



JAVA CORE #2

Ngọc Eri

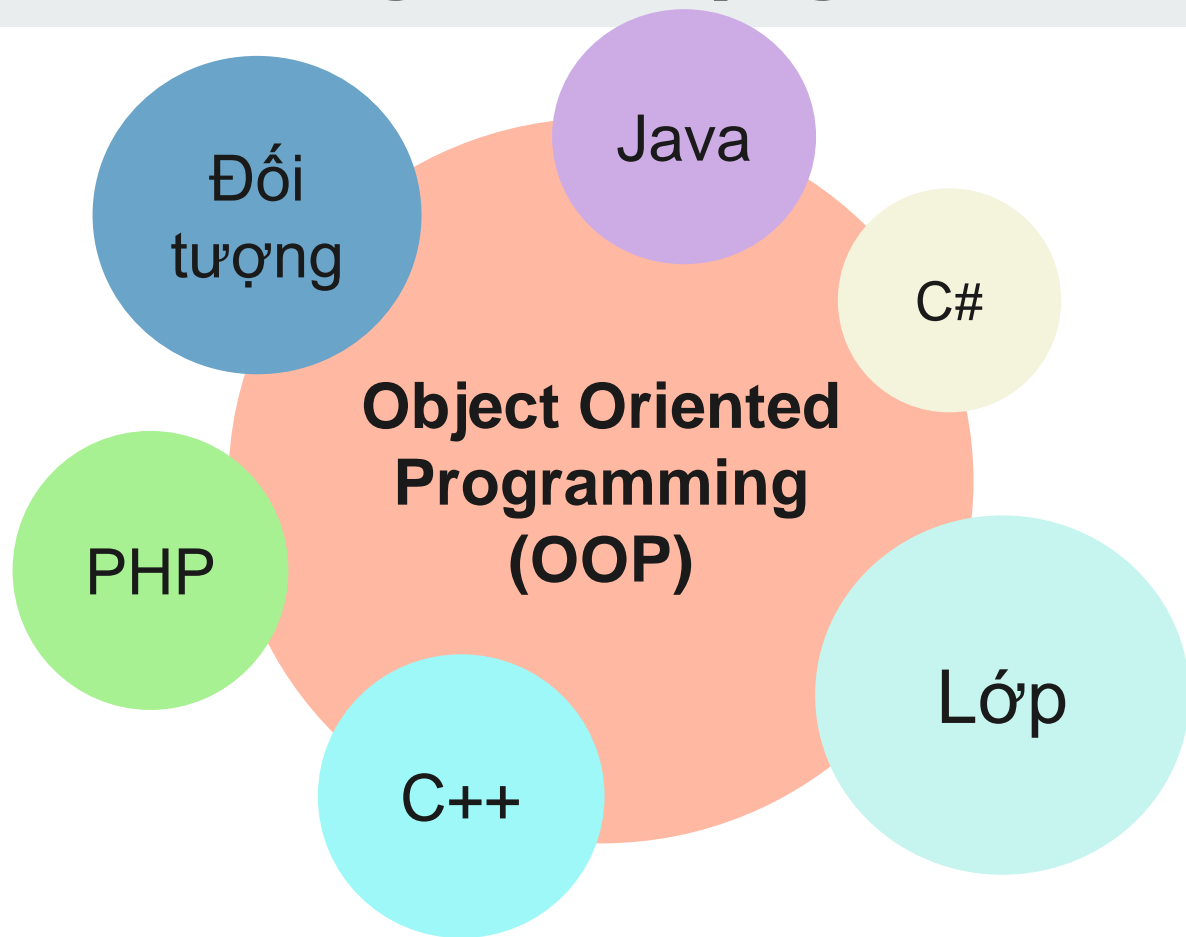
Lập trình hướng đối tượng - OOP

Lập trình tuần tự
(Ban đầu)

Lập trình có cấu trúc
(Năm 70-80)

Lập trình hướng
đối tượng (Năm
90 đến nay)

Lập trình hướng đối tượng - OOP



Lập trình hướng đối tượng - OOP

Khi viết chương trình theo phương pháp hướng đối tượng ta phải trả lời các câu hỏi:

- Chương trình liên quan tới những lớp đối tượng nào?
- Mỗi đối tượng cần có những dữ liệu và thao tác nào?
- Các đối tượng quan hệ với nhau như thế nào trong chương trình?

Các nguyên lý cơ bản của OOP

Tính đóng gói

Tính đa hình

Các nguyên lý cơ bản

Tính thừa kế

Tính trừu tượng

Ưu điểm của OOP

- Tính đóng gói làm giới hạn phạm vi sử dụng của các biến, nhờ đó việc quản lý giá trị của biến dễ dàng hơn, việc sử dụng mã an toàn hơn.
- Phương pháp này làm cho tốc độ phát triển các chương trình mới nhanh hơn vì mã được tái sử dụng và cải tiến dễ dàng, uyển chuyển.
- Phương pháp này tiến hành tiến trình phân tích, thiết kế chương trình thông qua việc xây dựng các đối tượng có sự tương hợp với các đối tượng thực tế. Điều này làm cho việc sửa đổi dễ dàng hơn khi cần thay đổi chương trình.

