |
| [TimePicker](https://vietjack.com/android/timepicker_trong_android.jsp) | TimePicker view cho phép người sử dụng lựa chọn thời gian của một ngày: hoặc chế độ 24 h hoặc chế độ AM/PM |
| [DatePicker](https://vietjack.com/android/datepicker_trong_android.jsp) | DatePicker view cho phép người dùng lựa chọn một date |

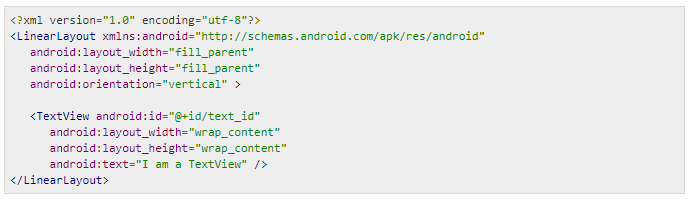
1. **Tạo UI Control trong Android**

Các điều khiển đầu vào là các thành phần tương tác trong giao diện UI của ứng dụng. Android cung cấp nhiều control đa dạng để bạn có thể sử dụng trong UI, như button, text field, seek bar, checkbox, zoom butto, toggle button, …

Một đối tượng View có thể có một ID duy nhất được gán cho nó mà sẽ nhận diện View một cách duy nhất bên trong cấu trúc cây. Cú pháp cho một ID, bên trong thẻ XML là:

Description: Capture

Để tạo một UI Control/View/Widger, phải định nghĩa một View/Widget trong layout file và gán cho nó một ID duy nhất, như sau:



Sau đó, tạo một sự thể hiện của đối tượng Control và nắm bắt nó từ layout, sử dụng:

Description: Capture

# CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG VÀ PHÂN TÍCH THIẾT KẾ ỨNG DỤNG

## Khảo sát

Qua khảo sát có rất nhiều ứng dụng tương tự:

* Ứng dụng freaking math tác giả Nguyễn Lương Bằng

Ảnh có chứa vẽ

Mô tả được tạo tự độngẢnh có chứa vẽ

Mô tả được tạo tự độngẢnh có chứa đồng hồ, máy tính

Mô tả được tạo tự động

* Ứng dụng freaking math tác giả ThuPV

Ảnh có chứa màn hình, đóng, ký hiệu, đường phố

Mô tả được tạo tự độngẢnh có chứa vẽ

Mô tả được tạo tự độngẢnh có chứa màn hình, TV, ảnh chụp màn hình, phòng

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, máy tính

Mô tả được tạo tự động

* Ứng dụng freaking math tác giả SingleHD

Ảnh có chứa đồng hồ

Mô tả được tạo tự độngẢnh có chứa vẽ, đồng hồ, ký hiệu, quả bóng

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, màn hình

Mô tả được tạo tự độngẢnh có chứa ảnh chụp màn hình, thiết bị điện tử, màn hình, máy tính

Mô tả được tạo tự động

Những ứng đều có 1 điểm chung là thời gian chạy rất nhanh não bộ của chúng ta chưa kịp tính toán đã hết thời gian.Với mong muốn xây dựng 1 ứng dụng tăng khả năng tính nhẩm ưa nhìn và dễ sử dụng cho học sinh em đã khắc phục nhược điểm này, thay vì hai đến ba giây nhưng những ứng dụng tương tự thì ứng dụng của em có thời gian khởi đầu là năm giây và sẽ rút ngắn dần khi tăng độ khó.

## Mô tả

### Mô tả bài toán

Trong thời buổi công nghệ thông tin phát triển 1 cách nhanh chóng, trẻ em, người lớn, người già đều có thể biết đến và sử dụng công nghệ thông tin 1 cách dễ dàng. Một hôm về nhà và giảng bài cho em gái hiện đang là học sinh thì thấy đầu óc của nó rất bị động, khi dạy toán cho em gái có những câu hỏi liên quan đến những phép tính cộng trừ nhân chia rất đơn giản nhưng khi hỏi thì em ấy vẫn không trả lời được và như một thói quen là cầm đến cái máy tính bỏ túi. Chính vì vậy nên em đã nghĩ tại sao mình không xây dựng 1 ứng dụng tính toán đơn giản giúp học sinh gia tăng khả năng tính nhẩm.

