

## Bài 6

# LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA

Company: DEVPRO VIỆT NAM

Website: [devpro.edu.vn](http://devpro.edu.vn)

Design by Minh An

1

## Nội dung

- Lập trình hướng đối tượng
- Trừu tượng hóa dữ liệu
- Khái niệm lớp, đối tượng
- Phương thức khởi tạo
- Biến đối tượng (thể hiện – Instance Variable)
- Các phương thức (Method)

Design by Minh An

2

## 1. Lập trình hướng đối tượng

- Object Oriented Programming - OOP
- Được phát triển dựa trên lập trình cấu trúc.
- Là phương pháp thiết kế và phát triển ứng dụng dựa trên kiến trúc lớp (**class**) và đối tượng (**object**).
- Chương trình được chia thành các đối tượng (đóng gói).

**Đối tượng = Phương thức + Dữ liệu**

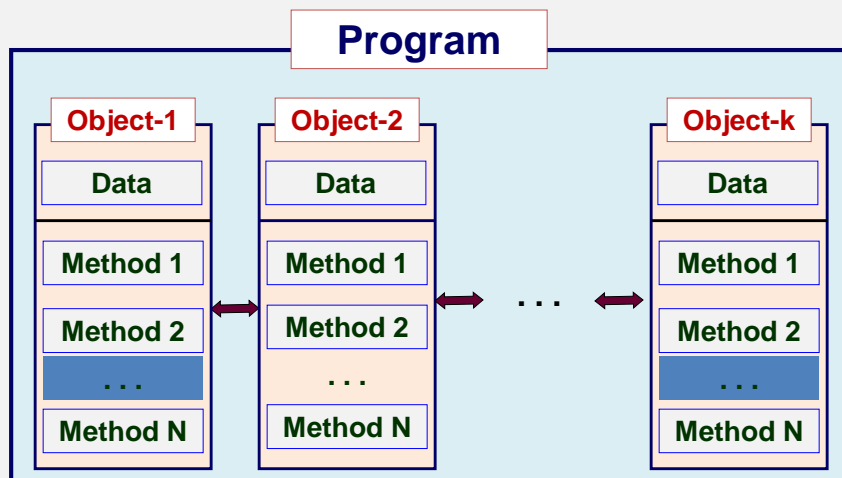
- Cho phép liên kết cấu trúc dữ liệu với các thao tác.
- Các đối tượng trao đổi dữ liệu với nhau bằng thông điệp.
- Khi phát triển ứng dụng lấy đối tượng của bài toán làm trung tâm.

Design by Minh An

3

## Lập trình hướng đối tượng (tt)

**Object = data + methods**



Design by Minh An

4

## Lập trình hướng đối tượng (tt)

- Các tính chất của lập trình hướng đối tượng
  - Tính trừu tượng – Abstraction
  - Tính đóng gói – Encapsulation
  - Tính kế thừa – Inheritance
  - Tính đa hình – polymorphism

Design by Minh An

5

## Tính trừu tượng

- Từ những đối tượng giống nhau, chúng ta có thể trừu tượng hóa thành một lớp đối tượng.
- Tính trừu tượng cho phép chúng ta loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.

Translated from APTECH WORLD  
WUDE book by Duong Thanh

Design by Minh An

Slide 3/6 of 17

6

## Tính đóng gói

- Tất cả các dữ liệu dùng để mô tả đối tượng cùng với các phương thức thao tác trên dữ liệu đó được đặt trong một khối (còn gọi là lớp).
- Với mỗi đối tượng người ta không thể truy nhập trực tiếp vào các thành phần dữ liệu của nó mà phải thông qua các thành phần chức năng (còn gọi là các phương thức) của chính đối tượng để làm việc đó.
- Tính đóng gói là tính chất đảm bảo sự toàn vẹn của đối tượng.

Design by Minh An

7

## Tính kế thừa

- Đặc tính này cho phép xây dựng một lớp mới có thể có sẵn các đặc tính mà các lớp khác đã có thông qua kế thừa.
- Điều này cho phép các lớp chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
- Tuy nhiên, không phải ngôn ngữ lập trình hướng đối tượng nào cũng có tính chất này.

Design by Minh An

8

## Tính đa hình

- Tính đa hình xuất hiện khi có khái niệm kế thừa.
- Hành vi cụ thể của một đối tượng chỉ được xác định trong ngữ cảnh cụ thể.
- Nếu có một lớp **đa giác** và hai lớp kế thừa từ lớp **đa giác** là lớp **tam giác** và lớp **tứ giác** (tam giác và tứ giác sẽ có đầy đủ các thuộc tính và tính chất của một đa giác).
  - Lúc này một đối tượng thuộc lớp hình **tam giác** hay **tứ giác** đều có thể hiểu rằng nó là một hình **đa giác**.
  - Mỗi **đa giác** ta có thể tính diện tích, chu vi của nó. Như vậy làm thế nào mà một **đa giác** có thể sử dụng đúng công thức để tính diện tích phù hợp với nó là hình **tam giác** hay **tứ giác**. Ta gọi đó là tính đa hình.

