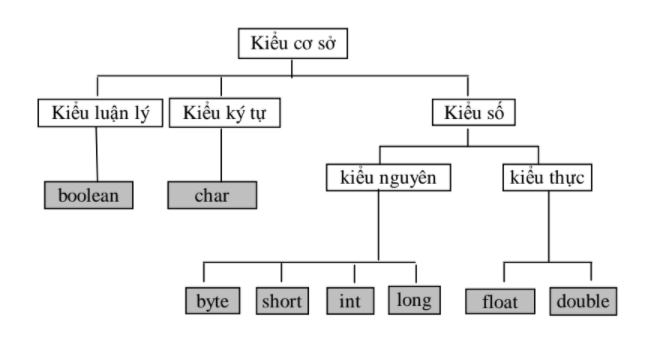
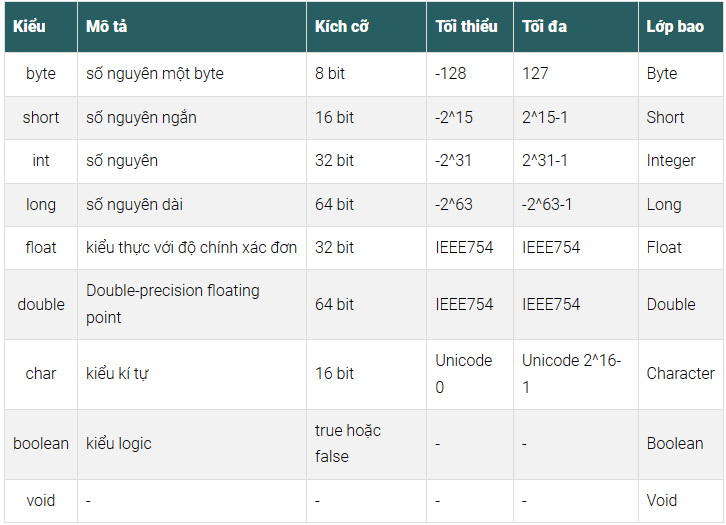
**Task1: Tìm hiểu về kiểu dữ liệu trong Java**

**1. Các kiểu dữ liệu nguyên thủy(Prinitive Data type):**

 Có 8 kiểu dữ liệu nguyên thủy(Prinitive Data type) đó là: Boolean, Byte, Char, Short, Int, Long, Float, Doule.

Các thông số chi tiết của các kiểu dữ liệu:



Đặc điểm của các biến có kiểu nguyên thủy là vùng nhớ của chúng được cấp phát ở phần stack. Do vậy việc truy xuất vào một biến kiểu nguyên thủy rất nhanh.

