* Biến là tên đặt cho vùng nhớ dùng để lưu trữ dữ liệu tạm thời nhằm phục vụ hoạt động của chương trình.
* Biến sẽ bị hủy khi chương trình kết thúc
* Biến được cấp phát bộ nhớ trong RAM
* Biến được phân thành các nhóm chính: biến cục bộ, biến của lớp, hằng số.

**Cú pháp khai báo biến**

* Cú pháp tổng quát: *kiểu danh\_sách\_biến;*
* Trong đó:
  + Kiểu là các kiểu dữ liệu hợp lệ trong ngôn ngữ Java, gồm các kiểu nguyên thủy và kiểu tham chiếu như int, float, bool hay các class, interface.
  + Danh sách biến có thể gồm 1 hoặc nhiều biến, nếu có nhiều biến thì các biến sẽ phân tách nhau bởi dấu phẩy.
  + Tên biến thường là danh từ hoặc cụm danh từ đặt tên sao cho tự bản thân biến mang ý nghĩa mà nó mô tả.
  + Tên biến không trùng với các từ khóa và các từ dành riêng trong Java.
  + Tên biến luôn bắt đầu bằng chữ cái thường, sau đó có thể là chữ cái, chữ số, dấu gạch dưới. Không bao giờ bắt đầu tên biến bằng một kí tự khác chữ cái thường.
  + Nếu tên biến có nhiều từ cấu thành thì viết liền nhau và viết hoa chữ cái đầu từ.
  + Tên biến phân biệt chữ hoa chữ thường. Tức là *age* khác với *Age* và khác *aGE.*
  + Luôn kết thúc câu lệnh khai báo bằng 1 dấu chấm phẩy.
  + Khuyến nghị khai báo mỗi biến trên 1 dòng.
* Ví dụ:

int age; *// tuổi*

float avgGrade; *// điểm TB*

double interestRate; *// lãi suất*

long ballance; *// số dư*

String fullName; *// họ tên đầy đủ*

String address; *// địa chỉ*

*// sau đây là các biến không hợp lệ:*

int 5year; *// tên bắt đầu bởi số*

float my bonus; *// trong tên có dấu cách*

double max-volume; *// trong tên có dấu -*

**Khởi tạo giá trị cho biến**

* Mọi biến trong Java phải khai báo trước khi nó có thể được sử dụng
* Trong quá trình khai báo biến mà ta gán cho biến 1 giá trị ban đầu cùng kiểu thì ta gọi là khởi tạo giá trị cho biến
* Cú pháp tổng quát: *kiểu tên\_biến = giá\_trị;*
* Trong đó:
  + Kiểu là các kiểu dữ liệu hợp lệ trong ngôn ngữ Java như int, float, boolean…
  + Tên biến đặt theo quy tắc đặt tên biến
  + Dấu = gọi là toán tử gán
  + Giá trị là các giá trị hợp lệ cùng kiểu với kiểu của biến
* Ví dụ:

int age = 0; *// khai báo và khởi tạo tuổi bằng 0*

float avgGrade = 0.0f; *// khai báo và khởi tạo điểm TB = 0.0*

*String fullName = ""; // khai báo và khởi tạo họ tên rỗng*

* Cú pháp khởi tạo giá trị cho biến luôn kèm theo kiểu của biến ở đầu

**Gán giá trị cho biến**

* Khi ta đã khai báo biến, sau đó mới cung cấp giá trị cùng kiểu cho các biến đó thì ta gọi cú pháp này là gán giá trị cho biến
* Cú pháp tổng quát: *tên\_biến = giá\_trị;*
* Trong đó:
  + Tên biến là các tên biến đã có trước đó
  + Dấu = gọi là toán tử gán, cả câu lệnh gồm hai vế trái và phải + toán tử gán gọi là gán giá trị cho biến
  + Giá trị là các giá trị cùng kiểu với kiểu của biến