### Mô tả ứng dụng

Ứng dụng bao gồm 2 giao diện:

* Giao diện thứ nhất là giao diện khởi động của game, có nút PLAY và hiển thị những người chơi đạt điểm cao
* Giao diện thứ hai là giao diện để chơi game

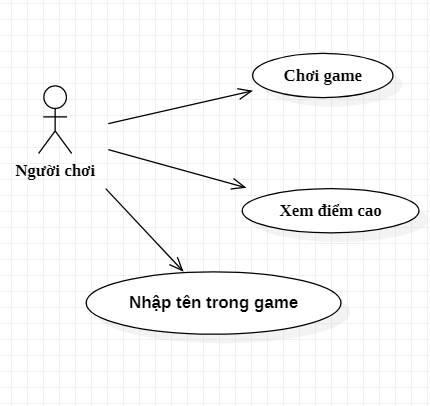
### Xây dựng kịch bản cho ứng dụng

* Ấn PLAY để chơi game
* Game sẽ hiển thị câu hỏi dạng A+B=C
* Chọn nút ĐÚNG hoặc SAI để trả lời câu hỏi
* Câu trả lời đúng thì điểm của bạn sẽ tăng thêm 1
* Khi điểm đạt đến từng mốc nhất định độ khó của game sẽ được tăng theo
  + Level 1: thời gian trả lời câu hỏi sẽ giảm xuống còn 4s
  + Level 2: thời gian trả lời câu hỏi sẽ giảm xuống còn 3s
  + Level 1: thời gian trả lời câu hỏi sẽ giảm xuống còn 3s
* Câu trả lời sai thì 1 thông báo sẽ được hiển thị thông báo tên và số điểm của người chơi
* Người chơi có thể chọn REPLAY để chơi lại hoặc có thể chọn HOME để quay về màn hình khởi động
* HIGHSCORE sẽ hiển thị 5 người có điểm cao nhất

## Phân tích thiết kế ứng dụng

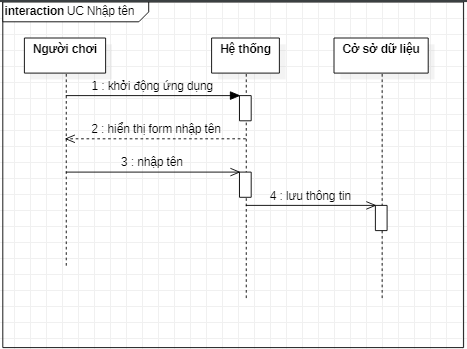
* Tác nhân: Người dùng
* Các chức năng:
  + Chơi game
  + Xem điểm cao

