



Bài 8

Lập trình hướng đối tượng

Module: BOOTCAMP PREPARATION

Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Lập trình hướng đối tượng”
Tóm tắt lại các phần đã học từ bài “Lập trình hướng đối tượng”

Mục tiêu

- Phân biệt lớp và đối tượng
- Trình bày được cách tạo lớp
- Trình bày được cách tạo đối tượng
- Trình bày được cách khai báo thuộc tính trong lớp
- Trình bày được cách khai báo phương thức trong lớp
- Tạo được đối tượng
- Truy cập được các thành phần của lớp



Thảo luận

Lớp

Đối tượng

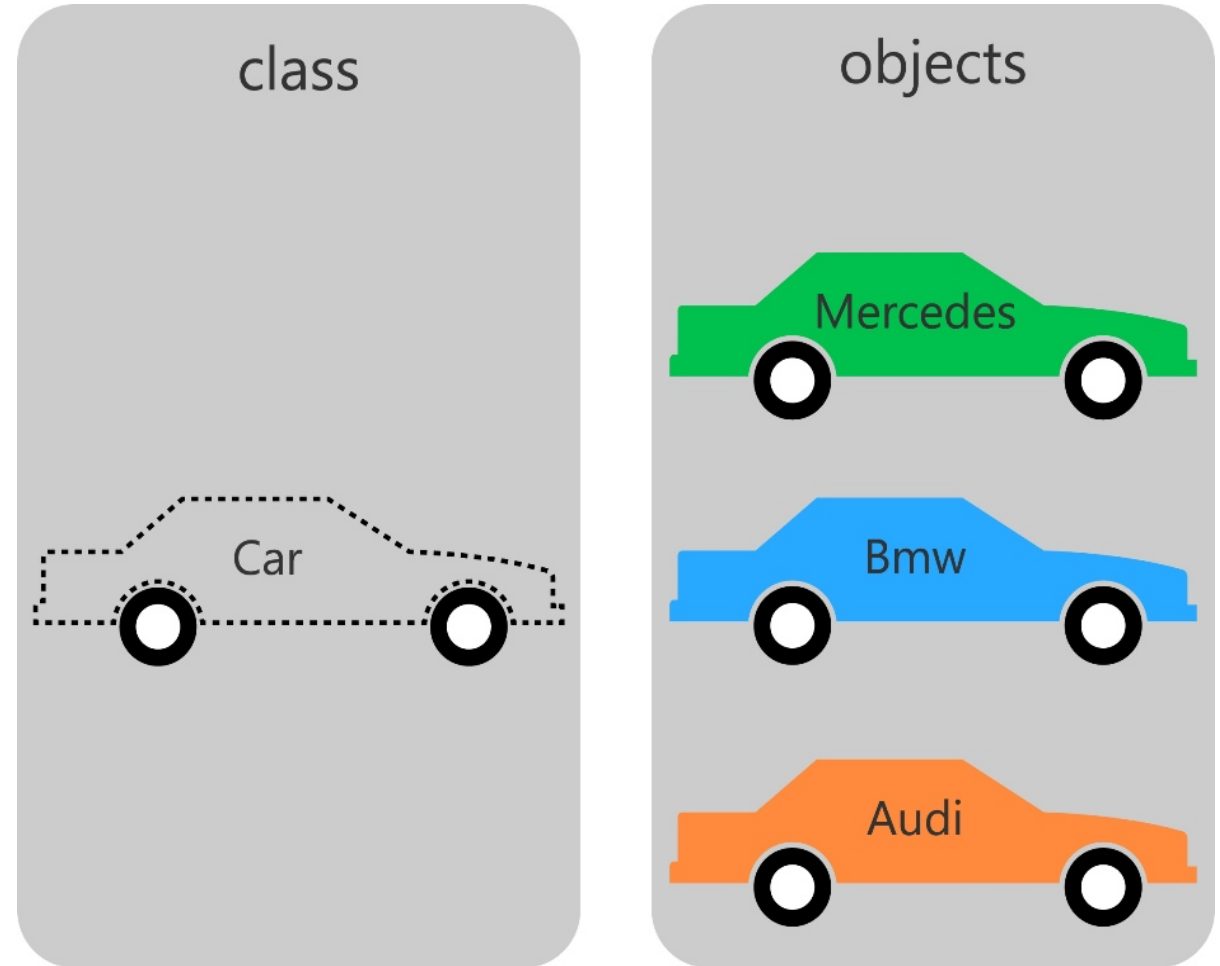
Thuộc tính

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

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>



Định nghĩa một lớp

- Trong OOP, định nghĩa một lớp đồng nghĩa với việc tạo ra một kiểu dữ liệu mới.
- VD định nghĩa một lớp với các thuộc tính:

```
function Car(name, model, weight, color) {  
    this.name    = name;           //Thuộc tính  
    this.model  = model;           //Thuộc tính  
    this.weight = weight;          //Thuộc tính  
    this.color  = color;           //Thuộc tính  
    //Phương thức  
    this.start = function() {  
        document.write("Starting...");  
    }  
}
```

- Từ khóa **this** để chỉ chính đối tượng đó, thuộc sở hữu của đối tượng đó.

Sơ đồ lớp Car



Car

- name: string
- model: string
- weight: double
- color: string

+ Car(name, model, weight, color)
+ start()
+ dive()
+ stop()
+ brake()
+ toString()

Trong sơ đồ lớp Car gồm:

- 4 thuộc tính name kiểu chuỗi, model kiểu chuỗi, weight kiểu số thực, color kiểu chuỗi.
- Phương thức khởi tạo Car() có 4 tham số truyền vào
- Các phương thức: start(), dive(), stop(), brake(), toString()



Demo

Lớp Car



Thảo luận

Quy tắc đặt tên lớp

Phương thức khởi tạo

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: 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, NullPointerException ...

Tạo đối tượng từ lớp



- Các đối tượng được khai báo như là một biến có kiểu là lớp tương ứng