Nhược điểm của OOP

- Các chương trình hướng đối tượng có xu hướng chậm hơn và sử dụng nhiều bộ nhớ
- Quá khái quát
- Các chương trình được xây dựng theo mô hình này có thể mất nhiều thời gian hơn

Các mô hình lập trình

Mô hình lập trình

```
graph LR; A[Mô hình lập trình] --- B[Lập trình mệnh lệnh]; A --- C[Lập trình khai báo]; B --- D[Lập trình hướng thủ tục]; B --- E[Lập trình hướng đối tượng]; C --- F[Lập trình hướng chức năng]; C --- G[Lập trình logic]; C --- H[Lập trình tính toán];
```

Lập trình mệnh lệnh

Lập trình hướng thủ tục

Lập trình hướng đối tượng

Lập trình khai báo

Lập trình hướng chức năng

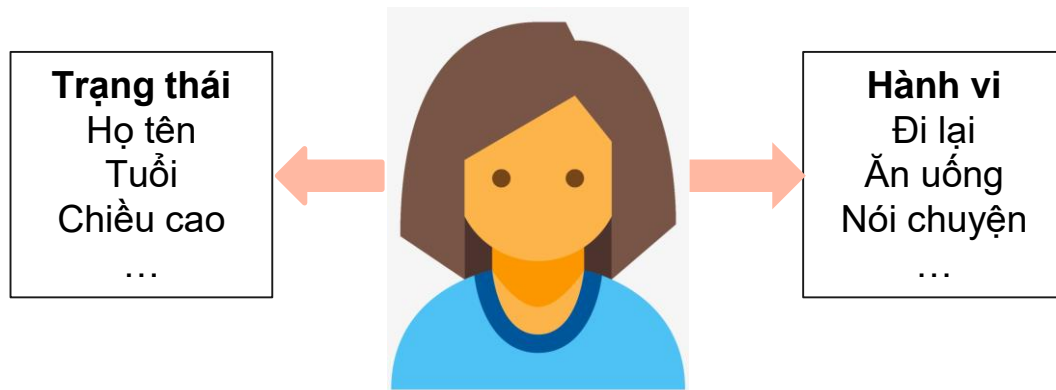
Lập trình logic

Lập trình tính toán

Đối tượng (Object)

Đối tượng là khái niệm cơ sở, quan trọng nhất của cách tiếp cận hướng đối tượng.

Đối tượng là thực thể của hệ thống, của CSDL và được xác định thông qua định danh ID của chúng



Lớp (Class)

Lớp là tập hợp các đối tượng có cùng trạng thái, hành vi hay là một nhóm các đối tượng có các thuộc tính chung.

Một lớp trong java có thể chứa:

- Thành viên dữ liệu
- Constructor
- Phương thức
- Khối lệnh
- Lớp và interface

So sánh đối tượng và lớp

Đối tượng	Lớp
Đối tượng là một thực thể trong thế giới thực: Con người, xe cộ, nhà cửa,...	Lớp bao gồm một nhóm các đối tượng tương tự nhau
Mang tính vật lý	Mang tính logic
Có thể được tạo nhiều lần	Được khai báo một lần duy nhất
Đối tượng được cấp bộ nhớ khi nó được tạo ra	Lớp không được cấp bộ nhớ khi nó được tạo ra
Có nhiều cách để tạo ra đối tượng trong java	Chỉ có một cách để định nghĩa lớp trong java là sử dụng từ khoá class

Tạo lớp

Cú pháp tạo lớp:

```
<Phạm vi truy cập> class <Tên lớp> {  
    <Thành viên truy cập>;  
    <Phương thức>;  
}
```

Tạo lớp

Cú pháp khai báo thuộc tính:

```
<Phạm vi truy cập> <Kiểu dữ liệu> <Tên thuộc tính>;
```

Cú pháp khai báo phương thức:

```
<Phạm vi truy cập> <Kiểu dữ liệu> <Tên phương thức>(){  
    //Code  
}
```

Tạo đối tượng

Tạo đối tượng bằng cách sử dụng từ khóa new, cú pháp:

```
<Tên lớp> <Tên biến tham chiếu> = new <Tên lớp>();
```

Cú pháp cung cấp thông tin cho thuộc tính:

```
<Tên biến tham chiếu>.<Thuộc tính> = <Giá trị>;
```

Constructor

Constructor là dạng đặc biệt của phương thức, được sử dụng để khởi tạo đối tượng.

Constructor được gọi ngay tại thời điểm tạo ra đối tượng

Đặc điểm của constructor

- Tên constructor giống tên class chứa nó
- Constructor không có kiểu trả về.

Nếu trong class không khai báo constructor thì trình biên dịch sẽ tự động tạo ra 1 constructor mặc định

Có 2 loại constructor:

- Constructor mặc định: Là constructor không có tham số, nhằm mục đích cung cấp các giá trị mặc định cho các đối tượng như 0, null,.. Tùy thuộc vào kiểu dữ liệu
- Constructor có tham số: Được sử dụng để cung cấp các giá trị khác nhau cho các đối tượng riêng biệt

Constructor

Cú pháp tạo constructor:

```
<Phạm vi truy cập> <Tên lớp>(<Các tham số>){  
    //code  
}
```