1.Tuần 01

## 1.1 Giới thiệu công nghệ.

## 1.1.1 Hệ điều hành Ubutu.

- Ubuntu là một hệ điều hành mã nguồn mở dựa trên kiến trúc của hệ điều hành Debian cũng chính là một nhánh khác của các phiên bản hệ điều hành của Linux. Ubuntu có mặt trên desktop, server, và trong các thiết bị iot, robot.

- Ubuntu trong tiếng Zulu có nghĩa là “tình người” mô tả triết lý của ubuntu “tôi được là chính mình là nhờ mọi người”. Logo ubuntu mô tả ba người nắm tay nhau và tạo thành một vòng tròn.

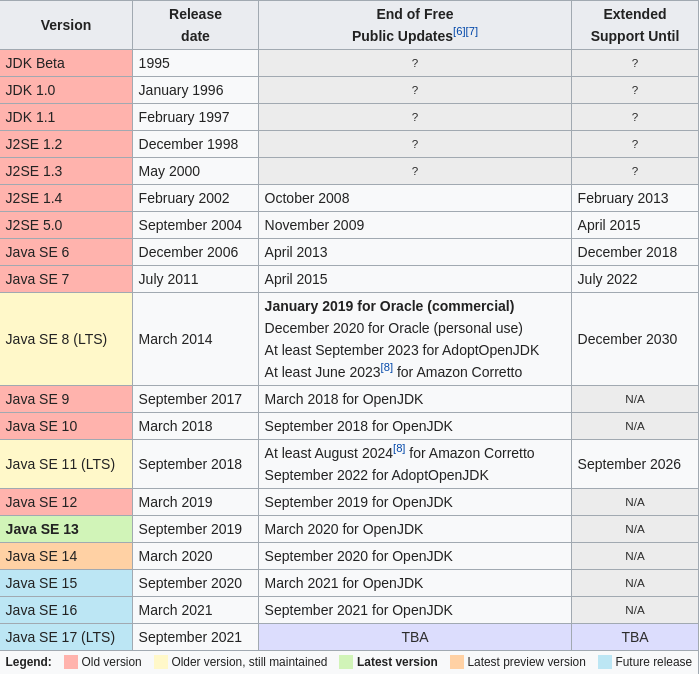
- Ubuntu có được thành công hơn các phiên bản Linux khác là vì được hỗ trợ bởi công ty Canocial. Khoảng những năm 2004, các phiên bản của Linux không hề thân thiện người dùng và được đánh giá là rất tệ khi không có được chăm sóc chuyên nghiệp từ cộng đồng. Tầm nhìn của Canocical về tương lai khi linux có được những người dùng trung thành cũng như là những người dùng mới khi và chỉ khi có một hệ điều hành được tạo ra và thân thiện với người dùng. Ubuntu được đầu tư về mặt quảng cáo, và công ty Canocial thu được các khoản lợi nhuận từ những công ty về dịch vụ điện toán và không thu phí người dùng. Điều đó tạo nên sự thành công của

Ubuntu.

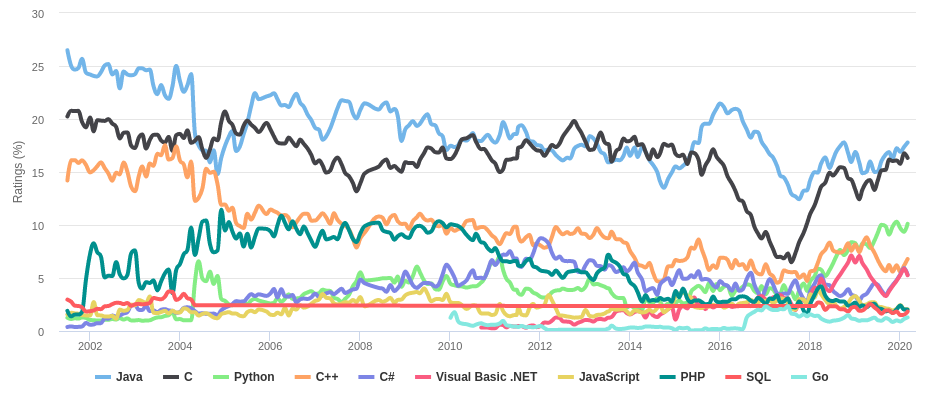


## 1.1.2 Ngôn ngữ Java và Object Oriented Programing.

- Ngôn ngữ Java được phát triển bởi James Gosling. Ông được biết đến là cha đẻ của Java vào năm 1995 nhưng sự thật là ông cùng các đồng nghiệp đã bắt đầu dự án từ đầu những năm 90. Java được tạo ra dựa trên các nền tảng cần thiết của ngôn ngữ lập trình là đơn giản, mạnh mẽ, dễ thay đổi, độc lập, bảo mật, hiệu năng cao, đa luồng, không phụ thuộc hệ điều hành, hướng đối tượng, linh hoạt. Java được sử dụng ở hầu hết các lĩnh vực trong nghành công nghệ thông tin.



