**Cấu trúc của một class**

Được khai báo bằng từ khoá ***class + tên-lớp***{}

**Hàm main**

public static void main(String[] args) {}

Hàm main là hàm chính trong lớp Java, nó xác định điểm bắt đầu các chương trình java khi thực thi

Từ khoá **public static void** đứng trước tên phương thức để xác định cách phương thức có thể được sử dụng ở mức độ nào.

**String [] args** là một mảng các chuỗi được truyền vào hàm main như là tham số.

**Chương trình Java đầu tiên**

In ra màn hình

public class Hello {

public static void main(String[] args) {

System.out.println("Hello World!");

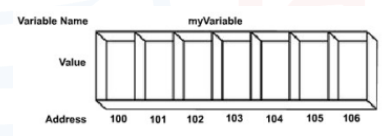
}

}

**Biến**

> Biến là một vị trí trong bộ nhớ máy tính mà ở đó giá trị được lưu trữ có thể được truy xuất sau đó

> Sử dụng để lưu trữ dữ liệu có thể thay đổi trong quá trình thực thi



> Tên biến phải là duy nhất trong 1 phạm vi

> Các biến cần được khai báo trước khi sử dụng

> Khai báo biến bao gồm đặc tả tên biến, và đặc tả kiểu dữ liệu biến đại diện

> Cú pháp:

**Type variable**

Hay

**Type variable = value;**

> Ví dụ:

**int var1;**

**int var2 = 10;**

> Kiểu của biến:

- Kiểu dữ liệu nguyên thuỷ

- Tên của một lớp

- Một mảng

> Để khai báo 1 biến mới, ta phải khai báo 1 lớp mới, sau đó kiểu biến mới được khai báo kiểu là lớp mới đó

> Các loại biến trong Java:

- Biến đối tượng

- Biến lớp

- Biến cục bộ

***Biến đối tượng***

> Dùng để định nghĩa thuộc tính, trạng thái cho 1 đối tượng

> Có thể là biến toàn cục của 1 đối tượng

> Khai báo:

**Type** variableName;

***Biến lớp***

> Tương tự biến đối tượng nhưng giá trị nằm trong chính lớp đó

> Ảnh hưởng toàn cục đến 1 lớp và tất cả các đối tượng trong lớp đó

> Thích hợp dùng để trao đổi thông tin giữa các đối tượng khác nhau trong cùng 1 lớp hay theo dõi trạng thái toàn cục của đối tượng

> Khai báo:

**static** **Type** variableName;

***Biến cục bộ***

> Được khai báo và sử dụng trong thân phương thức

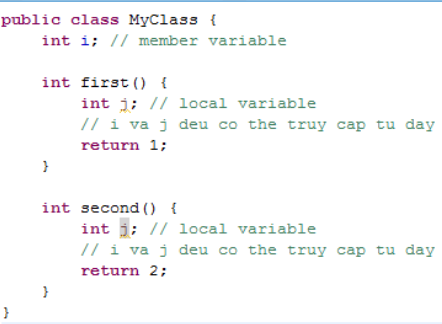
> Bắt buộc phải gán giá trị trước khi sử dụng

> Java không có biến toàn cục, biến đối tượng hoặc biến lớp được dùng để truyền thông tin toàn cục

***Phạm vi của biến***

> Biến đối tượng/lớp (toàn cục): có thể truy cập bất cứ đâu trong toàn bộ chương trình

> Biến toàn cục tồn tại giới hạn và quan hệ chỉ trong phần nhỏ của mã



**Cách đặt tên**

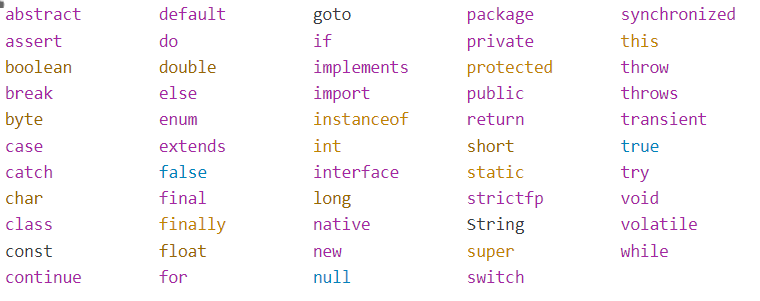
> Tên có thể được bắt đầu bằng một kí tự, hoặc dấu: $. \_