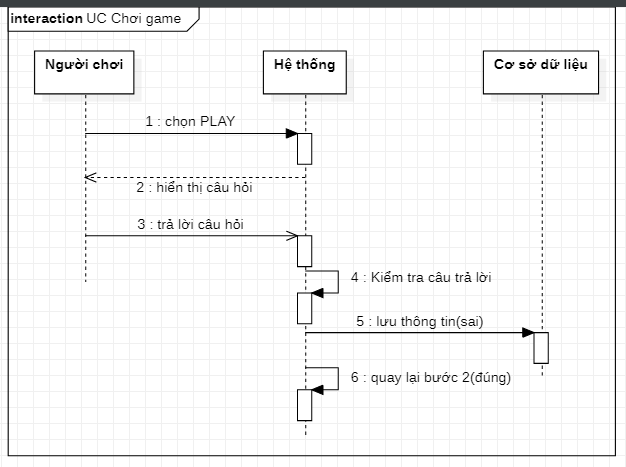
### Biểu đồ Use case

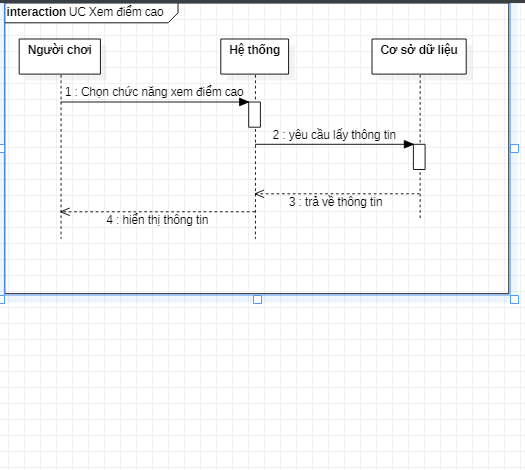


* Kịch bản use case:
  + Kịch bản cho chức năng Nhập tên trong game:
    - Mô tả: UC nhập tên cho phép người chơi nhập tên mình mong muốn trong game
    - UC bắt đầu khi người chơi chạy ứng dụng
    - Hệ thống sẽ hiển thì 1 ô để người chơi có thể nhập tên
    - Người chơi nhập tên vào ô trống
  + Kịch bản cho chức năng Chơi game:
    - Tiền điều kiện: Người chơi phải hoàn thành chức năng nhập tên trong game
    - UC bắt đầu khi người chơi nhấn nút PLAY
    - Hệ thống sẽ chuyển qua giao diện để chơi game
    - Hệ thống sẽ đưa ra câu hỏi (*Luồng 1*)
    - Người chơi đọc câu hỏi và chọn đúng hay sai:
    - Hệ thống kiểm tra câu trả lời
      * Nếu đúng hệ thống sẽ tăng điểm số của người chơi lên 1 và thực hiện *luồng 1*
      * Nếu sai hệ thông sẽ hiển thị 1 thông báo hiển thị điểm của người chơi:
        + Nếu người chơi chọn REPLAY hệ thống sẽ thực hiện *luồng 1*
        + Nếu người chơi chọn HOME hệ thống sẽ chuyển qua giao diện khởi động
  + Kịch bản chức năng xem điểm cao:
    - UC bắt đầu khi người chơi chọn nút HIGHSCORE
    - Hệ thống sẽ hiển thị danh sách 5 người chơi có thành tích tốt nhất

