

Bài 1: TỔNG QUAN VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Xem bài học trên website để ủng hộ Kteam: [Tổng quan về lập trình hướng đối tượng](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Đây là bài đầu tiên trong series Lập trình hướng đối tượng với C#. Trong bài này chúng ta sẽ cùng tìm hiểu xem **lập trình hướng đối tượng** là gì? Những đặc điểm cơ bản của lập trình hướng đối tượng?

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- **[BIẾN](#)** và **[KIỂU DỮ LIỆU](#)** trong C#
- **[TOÁN TỬ TRONG C#](#)**
- **[CÂU ĐIỀU KIỆN TRONG C#](#)**
- **[CẤU TRÚC CƠ BẢN CỦA VÒNG LẶP TRONG C#](#)**
- **[CẤU TRÚC HÀM CƠ BẢN TRONG C#](#)**
- **[MẢNG MỘT CHIỀU TRONG C#](#)**

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Lập trình hướng đối tượng là gì?
- Một số khái niệm cơ bản trong lập trình hướng đối tượng.
- Các đặc điểm của lập trình hướng đối tượng

Lập trình hướng đối tượng là gì?

Sự tiến hoá của các phương pháp lập trình

Đầu tiên là **lập trình không có cấu trúc**:

- Là phương pháp xuất hiện đầu tiên. Phương pháp này đơn giản chỉ là viết tất cả mã lệnh vào 1 hàm **main** duy nhất và chạy.
- Ngôn ngữ sử dụng phương pháp này là **Assembly** (hợp ngữ).
- Nhược điểm của phương pháp này:
 - Chỉ sử dụng biến toàn cục dẫn đến rất tốn bộ nhớ.
 - Vì có những đoạn chương trình cần sử dụng lại nhiều lần nên dẫn đến lạm dụng lệnh goto.
 - Khó hiểu, khó bảo trì, không thể tái sử dụng.
 - Khó phát triển các ứng dụng lớn.

Tiếp theo là **lập trình có cấu trúc** (lập trình thủ tục):

- Sau 1 thời gian tồn tại người ta dần nhận ra những nhược điểm của phương pháp trên và đưa ra giải pháp đó là chia chương trình lớn ra thành các chức năng, mỗi chức năng được đưa vào 1 hàm. Khi cần dùng đến chức năng nào thì ta sẽ gọi hàm tương ứng.
- Mỗi chương trình con lại có thể chia nhỏ ra nữa.
- Hầu hết các ngôn ngữ lập trình đều hỗ trợ phương pháp này.
- Ưu điểm:
 - Chương trình được **module hoá, dễ hiểu, dễ bảo trì**.
 - **Dễ dàng tạo** ra các **thư viện** phần mềm.
- Nhược điểm:
 - Dữ liệu và xử lý tách rời.
 - Khi cấu trúc dữ liệu thay đổi sẽ dẫn đến thuật toán bị thay đổi.
 - Không tự động khởi tạo, giải phóng dữ liệu động.
 - Không mô tả được đầy đủ, trung thực hệ thống trong thực tế.

Tiếp theo là **lập trình hướng đối tượng**, phương pháp mà chúng ta sẽ tìm hiểu trong series này.

Khái niệm lập trình hướng đối tượng

Với mong muốn xây dựng một phương pháp lập trình trực quan, mô tả trung thực hệ thống trong thực tế vì thế phương pháp **lập trình hướng đối tượng** ra đời.

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

Một định nghĩa khác về lập trình hướng đối tượng đó là phương pháp lập trình dựa trên kiến trúc **lớp** (class) và **đối tượng** (object).

Một số khái niệm cơ bản trong lập trình hướng đối tượng

Đối tượng

Trong lập trình hướng đối tượng, **đối tượng** được hiểu như là 1 thực thể: người, vật hoặc 1 bảng dữ liệu, ...

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: một người sẽ có họ tên, ngày sinh, màu da, kiểu tóc, ...
- **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

Các đối tượng có các đặc tính tương tự nhau được gom lại thành 1 **lớp đối tượng**.

Bên trong lớp cũng có 2 thành phần chính đó là thuộc tính và phương thức.

Ngoài ra, lớp còn được dùng để định nghĩa ra kiểu dữ liệu mới.

