

**ỦY BAN NHÂN DÂN TỈNH ĐỒNG THÁP  
TRƯỜNG CAO ĐẲNG CỘNG ĐỒNG ĐỒNG THÁP**



**GIÁO TRÌNH  
MÔN HỌC: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG  
NGÀNH, NGHỀ: CÔNG NGHỆ THÔNG TIN  
TRÌNH ĐỘ: CAO ĐẲNG**

(Ban hành kèm theo Quyết định số     /QĐ-CĐCĐ ngày     tháng     năm 20...  
của Hiệu trưởng trường Cao đẳng Cộng đồng Đồng Tháp)

**Đồng Tháp, năm 2017**

## **TUYÊN BỐ BẢN QUYỀN**

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

TaiLieu.vn

## **LỜI NÓI ĐẦU**

Bài giảng sẽ trình bày một cách hệ thống các khái niệm của lập trình hướng đối tượng được cài đặt trong C++ như: lớp, đối tượng, sự kế thừa, tính đa hình. Bên cạnh đó, bài giảng cũng giới thiệu chi tiết cách thiết kế một chương trình hướng đối tượng đến người học.

Bài giảng bao gồm các chương sau:

- Chương 1: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- Chương 2: CÁC ĐẶC ĐIỂM CỦA C++
- Chương 3: LỚP VÀ ĐỐI TƯỢNG
- Chương 4: THỪA KẾ
- Chương 5: TÍNH ĐA HÌNH
- Chương 6: THIẾT KẾ CHƯƠNG TRÌNH HƯỚNG ĐỐI TƯỢNG

Cuối cùng, xin gửi lời cảm ơn chân thành nhất đến bạn bè, đồng nghiệp đã có những ý kiến đóng góp sâu sắc cho bài giảng này. Mọi ý kiến đóng góp xin vui lòng gửi đến địa chỉ mail: [pqcuong@dtcc.edu.vn](mailto:pqcuong@dtcc.edu.vn) . Xin chân thành cảm ơn!

Đồng Tháp, ngày 01 tháng 07 năm 2017

Tác giả

Phan Quốc Cường

# MỤC LỤC

|                                                          |    |
|----------------------------------------------------------|----|
| LỜI NÓI ĐẦU .....                                        | i  |
| MỤC LỤC.....                                             | ii |
| DANH MỤC CÁC HÌNH.....                                   | v  |
| Chương 1 .....                                           | 1  |
| LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG .....                          | 1  |
| 1.1    Thế nào là lập trình hướng đối tượng .....        | 1  |
| 1.2    Một số khái niệm quan trọng.....                  | 3  |
| CÂU HỎI ÔN TẬP CHƯƠNG 1.....                             | 8  |
| Chương 2 .....                                           | 9  |
| CÁC ĐẶC ĐIỂM CỦA C++ .....                               | 9  |
| 2.1    Cấu trúc chương trình .....                       | 9  |
| 2.2    Cấu trúc điều khiển.....                          | 12 |
| 2.2.1 Cấu trúc điều kiện: <i>if</i> và <i>else</i> ..... | 13 |
| 2.2.2 Các cấu trúc lặp.....                              | 14 |
| 2.2.3 Các lệnh rẽ nhánh và lệnh nhảy .....               | 19 |
| 2.3    Con trỏ và chuỗi ký tự .....                      | 24 |
| 2.3.1 Con trỏ.....                                       | 24 |
| 2.3.2 Chuỗi ký tự.....                                   | 38 |
| 2.4    Tham số mặc định của hàm .....                    | 47 |
| 2.5    Hàm inline .....                                  | 50 |
| 2.6    Truyền tham số.....                               | 53 |
| 2.7    Tham chiếu.....                                   | 60 |
| 2.8    Struct .....                                      | 64 |
| BÀI TẬP THỰC HÀNH CHƯƠNG 2 .....                         | 73 |
| Chương 3 .....                                           | 76 |
| LỚP VÀ ĐỐI TƯỢNG .....                                   | 76 |
| 3.1    Lớp (Class) .....                                 | 76 |