Ví dụ:

int age = 0; *// khai báo và khởi tạo tuổi bằng 0*

float avgGrade = 0.0f; *// khai báo và khởi tạo điểm TB = 0.0*

double interestRate; *// lãi suất*

age = 20; *// gán giá trị 20 cho biến age*

avgGrade = 3.25f; *// gán giá trị 3.25 cho điểm TB*

interestRate = 5.25; *// gán giá trị cho lãi suất*

**Hằng số**

* Hằng số là các biến mà giá trị của nó đã biết trước, không có nhu cầu thay đổi và không thể thay đổi kể từ sau khi được gán giá trị
* Cú pháp tổng quát: final kiểu TÊN\_HẰNG\_SỐ = giá\_trị;
* Trong đó:
  + *final* là một từ khóa mặc định
  + Kiểu là bất kì kiều hợp lệ nào trong Java
  + Tên hằng số luôn viết hoa hoàn toàn, mang ý nghĩa nào đó. Nếu tên có nhiều từ cấu thành thì phân tách nhau bởi dấu gạch dưới.
  + Mọi hằng số không bắt buộc phải gán giá trị ngay khi khai báo nhưng sau khi đã gán giá trị lần đầu tiên thì giá trị của hằng số không thể thay đổi hoặc là chương trình sẽ báo lỗi.
* Ví dụ hằng số:

final double PI = 3.141592653589793; *// số PI*

final double E = 2.718281828459; *// số E*

final int MAX\_VALUE;

MAX\_VALUE = 100; *// gán giá trị 100 cho hằng số MAX\_VALUE*

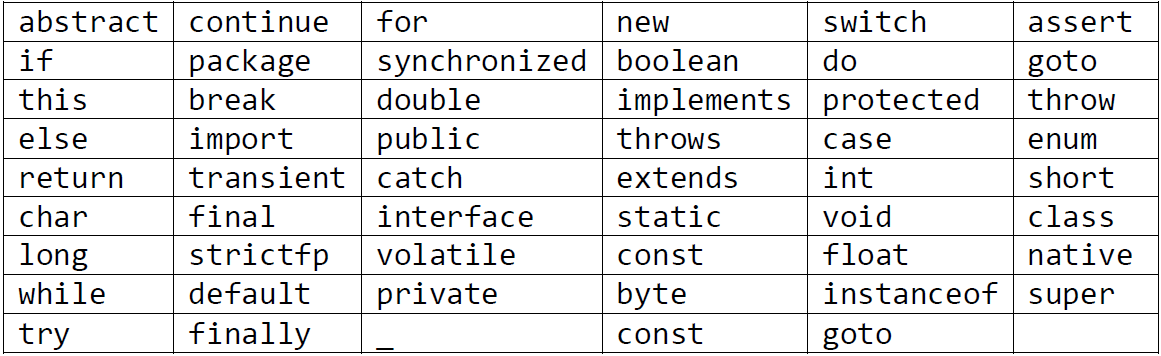
MAX\_VALUE = 500; *// error! giá trị của hằng số không thể thay đổi*