> Tên không được bắt đầu bằng 1 số

> Không được chứa dấu cách

> Java phân biệt chữ thường với chữ hoa

> Các tên biến không nên trùng với các từ khoá Java sau:



> Tên biến nên mô tả nội dung mà nó chứa

> Có thể gán giá trị ngay cho biến bằng dấu “=” ngay khi khai báo

Ví dụ: String thongBao = “Xin chao”;

**Comments**

|  |  |  |
| --- | --- | --- |
| Loại chú thích | Bắt đầu | Kết thúc |
| Trên 1 dòng | // |  |
| Trên nhiều dòng | /\* | \*/ |
| Dành cho Javadoc | /\*\* | \*/ |

**Lưu ý:**

* Nội dung ghi chú phải ngắn gọn, dễ hiểu
* Cập nhật ghi chú lại khi nội dung có thay đổi
* Không ghi chú lồng vào nhau

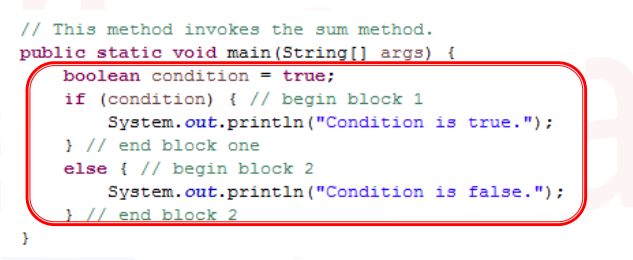
**Câu lệnh và khối lệnh**

> Câu lệnh kết thúc bằng dấu ;

> Các câu lệnh đơn có thể nối lại tạo thành khối lệnh thuộc 1 lớp

> Bộ lệnh không giới hạn trong cặp dấu ngoặc { và }

> Khối lệnh có thể đặt trong khối lệnh khác



**Tập kí tự**

**-** 26 chữ cái hoa: A…Z, 26 chữ cái thường: a…z

- 10 chữ số: 0…9

- Các kí hiệu toán học: +, -, \*, /, %, =, ()…

- Dấu nối: \_

- Các kí hiệu đặc biệt khác: :, {}, [], ?, \, &, !, #, $

- Java dùng bộ kí tự Unicode

**Từ khoá**

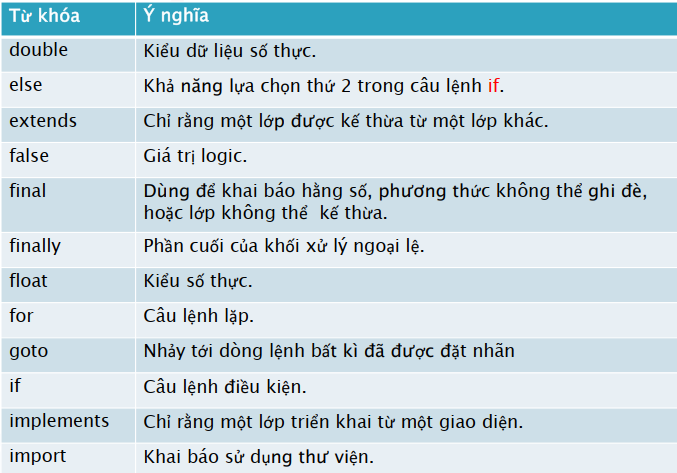
> Là những từ có ý nghĩa xác định

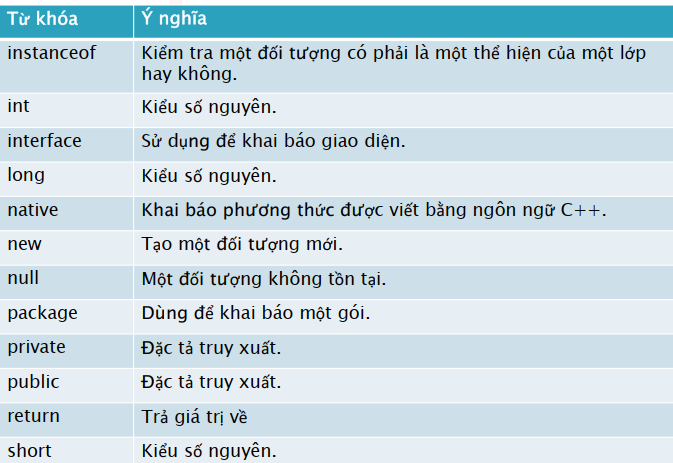
> Dùng để khai báo các kiểu dữ liệu, viết các toán tử và câu lệnh

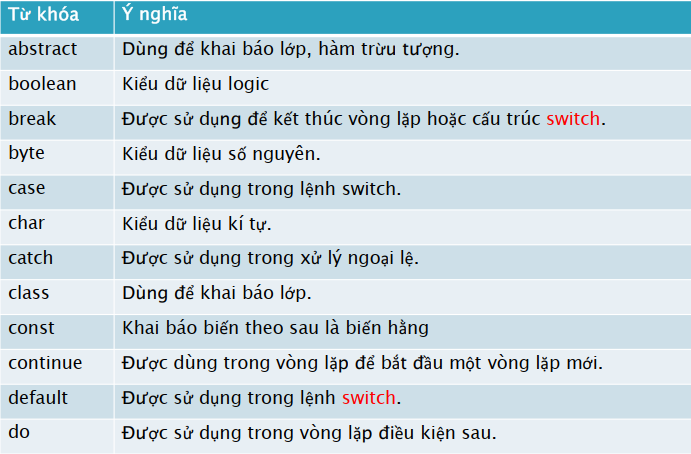
**CHÚ Ý:**

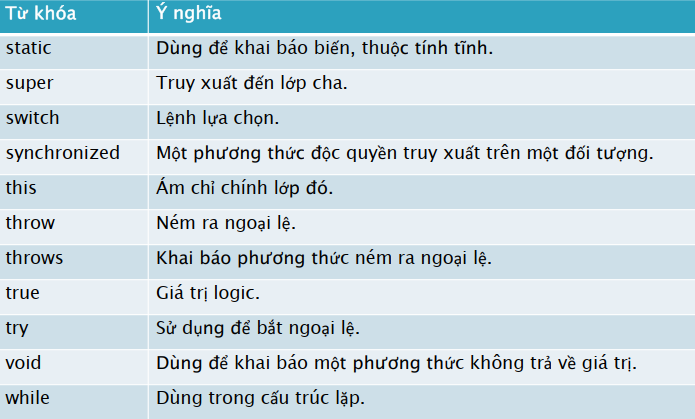
**Không được dùng từ khoá dể đặt tên cho hằng, biến mảng, hàm, …**

**Từ khoá được viết bằng chữ thường**

****

****

****



**Kiểu dữ liệu**

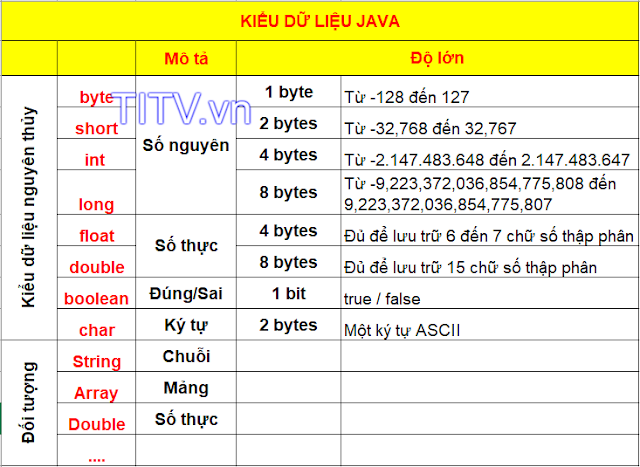
> Mỗi biến phải có 1 kiểu dữ liệu

- Kiểu dữ liệu nguyên thuỷ

- Kiểu dữ liệu dẫn xuất/đối tượng

> Kiểu dữ liệu xác định m iền giá trị cho biến

***Kiểu nguyên thuỷ***

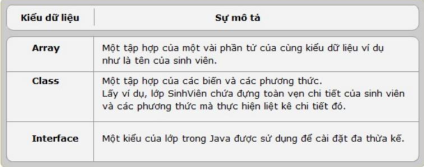


***Kiểu dữ liệu dẫn xuất***

> Tham chiếu tới một giá trị hay là tập hợp các giá trị mà biến khai báo



> Các kiểu dữ liệu dẫn xuất:



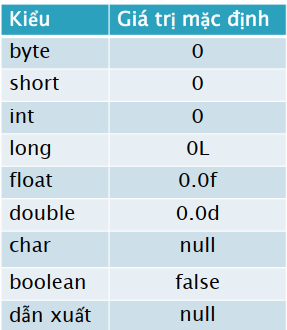
**Chú ý:**

* Kiểu dữ liệu nguyên thuỷ chữ đầu viết thường
* Kiểu dữ liệu dẫn xuất chữ đầu viết hoa
* Kiểu dữ liệu ***char*** khi khai báo sẽ nằm trong ‘…’
* Kiểu dữ liệu ***String*** khi khai báo sẽ nằm trong “…”

**Giá trị mặc định**

> Java hỗ trợ khởi tạo các giá trị mặc định cho các biến

> Lỗi hay gặp phải khi lập trình là sử dụng biến chưa khởi tạo giá trị (lỗi NullPointerException)



**Hằng số**

> Từ khoá **final** chỉ 1 biến không thể thay đổi giá trị

> Cách khai báo:

**final** data\_types = value

> Các hàm và lớp cũng có thể được khai báo **final**

**-** Phương thức **final** không thể viết chồng (**override**)

🡪 Nếu cố tình gán thêm giá trị khác sẽ gặp lỗi biên dịch

- Lớp **final** không thể có lớp con kế thừa (**extend**)

**Cách kiểm tra và xử lý lỗi biên dịch**

<https://www.titv.vn/2021/04/java-08-cach-kiem-tra-va-xu-ly-loi-bien.html>

**Nhập xuất dữ liệu**

***Scanner***

* Cho phép người dùng đọc giá trị của một vài kiểu dữ liệu từ Console
* Lớp Scanner cho phép người dùng nhập giá trị từ bàn phím đối với *một vài* kiểu dữ liệu
* Một số phương thức của Scanner với System.in là đối tượng dòng đầu vào

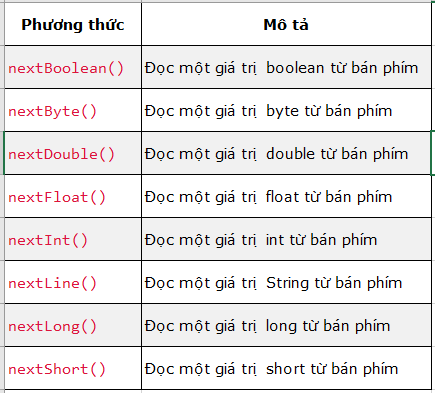
***Cách khai báo:***

**public static void** main (String args[]) {

Scanner scaner = new Scanner(System.*in*);

}

***Một số phương thức nhập:***

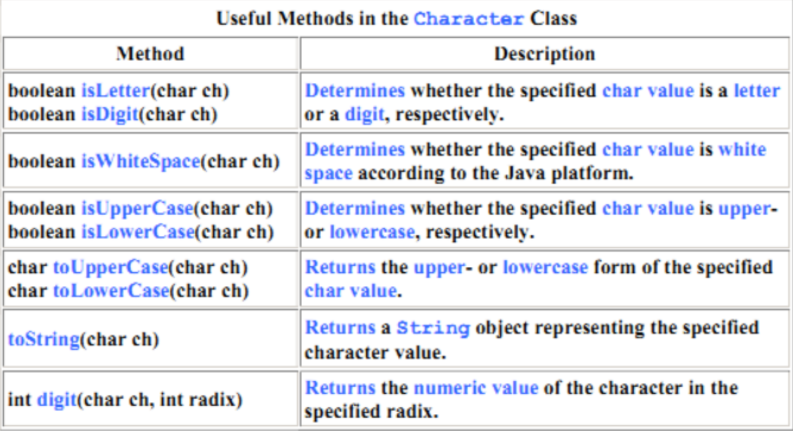


* Nhập xuất dữ liệu là tác vụ mức hệ thống
* Gói **java.io** chứa các lớp cho việc xuất nhập
* Java cung cấp class System mô tả hệ thống
* **System.out** là đối tượng xuất mặc định (màn hình)
* **System.in** là đối tượng nhập mặc định (bàn phím)
* **System.err** là đối tượng xuất lỗi (màn hình)
* Methods xuất dữ liệu ra màn hình:
  + System.out.print(Dữ liệu xuất);
  + System.out,println(Dữ liệu xuất);
* Dữ liệu xuất có thể là: ký tự, số, chuỗi, …

