**Tổng ôn lý thuyết**

**Đối tượng - object:** 1 đối tượng là 1 thực thể bao gồm thuộc tính và hành động

**Lớp - class:**

- Các đối tượng có các đặc tính tương tự nhau được gom chung thành một lớp đối tượng

- Một lớp đối tượng đặc trưng bằng các thuộc tính và các hành động của đối tượng thuộc lớp

- Một đối tượng cụ thể thuộc một lớp được gọi là một thể hiện ( intstance ) của lớp đó.

- Thuộc tính ( attribute ): một thành phần của đối tượng, có giá trị nhất định cho mỗi đối tượng tại mỗi thời điểm trong hệ thống

- Thao tác ( operation ): thể hiện hành vi của một đối tượng tác động qua lại với các đối tượng khác hoặc với chính nó

**Trừu tượng hóa - abstraction:** là các nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm ( bỏ qua những chi tiết không cần thiết ), tách biệt giao diện và phần thực thi

**Đóng gói - Encapsulation:** Đóng gói chính là quá trình ẩn đi phần Thực thi khỏi người dùng bên ngoài và giới hạn quyền truy cập vào nó. Người dùng của một Lớp chỉ có thể sử dụng Giao diện mà không có quyền truy cập vào phần Thực thi

**Kế thừa - Inheritance:** dùng để biểu diễn mối quan hệ đặc biệt hóa – tổng quát hóa giữa các lớp. Nó cho phép một lớp có thể được thừa hưởng các thuộc tính, phương thức từ một lớp khác.

Lớp kế thừa từ một lớp khác thì được gọi là lớp con (child class, subclass) hay lớp dẫn

xuất (derived class). Lớp được các lớp khác kế thừa được gọi là lớp cha (parent class,

superclass) hay lớp cơ sở (base class).

**Đa hình - Polymorphism:** là cơ chế cho phép tên một thao tác hoặc nhiều thuộc tính có thể được đinh nghĩa tại nhiều lớp và có thể có những cài đặt khác nhau tại các lớp đó

**Các mục tiêu:**

- Tính tái sử dụng ( reusability )

- Tính mở rộng ( extensibility )

- Tính mềm dẻo ( flexibility )

**Lớp đối tượng:** là một mô tả trừu tượng của nhóm các đối tượng cùng bản chất, ngược lại mỗi một đối tượng là một thể hiện cụ thể cho những mô tả trừu tượng đó

**Phạm vi truy xuất:**

- Mọi thành phần được liệt kê trong phần public đều có thể truy xuất trong bất kỳ hàm nào

- Những thành phần được liệt kê trong phần private chỉ được truy xuất bên trong lớp

**Con trỏ this:** tham chiếu đến đối tượng đang gọi thực hiện phương thức ( có thể là tham số ngầm định )

**Phép gán đối tượng:** là việc sao chép giá trị các thành phần dữ liệu từ đối tượng a sang đối tượng b tương ứng từng đôi một

**Phương thức thiết lập - Constructor:** phương thức đặc biệt dùng để khởi tạo thể hiện của lớp

Các đặc điểm:

• Phương thức thiết lập là một hàm thành viên đặc biệt có tên trùng với tên lớp.

• Không có giá trị trả về.

• Được tự động gọi thực hiện ngay khi đối tượng được khai báo.

Phân loại phương thức thiết lập:

• Phương thức thiết lập mặc định (default constructor).

• Phương thức thiết lập nhận tham số đầu vào (parameterized constructor).

• Phương thức thiết lập sao chép (copy constructor): hằng tham chiếu đến đối tượng thuộc lớp.

**Phương thức hủy bỏ - Destructor:** thực hiện dọn dẹp cần thiết trước khi đối tượng bị hủy bỏ khi hết phạm vi hoạt động.

Đặc điểm:

• Tên phương thức trùng với tên lớp nhưng có dấu ngã ở đằng trước.

• Không có giá trị trả về.

• Không có tham số đầu vào.

• Được tự động gọi thực hiện trước khi đối tượng bị hủy.

• Có và chỉ có duy nhất một phương thức phá huỷ trong 1 lớp.