- Sử dụng từ khóa `new` để tạo mới một đối tượng từ một lớp đã định nghĩa trước đó.
- Từ khoá `new` dùng để cấp phát động một vùng nhớ cho đối tượng và trả về một tham chiếu đến đối tượng đó.
- VD tạo đối tượng từ lớp `Car` và gọi phương thức `start()` của nó:

```
var carobj = new Car("Hyundai", "i10", 1200, "White");  
carobj.start();
```

Phương thức khởi tạo



- Phương thức khởi tạo (Constructor) là một phương thức đặc biệt với các đặc điểm:
 - Tên phương thức trùng với tên lớp
 - Không có kiểu dữ liệu trả về
 - Có các tham số đầu vào như phương thức thông thường khác
- Phương thức khởi tạo được sử dụng để:
 - Tạo đối tượng
 - Khởi tạo các giá trị ban đầu cho các thuộc tính của lớp

Phương thức khởi tạo



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



Thảo luận

Các cách tạo đối tượng

Sơ đồ lớp Apple



Apple

```
- type: string
- color: string

+ Apple(type)
+ getInfo()
+ setColor(color)
+ setType(type)
```

Trong sơ đồ lớp Apple gồm:

- 2 thuộc tính type kiểu chuỗi, color kiểu chuỗi.
- Phương thức khởi tạo Apple() có 1 tham số truyền vào
- Các phương thức:
 - getInfo() trả về thông tin của táo gồm type và color
 - setColor() thiết đặt màu khác cho táo
 - setType() thiết đặt loại khác cho táo

Sử dụng hàm để tạo đối tượng



```
//Khai báo lớp Apple với các thuộc tính type, color,  
//phương thức getAppleInfo()  
function Apple (type) {  
    this.type = type;  
    this.color = "red";  
    this.getInfo = getAppleInfo;  
}  
  
//Khai báo hàm getAppleInfo()  
//Khuyến nghị: đây là cách không nên dùng,  
//đọc tiếp sẽ rõ  
function getAppleInfo() {  
    return this.color + ' ' + this.type + ' apple';  
}
```

Sử dụng từ khóa new để tạo đối tượng:

```
var apple = new Apple('macintosh');
```

Truy cập thuộc tính & phương thức:

```
apple.color = "reddish";  
alert(apple.getInfo());
```

Sử dụng hàm bên trong



```
function Apple (type) {  
    this.type = type;  
    this.color = "red";  
    this.getInfo = function() {  
        return this.color + ' ' + this.type + ' apple';  
    };  
}
```

