

## Mục tiêu

Kết thúc chương, học viên có thể:

- Định nghĩa Lập trình hướng Đối tượng (Object-oriented Programming).
- Nhận thức về Trừu tượng hóa Dữ liệu (Data Abstraction).
- Định nghĩa một Lớp (Class).
- Định nghĩa một Đối tượng (Object).
- Nhận thức được sự khác biệt giữa Lớp và Đối tượng.
- Nhận thức được sự cần thiết đối với Thiết lập (Construction) và Hủy (Destruction).
- Định nghĩa tính Bền vững (Persistence).
- Hiểu biết về tính Thừa kế (Inheritance).
- Định nghĩa tính Đa hình (Polymorphism).
- Liệt kê những thuận lợi của phương pháp hướng Đối tượng.

### 1.1 Giới thiệu về Lập trình hướng Đối tượng (Object-oriented Programming)

Lập trình hướng Đối tượng (OOP) là một phương pháp thiết kế và phát triển phần mềm. Những ngôn ngữ OOP không chỉ bao gồm cú pháp và một trình biên dịch (compiler) mà còn có một môi trường phát triển toàn diện. Môi trường này bao gồm một thư viện được thiết kế tốt, thuận lợi cho việc sử dụng các đối tượng.

Đối với một ngôn ngữ lập trình hỗ trợ OOP thì việc triển khai kỹ thuật lập trình hướng đối tượng sẽ dễ dàng hơn. Kỹ thuật lập trình hướng đối tượng cải tiến việc phát triển các hệ thống phần mềm. Kỹ thuật ấy đề cao nhân tố chức năng (functionality) và các mối quan hệ dữ liệu.

OOP là phương thức tư duy mới để giải quyết vấn đề bằng máy tính. Để đạt kết quả, lập trình viên phải nắn vấn đề thành một thực thể quen thuộc với máy tính. Cách tiếp cận hướng đối tượng cung cấp một giải pháp toàn vẹn để giải quyết vấn đề.

Hãy xem xét một tình huống cần được triển khai thành một hệ thống trên máy vi tính: việc mua bán xe hơi. Vấn đề vi tính hóa việc mua bán xe hơi bao gồm những gì?

Những yếu tố rõ ràng nhất liên quan đến việc mua bán xe hơi là:

- 1) Các kiểu xe hơi (model).

- 2) Nhân viên bán hàng.
- 3) Khách hàng.

Những hoạt động liên quan đến việc mua bán:

- 1) Nhân viên bán hàng đưa khách hàng tham quan phòng trưng bày.
- 2) Khách hàng chọn lựa một xe hơi.
- 3) Khách hàng đặt hóa đơn.
- 4) Khách hàng trả tiền.
- 5) Chiếc xe được trao cho khách hàng.

Mỗi vấn đề được chia ra thành nhiều yếu tố, được gọi là các Đối tượng (Objects) hoặc các Thực thể (Entities). Chẳng hạn như ở ví dụ trên, khách hàng, xe hơi và nhân viên bán hàng là những đối tượng hoặc thực thể.

Lập trình viên luôn luôn cố gắng tạo ra những kịch bản (scenarios) thật quen thuộc với những tình huống đời sống thực. Bước thứ nhất trong đường hướng này là làm cho máy tính liên kết với những đối tượng thế giới thực.

Tuy nhiên, máy tính chỉ là một cỗ máy. Nó chỉ thực hiện những công việc được lập trình mà thôi. Vì thế, trách nhiệm của lập trình viên là cung cấp cho máy tính những thông tin theo cách thức mà nó cũng nhận thức được cùng những thực thể như chúng ta nhận thức.

Đó chính là lãnh vực của kỹ thuật hướng đối tượng. Chúng ta sử dụng kỹ thuật hướng đối tượng để ánh xạ những thực thể chúng ta gặp phải trong đời sống thực thành những thực thể tương tự trong máy tính.

Phát triển phần mềm theo kỹ thuật lập trình hướng đối tượng có khả năng giảm thiểu sự lẫn lộn thường xảy ra giữa hệ thống và lãnh vực ứng dụng.