|                                  |                                                                 |     |
|----------------------------------|-----------------------------------------------------------------|-----|
| 3.2                              | Đối tượng.....                                                  | 79  |
| 3.3                              | Phạm vi lớp và truy cập các thành viên lớp.....                 | 82  |
| 3.4                              | Hàm tạo và hàm hủy.....                                         | 86  |
| 3.4.1                            | Hàm tạo.....                                                    | 87  |
| 3.4.2                            | Dùng hàm tạo trong khai báo.....                                | 88  |
| 3.4.3                            | Dùng hàm tạo trong cấp phát bộ nhớ.....                         | 89  |
| 3.4.4                            | Dùng hàm tạo để biểu diễn các đối tượng hằng.....               | 90  |
| 3.4.5                            | Hàm tạo mặc định.....                                           | 92  |
| 3.4.6                            | Hàm hủy.....                                                    | 96  |
| 3.4.7                            | Hàm tạo và hàm hủy được gọi khi nào? .....                      | 98  |
| 3.5                              | Đối tượng hằng và hàm thành viên hằng.....                      | 101 |
| 3.6                              | Hàm bạn và lớp bạn.....                                         | 109 |
| 3.6.1                            | Hàm bạn.....                                                    | 109 |
| 3.6.2                            | Lớp bạn.....                                                    | 121 |
| 3.7                              | Hàm thành viên tĩnh.....                                        | 128 |
| 3.8                              | Con trỏ this.....                                               | 134 |
| BÀI TẬP THỰC HÀNH CHƯƠNG 3 ..... |                                                                 | 142 |
| Chương 4.....                    |                                                                 | 145 |
| THỪA KẾ.....                     |                                                                 | 145 |
| 4.1                              | Kế thừa đơn.....                                                | 145 |
| 4.1.1                            | Ép kiểu các con trỏ lớp cơ sở tới các con trỏ lớp dẫn xuất..... | 148 |
| 4.1.2                            | Các lớp cơ sở public, protected và private.....                 | 155 |
| 4.1.3                            | Các constructor và destructor lớp dẫn xuất.....                 | 156 |
| 4.2                              | Đa kế thừa.....                                                 | 161 |
| 4.3                              | Các lớp cơ sở ảo (virtual base classes).....                    | 167 |
| BÀI TẬP THỰC HÀNH CHƯƠNG 4 ..... |                                                                 | 172 |
| Chương 5.....                    |                                                                 | 178 |
| TÍNH ĐA HÌNH.....                |                                                                 | 178 |

|                                                       |     |
|-------------------------------------------------------|-----|
| 5.1 Giới thiệu.....                                   | 178 |
| 5.2 Phương thức ảo (virtual function).....            | 179 |
| 5.3 Các đặc trưng của phương thức ảo.....             | 182 |
| 5.4 Lớp trừu tượng (Abstract class) .....             | 184 |
| 5.5 Các thành viên ảo của một lớp .....               | 188 |
| 5.5.1 Toán tử ảo .....                                | 188 |
| 5.5.2 Có constructor và destructor ảo hay không?..... | 194 |
| BÀI TẬP THỰC HÀNH CHƯƠNG 5 .....                      | 197 |
| Chương 6.....                                         | 200 |
| THIẾT KẾ CHƯƠNG TRÌNH THEO HƯỚNG ĐỐI TƯỢNG .....      | 200 |
| 6.1 Các giai đoạn phát triển hệ thống.....            | 200 |
| 6.2 Cách tìm lớp .....                                | 202 |
| 6.3 Các bước cần thiết để thiết kế chương trình ..... | 202 |
| 6.4 Một số ví dụ minh họa .....                       | 204 |
| BÀI TẬP THỰC HÀNH CHƯƠNG 6 .....                      | 221 |
| TÀI LIỆU THAM KHẢO.....                               | 222 |

