

# Bài 16: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA

Xem bài học trên website để ủng hộ Kteam: [Lập trình hướng đối tượng trong Java](#)

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

Như vậy, Kteam đã hướng dẫn cho các bạn những kiến thức cơ bản trong lập trình. Bây giờ, Kteam sẽ chuyển sang phần quan trọng nhất là **lập trình hướng đối tượng**, đây là yêu cầu quan trọng đối với lập trình viên hiện giờ. Kteam sẽ giải thích cho các bạn tốt nhất

---

## Nội dung

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

- [CÁC BIẾN TRONG JAVA.](#)
- [CÁC KIỂU DỮ LIỆU TRONG JAVA.](#)
- [CÁC HẠNG TOÁN TỬ TRONG JAVA](#)
- [CẤU TRÚC Rẽ NHÁNH TRONG JAVA](#)
- [VÒNG LẶP WHILE TRONG JAVA](#)
- [VÒNG LẶP FOR TRONG JAVA](#)
- [MẢNG TRONG JAVA](#)
- [VÒNG LẶP FOR-EACH TRONG JAVA](#)
- [VAI TRÒ BREAK, CONTINUE TRONG VÒNG LẶP JAVA](#)
- [SWITCH TRONG JAVA](#)

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Lập trình hướng đối tượng là gì?

- Những khái niệm cơ bản của lập trình hướng đối tượng
- Hướng đối tượng trong Java

## Lập trình hướng đối tượng là gì?

Trong những bài viết trước đây, chúng ta lập trình bằng cách viết toàn bộ code vào chương trình **main** duy nhất và chạy, và ta tự định nghĩa các biến theo suy nghĩ cá nhân. Ví dụ như bài học trước đây, ta tự định nghĩa các biến lưu thông tin con người như sau:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        String name;  
        float height;  
        int age;  
        name = "Chau Kter";  
        height = 1.7f;  
        age = 21;  
        System.out.println(name);  
        System.out.println(height);  
        System.out.println(age);  
    }  
}
```

Như trên, việc định nghĩa biến `name`, `height` và `age` là do cá nhân người viết lúc đó tự định nghĩa ra. Về cốt lõi chương trình, khó mà nhận biết được mối liên quan các biến trên, giả sử trong 1 chương trình có lưu thông tin đến 3-4 người, như vậy ta phải tăng số lượng biến lưu thông tin gấp 3-4 lần, khó mà đảm bảo được biến nào lưu thông tin cho đối tượng nào. Chưa kể, trong một dự án với nhiều người, việc tự định nghĩa lập trình theo cá nhân sẽ ảnh hưởng đến cách làm việc của tập thể.

Việc lập trình như vậy ta cần phải gom lại trực quan hơn, để mô tả trung thực hệ thống. Ta sẽ quy lại thành một đối tượng, như ví dụ trên: Việc các 3 biến **name**, **height**, **age** là lưu thông tin của một người, ta sẽ tạo ra một đối tượng là con người và trong đối tượng đó sẽ có thông tin 3 biến trên.

# Những khái niệm cơ bản của lập trình hướng đối tượng

## Đối tượng (Object)

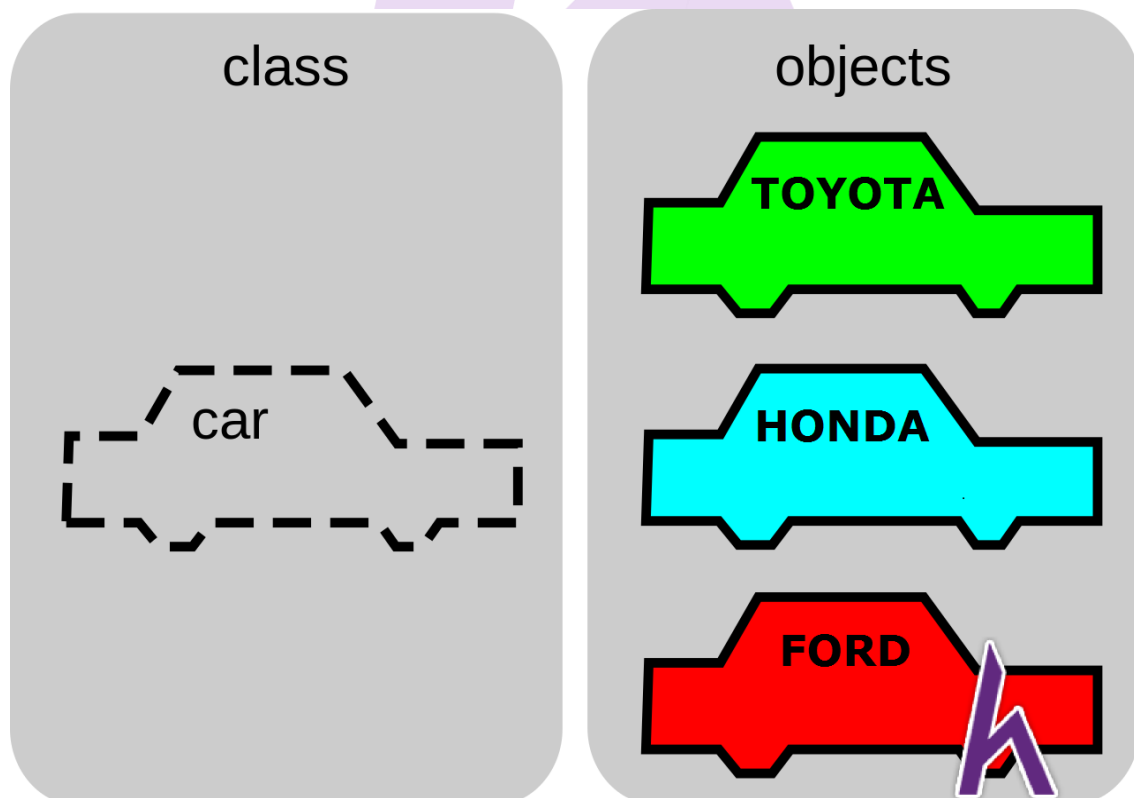
Đối tượng ở đây ta thể hiểu như khái niệm bên ngoài: Con người, Xe máy, Nhà cửa...

Trong một đối tượng sẽ bao gồm 2 thông tin: **thuộc tính** và **phương thức**.

- **Thuộc tính:** là những thông tin của đối tượng. Ví dụ: con người có họ tên, chiều cao, độ tuổi,...
- **Phương thức:** là những thao tác, hành động mà đối tượng đó có thể thực hiện. Ví dụ: con người có những hành động ăn, ngủ, đi lại,...

## Lớp (Class)

Lớp chính là định nghĩa của đối tượng, ta sẽ xây dựng lớp để tạo ra những đối tượng khác nhau. Ví dụ như: Bạn **Nguyễn Văn A** và **Lê Văn B** đều là **con người**, mà con người thì đều có tên, tuổi, chiều cao,.. tuy nhiên thông tin lại khác nhau như ngoài tên, bạn A 20 tuổi còn bạn B 22 tuổi. Như vậy **con người** chính là **lớp**, **Nguyễn Văn A** và **Lê Văn B** là **đối tượng**.

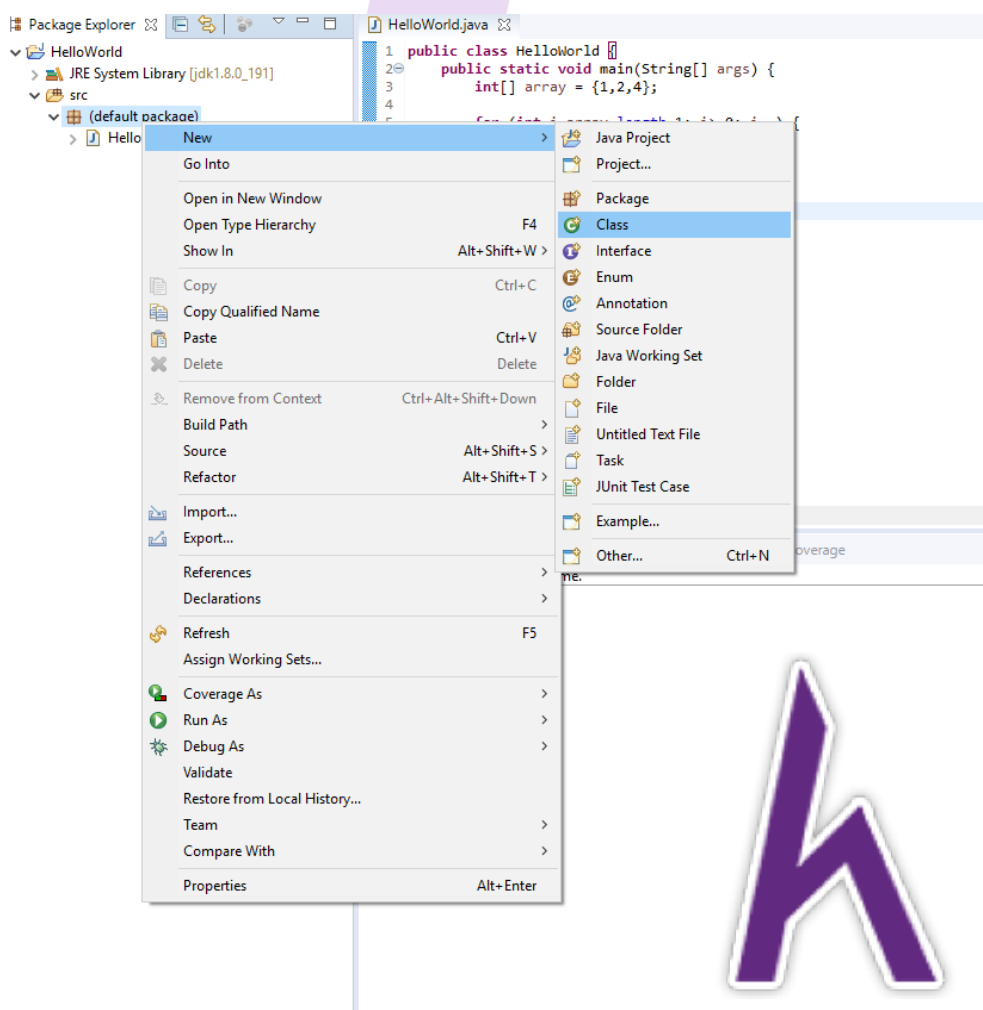


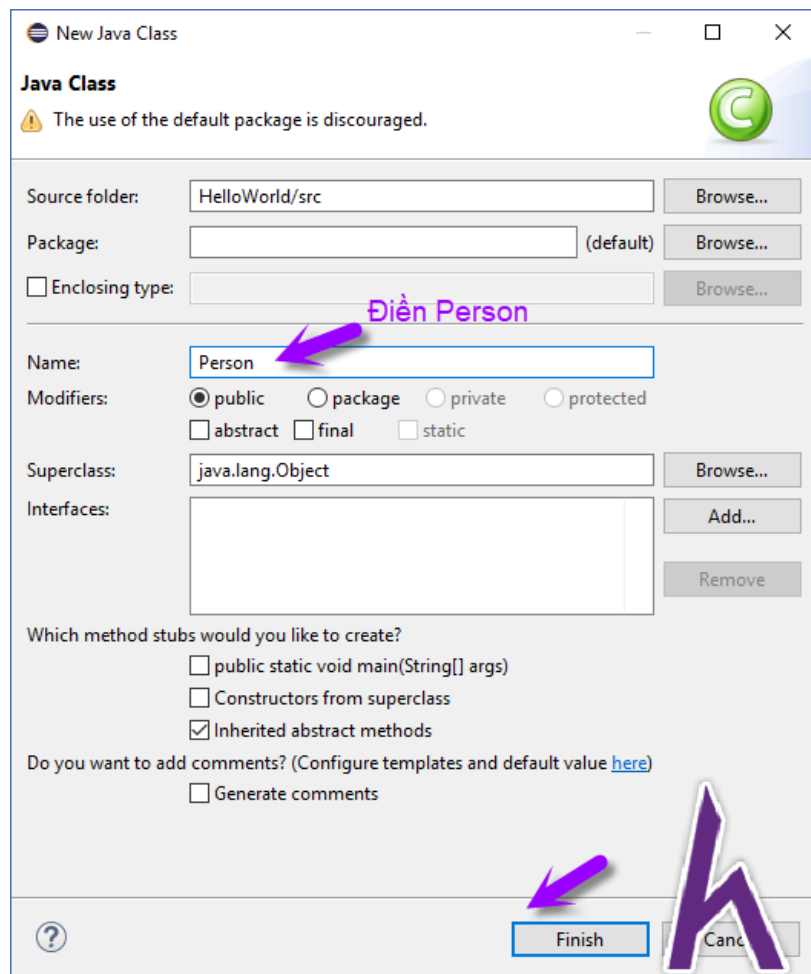
## Hướng đối tượng trong Java

Bản chất Java là ngôn ngữ thuần hướng đối tượng, vì vậy đây là ngôn ngữ bậc cao nên việc học lập trình ngay từ đầu bạn sẽ thấy khó hiểu với những từ khóa class, new,... Đó là lý do Kteam không khuyến khích những bạn mới bắt đầu học lập trình lựa chọn **Java**.

Bây giờ, Kteam sẽ ví dụ qua cách chuyển đoạn code trên thành đoạn code hướng đối tượng. Các bài sau sẽ giải thích sâu hơn:

Đầu tiên ta sẽ khai báo một lớp là con người, trong con người có những thuộc tính là tên, chiều cao và tuổi: Ta sẽ tạo một file class **.java** lưu riêng, nếu bạn đang dùng Eclipse hãy dùng cách sau:





Hoặc bạn có thể tạo file class **Person.java** theo cách truyền thống. Sau đó, khai báo các thuộc tính lớp Person như sau:

```
public class Person {
    String name;
    int age;
    float height;
}
```

Tiếp theo, ta sẽ tạo một đối tượng từ lớp con người và cung cấp thông tin cho nó: Ta sẽ dùng cú pháp **Person a = new Person();** có nghĩa là tạo đối tượng a thuộc lớp con người.

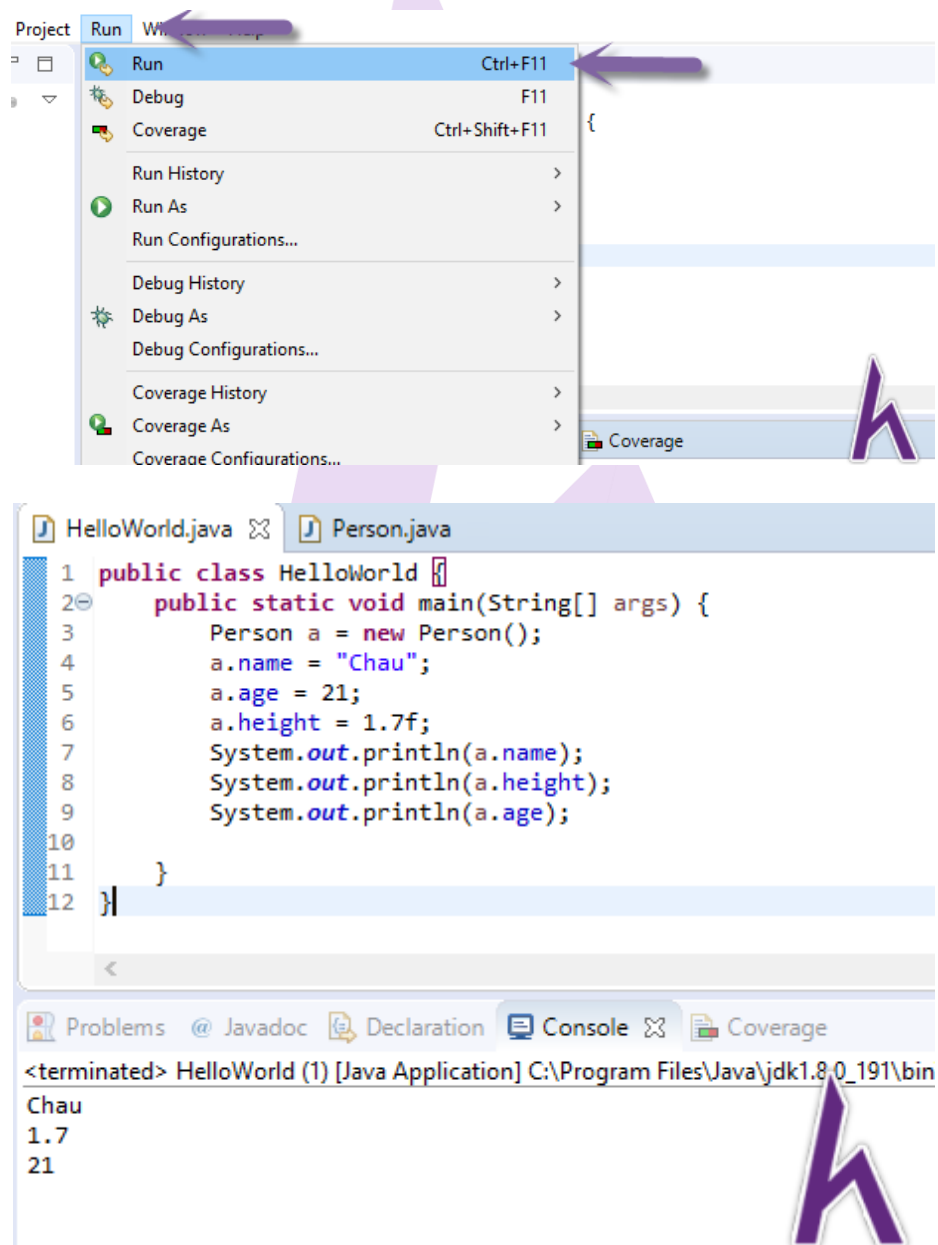
Ta sẽ cung cấp thông tin cho các thuộc tính bằng cú pháp:

**<đối tượng>.<thuộc tính> = <giá trị>;**

```
public class HelloWorld {
    public static void main(String[] args) {
```

```
    Person a = new Person();  
    a.name = "Chau";  
    a.age = 21;  
    a.height = 1.7f;  
    System.out.println(a.name);  
System.out.println(a.height);  
System.out.println(a.age);  
}
```

Ta sẽ chạy chương trình bằng hỗ trợ Eclipse:



Hoặc với cách truyền thống:

```
cmd C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.472]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Windows 10\eclipse-workspace\HelloWorld\src>javac HelloWorld.java Person.java

C:\Users\Windows 10\eclipse-workspace\HelloWorld\src>java HelloWorld
Chau
1.7
21
```

Như vậy, với cách này ta dễ dàng quản lý chương trình hơn vì ta biết rõ thông tin nào thuộc đối tượng nào. Như bạn đối tượng **a** có tên gì, chiều cao và độ tuổi bao nhiêu; và giả sử có đối tượng **b** thì khó nhầm lẫn thông tin với đối tượng **a** được.

---

## Kết

Như vậy chúng ta đã tìm hiểu lập trình hướng đối tượng

Ở bài sau, Kteam sẽ giới thiệu đến bạn 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"**.