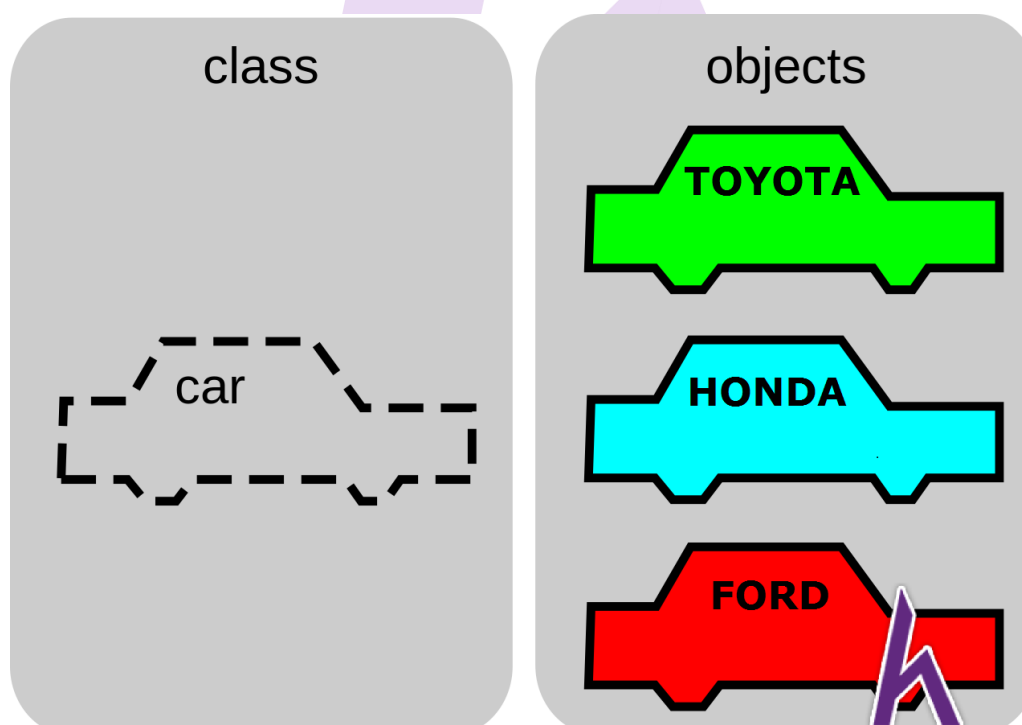
Sự khác nhau giữa đối tượng và lớp

Lớp là một khuôn mẫu còn **đối tượng** là một thể hiện cụ thể dựa trên khuôn mẫu đó.

Để dễ hiểu hơn mình sẽ lấy một ví dụ thực tế:

- Nói về con mèo thì lớp chính là loài mèo. Loài mèo có:
 - Các thông tin, đặc điểm như 4 chân, 2 mắt, có đuôi, có chiều cao, có cân nặng, màu lông . . .
 - Các hành động như: kêu meo meo, đi, ăn, ngủ, . . .
- Như vậy mọi động vật thuộc loài mèo sẽ có những đặc điểm như trên.
- Đối tượng chính là một con mèo cụ thể nào đó như con mèo con đang nằm dưới chân mình.

Một ví dụ khác. Ví dụ này mình sẽ dùng 1 hình ảnh để minh họa:



- Bạn thấy đấy khi nói đến xe ô tô thì lớp chính khuôn mẫu của cái xe với các đặc trưng như có 4 bánh và có thiết kế tương tự như hình vẽ. Đối

tượng chính là các chiếc xe ô tô cụ thể như Toyota camry, Honda city, Ford ranger.

Sự xuất hiện của 2 khái niệm mới là **lớp** và **đối tượng** chính là **đặc trưng của phương pháp lập trình hướng đối tượng**. Nó đã giải quyết được các khuyết điểm của phương pháp lập trình hướng cấu trúc để lại. Ngoài ra 2 khái niệm này đã giúp biểu diễn tốt hơn thế giới thực trên máy tính.

Các đặc điểm của lập trình hướng đối tượng

Lập trình hướng đối tượng có 4 đặc điểm chính:

Tính đóng gói:

- Các dữ liệu và phương thức có liên quan với nhau được đóng gói thành các lớp để tiện cho việc quản lý và sử dụng.
- Ngoài ra, đóng gói còn để che giấu một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không thể nhìn thấy.

Tính trừu tượng: Khi viết chương trình theo phong cách hướng đối tượng, việc thiết kế các đối tượng ta cần rút tĩa ra những đặc trưng chung của chúng rồi trừu tượng thành các [interface](#) (khái niệm [interface](#) sẽ được trình bày trong bài [INTERFACE TRONG C#](#)) và thiết kế xem chúng sẽ tương tác với nhau như thế nào.

Tính kế thừa: Lớp cha có thể chia sẻ dữ liệu và phương thức cho các lớp con, các lớp con khỏi phải định nghĩa lại, giúp chương trình ngắn gọn. Chi tiết sẽ được trình bày trong bài [TÍNH KẾ THỪA TRONG C#](#).

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. Chi tiết sẽ được trình bày và giải thích trong bài [TÍNH ĐA HÌNH TRONG C#](#).

Kết luận

Nội dung bài này giúp các bạn nắm được:

- Lập trình hướng đối tượng là gì?

Bài tiếp theo chúng ta sẽ cùng tìm hiểu về **CLASS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.