## DANH MỤC CÁC HÌNH

|                                                                                                 |     |
|-------------------------------------------------------------------------------------------------|-----|
| Hình 1 - Lớp ẩn phẩm và các lớp dẫn xuất của nó.....                                            | 5   |
| Hình 2 – Minh họa tính đa hình đối với ẩn phẩm và các lớp dẫn xuất của nó .....                 | 7   |
| Hình 3 – Kết quả tính thể tích hình hộp .....                                                   | 49  |
| Hình 4 – Kết quả tính thể tích hình lập phương.....                                             | 52  |
| Hình 5 – Một đối tượng phần mềm.....                                                            | 81  |
| Hình 6 – Đối tượng phần mềm xe đạp .....                                                        | 82  |
| Hình 7 – Kết quả minh họa phạm vi lớp .....                                                     | 84  |
| Hình 8 – TBD báo lỗi.....                                                                       | 86  |
| Hình 9 – Kết quả chương trình có sử dụng hàm hủy .....                                          | 98  |
| Hình 10 – Kết quả chương trình minh họa thứ tự các constructor và destructor được gọi .....     | 101 |
| Hình 11 – Lỗi biên dịch .....                                                                   | 106 |
| Hình 12 – Kết quả khởi tạo một hàng của kiểu dữ liệu có sẵn .....                               | 109 |
| Hình 13 – Trình biên dịch thông báo lỗi.....                                                    | 109 |
| Hình 14 – Kết quả minh họa sử dụng tường minh của con trỏ this .....                            | 136 |
| Hình 15 – Kết quả trả về một tham chiếu tới một đối tượng Time.....                             | 141 |
| Hình 16 – Minh họa phân cấp kế thừa đơn.....                                                    | 146 |
| Hình 17 – Phân cấp lớp Shape .....                                                              | 146 |
| Hình 18 – Kết quả minh họa .....                                                                | 153 |
| Hình 19 - Minh họa thứ tự các constructor và destructor lớp cơ sở và lớp dẫn xuất được gọi..... | 160 |
| Hình 20 – Kết quả minh họa .....                                                                | 166 |
| Hình 21 – Kết quả sau khi lớp cơ sở ảo .....                                                    | 171 |
| Hình 22 – Kết quả chương trình.....                                                             | 180 |
| Hình 23 – Kết quả minh họa các tính chất phương thức ảo.....                                    | 184 |
| Hình 24 – Mô tả tính đa hình của hàm Print() .....                                              | 185 |
| Hình 25 – Kết quả ví dụ minh họa tính đa hình hàm Print() .....                                 | 188 |

|                                                                                                              |     |
|--------------------------------------------------------------------------------------------------------------|-----|
| Hình 26 – Kết quả ví dụ minh họa đa năng hóa toán tử với hàm toán tử là phương thức ảo .....                 | 194 |
| Hình 27 – Kết quả minh họa destructor ảo .....                                                               | 196 |
| Hình 28 - Tìm ra những gì tồn tại chung giữa các đối tượng .....                                             | 203 |
| Hình 29 - Tiếp tục theo hướng từ dưới lên .....                                                              | 204 |
| Hình 30 – Các lớp Nguoi, BienChe, HopDong .....                                                              | 205 |
| Hình 31 – Kết quả chương trình tính tiền lương của các nhân viên trong cơ quan theo các dạng khác nhau ..... | 212 |
| Hình 32 – Các lớp Nguoi, SinhVien, GiangVien.....                                                            | 213 |
| Hình 33 – Kết quả ví dụ 2 .....                                                                              | 220 |

# **Chương 1**

## **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

### **MỤC ĐÍCH**

Phân biệt sự khác nhau giữa lập trình cổ điển và lập trình hướng đối tượng; giới thiệu các khái niệm quan trọng trong lập trình hướng đối tượng như: đối tượng, thể hiện, lớp; và giới thiệu các đặc điểm chính của lập trình hướng đối tượng như: tính đóng gói, tính kế thừa, tính đa hình.

### **Nhiệm vụ tự học và tài liệu tham khảo của sinh viên trong chương 1**

Để đạt được mục đích đề ra trong chương 1, sinh viên phải thực hiện tốt các yêu cầu sau đây:

- Xem trước đề cương chi tiết của Chương 1 để biết được mục đích, nội dung sẽ được trình bày trong chương này
- Phải đọc trước nội dung Chương 1 trong quyển bài giảng này trước khi lên lớp

### **1.1 Thế nào là lập trình hướng đối tượng**

#### **Phương pháp lập trình cấu trúc**

- Tư tưởng chính của lập trình cấu trúc là tổ chức chương trình thành các chương trình con. Trong PASCAL có 2 kiểu chương trình con là thủ tục và hàm. Trong C chỉ có một loại chương trình con là hàm. Hàm là một đơn vị chương trình độc lập dùng để thực hiện một phần việc nào đó như: Nhập số liệu, in kết quả hay thực hiện một số tính toán. Hàm cần có đối và các biến, mảng cục bộ dùng riêng cho hàm.

- Các ngôn ngữ như C, PASCAL, FOXPRO là các ngôn ngữ cho phép triển khai phương pháp lập trình cấu trúc. Một chương trình cấu trúc gồm các cấu trúc dữ liệu (như biến, mảng, bản ghi) và các hàm, thủ tục. Nhiệm vụ chính của việc tổ chức thiết kế chương trình cấu trúc là tổ chức chương trình thành các hàm, thủ tục: Chương trình sẽ bao gồm các hàm, thủ tục nào.