final String MONDAY = "MONDAY"; *// ok*

MONDAY = "FRIDAY"; *// error!*

**Các từ khóa**

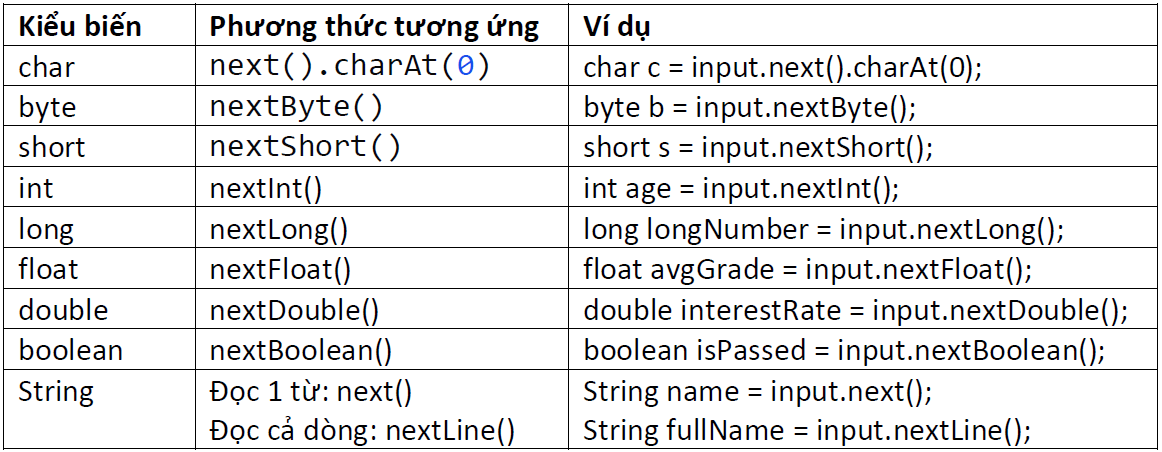
* Từ khóa, từ dành riêng là các từ được gán sẵn 1 chức năng cụ thể nào đó và chỉ sử dụng với mục đích thực hiện chức năng đó mà thôi.
* Các từ khóa và từ dành riêng luôn viết thường hoàn toàn
* Sau đây là các từ khóa và từ dành riêng trong Java. Khi khai báo biến mà thấy tên biến in đậm lên hoặc có màu khác thường thì đồng nghĩa với việc tên biến đã vi phạm vào các từ khóa, từ dành riêng.

[](https://braniumacademy.net/wp-content/uploads/2020/08/c%C3%A1c-t%E1%BB%AB-kh%C3%B3a-v%C3%A0-t%E1%BB%AB-d%C3%A0nh-ri%C3%AAng-trong-Java.png)

* Dấu gạch dưới \_ là một từ dành riêng sẽ được sử dụng trong tương lai
* Hai giá trị *true, false* không phải từ khóa nhưng nó là giá trị của riêng kiểu boolean cũng không được phép sử dụng làm tên biến
* Biểu diễn giá trị *null* cũng không được dùng làm tên biến
* *var, yiel* cũng không được dùng làm tên biến
* Các từ khóa bị giới hạn khác là: *open, module, requires, transitive, exports, opens, to, uses, provides, with*

**Nhập dữ liệu cho biến**

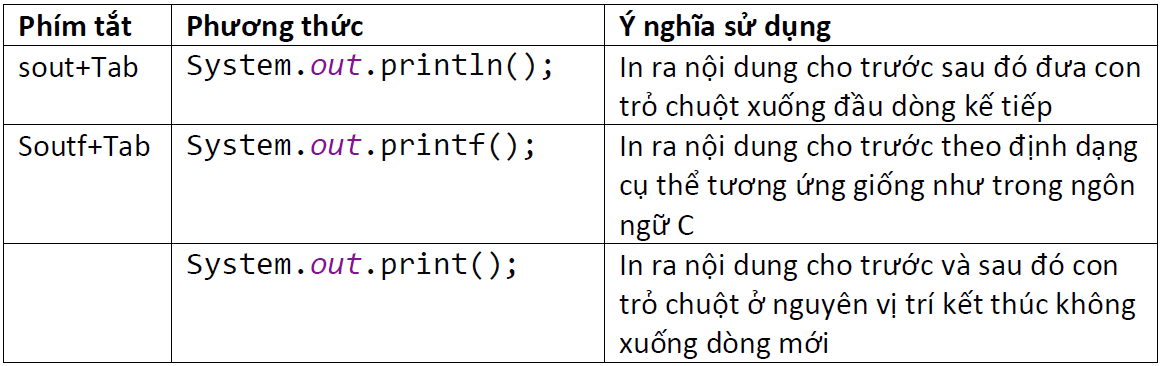
* Để nhập dữ liệu vào từ bàn phím và gán cho các biến, ta sử dụng đối tượng của lớp Scanner: Scanner input = new Scanner(System.in);
* in là chuẩn đầu vào dữ liệu đọc từ bàn phím
* Để đọc dữ liệu kiểu nào ta dùng phương thức nextX trong đó X là tên kiểu của biến cần lấy giá trị. Cụ thể cho trong bảng sau:

[](https://braniumacademy.net/wp-content/uploads/2020/08/c%C3%BA-ph%C3%A1p-nh%E1%BA%ADp-li%E1%BB%87u-cho-t%E1%BB%ABng-ki%E1%BB%83u.png)

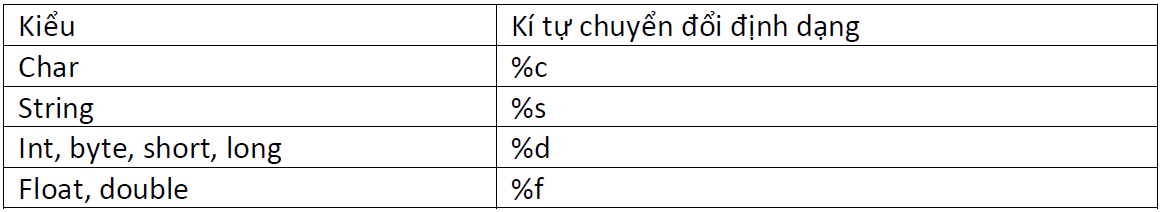
* Thường đưa ra lời nhắc cho người dùng nhập đúng dữ liệu của kiểu ta muốn nhận vào.
* Khi đọc vào 1 từ hoặc các giá trị khác, việc đọc sẽ dừng khi gặp khoảng trắng như dấu cách, dấu tab, hoặc khi bạn ấn enter xuống dòng(dấu hiệu kết thúc dòng).

**Xuất dữ liệu ra màn hình**

* Để xuất dữ liệu trong các biến ra màn hình ta dùng phương thức print, printf hoặc println của System.out
* out là chuẩn đầu ra dữ liệu mặc định của Java – màn hình console
* Cú pháp để thực hiện việc xuất dữ liệu ra với công cụ IntelliJ như sau:

[](https://braniumacademy.net/wp-content/uploads/2020/08/c%C3%BA-ph%C3%A1p-xu%E1%BA%A5t-d%E1%BB%AF-li%E1%BB%87u-ra-m%C3%A0n-h%C3%ACnh.png)

* Một số kí tự định dạng với printf:

[](https://braniumacademy.net/wp-content/uploads/2020/08/M%E1%BB%99t-s%E1%BB%91-%C4%91%E1%BB%8Bnh-d%E1%BA%A1ng-chuy%E1%BB%83n-%C4%91%E1%BB%95i.png)

Ví dụ áp dụng:

import java.util.Scanner;

public class Bai3 {

public static void main(String[] args) {

// hằng số

final int MAX = 500;

final double PI;

PI = 3.141592;

final String MONDAY = "MONDAY";

final boolean OK = true;

// biến

int age = 0;

String fullName = "";

long ballance;

// tạo đối tượng đọc dữ liệu vào từ bàn phím

Scanner input = new Scanner(System.in);

System.out.println("Nhập tuổi của bạn: ");

age = input.nextInt();

input.nextLine(); // đọc bỏ dòng thừa

System.out.println("Nhập tên của bạn: ");

fullName = input.nextLine();

System.out.println("Số dư là: ");

ballance = input.nextLong();

// hiển thị kết quả với print

System.out.print("Hello " + fullName + "\n");

System.out.print("Bạn " + age + " tuổi.");

System.out.print("Số dư của bạn là: " + ballance + "đ");

System.out.printf("Hello %s!\n", MONDAY);

// hiển thị kết quả với printf

System.out.printf("Tuổi của bạn là: %d\n", age);

System.out.printf("Số dư: %d\n", ballance);

}

}

Kết quả:

What's your full name?

Nguyen Van A

How old are you?

20

Where do you live?

Ha Noi City

Hello Nguyen Van A, nice to meet you!

Your address is Ha Noi City and you are 20 years old.