

# Giới thiệu về OOP

## 1. Object (Đối tượng).

---

Các bạn có thể hình dung về khái niệm đối tượng như sau: Trong lập trình hướng đối tượng, một thực thể được gọi là đối tượng nếu thực thể đó có 2 đặc điểm: trạng thái (*state*) và hành vi (*behavior*).

Ví dụ: là con người thì chúng ta sẽ có các trạng thái như: màu da, màu mắt, màu tóc, chiều cao, cân nặng,..., có các hành vi như: ăn, uống, làm việc,... ⇒ con người là 1 đối tượng.

Như vậy, thông qua ví dụ trên chúng ta có thể định nghĩa đối tượng như sau: **đối tượng là một khái niệm dùng để chỉ một thực thể cụ thể có trạng thái và hành vi. Một thực thể sẽ không được gọi là đối tượng nếu thực thể đó không thỏa mãn một trong các điều kiện trên.**

**Trạng thái:** Những đặc điểm của một đối tượng, chẳng hạn như đối tượng Sinh viên thì có mã, họ tên, ngày tháng năm sinh,...

**Hành vi:** Những hành động mà một đối tượng thực hiện, ví dụ: Sinh viên thì có các hành vi như đi học, đi dã ngoại vui chơi giải trí,...

## 2. Class (Lớp).

---

**Lớp là một tập hợp các đối tượng có cùng trạng thái và hành vi, vì vậy nó định nghĩa các tính chất của một tập hợp các đối tượng cùng kiểu.** Ví dụ: Lớp Student sẽ bao gồm một tập hợp các sinh viên của một lớp học, lớp Mammals sẽ bao gồm một tập hợp các động vật có vú trên thế giới,...

Mỗi đối tượng là một thể hiện (*instance*) của lớp. Thông thường, các đối tượng trong cùng một lớp sẽ có cùng một hành vi (*behavior*), nghĩa là có cách thức hoạt động tương tự nhau. Cách thức hoạt động của các đối tượng thuộc về một lớp được thiết lập nhờ vào những phương thức (*method*) của lớp đó.

**access\_modifier:** phạm vi truy cập của lớp, thuộc tính và phương thức.

**Tên lớp (class name):** mỗi lớp có một tên duy nhất để phân biệt với các lớp khác trong cùng một phạm vi.

**Các thuộc tính (attributes):** mô tả các trường để lưu dữ liệu cho mỗi đối tượng của lớp đang mô tả hay là lưu các tham chiếu đến các đối tượng của lớp khác. Sau này khi tạo lập một đối tượng của lớp thì mỗi thành phần dữ liệu trong đối tượng sẽ chứa hay liên kết với đối tượng dữ liệu cần thiết.

**Hệ thống các phương thức của lớp (methods):** mỗi phương thức của lớp thực chất là một hàm được viết riêng cho các đối tượng của lớp, chỉ được phép gọi để tác động lên chính các đối tượng của lớp này.

Dưới đây là ví dụ minh họa của 1 lớp.

```
1 public class Student {
2     // thuộc tính (instance variable)
3     private String id;
4     private String name;
5     private String information;
6
7     // Phương thức này có tên inputInformation và không có giá trị trả về
8     public void inputInformation() {
9
10    }
11
12    // Phương thức này có tên showInformation và có kiểu trả về là String
13    public String showInformation() {
14
15    }
16
17    // hàm main
18    public static void main(String[] args) {
19
20    }
21
22 }
```

---

Sau đây là một số vấn đề mà chúng ta cần nắm khi xem qua cấu trúc của lớp. Những vấn đề này chúng ta sẽ tìm hiểu rõ hơn trong các bài sau.

## Phạm vi truy cập (*Access Modifier*) của lớp, thuộc tính và phương thức

Trong Java, có 4 phạm vi truy cập sau: `public`, `private`, `protected` và `default` (*mặc định*). Bảng dưới đây sẽ giới thiệu tổng quan về phạm vi của 4 loại quyền truy cập này. Sang các bài sau, tôi sẽ nói rõ hơn về phạm vi truy cập trong Java.