## Phương pháp lập trình hướng đối tượng

Khái niệm trung tâm của lập trình hướng đối tượng là lớp (class). Có thể xem lớp là sự kết hợp các thành phần dữ liệu và các hàm. Cũng có thể xem lớp là sự mở rộng của cấu trúc trong C (struct) bằng cách đưa thêm vào các phương thức (method) hay cũng gọi là hàm thành viên (member function). Một lớp được định nghĩa như sau:

```
class Tên_Lớp
{
// Khai báo các thành phần dữ liệu
// Khai báo các phương thức
};
```

Các phương thức có thể được viết (xây dựng) bên trong hoặc bên ngoài (phía dưới) phần định nghĩa lớp. Cấu trúc (cách viết) phương thức tương tự như hàm ngoại trừ quy tắc sau: Khi xây dựng một phương thức bên ngoài định nghĩa lớp thì trong dòng đầu tiên cần dùng tên lớp và dấu :: đặt trước tên phương thức để chỉ phương thức thuộc lớp nào.

Sử dụng các thành phần dữ liệu trong phương thức: Vì phương thức và các thành phần dữ liệu thuộc cùng một lớp và vì phương thức được lập lên cốt để xử lý các thành phần dữ liệu, nên trong thân của phương thức có quyền truy nhập đến các thành phần dữ liệu (của cùng lớp).

*Biến lớp:* Sau khi định nghĩa một lớp, có thể dùng tên lớp để khai báo các biến kiểu lớp hay cũng gọi là đối tượng. Mỗi đối tượng sẽ có các thành phần dữ liệu và các phương thức. Lời gọi một phương thức cần chứa tên đối tượng để xác định phương thức thực hiện từ đối tượng nào.

Một chương trình hướng đối tượng sẽ bao gồm các lớp có quan hệ với nhau. Việc phân tích, thiết kế chương trình theo phương pháp hướng đối tượng nhằm thiết kế, xây dựng

các lớp. Từ khái niệm lớp nảy sinh hàng loạt khái niệm khác như: Thành phần dữ liệu, phương thức, phạm vi, sự đóng gói, hàm tạo, hàm huỷ, sự thừa kế, lớp cơ sở, lớp dẫn xuất, tương ứng bội, phương thức ảo, ...

Ưu điểm của việc thiết kế hướng đối tượng là tập trung xác định các lớp để mô tả các thực thể của bài toán. Mỗi lớp đưa vào các thành phần dữ liệu của thực thể và xây dựng luôn các phương thức để xử lý dữ liệu. Như vậy việc thiết kế chương trình xuất phát từ các nội dung, các vấn đề của bài toán.

Các ngôn ngữ thuần túy hướng đối tượng (như Smalltalk) chỉ hỗ trợ các khái niệm về lớp, không có các khái niệm hàm. C++ là ngôn ngữ lai, nó cho phép sử dụng cả các công cụ của lớp và hàm.

## **1.2 Một số khái niệm quan trọng**

Trong phần này, chúng ta tìm hiểu các khái niệm như sự đóng gói, tính kế thừa và tính đa hình. Đây là các khái niệm căn bản, là nền tảng tư tưởng của lập trình hướng đối tượng. Hiểu được khái niệm này, chúng ta bước đầu tiếp cận với phong cách lập trình mới, phong cách lập trình dựa vào đối tượng làm nền tảng mà trong đó quan điểm che giấu thông tin thông qua sự đóng gói là quan điểm trung tâm của vấn đề.

### **Sự đóng gói (Encapsulation)**

Sự đóng gói là cơ chế ràng buộc dữ liệu và thao tác trên dữ liệu đó thành một thể thống nhất, tránh được các tác động bất ngờ từ bên ngoài. Thể thống nhất này gọi là đối tượng. Tất cả các thông tin của một hệ thống định hướng đối tượng được lưu trữ bên trong đối tượng của nó và chỉ có thể hành động khi các đối tượng đó được ra lệnh thực hiện các thao tác.

Như vậy, sự đóng gói không chỉ đơn thuần là sự gom chung dữ liệu và chương trình vào trong một khối, chúng còn được hiểu theo nghĩa là sự đồng nhất giữa dữ liệu và các thao tác tác động lên dữ liệu đó.

Trong một đối tượng, dữ liệu hay thao tác hay cả hai có thể là riêng (private) hoặc chung (public) của đối tượng đó. Thao tác hay dữ liệu riêng là thuộc về đối tượng đó chỉ được truy cập bởi các thành phần của đối tượng, điều này nghĩa là thao tác hay dữ liệu riêng không thể truy cập bởi các phần khác của chương trình tồn tại ngoài đối tượng.