**Dãy Escape**

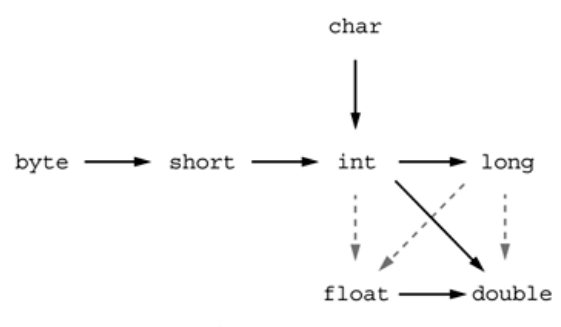


**Các phương thức hữu dụng trong Character**



**Ép kiểu**

> Chuyển đổi giữa các loại dữ liệu số:

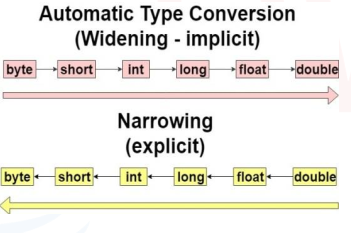


> Dùng để chuyển từ một kiểu dữ liệu này sang kiểu dữ liệu khác

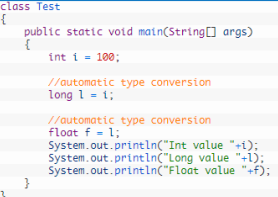
> Có 2 dạng ép kiểu:

- Ép kiểu ngầm định (implicit)

- Ép kiểu tường minh (explicit)



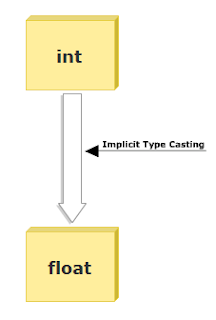
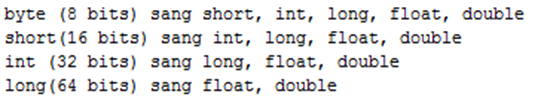
***Ép kiểu ngầm định***

> Khi một kiểu dữ liệu được gán cho 1 biến của 1 kiểu khác 🡪 tự động chuyển kiểu

> Điều kiện:

- Hai kiểu phải tương thích

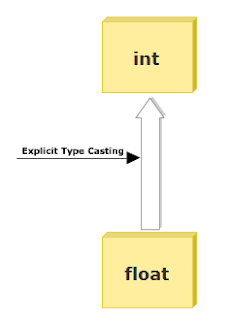
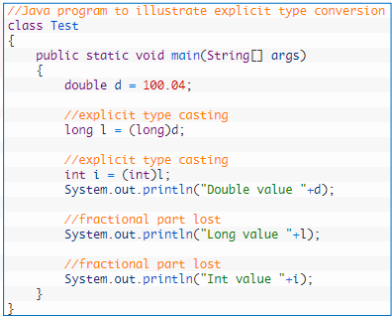
- Kiểu đích phải lớn hơn kiểu nguồn

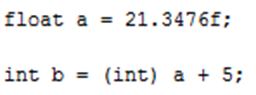


***Ép kiểu tường minh***

> Gán một giá trị của kiểu dữ liệu lớn hơn sang kiểu dữ liệu nhỏ hơn

> Khi cần chuyển sang kiểu có độ chính xác cao hơn





***Cách chuyển đổi kiểu dữ liệu***

> Có 3 dạng chuyển đổi kiểu dữ liệu:

- Chuyển đổi cho các kiểu dữ liệu cơ bản

**(New Type) value;**

- Chuyển đổi kiểu cho các đối tượng: các lớp chuyển đổi phải kế thừa nhau

**(New Class) object;**

- Chuyển đổi cho các kiểu dữ liệu cơ bản sang đối tượng và ngược lại

Chỉ chuyển đổi giữa các đối tượng có sẵn trong gói java.lang tương ứng với

các dữ liệu nguyên thuỷ

**int intObject = new Integer(32);**