Phiên bản JDK theo từng năm.



Bảng xếp hạng các ngôn ngữ lập trình đến năm 2020

Object Oriented Programing(OOP) lập trình hướng đối tượng được tạo ra để đáp ứng các nhu cầu về mô tả dữ liệu thông qua các trường( field). Mộ Object có nhiều field mỗi field sẽ là mỗi thuộc tính riêng biệt. Hướng lập trình của OOP rất đa dạng nhưng nổi tiếng nhất là hướng lập trình hướng đối tượng dựa trên lớp( Class based).

Các ngôn ngữ có hỗ trợ lập trình hướng đối tượng là C++, Java, Python đồng thời ba ngôn ngữ này cũng hỗ trơ các dạng lập trình như imperative language, procedural programingocedural. Bên cạnh đó còn rất nhiều các loại ngôn ngữ khác hỗ trợ lập trình hướng đối tượng [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)" \o "C Sharp (programming language)), [PHP](https://en.wikipedia.org/wiki/PHP" \o "PHP), [JavaScript](https://en.wikipedia.org/wiki/JavaScript" \o "JavaScript), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)" \o "Ruby (programming language)), [Perl](https://en.wikipedia.org/wiki/Perl" \o "Perl), [Object Pascal](https://en.wikipedia.org/wiki/Object_Pascal" \o "Object Pascal),  [Objective-C](https://en.wikipedia.org/wiki/Objective-C" \o "Objective-C), [Dart](https://en.wikipedia.org/wiki/Dart_(programming_language)" \o "Dart (programming language)), [Swift](https://en.wikipedia.org/wiki/Swift_(programming_language)" \o "Swift (programming language)), [Scala](https://en.wikipedia.org/wiki/Scala_(programming_language)" \o "Scala (programming language)), [Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp" \o "Common Lisp), [MATLAB](https://en.wikipedia.org/wiki/MATLAB" \o "MATLAB), and [Smalltalk](https://en.wikipedia.org/wiki/Smalltalk" \o "Smalltalk).

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| - Một team làm việc đồng thời và độc lập cho các tính năng cùng dự án dễ dàng.  - Cấu trúc của class trong OOP giúp nó có thể tái sử dụng.  - Ngôn ngữ được một cộng đồng rất lớn hỗ trợ cũng như là có bề dày lịch sử nên sẽ dễ bảo trì nếu có vấn đề xảy ra. | - Hướng đối tượng do cần tạo ra nhiều class nên sử dụng nhiều tài nguyên hơn các dạng lập trình khác.  - Trong trường hợp đã mở rộng các tính năng quá nhiều và quản lý không tốt thì một lượng lớn code không được tái sử dụng ảnh hượng đến hiệu năng và gây ra phí tổn thất là lớn.  - OOP dễ thiết kế hơn là hiện thực, trong quá trình hiện thực quá trình các tiến trình bị lặp lại mà không được tái sử dụng có khả năng xảy ra là cao. |

## 1.1.3. Ngôn ngữ Scala và Functional Programing.

- Scala được tạo ra để phục vụ cho nhiều hướng phát triển của ngôn ngữ lập trình. Martin Odersky là người phát triển cũng như là cha đẻ của ngôn ngữ scala. Scala không phải là phần mở rộng của ngon ngữ java nhưng giữa hai ngôn ngữ này hoàn toàn trao đổi dữ liệu trong quá trình hoạt động. Trong quá trình complie dữ liệu, scala truyền file về Java bytecode và chạy trên JVM( Java virtual machine).

Scala được thiết kế để phục lập trình hướng đối tượng và cả lập trình hướng hàm, tất cả mọi giá trị trong scala đều có kiểu object và tất cả mọi hàm trong scala đều trả một giá trị. Scala được lấy cảm hứng từ chữ scalable nghĩa là có thể mở rộng dễ dàng dựa trên nhu cầu của người dùng.

Martin Ordesky bắt đầu dự án scala từ năm 2001 tại viện Ecole Polytechnique Federale de Lausanne (EPFL). Chính thức công bố ngôn ngữ là vào năm 2004.

Ngôn ngữ scala dần nổi tiếng và được sử dụng trong các phần quan trọng của các công ty công nghệ lớn.

* Twitter thông báo chuyển toàn bộ hệ thống backend từ Ruby về Scala.
* Apple sử dụng scala ở một số team cùng với Java Play Framework.
* Năm 2014 New York Times tuyên bố sử dụng scala, akka, play framework để xây dựng hệ thống quản lý nội dung Blackbeard.
* Google sử dụng scala để xây dựng Firebase và Nest vì cần scale lượng dữ liệu khổng lồ.
* Walmart sử dụng scala cho backend.



## 1.1.4. SQL và NoSQL.

## 1.1.5. Hệ cơ sở dữ liệu MySQL.