Khi thao tác hay dữ liệu là chung, các phần khác của chương trình có thể truy cập nó mặc dù nó được định nghĩa trong một đối tượng. Các thành phần chung của một đối tượng dùng để cung cấp một giao diện có điều khiển cho các thành phần riêng của đối tượng. Cơ chế đóng gói là phương thức tốt để thực hiện cơ chế che dấu thông tin so với các ngôn ngữ lập trình cấu trúc.

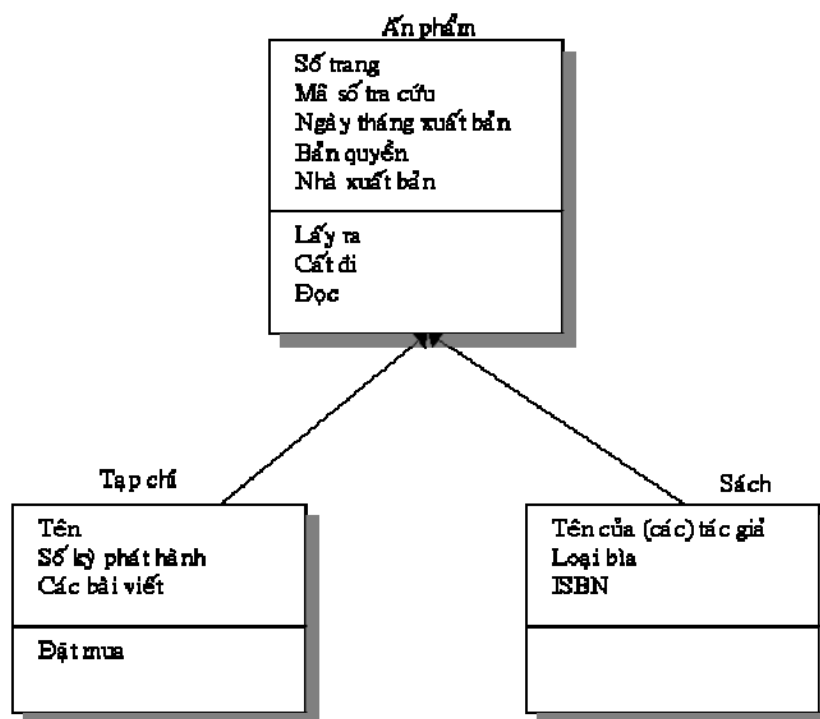
### **Tính kế thừa (Inheritance)**

Chúng ta có thể xây dựng các lớp mới từ các lớp cũ thông qua sự kế thừa. Một lớp mới còn gọi là lớp dẫn xuất (derived class), có thể thừa hưởng dữ liệu và các phương thức của lớp cơ sở (base class) ban đầu. Trong lớp này, có thể bổ sung các thành phần dữ liệu và các phương thức mới vào những thành phần dữ liệu và các phương thức mà nó thừa hưởng từ lớp cơ sở.

Mỗi lớp (kể cả lớp dẫn xuất) có thể có một số lượng bất kỳ các lớp dẫn xuất. Qua cơ cấu kế thừa này, dạng hình cây của các lớp được hình thành. Dạng cây của các lớp trông giống như các cây gia phả vì thế các lớp cơ sở còn được gọi là lớp cha (parent class) và các lớp dẫn xuất được gọi là lớp con (child class).

Ví dụ: Chúng ta sẽ xây dựng một tập các lớp mô tả cho thư viện các ấn phẩm. Có hai kiểu ấn phẩm: tạp chí và sách. Chúng ta có thể tạo một ấn phẩm tổng quát bằng cách định nghĩa các thành phần dữ liệu tương ứng với số trang, mã số tra cứu, ngày tháng xuất bản, bản quyền và nhà xuất bản. Các ấn phẩm có thể được lấy ra, cất đi và đọc. Đó là các phương thức thực hiện trên một ấn phẩm. Tiếp đó chúng ta định nghĩa hai lớp dẫn xuất tên là tạp chí và sách. Tạp chí có tên, số ký phát hành và chứa nhiều bài của

các tác giả khác nhau. Các thành phần dữ liệu tương ứng với các yếu tố này được đặt vào định nghĩa của lớp tạp chí. Tạp chí cũng cần có một phương thức nữa đó là đặt mua. Các thành phần dữ liệu xác định cho sách sẽ bao gồm tên của (các) tác giả, loại bìa (cứng hay mềm) và số hiệu ISBN của nó. Như vậy chúng ta có thể thấy, sách và tạp chí có chung các đặc trưng ẩn phẩm, trong khi vẫn có các thuộc tính riêng của chúng.



