



Lớp và Đối tượng

Khóa học: NỀN TẢNG LẬP TRÌNH

Mục tiêu



- Trình bày được mô hình lập trình hướng đối tượng
- Trình bày được các khái niệm:
 - lớp, đối tượng
 - phương thức, thuộc tính
 - phương thức khởi tạo
- Trình bày được cú pháp khai báo lớp
- Trình bày được cú pháp khởi tạo đối tượng
- Tạo và sử dụng được các đối tượng đơn giản
- Phân biệt lớp và đối tượng
- Trình bày được cách khai báo thuộc tính trong lớp
- Trình bày được cách khai báo phương thức trong lớp
- Trình bày được cách truy xuất thuộc tính và gọi phương thức của lớp



Thảo luận

Lập trình hướng đối tượng

Ngôn ngữ lập trình



- Ngôn ngữ lập trình cho phép Lập trình viên phát triển các phần mềm
- Các dòng ngôn ngữ lập trình phổ biến:
 - Ngôn ngữ máy (machine languages)
 - Ngôn ngữ assembly
 - Ngôn ngữ bậc cao (high-level languages)

Ngôn ngữ máy



- Được tạo thành từ các số 0 và số 1
- Là ngôn ngữ “native” của máy tính
- Khá khó để lập trình
- Ví dụ:

1110100010101 111010101110

10111010110100 10100011110111

Ngôn ngữ assembly

- Ngôn ngữ assembly dễ lập trình hơn so với ngôn ngữ máy
- Bao gồm một tập các câu lệnh (command) cần thiết được quy định trong các bộ vi xử lý
- Ngôn ngữ assembly cần thiết được chuyển sang ngôn ngữ máy trước khi được thực thi
- Ví dụ:

`ADD 1001010, 1011010`



Ngôn ngữ bậc cao

- Ngôn ngữ bậc cao rất dễ để lập trình so với ngôn ngữ assembly
- Cú pháp của ngôn ngữ bậc cao khá gần với tiếng Anh.
- Ngôn ngữ bậc cao có thể được phân loại thành 2 thể loại phổ biến:
 - Ngôn ngữ thủ tục (procedural languages)
 - Ngôn ngữ Lập trình Hướng đối tượng (OOP – Object Oriented Language)



Ngôn ngữ Thủ tục

- Các ngôn ngữ thủ tục thường là các ngôn ngữ bậc cao ra đời từ rất sớm
- Đặc trưng bởi các tập lệnh tuyến tính nối tiếp nhau
- Tập trung vào cấu trúc (structure) của chương trình
- Ví dụ: C, COBOL, Fortran, LISP, Perl, VBScript

Ngôn ngữ Hướng đối tượng (OOP)



- Không tập trung vào cấu trúc như ngôn ngữ thủ tục, mà tập trung vào mô hình hoá dữ liệu (data modeling)
- Lập trình viên sử dụng các Lớp (class) khi lập trình
- Mô phỏng các đối tượng trong thế giới thực vào trong các chương trình.
- Class: khuôn mẫu của dữ liệu được mô hình hoá
- Ví dụ: C++, C#, Java, PHP, Javascript...



Các khái niệm

- Đối tượng (Object)
- Lớp (Class)
- Thuộc tính (Property)
- Phương thức (Method)/Hành vi (Behavior)/Hành động (Action)/Khả năng (Capability)
- Các đặc tính cơ bản:
 - Tính bao gói
 - Tính kế thừa
 - Tính trừu tượng
 - Tính đa hình