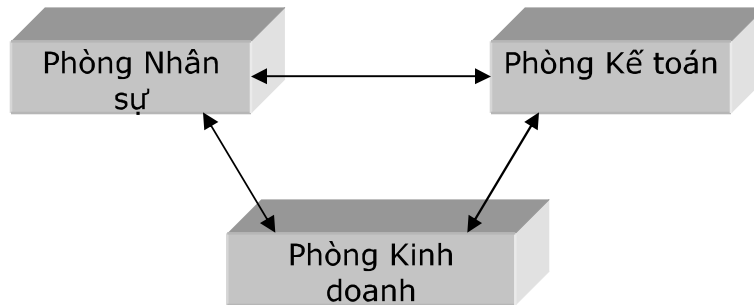
Lập trình hướng đối tượng đề cập đến dữ liệu và thủ tục xử lý dữ liệu theo quan điểm là một đối tượng duy nhất. Lập trình hướng đối tượng xem xét dữ liệu như là một thực thể hay là một đơn vị độc lập, với bản chất riêng và những đặc tính của thực thể ấy. Bây giờ chúng ta hãy khảo sát những hạn từ 'đối tượng' (object), 'dữ liệu' (data) và 'phương thức' (method).

Có nhiều loại đối tượng khác nhau. Chúng ta có thể xem các bộ phận khác nhau trong một cơ quan là các đối tượng. Điển hình là một cơ quan có những bộ phận liên quan đến việc quản trị, đến việc kinh doanh, đến việc kế toán, đến việc tiếp thị ... Mỗi bộ phận có nhân sự riêng, các nhân sự được trao cho những trách nhiệm rõ ràng. Mỗi bộ phận cũng có những dữ liệu riêng chẳng hạn như thông tin cá nhân, bảng kiểm kê, những thể thức kinh doanh, hoặc bất kỳ dữ liệu nào liên quan đến chức năng của bộ phận đó. Rõ ràng là một cơ quan được chia thành nhiều bộ phận thì việc quản trị nhân sự và những hoạt động doanh nghiệp dễ dàng hơn. Nhân sự của cơ quan điều khiển và xử lý dữ liệu liên quan đến bộ phận của mình.

Chẳng hạn như bộ phận kế toán chịu trách nhiệm về lương bổng đối với cơ quan. Nếu một người ở đơn vị tiếp thị cần những chi tiết liên quan đến lương bổng của đơn vị mình, người ấy chỉ cần liên hệ với bộ phận kế toán. Một người có thẩm quyền trong bộ phận kế toán sẽ

cung cấp thông tin cần biết, nếu như thông tin ấy có thể chia sẻ được. Một người không có thẩm quyền từ một bộ phận khác thì không thể truy cập dữ liệu, hoặc không thể thay đổi làm hư hỏng dữ liệu. Ví dụ này minh chứng các đối tượng là hữu dụng trong việc phân cấp và tổ chức dữ liệu.

Hình 1.1 Minh họa cấu trúc của một cơ quan điển hình.



**Hình 1.1**

Khái niệm về đối tượng có thể được mở rộng đến hầu hết các lãnh vực đời sống, và hơn nữa, đến thế giới lập trình. Bất kỳ ứng dụng nào đều có thể được định nghĩa theo hạn từ thực thể hoặc đối tượng để tạo ra tiến trình xử lý mô phỏng theo tiến trình xử lý mà con người nghĩ ra.

Phương pháp giải quyết 'top-down' (từ trên xuống) cũng còn được gọi là 'lập trình hướng cấu trúc' (structured programming). Nó xác định những chức năng chính của một chương trình và những chức năng này được phân thành những đơn vị nhỏ hơn cho đến mức độ thấp nhất. Bằng kỹ thuật này, các chương trình được cấu trúc theo hệ thống phân cấp các module. Mỗi một module có một đầu vào riêng và một đầu ra riêng. Trong mỗi module, sự điều khiển có chiều hướng đi xuống theo cấu trúc chứ không có chiều hướng đi lên.

Phương pháp OOP cố gắng quản lý việc thừa kế phức tạp trong những vấn đề đời thực. Để làm được việc này, phương thức OOP che giấu một vài thông tin bên trong các đối tượng. OOP tập trung trước hết trên dữ liệu. Rồi gắn kết các phương thức thao tác trên dữ liệu, việc này được xem như là phần thừa kế của việc định nghĩa dữ liệu. Bảng 1.1 minh họa sự khác biệt giữa hai phương pháp:

<b>Phương pháp Top-Down</b>	<b>OOP</b>
Chúng ta sẽ xây dựng một khách sạn.	Chúng ta sẽ xây dựng một tòa nhà 10 tầng với những dãy phòng trung bình, sang trọng, và một phòng họp lớn.
Chúng ta sẽ thiết kế các tầng lầu, các phòng và phòng họp.	Chúng ta sẽ xây dựng một khách sạn với những thành phần trên.