Hình 1 - Lớp ấn phẩm và các lớp dẫn xuất của nó

Với tính kế thừa, chúng ta không phải mất công xây dựng lại từ đầu các lớp mới, chỉ cần bổ sung để có được trong các lớp dẫn xuất các đặc trưng cần thiết.

### Tính đa hình (Polymorphism)

Đó là khả năng để cho một thông điệp có thể thay đổi cách thực hiện của nó theo lớp cụ thể của đối tượng nhận thông điệp. Khi một lớp dẫn xuất được tạo ra, nó có thể thay đổi cách thực hiện các phương thức nào đó mà nó thừa hưởng từ lớp cơ sở của nó. Một thông điệp khi được gọi đến một đối tượng của lớp cơ sở, sẽ dùng phương thức đã định

nghĩa cho nó trong lớp cơ sở. Nếu một lớp dẫn xuất định nghĩa lại một phương thức thừa hưởng từ lớp cơ sở của nó thì một thông điệp có cùng tên với phương thức này, khi được gởi tới một đối tượng của lớp dẫn xuất sẽ gọi phương thức đã định nghĩa cho lớp dẫn xuất.

Như vậy đa hình là khả năng cho phép gởi cùng một thông điệp đến những đối tượng khác nhau có cùng chung một đặc điểm, nói cách khác thông điệp được gởi đi không cần biết thực thể nhận thuộc lớp nào, chỉ biết rằng tập hợp các thực thể nhận có chung một tính chất nào đó. Chẳng hạn, thông điệp “vẽ hình” được gởi đến cả hai đối tượng hình hộp và hình tròn. Trong hai đối tượng này đều có chung phương thức vẽ hình, tuy nhiên tùy theo thời điểm mà đối tượng nhận thông điệp, hình tương ứng sẽ được vẽ lên.

Trong các ngôn ngữ lập trình OOP, tính đa hình thể hiện qua khả năng cho phép mô tả những phương thức có tên giống nhau trong các lớp khác nhau. Đặc điểm này giúp người lập trình không phải viết những cấu trúc điều khiển rườm rà trong chương trình, các khả năng khác nhau của thông điệp chỉ thực sự đòi hỏi khi chương trình thực hiện.

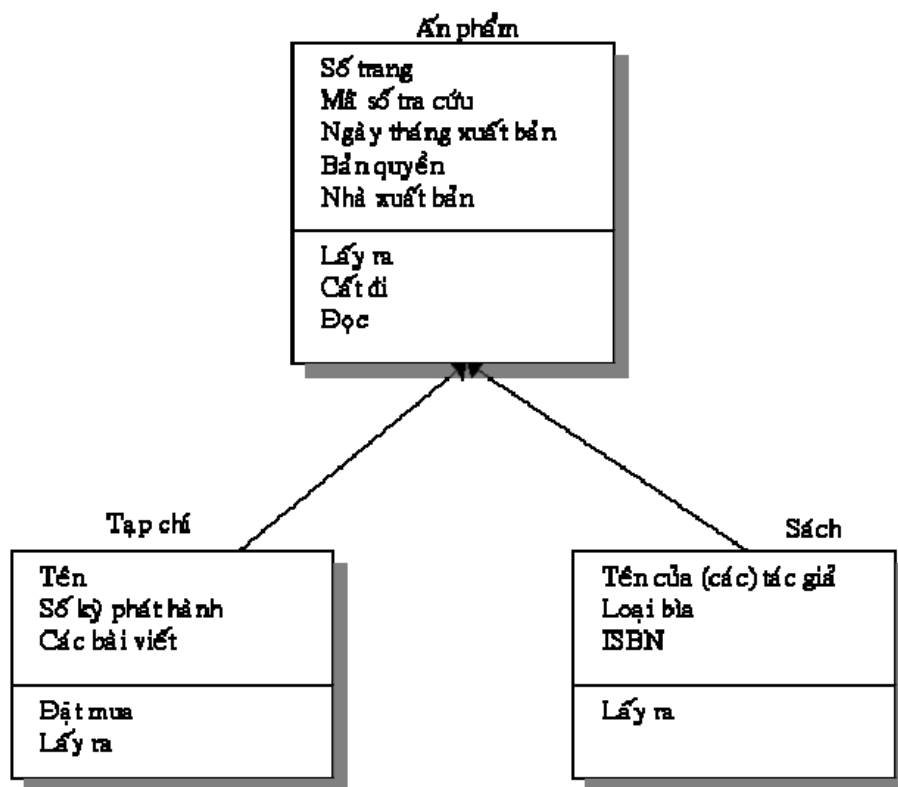
Ví dụ: Xét lại ví dụ trước, chúng ta thấy rằng cả tạp chí và sách đều phải có khả năng lấy ra. Tuy nhiên phương pháp lấy ra cho tạp chí có khác so với phương pháp lấy ra cho sách, mặc dù kết quả cuối cùng giống nhau.

