Lý thuyết OOP đã tổng hợp

Nhập môn lập trình (Trường Đại học Công nghệ thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh)

ÔN TẬP
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG LÝ THUYẾT
HỌC THUỘC

1. Lập trình hướng đối thượng là gì?

Lập trình hướng đối tượng là phương pháp lập trình lấy đối tượng làm nền tảng để xây dựng thuật giải xây dựng chương trình.

2. Lớp là gì? Đối tượng là gì? Phân biệt lớp và đối tượng.

- Một đối tượng là một thực thể bao gồm thuộc tính và hành động.
- Các đối tượng có các đặc tính tương tự nhau được gom chung thành lớp đối tượng. Một lớp đối tượng đặc trưng bằng các thuộc tính và các thao tác.

- Thuộc tính: một thành phần của đối tượng, có giá trị nhất định cho một đối tượng tại mỗi thời điểm trong hệ thống.
- Thao tác: thể hiện hành vi của một đối tượng tác động qua lại với các đối tượng khác hoặc với chính nó.

Lớp	Đối Tượng
Là một template chung cho tất cả các đối tượng, là một mô tả trừu tượng	Là một thể hiện của lớp
Chỉ được khai báo một lần	Có thể có nhiều đối tượng thuộc lớp
Khi khai báo thì không được cấp phát vùng nhớ	Khi khai báo thì được cấp phát vùng nhớ
Là một nhóm đối tượng giống nhau	Là những đối tượng cụ thể có thật

3. Trình bày các đặc điểm quan trọng của LTHĐT.

- Trừu tượng hóa – Abstraction: cách nhìn khái quát hóa về một tập các đối tượng có chung các đặc điểm được quan tâm (và bỏ qua những chi tiết không cần thiết).
- Đóng gói – Encapsulation: nhóm những gì có liên quan với nhau vào làm một, để sau này có thể dùng một cái tên để gọi đến; đồng thời che một số thông tin và chi tiết cài dặt nội bộ để bên ngoài không nhìn thấy. Vd: các hàm/ thủ tục đóng gói các câu lệnh, các đối tượng đóng gói dữ liệu của chúng và các thủ tục có liên quan.
- Thừa kế - Inheritance: cho phép một lớp D có được các thuộc tính và thao tác của lớp C, như thể các thuộc tính và thao tác đó đã được định nghĩa tại lớp D.
  Cho phép cài đặt nhiều quan hệ giữa các đối tượng: đặc biệt hóa – tổng quát hóa.

- Đa hình – Polymorphism: là cơ chế cho phép một tên thao tác hoặc thuộc tính có thể được định nghĩa tại nhiều lớp và có thể có nhiều cài đặt khác nhau tại mỗi lớp trong các lớp đó.

4. Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp - Set: Vì dữ liệu được khai báo trong nhãn private chỉ được truy xuất trong phạm vi lớp đó, nếu muốn thay đổi dữ liệu đó ở ngoài phạm vi lớp thì phải dùng hàm set, ngoài ra còn để kiểm soát thông tin nhập vào biến đó có phù hợp với đề bài hay không.

- Get: Tương tự như set, muốn lấy dữ liệu được khai báo trong nhãn private thì chỉ được truy xuất trong phạm vi lớp, nên nếu muốn truy xuất dữ liệu đó ở ngoài phạm vi lớp phải dùng hàm get để lấy dữ liệu ra.

5. Phân biệt các phạm vi truy cập private, protected và public.

- Private: mọi thành phần được khai báo trong private chỉ được truy xuất bên trong phạm vi lớp và hàm bạn, lớp bạn.
- Public: mọi thành phần được khai báo trong public đều được truy xuất ở bất kỳ hàm nào.
- Protected: mọi thành phần được khai báo trong protected chỉ được truy xuất bên trong phạm vi lớp và có thể truy cập từ lớp dẫn xuất, hàm bạn, lớp bạn.

6. Nêu khái niệm Constructor, Destructor. Phân biệt Constructor mặc định và Contructor khác. Phân biệt Constructor và Destructor.