Design by Minh An

9

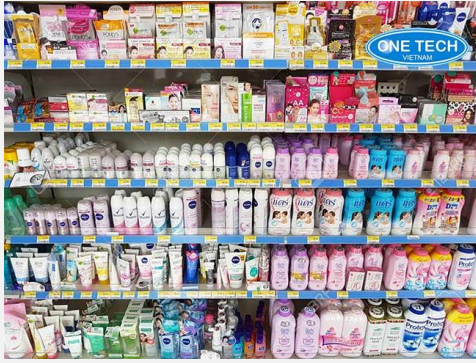
## 2. Trừu tượng hóa dữ liệu

- Xem xét khái quát cấu trúc và hành vi của các đối tượng mà không quan tâm đến việc biểu diễn cụ thể chúng.
- Thực hiện mô hình hóa các đối tượng.
- Những ưu điểm của việc Trừu tượng hóa là:
  - Tập trung vào vấn đề cần giải quyết.
  - Xác định những đặc tính thiết yếu và những hành động cần thiết.
  - Giảm thiểu những chi tiết không cần thiết, hoặc những giải thích.

Design by Minh An

10

## Trừu tượng hóa dữ liệu (tt)



- ✓ Tên sản phẩm
- ✓ Số lượng
- ✓ Đơn giá
- ✓ Hạn sử dụng
- ✓ ...

- Nhập thông tin SP
- Xem chi tiết SP
- Sửa thông tin SP
- Xóa thông tin SP
- ...

Design by Minh An

11

## 3. Đối tượng - Object

- Một người nào đó, một cây nào đó, một gia đình nào đó, một sinh viên nào đó, một phương trình bậc hai nào đó, một tam giác nào đó, một số nào đó, một chuỗi nào đó, ... chưa xác định cụ thể gọi là một đối tượng.
- Mỗi đối tượng có một số đặc trưng (thuộc tính) và một số hành vi (phương thức) để nhận biết đối tượng và phân biệt với các đối tượng khác.
- Ví dụ:
  - Một người: có các thuộc tính họ tên, tuổi, chiều cao, cân nặng, giới tính, dấu vết riêng và các phương thức ăn, ngủ, học, đi lại.
  - Một phương trình bậc hai: có các thuộc tính là ba hệ số và các phương thức nhận vào giá trị các hệ số, giải phương trình.
  - ...

Design by Minh An

12

## 4. Lớp - Class

- Một tập các đối tượng có cùng các thuộc tính và các phương thức tạo thành một lớp.
- Trong chương trình:
  - Lớp (được xem như) là một kiểu dữ liệu.
  - Đối tượng (được xem như) là một biến được gọi là một thể hiện – instance variable.
- Khai báo lớp

```
[modifier] class ClassName {  
    //Các thuộc tính  
    //Các phương thức  
}
```

Design by Minh An

13

## Lớp

- Ví dụ:

```
public class Person {  
    //Các thuộc tính  
    private String name;  
    private int age;  
    private String gender;  
    //Các phương thức  
    public void fillInfo();  
    public void display();  
    public int getAge();  
}
```

- Chương trình cho người một người điền thông tin của mình vào máy tính và xem lại thông tin vừa điền.

Design by Minh An

14

## 5. Modifier

- Chỉ định phạm vi của mã lệnh trong chương trình được phép truy xuất đến các thuộc tính, phương thức của lớp.

modifier	Trong class	Trong package	Ngoài package bởi class con	Ngoài package không thuộc class con
private	Yes			
Mặc định (không viết)	Yes	Yes		
protected	Yes	Yes	Yes	
public	Yes	Yes	Yes	Yes

Design by Minh An

15

## 6. Phương thức khởi tạo - Constructor

- Constructors là các phương thức của lớp để tạo ra các đối tượng (instances) trong chương trình và khởi tạo giá trị ban đầu cho các thuộc tính của đối tượng.
- Tên Constructors phải trùng với tên lớp.
- Constructor có thể có hoặc không có đối.
- Constructors không có giá trị trả về và cũng không có kiểu trả về.
- Constructor không đối là constructor mặc định, được cung cấp tự động bởi trình biên dịch nếu như trong lớp không định nghĩa constructor.

Design by Minh An

16



## Phương thức khởi tạo - Constructor

```
public class Car {  
    private int id;  
    private String brand;  
    private double price;  
    private String manufacturer;  
    public Car() {  
        this.id = 1; this.brand = "vios";  
        this.price = 25000;  
        this.manufacturer = "toyota";  
    }  
    public Car(int id, String brand,  
        double price, String manufacturer) {  
        this.id = id; this.brand = brand;  
        this.price = price;  
        this.manufacturer = manufacturer;  
    }  
}
```

Design by Minh An

17

## Tạo các đối tượng trong chương trình

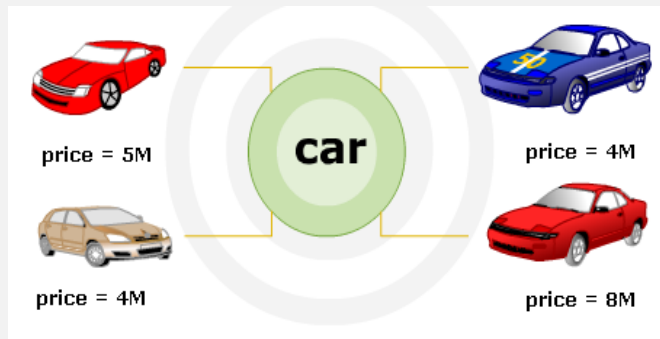
- Để tạo đối tượng, ta dùng từ khóa **new**.
- Ví dụ:  
  
`Car toyCar = new Car();`  
`Car mazCar = new Car(2, "CX5", 3100, "Mazda");`
- JVM (máy ảo java) triệu gọi constructor của lớp.
- Một tham chiếu đến bộ nhớ được cấp phát cho đối tượng.

Design by Minh An

18

## 7. Biến đối tượng - instance variable

- Instance variables hay object variables được sử dụng để lưu trữ thông tin về một thực thể (entity).
- instance variable được khai báo và sử dụng để lưu các thể hiện (instances) hay đối tượng (object) của lớp.



Design by Minh An

19

## Truy xuất các thành phần của instance

Cách viết:

```
instanceName.memberName cho thuộc tính  
instanceName.memberName(...) cho phương thức
```

Ví dụ:

```
Person p = new Person();  
System.out.println("Name: " + p.name);  
System.out.println("Age: " + p.age);  
p.fillInfo();  
p.display();
```

Design by Minh An

20

## 8. Các setter() và getter()

- **setter()** : là phương thức thiết lập giá trị mới cho thuộc tính của đối tượng.
- **getter()** : là phương thức để lấy giá trị từ thuộc tính của đối tượng.

**Ví dụ:**

```
public String getName() {  
    return name;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
//...  
  
p.setName("David Dũng");  
System.out.println("New name: "+ p.getName());
```

Design by Minh An

21

## Bài tập

1. Xây dựng chương trình cho chủ cửa hàng xe hơi thực hiện được các thao tác sau:
  - Tạo một xe hơi gồm các thông tin: mã xe, nhãn hiệu, hãng sản xuất, kiểu dáng, giá tiền, xuất xứ.
  - Hiển thị thông tin về xe vừa tạo.
  - Sửa các thông tin của xe gồm: nhãn hiệu, hãng sản xuất, kiểu dáng, giá tiền, xuất xứ.
  - Chương trình có menu để chọn các chức năng.

Design by Minh An

22

## Bài tập

2. Xây dựng chương trình cho người quản sinh thực hiện được các thao tác sau:
  - Tạo một sinh viên gồm các thông tin: mã sinh viên, họ đệm, tên, năm sinh, giới tính, danh sách n học phần mà sinh viên đã học (mỗi học phần gồm mã học phần, tên học phần, số tín chỉ, điểm kết thúc học phần).
  - Hiển thị thông tin về sinh viên vừa tạo cùng với điểm tổng kết, biết điểm tổng kết được tính theo điểm kết thúc học phần và số tín chỉ của học phần.
  - Sửa các thông tin của sinh viên gồm: họ đệm, tên, năm sinh, giới tính.
  - Tìm và sửa điểm học phần theo mã học phần.

Design by Minh An

23

## Bài tập

### 3. Cho bảng điểm cá nhân như hình dưới đây

BẢNG ĐIỂM CÁ NHÂN				
Mã sinh viên: 1001222		Họ và tên: Nguyễn Văn B		
Ngày sinh: 21/08/1998		Ngành: Khoa học máy tính		
STT	Mã học phần	Tên học phần	Số tín chỉ	Điểm học phần
1	0503127	Lập trình hướng đối tượng	3	7.5
2	0503130	Kỹ thuật lập trình	4	6.6
3	0503174	Cơ sở dữ liệu	3	8.2
...				

Điểm tổng kết: (Là điểm TB chung HP theo tín chỉ)

Design by Minh An

24

## Bài tập

Viết chương trình thực hiện các yêu cầu sau:

- Khởi tạo thông tin đầy đủ cho bảng điểm (5 – 10 học phần).
- Hiển thị bảng điểm ra màn hình cùng với điểm tổng kết, sao cho danh sách các học phần theo thứ tự từ điển.
- Hiển thị ra màn hình danh sách các học phần có điểm dưới 6.0.
- Chương trình cho phép chọn và sửa thông tin về sinh viên gồm họ tên, ngày tháng năm sinh, ngành.
- Chương trình cho phép chọn và sửa thông tin về học phần gồm tên học phần, số tín chỉ, điểm học phần, tính lại điểm tổng kết nếu sửa điểm học phần hoặc số tín chỉ.

Design by Minh An

25

## Bài tập

4. Cài đặt lại chương trình quản lý tiền điện ở day04 theo class & object.
5. Cài đặt lại chương trình quản lý vận động viên ở day04 theo class & object.

Design by Minh An

26