Khi phải lấy ra tạp chí, thì phải sử dụng phương pháp lấy ra riêng cho tạp chí (dựa trên một bản tra cứu) nhưng khi lấy ra sách thì lại phải sử dụng phương pháp lấy ra riêng cho sách (dựa trên hệ thống phiếu lưu trữ).

Tính đa hình cho phép chúng ta xác định một phương thức để lấy ra một tạp chí hay một cuốn sách. Khi lấy ra một tạp chí nó sẽ dùng phương thức lấy ra dành riêng cho tạp chí, còn khi lấy ra một cuốn sách thì nó sử dụng phương thức lấy ra tương ứng với sách.

Kết quả là chỉ cần một tên phương thức duy nhất được dùng cho cả hai công việc tiến hành trên hai lớp dẫn xuất có liên quan, mặc dù việc thực hiện của phương thức đó thay đổi tùy theo từng lớp.

Tính đa hình dựa trên sự nối kết (Binding), đó là quá trình gắn một phương thức với một hàm thực sự. Khi các phương thức kiểu đa hình được sử dụng thì trình biên dịch chưa thể xác định hàm nào tương ứng với phương thức nào sẽ được gọi. Hàm cụ thể được gọi sẽ tùy thuộc vào việc phân tử nhận thông điệp lúc đó là thuộc lớp nào, do đó hàm được gọi chỉ xác định được vào lúc chương trình chạy. Điều này gọi là sự kết nối muộn (Late binding) hay kết nối lúc chạy (Runtime binding) vì nó xảy ra khi chương trình đang thực hiện.



Hình 2 – Minh họa tính đa hình đối với ấn phẩm và các lớp dẫn xuất của nó

## **CÂU HỎI ÔN TẬP CHƯƠNG 1**

1. Thế nào là lập trình hướng đối tượng?
2. Phân biệt lập trình hướng cấu trúc và lập trình hướng đối tượng?

Tailieu.vn

## Chương 2

### CÁC ĐẶC ĐIỂM CỦA C++

#### MỤC ĐÍCH

Nhắc lại một số kiến thức cơ bản và lập trình cấu trúc trong C++ và giới thiệu các đặc điểm mới (hướng đối tượng) trong C++.

#### Nhiệm vụ tự học và tài liệu tham khảo của sinh viên trong chương 2

Để đạt được mục đích đề ra trong chương 2, sinh viên phải thực hiện tốt các yêu cầu sau đây:

- Xem trước đề cương chi tiết của Chương 2 để biết được mục đích, nội dung sẽ được trình bày trong chương này
- Phải đọc trước nội dung Chương 2 trong quyển bài giảng này trước khi lên lớp

#### 2.1 Cấu trúc chương trình

Có lẽ một trong những cách tốt nhất để bắt đầu học một ngôn ngữ lập trình là bằng một chương trình. Vậy đây là chương trình đầu tiên của chúng ta:

```
// my first program in C++
#include <iostream.h>
int main ()
{
    cout << "Hello World!";
    return 0;
}
```

Chương trình trên đây là chương trình đầu tiên mà hầu hết những người học nghề lập trình viết đầu tiên và kết quả của nó là viết câu "Hello World" lên màn hình. Đây là một trong những chương trình đơn giản nhất có thể viết bằng C++ nhưng nó đã bao

gồm những phần cơ bản mà mọi chương trình C++ có. Hãy cùng xem xét từng dòng một:

```
// my first program in C++
```

Đây là dòng chú thích. Tất cả các dòng bắt đầu bằng hai dấu sổ (//) được coi là chú thích mà chúng không có bất kì một ảnh hưởng nào đến hoạt động của chương trình. Chúng có thể được các lập trình viên dùng để giải thích hay bình phẩm bên trong mã nguồn của chương trình. Trong trường hợp này, dòng chú thích là một giải thích ngắn gọn những gì mà chương trình chúng ta làm.

```
#include <iostream.h>
```

Các câu bắt đầu bằng dấu (#) được dùng cho preprocessor (ai dịch hộ tôi từ này với). Chúng không phải là những dòng mã thực hiện nhưng được dùng để báo hiệu cho trình dịch. Ở đây câu lệnh *#include <iostream.h>* báo cho trình dịch biết cần phải "include" thư viện iostream. Đây là một thư viện vào ra cơ bản trong C++ và nó phải được "include" vì nó sẽ được dùng trong chương trình. Đây là cách cổ điển để sử dụng thư viện iostream.

```
int main ()
```

Dòng này tương ứng với phần bắt đầu khai báo hàm main. Hàm main là điểm mà tất cả các chương trình C++ bắt đầu thực hiện. Nó không phụ thuộc vào vị trí của hàm này (ở đầu, cuối hay ở giữa của mã nguồn) mà nội dung của nó luôn được thực hiện đầu tiên khi chương trình bắt đầu. Thêm vào đó, do nguyên nhân nói trên, mọi chương trình C++ đều phải tồn tại một hàm main.

Theo sau main là một cặp ngoặc đơn bởi vì nó là một hàm. Trong C++, tất cả các hàm mà sau đó là một cặp ngoặc đơn () thì có nghĩa là nó có thể có hoặc không có tham số (không bắt buộc). Nội dung của hàm main tiếp ngay sau phần khai báo chính thức được bao trong các ngoặc nhọn ( { } ) như trong ví dụ của chúng ta.

```
cout << "Hello World";
```

Dòng lệnh này làm việc quan trọng nhất của chương trình, cout là một dòng (stream) output chuẩn trong C++ được định nghĩa trong thư viện iostream và những gì mà dòng lệnh này làm là gửi chuỗi kí tự "Hello World" ra màn hình.

Chú ý rằng dòng này kết thúc bằng dấu chấm phẩy ( ; ). Kí tự này được dùng để kết thúc một lệnh và bắt buộc phải có sau mỗi lệnh trong chương trình C++ của bạn (một trong những lỗi phổ biến nhất của những lập trình viên C++ là quên mất dấu chấm phẩy).

```
return 0;
```

Lệnh return kết thúc hàm main và trả về mã đi sau nó, trong trường hợp này là 0. Đây là một kết thúc bình thường của một chương trình không có một lỗi nào trong quá trình thực hiện. Như bạn sẽ thấy trong các ví dụ tiếp theo, đây là một cách phổ biến nhất để kết thúc một chương trình C++.

Chương trình được cấu trúc thành những dòng khác nhau để nó trở nên dễ đọc hơn nhưng hoàn toàn không phải bắt buộc phải làm vậy. Ví dụ, thay vì viết:

```
int main ()  
{  
    cout << " Hello World ";  
    return 0;  
}
```

Ta có thể viết:

```
int main () { cout << " Hello World "; return 0; }
```

cũng cho một kết quả chính xác như nhau.

Trong C++, các dòng lệnh được phân cách bằng dấu chấm phẩy ( ;). Việc chia chương trình thành các dòng chỉ nhằm để cho nó dễ đọc hơn mà thôi.

Các chú thích được các lập trình viên sử dụng để ghi chú hay mô tả trong các phần của chương trình. Trong C++ có hai cách để chú thích:

// Chú thích theo dòng

/\* Chú thích theo khối \*/

Chú thích theo dòng bắt đầu từ cặp dấu xỏ (//) cho đến cuối dòng. Chú thích theo khối bắt đầu bằng /\* và kết thúc bằng \*/ và có thể bao gồm nhiều dòng. Chúng ta sẽ thêm các chú thích cho chương trình:

```
/* my second program in C++  
   with more comments */  
#include <iostream.h>  
int main ()  
{  
    cout << "Hello World! ";    // says Hello World!  
    cout << "I'm a C++ program"; // says I'm a C++ program  
    return 0;  
}
```

Chương trình trên sẽ cho kết quả: Hello World! I'm a C++ program

## 2.2 Cấu trúc điều khiển

Một chương trình thường không chỉ bao gồm các lệnh tuần tự nối tiếp nhau. Trong quá trình chạy nó có thể rẽ nhánh hay lặp lại một đoạn mã nào đó. Để làm điều này chúng ta sử dụng các cấu trúc điều khiển.