**Bảng 1.1 Một ví dụ về hai phương pháp giải quyết OOP và Structured**

## 1.2 Trừu tượng hóa dữ liệu (Data Abstraction)

Khi một lập trình viên phải phát triển một chương trình ứng dụng thì **không có nghĩa là người ấy lập tức viết mã cho ứng dụng ấy. Trước hết, người ấy phải nghiên cứu ứng dụng**

và xác định những thành phần tạo nên ứng dụng. Kế tiếp, người ấy phải xác định những thông tin cần thiết về mỗi thành phần.

Hãy khảo sát chương trình ứng dụng cho việc mua bán xe hơi nói trên. Chương trình phải xuất hóa đơn cho những xe hơi đã bán cho khách hàng. Để xuất một hóa đơn, chúng ta cần những thông tin chi tiết về khách hàng. Vậy bước thứ nhất là xác định những đặc tính của khách hàng.

Một vài đặc tính gắn kết với khách hàng là:

- Tên.
- Địa chỉ.
- Tuổi.
- Chiều cao.
- Màu tóc.

Từ danh sách kể trên, chúng ta xác định những đặc tính thiết yếu đối với ứng dụng. Bởi vì chúng ta đang đề cập đến những khách hàng mua xe, vì thế những chi tiết thiết yếu là:

- Tên.
- Địa chỉ.

Còn những chi tiết khác (chiều cao, màu tóc ...) là không quan trọng đối với ứng dụng. Tuy nhiên, nếu chúng ta phát triển một ứng dụng hỗ trợ cho việc điều tra tội phạm thì những thông tin chẳng hạn như màu tóc là thiết yếu.

Bên cạnh những chi tiết về khách hàng, những thông tin sau cũng cần thiết:

- Kiểu xe được bán.
- Nhân viên nào bán xe.

Bên cạnh những đặc tính của khách hàng, xe hơi và nhân viên bán hàng, chúng ta cũng cần liệt kê những hành động được thực hiện.

Công việc xuất hóa đơn đòi hỏi những hành động sau:

- Nhập tên của khách hàng.
- Nhập địa chỉ của khách hàng.
- Nhập kiểu xe.
- Nhập tên của nhân viên bán xe.
- Xuất hóa đơn với định dạng đòi hỏi.

Khung thông tin bên dưới cho thấy những thuộc tính và những hành động liên quan đến một hóa đơn:

<b>Các thuộc tính</b>
Tên của khách hàng
Địa chỉ của khách hàng

Kiểu xe bán
Nhân viên bán xe

<b>Các hành động</b>
Nhập tên
Nhập địa chỉ
Nhập kiểu xe
Nhập tên nhân viên bán xe
Xuất hóa đơn

## Định nghĩa

**Sự trừu tượng hóa dữ liệu** là tiến trình xác định và nhóm các thuộc tính và các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.

Tiếp theo, chúng ta muốn ứng dụng tính toán tiền hoa hồng cho nhân viên bán hàng.

Những thuộc tính liên kết với nhân viên bán hàng có tương quan với ứng dụng này là:

- Tên.
- Số lượng xe bán được.
- Tiền hoa hồng.

Những hành động đòi buộc đối với công việc này là:

- Nhập tên nhân viên bán xe.
- Nhập số lượng xe bán được.
- Tính tiền hoa hồng kiếm được.

<b>Những thuộc tính</b>
Tên
Số lượng xe bán được
Tiền hoa hồng

<b>Những hành động</b>
Nhập tên
Nhập số lượng xe bán được
Tính tiền hoa hồng

Như thế, việc trừu tượng hóa dữ liệu tra đặt ra câu hỏi 'Đâu là những thuộc tính và những hành động cần thiết cho một vấn đề đặt ra?'

### 1.2.1 Những thuận lợi của việc Trừu tượng hóa

Những thuận lợi của việc Trừu tượng hóa là: