**Khái niệm OOP**

Lập trình hướng đối tượng (OOP) là một kỹ thuật lập trình cho phép lập trình viên tạo ra các đối tượng trong code trừu tượng hóa các đối tượng.

**Đối tượng**

Một đối tượng bao gồm 2 thông tin: thuộc tính và phương thức.

* Thuộc tính chính là những thông tin, đặc điểm của đối tượng. Ví dụ: con người có các đặc tính như mắt, mũi, tay, chân…
* Phương thức là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: một người sẽ có thể thực hiện hành động nói, đi, ăn, uống, . . .

**Lớp**

Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.

**Tính kế thừa (Inheritance)**

Nó cho phép xây dựng một lớp mới dựa trên các định nghĩa của lớp đã có. Có nghĩa là lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con. Các lớp con khỏi phải định nghĩa lại, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới. Tái sử dụng mã nguồn 1 cách tối ưu, tận dụng được mã nguồn. Một số loại kế loại kế thừa thường gặp: đơn kế thừa, đa kế thừa, kế thừa đa cấp, kế thừa thứ bậc. Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu thiết kế định nghĩa các lớp trước. Thông thường một số lớp có quan hệ với những lớp khác, chúng có những đặc tính giống nhau. VD: 2 lớp Android, iPhone Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau như gọi điện, nhắn tin, chụp hình. Thay vì sao chép những thuộc tính này, ta nên đặt chúng vào một lớp chung gọi là lớp cha. Chúng ta có thể định nghĩa lớp cha – trong trường hợp này là Smartphone và có những lớp con kế thừa từ nó, tạo ra một mối quan hệ cha/con. Tính đa hình

**Tính đa hình (Polymorphism)**

Tính đa hình là một hành động có thể được thực hiện bằng nhiều cách khác nhau. Đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng. Hiểu một cách đơn giản hơn: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau. Ví dụ như ở phần trên, mỗi một smartphone kế thừa từ lớp cha Smartphone nhưng có thể lưu trữ dữ liệu trên cloud theo những cách khác nhau. Android lưu trữ bằng Google Drive iPhone lưu trên iCloud. Một ví dụ về đa hình trong thực tế. Ta có 2 con vật: chó, mèo. Cả 2 con vật này đều là lớp động vật. Nhưng khi ta bảo cả 2 động vật kêu thì con chó sẽ kêu gâu gâu, con mèo sẽ kêu meo meo. Vậy trong ví dụ chó, mèo xem như là các đối tượng. 2 con vật có thể hiểu cùng kêu nhưng theo các cách khác nhau.

**Tính trừu tượng (Abstraction**)

Trừu tượng có nghĩ là tổng quát hóa một cái gì đó lên, không cần chú ý chi tiết bên trong. Nó không màng đến chi tiết bên trong là gì và người ta vẫn hiểu nó mỗi khi nghe về nó. Ví dụ: Bạn chạy xe tay ga thì có hành động là tăng ga để tăng tốc, thì chức năng tăng ga là đại diện cho trừu tượng (abstraction). Người dùng chỉ cần biết là tăng ga thì xe tăng tốc, không cần biết bên trong nó làm thế nào. Ở đây trong lập trình OOP, tính trừu tượng nghĩa là chọn ra các thuộc tính, phương thức của đối tượng cần cho việc giải quyết bài toán đang lập trình. Vì một đối tượng có rất nhiều thuộc tính phương thức, nhưng với bài toán cụ thể không nhất thiết phải chọn tất cả. Ví dụ: Bài toán quản lý sinh viên chúng ta chỉ cần quản lý các thông tin như Họ tên Ngày sinh Giới tính … Điểm thi mà lại không cần quản lý thêm các thông tin: Màu tóc Sở thích Chiều cao Tại vì chúng thực sự không cần thiết.

**Tính đóng gói (Encapsulation)**

Đơn giản là việc kết hợp một bộ các dữ liệu (data) liên quan đến nhau cùng với một bộ các hàm/phương thức (functions/methods) hoạt động trên các dữ liệu đó, “gói” tất cả vào trong một cái gọi là class. Các thực thể của các class thì được gọi là các đối tượng (objects) trong khi class giống như một công thức được sử dụng để tạo ra các đối tượng đó.