- Constructor là hàm dùng để khởi tạo giá trị cho đối tượng. Được khai báo như một phương thức, tên phương thức trùng với tên lớp nhưng không có kiểu dữ liệu trả về. Một class có thể có nhiều constructor. Constructor phải có thuộc tính public.
- Destructor là hàm dùng để hủy đối tượng khi hết phạm vi sử dụng. Được khai báo như một phương thức, tên phương thức trùng với tên lớp và có dấu ~ ở trước. Không có kiểu dữ liệu trả về. Destructor phải được khai báo ở nhãn public và chỉ có duy nhất 1 destructor trong class.
- Phân biệt constructor mặc định và các constructor khác

<table>
<tr><th>Constructor mặc định</th><th>Constructor sao chép hoặc contructor có tham số đầu vào</th></tr>
<tr><td colspan="2">Là hàm dùng để khởi tạo giá trị cho đối tượng
Được khai báo như một phương thức có tên trùng tên lớp
Không có kiểu dữ liệu trả về</td></tr>
<tr><td colspan="2">Khai báo ở nhãn public</td></tr>
<tr><td>Không có tham số truyền vào hoặc tất cả tham số điều có giá trị mặc nhiên.</td><td>Có một hoặc nhiều tham số đầu vào.</td></tr>
</table>

Khi khởi tạo đối tượng thì không cần truyền tham số vào	Khi khai báo đối tượng thì phải truyền tham số nếu không trình biên dịch sẽ dùng constructor mặc định.
Khởi tạo các đối tượng có cùng dữ liệu được khai báo trong constructor mặc định	Khởi tạo những đối tượng khác dữ liệu tùy thuộc vào tham số mà ta truyền vào
Chỉ có 1 constructor mặc định	Có thể có nhiều constructor có tham số

* Nếu không khai báo constructor thì trình biên dịch sẽ tự động sinh constructor mặc định, nhưng một khi đã khai báo các constructor khác rồi mà không khai báo constructor mặc định thì chương trình sẽ báo lỗi.

- Phân biệt construcor và destructor

Constructor	Destructor
Không có kiểu dữ liệu trả về Khai báo ở nhãn public Được gọi tự động	
Tên trùng tên lớp	Tên trùng tên lớp nhưng thêm ~ ở trước
Khởi tạo giá trị cho đối tượng	Hủy đối tượng
Có nhiều constructor	Chỉ có 1 destructor

7. Nêu khái niệm về sự kế thừa và những ưu điểm của kế thừa trong lập trình.

- 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 nói trên.

- ***Ưu điểm của kế thừa***

- Cho phép xây dựng 1 lớp mới từ lớp đã có.

  Lớp mới gọi là lớp con (subclass) hay lớp dẫn xuất (derived class).

  Lớp đã có gọi là lớp cha (superclass) hay lớp cơ sở (base class).

- Cho phép chia sẽ các thông tin chung nhằm tái sử dụng và đồng thời giúp ta dễ dàng nâng cấp, dễ dàng bảo trì.
- Định nghĩa sự tương thích giữa các lớp, nhờ đó ta có thể chuyển kiểu tự động

Ví dụ: Giả sử ta có lớp TamGiac chứa thông tin toạ độ của 3 điểm A, B, C. Ta biết rằng tam giác cân là 1 trường hợp đặc biệt của tam giác (ngược lại tam giác là trường hợp tổng quát của tam

giác cân). Từ đó ta có thể cho lớp TamGiacCan kế thừa lại lớp TamGiac để có thể sử dụng lại các thông tin như toạ độ 3 điểm A, B, C mà không cần phải khai báo.

8. Phân biệt các kiểu kế thừa private, protected và public.

Phạm vi truy xuất \ Từ khóa dẫn xuất	Private	Protected	public
Private	X	X	X
Protected	Private	Protected	Protected
Public	Private	Protected	Public

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con.
- Nếu từ khóa dẫn xuất là public thì thành phần protected lớp cha trở thành protected của lớp con, thành phần public của lớp cha trở thành public của lớp con.
- Nếu từ khóa dẫn xuất là private thì thành phần protected và public của lớp cha trở thành private của lớp con.
- Nếu từ khóa dẫn xuất là protected thì thành phần protected và public của lớp cha trở thành protected của lớp con.

9. Lớp cơ sở trừu tượng là gì? Được cài đặt trong C++ như thế nào?

Phương thức thuần ảo có ý nghĩa trong việc tổ chức phân cấp các lớp: phương thức thuần ảo trong lớp cơ sở sẽ là phương thức chung cho các lớp dẫn xuất thừa kế và cài đặt. Phương thức ảo thuần túy là phương thức không có nội dung, khai báo với từ khóa virtual ở đầu và được gán bằng 0.