• Trong một quá trình sống của đối tượng có và chỉ có một lần phương thức phá hủy

được gọi thực hiện mà thôi.

Ý nghĩa:

• Một cách dùng của phương thức phá hủy là để giải phóng bộ nhớ của các thuộc tính

được cấp phát động trong một đối tượng. Nếu chúng ta không giải phóng các vùng

nhớ này, nó sẽ bị tồn đọng lại và chiếm không gian không cần thiết.

Phương thức truy vấn: Getter - trả

Phương thức cập nhật: Setter - không trả

**Thành viên tĩnh - static member:**

- Đối với class, static dùng để khai báo thành viên dữ liệu dùng chung cho mọi thể hiện của lớp:

* 1 bản duy nhất tồn tại trong suốt quá trình chạy của chương trình
* dùng chung cho tất cả các thể hiện của lớp
* bất kể lớp đó có bao nhiêu thể hiện

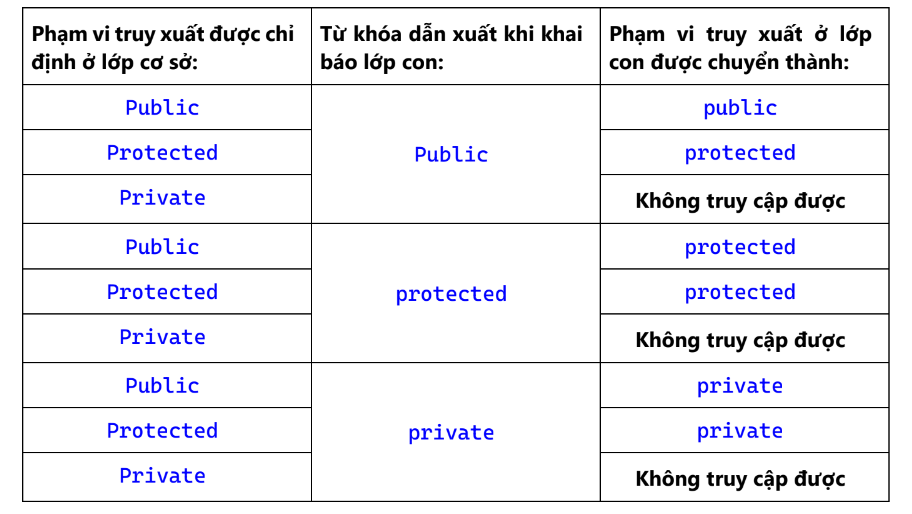
ex:

static void …();

static int count;

**Đối tượng là thành phần của lớp:** đối tượng có thể là thành phần của đối tượng khác, khi một đối tượng thuộc lớp “lớn” được tạo ra, các thành phần của nó cũng được tạo ra.

**Hàm bạn**: không thuộc lớp, tuy nhiên, có quyền truy cập các thành viên private - quan hệ 1 chiều



**Sự kế thừa:** là một mức cao hơn của trừu tượng hóa, cung cấp một cơ chế gom chung các lớp có liên quan với nhau thành một mức khái quát hóa đặc trưng cho toàn bộ các lớp trên

**Con trỏ và kế thừa:**

- con trỏ đến đối tượng thuộc lớp base thì có thể trỏ đến các đối tượng thuộc lớp con

- con trỏ thuộc lớp con thì không thể trỏ đến đối tượng thuộc lớp cha

- có thể ép kiểu để con trỏ thuộc lớp con đến lớp base tuy nhiên có thể rủi ro

**Đa kế thừa:** cho phép một lớp có thể là dẫn xuất của nhiều lớp cơ sở.

**Đa hình:** là hiện tượng các đối tượng thuộc các lớp khác nhau có khả năng hiểu cùng một thông điệp theo các cách khác nhau.

**Phương thức ảo:** là cách thể hiện tính đa hình, các method ở baseclass có tính đa hình được định nghĩa bằng virtual methods và các lớp dẫn xuất có thể override (ghi đè).

**Lớp cơ sở trừu tượng:** là lớp cơ sở không có đối tượng nào thuộc chính nó