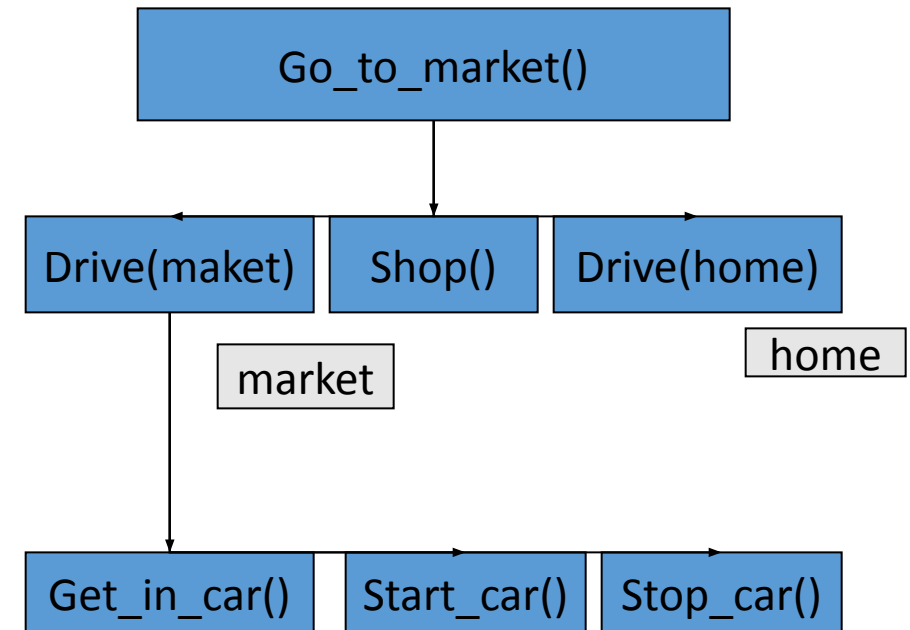
Ngôn ngữ Hướng đối tượng (OOP)

- OOP (Object Oriented Programming) là một triết lý thiết kế chương trình
- Có nhiều ngôn ngữ lập trình hỗ trợ OOP
- Tất cả mọi thứ trong OOP đều là “đối tượng”. Một chương trình phần mềm được coi như là một thế giới bao gồm các đối tượng tương tác với nhau
- Mã lệnh và dữ liệu được kết hợp trong một thể thống nhất – đó là *đối tượng*.
- Đối tượng bao gồm:
 - Thuộc tính: các dữ liệu, tính chất của đối tượng
 - Hành vi: các khả năng, hành động mà đối tượng có thể thực hiện
- Các đối tượng có thể có quan hệ với nhau

So sánh Lập trình Thủ tục và OOP - 1



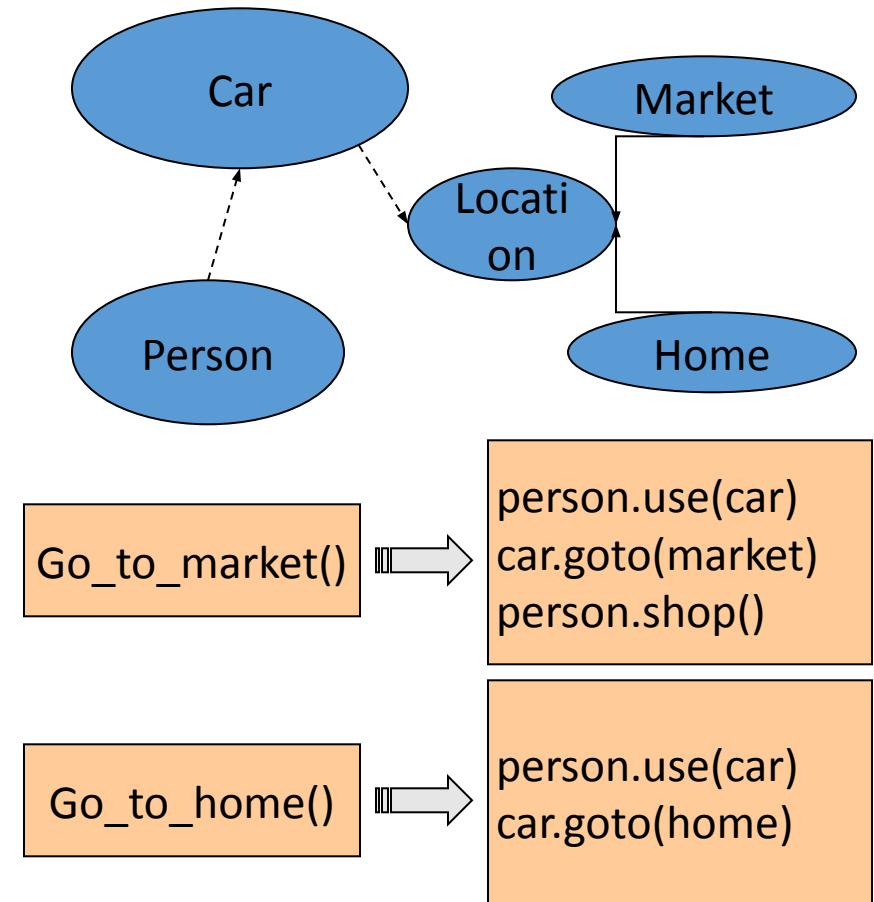
- Lập trình hướng cấu trúc:
 - Hướng tiếp cận: từ trên xuống (top down)
 - Chia nhỏ bài toán thành các module chức năng.
 - Dữ liệu và mã lệnh phân tán.
- Hạn chế:
 - Tính ổn định giảm khi hệ thống phát triển
 - Khó bảo trì và tái sử dụng
 - Chi phí phát triển cao



So sánh Lập trình Thủ tục và OOP - 2



- Hướng đối tượng
 - Hướng tiếp cận đa dạng gần với thực tế.
 - Tìm và phân tích mối quan hệ giữa các đối tượng trong bài toán
 - Mã lệnh và dữ liệu liên kết trong thể thống nhất.
- Ưu thế:
 - Khả năng tái sử dụng cao
 - Ổn định và dễ bảo trì
 - Chi phí giảm dần.



Thảo luận

Lớp

Đối tượng

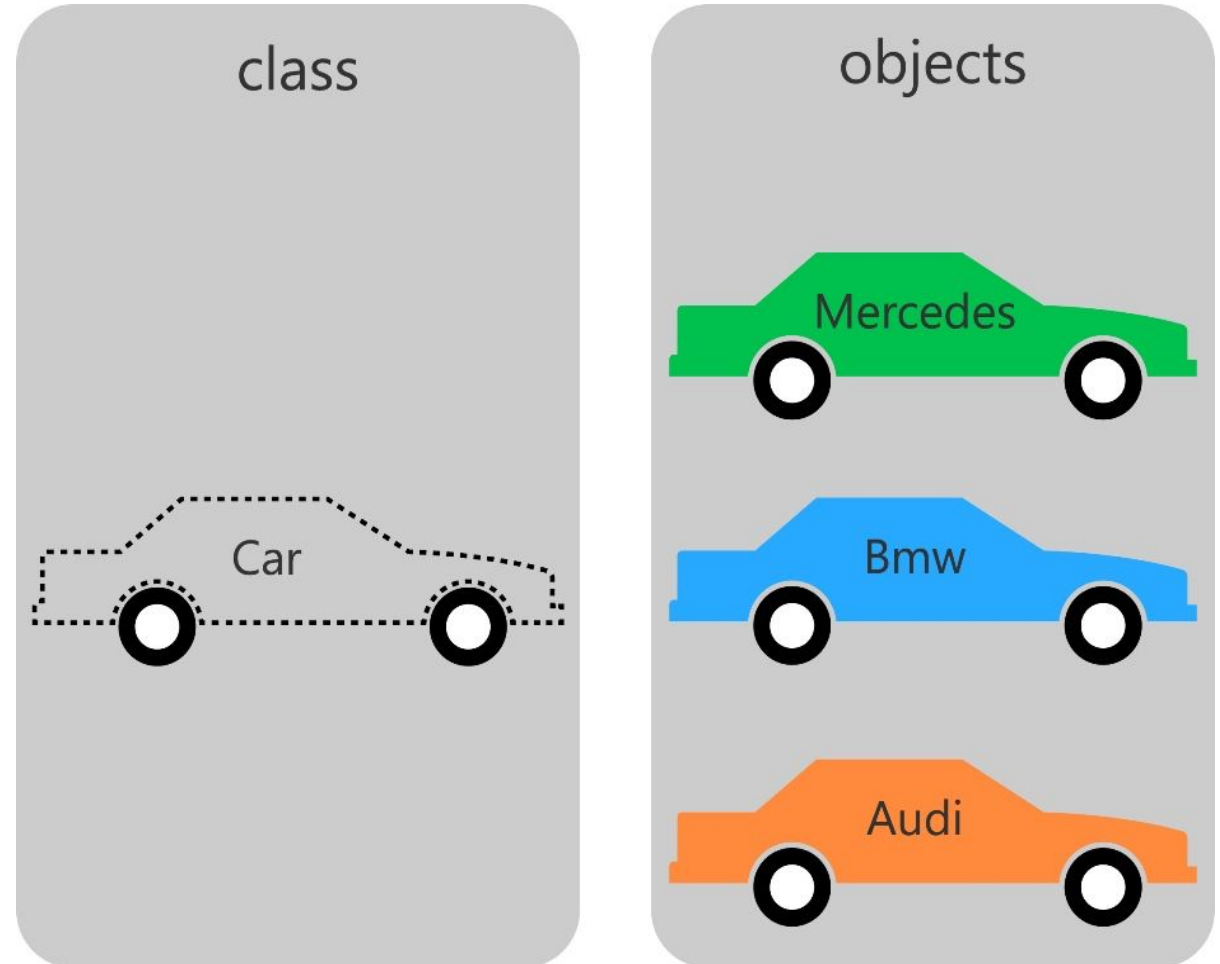
Thuộc tính

Phương thức

Lớp & Đối tượng




- Lớp là khuôn mẫu chứa những khai báo định dạng và nguyên tắc hoạt động
- Đối tượng là sự thể hiện của một lớp và được coi như những sản phẩm thực sự được tạo ra từ khuôn mẫu đó (lớp).



Đối tượng


- Các **đối** tượng là điểm cốt lõi để hiểu về công nghệ hướng đối tượng.
- Đối tượng trong thế giới thực có chung hai đặc điểm: trạng thái và hành vi.
- OOP xây **dựng** những đối tượng trong phần mềm với hai đặc điểm: thuộc tính và phương thức.
- Các đối tượng được tạo ra từ lớp (class) được xây dựng trước đó.

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

Thuộc tính




- Đối tượng có các thuộc tính (property) dùng để xác định các đặc tính của nó.
- VD: *xe hơi có thuộc tính dòng xe, mẫu xe, trọng lượng, màu sắc*

Object	Properties	Methods
	<div>car.name = Fiat car.model = 500 car.weight = 850kg car.color = white</div>	<div>car.start() car.drive() car.brake() car.stop()</div>

Phương thức



- Đối tượng còn có các phương thức (method) dùng để định nghĩa cách thức hoạt động của nó.
- VD: xe hơi có phương thức *start()*, *drive()*, *break()*, *stop()*

Object	Properties	Methods
	<code>car.name = Fiat</code> <code>car.model = 500</code> <code>car.weight = 850kg</code> <code>car.color = white</code>	<code>car.start()</code> <code>car.drive()</code> <code>car.brake()</code> <code>car.stop()</code>

Đối tượng

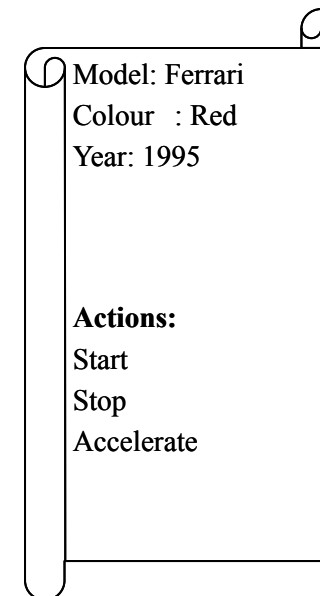
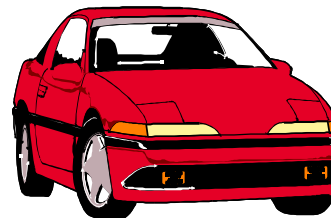
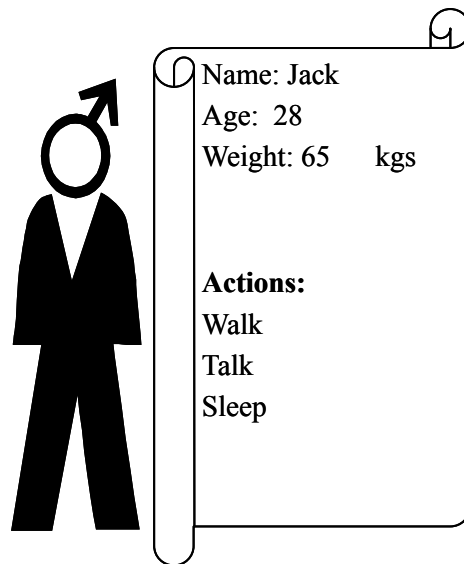


- Mỗi đối tượng có những **thuộc tính** hay những đặc điểm mô tả và những **hành vi** riêng nhằm phân biệt nó với các đối tượng khác.