Lớp cơ sở trừu tượng là lớp chứa ít nhất một phương thức ảo thuần túy. Lớp cơ sở trừu tượng không có đối tượng nào có kiểu dữ liệu là chính nó. Cài đặt

```
class baseClass { public:
      baseClass();
      ~baseClass();
 virtual void ExampleMethod() = 0;
}
```

10. Phân biệt khái niệm overload và override

Overload đơn giản là có vài phương thức trùng tên nhưng khác nhau về đối số. Cài chồng phương thức cho phép ta tạo nhiều phiên bản của một phương thức, mỗi phiên bản chấp nhận một danh sách đối số khác nhau, nhằm tạo thuận lợi cho việc gọi phương thức.
Override là một tính năng cho phép một lớp con hoặc lớp con cung cấp một triển khai cụ thể của một phương thức đã được cung cấp bởi một trong các lớp siêu hoặc các lớp cha của nó. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.

	Override	Overload
Khái niệm	Là một tính năng cho phép một lớp con hoặc lớp dẫn xuất cung cấp một triển khai cụ thể của một phương thức đã được cung cấp bởi một trong các lớp siêu hoặc các lớp cha của nó. Nói cách khác, nếu lớp con cung cấp trình triển khai cụ thể của phương thức mà đã được cung cấp bởi một trong các lớp cha của nó, thì đó là ghi đè phương thức.	Nạp chồng phương thức đơn giản là có vài phương thức trùng tên nhưng khác nhau về đối số. Cài chồng phương thức cho phép ta tạo nhiều phiên bản của một phương thức, mỗi phiên bản chấp nhận một danh sách đối số khác nhau, nhằm tạo thuận lợi cho việc gọi phương thức.
Hành vi	Thêm hoặc mở rộng cho hành vi của phương thức.	Thay đổi hành vi hiện tại của phương thức.
Đa hình	Thể hiện tính đa hình tại run time.	Thể hiện tính đa hình tại compile
Danh sách tham số	Danh sách tham số phải giống nhau.	Danh sách tham số khác nhau (số lượng, thứ tự, kiểu dữ liệu)
Giá trị trả về	Kiểu trả về bắt buộc phải giống nhau.	Kiểu trả về có thể khác nhau.
Phạm vi	Xảy ra giữa 2 class có quan hệ kế thừa	Xảy ra trong phạm vi cùng 1 class.

Overloading	Overriding
-Ngôn ngữ C++ phân biệt tên hàm thông qua kiểu trả về, số lượng và kiểu đối số -Cho phép nạp chồng toàn tử (có thể tự định nghĩa lại toán tử do người cài đặt tạo ra nhưng không làm thay đổi bản chất của toán tử) -Các hàm cùng tồn tại song song	-Thêm hành vi cho phương thức -Chỉ sử dụng trong các class có kế thừa, đa hình -Giữ nguyên kiểu hàm, kiểu trả về, kiểu dữ liệu và số lượng đối số -Áp dụng thuật toán khác nhau -Phải tạo ra ở lớp dẫn xuất (nếu không tạo

-Được cài đặt chung trong một lớp -Thay đổi hành vi hiện tại của phương thức	ra ở lớp dẫn xuất thì nó vẫn kế thừa virtual ở lớp cơ sở) -Có từ khóa virtual -Xảy ra ở các lớp khác nhau
Cho phép với tất cả các kỹ thuật lập trình **Cho phép các hàm trùng tên**	**Chỉ áp dụng với OOP**

11. Trình bày khái niệm đa hình trong LTHĐT.

Tính đa hình là hiện tượng các đối tượng thuộc các lớp khác nhau có thể hiểu cùng một thông điệp theo các cách khác nhau.

Ví dụ: có 3 con vật: chó, mèo, lợn. Khi ta bảo "kêu" thì con chó sẽ kêu gâu gâu, con mèo sẽ kêu meo meo và con heo sẽ kêu ẹt ẹt. Cả 3 con vật có thể hiểu cùng một thông điệp là "kêu" nhưng thực hiện theo các cách khác nhau.