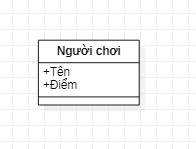
### Biểu đồ tương tác



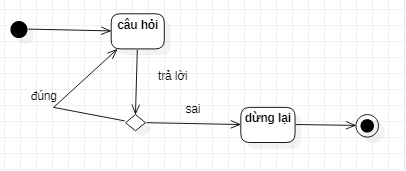




### Biểu đồ lớp



### Biểu đồ chuyển trạng thái



# CHƯƠNG 3: NGHIÊN CỨU HỆ ĐIỀU HÀNH ANDROID VÀ XÂY DỰNG ỨNG DỤNG FREAKING MATH

## Giao diện của game

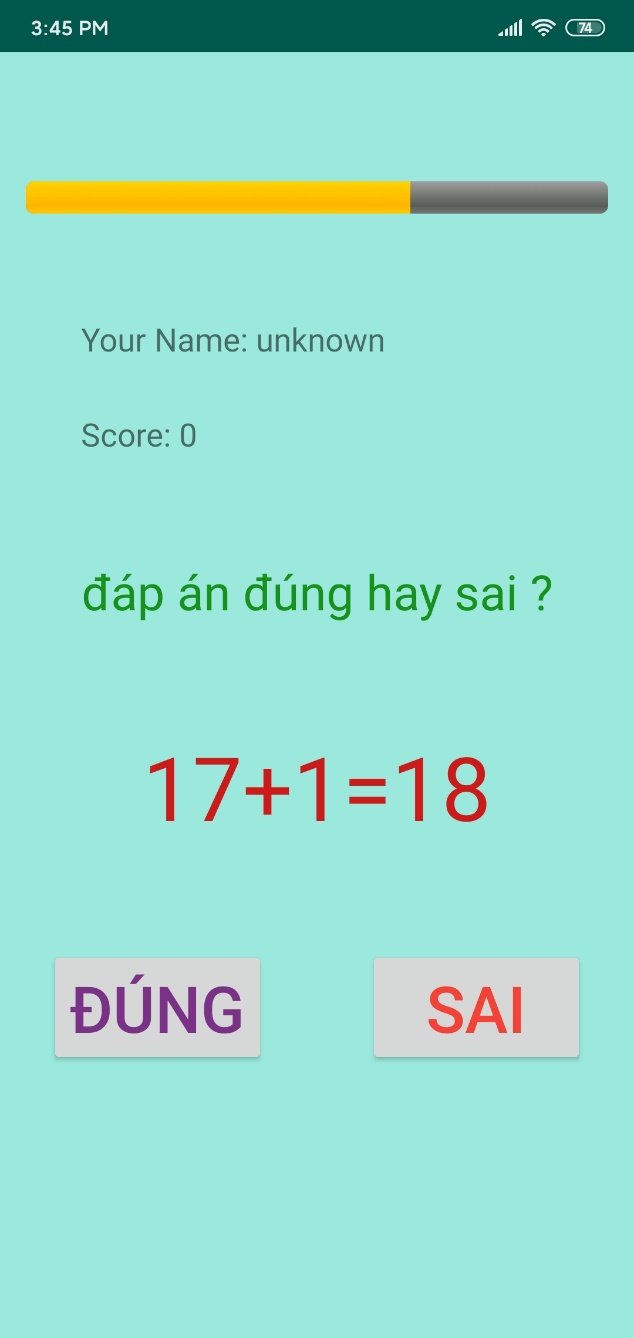
### Giao diện khởi động

Ảnh có chứa ảnh chụp màn hình, máy tính, bàn phím

Mô tả được tạo tự động

* Đây là giao diện khai vừa vào ứng dụng ta có thể nhập tên chọn bắt đầu chơi hoặc chọn HIGHSCORE để xem điểm cao

### Giao diện chơi game



* Đây là giao diện sau khi ấn PLAY chúng ta sẽ chọn ĐÚNG hoặc SAI để chơi

### Giao diện GameOver

Ảnh có chứa ảnh chụp màn hình, đồng hồ

Mô tả được tạo tự động

* Đây là giao diện khi trả lời sai hoặc hết thời gian

### Giao diện xem điểm cao

Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo tự động

* Khi chọn HIGHSCORE 1 bảng điểm sẽ hiện ra để người chơi có thể xem 5 điểm cao nhất.

# TỔNG KẾT

Sau nỗ lực, cố gắng của bản thân và sự hỗ trợ của cô Hoàng Thị Cành – giáo viên hướng dẫn xây dựng demo ứng dụng FreakingMath đã cơ bản hoàn thiện. Trong khoảng thời gian nhất định dành cho việc hoàn thành đề tài, do một số vấn đề về trình độ nên phần mềm vẫn chưa hoàn chỉnh và chính xác. Tuy nhiên, em đã đặt được một số kết quả:

* Về lý thuyết: đã tìm hiểu, nghiên cứu được về java, android studio và các kỹ thuật liên quan cho việc thực hiện đề tài, bước đầu biết sử dụng SQLite.
* Về thực nghiệm: đã xây dựng cơ bản thành công ứng dụng FreakingMath.

Trong tương lại em sẽ tìm hiểu, hoàn thiện và phát triển ứng dụng hơn nữa.

**Tài liệu tham khảo**

**Tài liệu tham khảo trực tuyến:**

1. [**https://niithanoi.edu.vn/ung-dung-thuc-te-cua-java.html**](https://niithanoi.edu.vn/ung-dung-thuc-te-cua-java.html)
2. [**https://freetuts.net/java-la-gi-gioi-thieu-ngon-ngu-java-1023.html**](https://freetuts.net/java-la-gi-gioi-thieu-ngon-ngu-java-1023.html)