Sử dụng từ khóa new để tạo đối tượng:

```
var apple = new Apple('macintosh');
```

Truy cập thuộc tính & phương thức:

```
apple.color = "reddish";
```

```
alert(apple.getInfo());
```

Sử dụng object literals



```
var apple = {  
  type: "macintosh",  
  color: "red",  
  getInfo: function () {  
    return this.color + ' ' + this.type + ' apple';  
  }  
}
```

Truy cập thuộc tính & phương thức:

```
apple.color = "reddish";  
alert(apple.getInfo());
```

Sử dụng Singleton



```
var apple = new function() {  
    this.type = "macintosh";  
    this.color = "red";  
    this.getInfo = function () {  
        return this.color + ' ' + this.type + ' apple';  
    };  
}
```

Truy cập thuộc tính & phương thức:

```
apple.color = "reddish";
```

```
alert(apple.getInfo());
```



Demo

Các tạo lớp Apple theo 3 cách



Thảo luận

Các đối tượng có sẵn trong JavaScript

Các đối tượng có sẵn



- Một số đối tượng có sẵn cơ bản:
 - Date: lưu trữ thông tin về thời gian của hệ thống hoặc do người dùng tạo
 - String: lưu chuỗi, chứa các hàm làm việc với chuỗi như cắt chuỗi, lấy từng ký tự trong chuỗi
 - Math: chứa các hàm và thuộc tính để thực hiện tính toán số học như tính bình phương, tính căn bậc hai ...

Đối tượng Date



- Đối tượng Date chứa thông tin về thời gian của hệ thống.
- Khởi tạo đối tượng Date:

```
new Date()  
new Date(milliseconds)  
new Date(dateString)  
new Date(year, month, day, hours, minutes, seconds, milliseconds)
```

- Ví dụ:

```
var d = new Date();  
document.getElementById("demo").innerHTML = d;  
  
var d = new Date("October 13, 2014 11:13:00");  
document.getElementById("demo").innerHTML = d;
```

Phương thức của đối tượng Date



Phương thức	Mô tả
getDate()	Lấy về ngày trong tháng (1-31)
getDay()	Lấy về ngày trong tuần (0-6)
getFullYear()	Lấy về năm (yyyy)
getHours()	Lấy về giờ (0-23)
getMilliseconds()	Lấy về milisecond giây (0-999)
getMinutes()	Lấy về phút (0-59)
getMonth()	Lấy về tháng trong năm (0-11)
getSeconds()	Lấy về giây (0-59)
getTime()	Lấy về thời gian (milliseconds since January 1, 1970)

Đối tượng Math



- Đối tượng Math gồm các phương thức thực hiện phép tính số học.
- Các phương thức thông dụng của đối tượng Math:
 - `abs()`, `ceil()`, `floor()`, `max()`, `min()`, `pow()`, `random()`, `round()`, `sqrt()`
- Các hằng số thông dụng của Math
 - `PI`
 - `E`
- Truy xuất thuộc tính và phương thức:
 - `Math.PI`
 - `Math.E`
 - `Math.abs(-4)`
 - `Math.ceil(4.5)`



Đối tượng String

- Đối tượng String lưu trữ một tập các ký tự

- Tạo đối tượng String

```
var str = "text data";
```

```
var str = new String("text data");
```

- Thuộc tính length trả về độ dài chuỗi

```
var str = "Hello World!";
```

```
var n = str.length;
```

```
document.getElementById("demo").innerHTML = n;
```

Kết quả: 12

Phương thức của đối tượng String



Phương thức	Mô tả
charAt()	Lấy về ký tự thứ i trong chuỗi
concat()	Nối chuỗi
startsWith()	Lấy về ký tự đầu tiên
endsWith()	Lấy về ký tự cuối cùng
indexOf()	Trả về vị trí chuỗi con trong chuỗi đã cho
substring()	Lấy chuỗi con
toLowerCase()	Chuyển chuỗi thành chữ in thường
toUpperCase()	Chuyển chuỗi thành chữ in hoa
split(pattern)	Cắt chuỗi thành các chuỗi nhỏ theo pattern



Demo

Đối tượng Date

Đối tượng Math

Đối tượng String

Tóm tắt bài học



- Lớp
- Đối tượng



Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: *Chuỗi*