**1.1 Cách sử dụng các kiểu dữ liệu:**

* Kiểu dữ liệu Byte:
* Dùng để lưu dữ liệu kiểu số nguyên có kích thước một byte (8 bit)
* Giá trị nhỏ nhất là -128 (-2^7) và giá trị lớn nhất là 127. (2^7 -1)
* Giá trị mặc định là 0
* Kiểu dữ liệu byte được sử dụng để lưu giữ khoảng trống trong các mảng lớn, chủ yếu là các số nguyên.
* Ví dụ: byte x = 20 , byte y = -10
* Kiểu dữ liệu Short:
* Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 2 byte (16 bit).
* Giá trị nhỏ nhất là -32,768 (-2^15) và giá trị lớn nhất là 32,767 (2^15 -1).
* Kiểu dữ liệu short cũng có thể được sử dụng để lưu bộ nhớ như kiểu dữ liệu byte.
* Giá trị mặc định là 0.
* Ví dụ: short t = 50, short z = -10
* Kiểu dữ liệu int:
* Dùng để lưu dữ liệu có kiểu số nguyên, kích cỡ 4 byte (32 bit).
* Giá trị nhỏ nhất là - 2,147,483,648.(-2^31) và giá trị lớn nhất là 2,147,483,647 (2^31 -1)
* Nói chung, int được sử dụng như là kiểu dữ liệu mặc định cho các giá trị nguyên.
* Giá trị mặc định là 0.
* Ví dụ: int a = 5, int b = -50
* Kiểu dữ liệu Long:
* Dùng để lưu dữ liệu có kiểu số nguyên có kích thước lên đến 8 byte.
* Giá trị nhỏ nhất là -9,223,372,036,854,775,808.(-2^63) và lớn nhất là 9,223,372,036,854,775,807. (2^63 -1)
* Kiểu này được sử dụng khi cần một dải giá trị rộng hơn int.
* Giá trị mặc định là 0L.
* Ví dụ: long a = 100000L, int b = -200000L
* Kiểu dữ liệu Float:
* Dùng để lưu dữ liệu có kiểu số thực, kích cỡ 4 byte (32 bit)
* Kiểu Float được sử dụng chủ yếu để lưu bộ nhớ trong các mảng rộng hơn các số thực dấu chấm động.
* Giá trị mặc định là 0.0f.
* Kiểu Float không bao giờ được sử dụng cho các giá trị chính xác như currency.
* Ví dụ: float usd = 22.5f
* Kiểu dữ liệu Double:
* Kiểu dữ liệu double được sử dụng để lưu dữ liệu có kiểu số thực có kích thước lên đến 8 byte
* Nói chung, kiểu dữ liệu này được sử dụng như là kiểu mặc định cho các giá trị decimal.
* Kiểu double không bao giờ được sử dụng cho các giá trị chính xác như currency.
* Giá trị mặc định là 0.0d.
* Ví dụ: double ct = 676.7
* Kiểu dữ liệu Boolean:
* Độ lớn chỉ có 1 bit
* Dùng để lưu dữ liệu chỉ có hai trạng thái true hoặc false
* Giá trị mặc định là false.
* Ví dụ: boolean switch1 = true
* Kiểu dữ liệu Char
* Dùng để lưu dữ liệu kiểu kí tự hoặc số nguyên không âm có kích thước 2 byte (16 bit)
* Giá trị nhỏ nhất là '\u0000' (hoặc 0) và giá trị lớn nhất là '\uffff' (hoặc 65,535)..
* Ví dụ: char nam ='T'

**1.3. Kiểu số nguyên trong Java**

* Java cung cấp 4 kiểu số nguyên khác nhau là: *byte*, *short*, *int*, *long.* Kích thước, giá trị nhỏ nhất, lớn nhất, cũng như giá trị mặc định của các kiểu dữ liệu số nguyên được mô tả chi tiết trong bảng trên.
* Kiểu mặc định của các số nguyên là kiểu *int*.
* Các số nguyên kiểu *byte*và *short* rất ít khi được dùng.
* Trong java không có kiểu số nguyên không dấu như trong ngôn ngữ C/C++.

**2. Kiểu dữ liệu object(Non-primitive data type)**

Là những kiểu dữ liệu không thuộc kiểu dữ liệu nguyên thủy, những kiểu dữ liệu thuộc Object gồm : Object, Array object, Regular Expression, Function.

* Object : chứa các cặp *key* và *value*, nó không giống như Array, các cặp *key* và *value* trong object có thể có nhiều kiểu dữ liệu khác nhau. Ta có thể truy cập vào giá trị của một object bằng cách ghi tên object đó ra sau đó là *key* mà bạn muốn lấy giá trị.
* Array : hay còn gọi là mảng, nó chứa một danh sách các giá trị thuộc cùng một kiểu dữ liệu, các giá trị trong mảng có thể thuộc bất kỳ dữ liệu nào. Để truy cập đến một giá trị bất kỳ trong mảng thì bạn phải biết vị trí *index* của giá trị đó trong mảng(Trong JavaScript, mảng bắt đầu từ 0).
* Regular Expression : hay còn được gọi là Biểu thức chính quy.
* Fucion: Hàm