Thuộc tính và Hành vi

- Thuộc tính là những đặc điểm đặc trưng của đối tượng, thể hiện thông qua những giá trị cụ thể.
- Hành vi là những cách thức mà qua đó đối tượng thể hiện sự hoạt động hay chức năng của chúng.





Định nghĩa lớp

- Tạo lớp mới với thuộc tính và phương thức

```
class Car {  
  constructor(name, year) {  
    this.name = name;  
    this.year = year;  
  }  
  age() {  
    let date = new Date();  
    return date.getFullYear() - this.year;  
  }  
}
```

- Tạo đối tượng

```
let myCar = new Car("Ford", 2014);
```



Sử dụng thuộc tính

- Truy xuất giá trị của thuộc tính của đối tượng

```
objectName.property           // person.age
```

- Sử dụng câu lệnh for-in để truy xuất đến tất cả các thuộc tính của đối tượng

```
objectName["property"]       // person["age"]
```

```
for (x in person) {  
    txt += person[x];  
}
```



Sử dụng phương thức

- Gọi một phương thức của đối tượng
`objectName.methodName(parameters);`
- Truyền tham số khi gọi phương thức của đối tượng

```
myMother.changeName("Doe");
```

Demo: Tạo và sử dụng lớp Hình Tròn



Hãy xây dựng lớp hình tròn theo sơ đồ lớp sau:

Circle

- radius: number
- color: string

+ constructor()
+ getRadius(): number
+ getArea(): number

Lớp Circle (hình tròn) gồm:

- Thuộc tính:
 - bán kính (radius) sẽ nhận vào giá trị dạng số thực
 - màu sắc (color) sẽ nhận vào giá trị dạng chuỗi.
- Phương thức:
 - Circle() là phương thức khởi tạo để tạo đối tượng không tham số.
 - getRadius() là phương thức trả về bán kính của hình tròn
 - getArea() là phương thức trả về diện tích hình tròn theo công thức $S = \text{Math.PI} * \text{radius} * \text{radius}$

Quy tắc đặt tên lớp



- Tên lớp nên là một danh từ
- Ký tự đầu tiên của tên lớp phải viết hoa và được đặt tên theo quy tắc Pascal tức ký tự đầu tiên của mỗi từ phải viết hoa
- Tên lớp đơn giản, có ý nghĩa
- Không dùng từ khoá để đặt tên lớp (từ khoá: let, var, function, new ...)
- Tên lớp không được bắt đầu bởi một số, nên bắt đầu là một ký tự trong bảng chữ cái (A-Z_)
- Ví dụ: Student, Employee,...

Phương thức khởi tạo



- Phương thức khởi tạo được gọi ngay khi một đối tượng được tạo ra bởi từ khóa new.
- Ví dụ:
 - Trong lớp Circle có phương thức khởi tạo là constructor(name, model, weight, color)

Demo: Tạo và sử dụng lớp Hình Tròn



```
class Circle {  
    constructor(radius, color) {  
        this.radius = radius;  
        this.color = color;  
    }  
    getRadius() {  
        return this.radius;  
    }  
    getArea() {  
        return Math.PI * this.radius * this.radius;  
    }  
};  
  
var circle = new Circle(2);  
circle.getRadius(); // 2  
circle.getArea();   // 12.566370614359172
```



Demo

Tạo lớp và đối tượng

Tổng kết



Qua bài học này, học viên đã được tìm hiểu các khái niệm sau:

- Mô hình lập trình hướng đối tượng
- Lớp, đối tượng
- Phương thức, thuộc tính
- Phương thức khởi tạo
- Phân biệt lớp và đối tượng

Học viên cần đạt được những kỹ năng sau:

- Khai báo được lớp
- Khởi tạo đối tượng
- Khai báo thuộc tính trong lớp
- Khai báo phương thức trong lớp
- Truy cập được các thành phần của lớp