`public`: có thể truy cập ở mọi nơi trong Project.

`protected`: truy cập được từ trong lớp khai báo, lớp con của lớp khai báo và các lớp cùng gói với lớp khai báo.

`default`: truy cập được từ trong lớp khai báo và các lớp cùng gói với lớp khai báo.

`private`: chỉ có thể truy cập bên trong lớp.

## Thuộc tính (*Instance variable*).

Thuộc tính là 1 biến được khai báo bên trong lớp nhưng ở bên ngoài một phương thức, hàm tạo hoặc 1 khối lệnh. Thuộc tính được khởi tạo khi một lớp được khởi tạo và có thể được sử dụng ở bên trong một hàm, hàm tạo hoặc trong một khối lệnh trong lớp đó.

## Phương thức

**Phương thức là một hàm được viết riêng cho các đối tượng của lớp, chỉ được phép gọi để tác động lên chính các đối tượng của lớp này.** Một phương thức của lớp bao gồm 3 thành phần chính đó là phạm vi truy cập, giá trị trả về và tên của phương thức đó.

Một phương thức là một tập hợp các câu lệnh được nhóm lại với nhau để thực hiện một chức năng nào đó. Sau đây tôi sẽ giới thiệu đến các bạn cách tạo và gọi phương thức trong chương trình. Trong bài ngày hôm nay, tôi chỉ minh họa cách khai báo và gọi phương thức thông thường (*chưa đề cập đến cách gọi phương thức thông qua đối tượng của lớp*).

### Tạo phương thức

Cú pháp tạo phương thức như sau:

```
1  [quyền_truy_cập] [kiểu_của_phương_thức] tênPhươngThức(Danh_sách_các_tham_số_truyền_vào) {
2      [Khai báo các biến bên trong phương thức];
3      [Các câu lệnh bên trong];
4      return [giá_trị_trả_về];
5  }
```

### Hàm tạo (*Constructor*).

Trong lập trình hướng đối tượng, chúng ta có 1 loại hàm đặc biệt đó là hàm tạo. **Hàm tạo trong Java là hàm có cùng tên với lớp và không có giá trị trả về.** Chúng ta hãy xem đoạn chương trình sau:

```
1  public class Student {
2      // thuộc tính (instance variable)
3      private String name;
4
5      // hàm tạo mặc định
6      public Student() {
7
8      }
9
10     // hàm tạo có đối số truyền vào có kiểu String và có tên là name
11     public Student(String name) {
12
13     }
14
15     // hàm main
16     public static void main(String[] args) {
17
18     }
19
20 }
```

---

Một lớp trong Java có thể có một hoặc nhiều hàm tạo. Có 2 loại hàm tạo chúng ta thường gặp đó là: **hàm tạo mặc định (*không có đối số*)** và **hàm tạo có đối số**.

- Hàm tạo mặc định là hàm tạo không có tham số đầu vào. Trong trường hợp người lập trình không định nghĩa bất kỳ hàm tạo nào thì trình biên dịch sẽ tự động tạo ra một phương thức tạo lập mặc định cho lớp này.
- Hàm tạo có đối số là các hàm tạo do người dùng định nghĩa với các tham số đầu vào khác nhau để khởi tạo dữ liệu cho đối tượng.

Ví dụ trong đoạn chương trình trên thì lớp **Student** có hàm **public Student()** và **public Student(String name)** là hàm tạo. Trong khi viết chương trình, nếu chúng ta không khởi tạo hàm tạo cho lớp thì khi đó trình biên dịch Java sẽ xây dựng 1 hàm tạo mặc định cho lớp đó.

Ý nghĩa của hàm tạo: tự động khởi tạo các thông tin mặc định khi đối tượng được tạo ra.

**Lưu ý:** Khi người lập trình khai báo bất kỳ một hàm tạo nào cho lớp thì hàm tạo mặc định sẽ không được phát sinh bởi trình biên dịch cho lớp này nữa. Trong trường hợp này, nếu muốn sử dụng hàm tạo mặc định, người lập trình phải tự viết thêm vào.