



[Home](#) » [Java Core](#) » Khai báo interface trong Java, hằng số với interface

## Khai báo interface trong Java, hằng số với interface

Posted on Tháng Một 2, 2016

Khai báo interface trong Java, hằng số với interface.

(Xem thêm: [Khai báo class trong Java](#), [Nguyên tắc khai báo class](#).)

(Xem thêm: [Sự khác nhau giữa Abstract class với Interface trong Java](#))

## Khai báo interface

Một interface là một source file được khai báo với từ khóa *interface* chứ không phải *class*

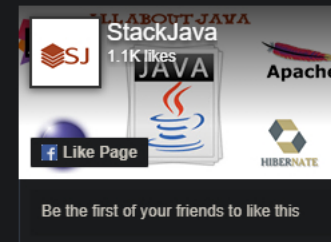
Khi bạn tạo một interface tức là bạn đang định nghĩa những gì mà class có thể làm mà không nói gì đến nó làm như thế nào.

Ví dụ bạn định nghĩa interface PeopleAction với các hành động như work(), study()... Class Student và Employee thực hiện cài đặt (implements) lại interface PeopleAction tức là 2 class Student và Employee có thể và bắt buộc cài đặt lại word() và study() còn cài đặt như nào thì mỗi class lại có cách cài đặt khác nhau.

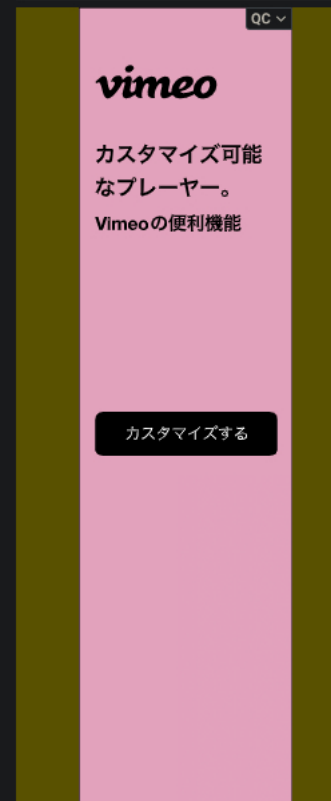
- Một interface có thể extends nhiều interface khác (không thể implements interface)
- Một class có thể implements nhiều interface.
- Không thể sử dụng từ khóa final khi khai báo interface (vì interface sinh ra để những thằng khác thừa kế)
- Tất cả các method trong interface đều được ngầm hiểu là public abstract (Từ [Java 8](#) cho phép thêm method không phải là abstract vào interface bằng từ khóa **default**; Từ Java 9 cho phép thêm method private và private static vào Interface)
- Các method trong interface không thể khai báo với từ khóa static
- Vì các method trong interface là abstract nên nó cũng không thể khai báo với từ khóa final, strictfp, native
- Tất cả các biến trong interface đều được ngầm hiểu là public static final.

```
public interface PeopleAction {  
    void work();  
    void study();  
}
```

Thứ bạn khai báo



Tìm kiếm ...



JAVA CORE

Biến trong Java là gì? Khai báo biến

```


public interface PeopleAction {
    public abstract void work();
    public abstract void study();
}

public class Student implements PeopleAction{
    @Override
    public void work() {
        // TODO
    }
    @Override
    public void study() {
        // TODO
    }
}

```

Thứ compiler hiểu

Tất cả các method của interface đều phải được cài đặt lại



## Khai báo hằng số trong interface

Bạn được phép khai báo các hằng số bên trong một interface. Bằng cách này bạn đảm bảo được rằng các class cùng thừa kế interface sẽ được truy cập vào cùng 1 hằng số.

Ví dụ:

```

public interface Common {
    int MAX_CONNECTION = 100; // tương đương với public static final int MAX_CONNECTION = 100;
}

```

Khi cần sử dụng hằng số `MAX_CONNECTION` ta chỉ cần gọi `Common.MAX_CONNECTION`

Việc khai báo hằng số trong interface lợi thế hơn việc khai báo hằng số trong class vì tất cả các biến trong interface đều được ngầm hiểu là `public static final`, ta có thể chắc chắn nó là hằng số và không bị thay đổi giá trị.

(Xem thêm: [default interface trong java 8](#))

Xem thêm: <https://stackoverflow.com/java-core>

References:

[SCJP pdf](#)

This entry was posted in *Java Core*. Bookmark the *permalink*.

Chức năng của hàm khởi tạo trong Java?

Khai báo hàm khởi tạo trong Java – Constructor Declarations

Class trong Java là gì? Object trong Java là gì?

Ngôn ngữ Java là gì? Đặc trưng của Java

Quy ước, chuẩn đặt tên định danh trong Java

Định danh trong Java, nguyên tắc đặt tên định danh

Các loại access modifier trong Java (public, protected, default, private)

Khai báo interface trong Java, hằng số với interface

Khai báo class trong Java, Nguyên tắc khai báo class

strictfp là gì, Từ khóa strictfp trong Java, ví dụ

SCJP, OCPJP là gì, tài liệu SCJP, OCPJP tiếng việt

Java 2 là gì? Các version/Phiên bản của Java

Phân biệt giữa Java ME, Java SE và Java EE

So sánh sự khác nhau giữa J2ME, J2SE và J2EE

## CHUYÊN MỤC

Algorithm

Apache

Apache JMeter

← Khai báo class trong Java, Nguyên tắc khai báo class

Các loại access modifier trong Java (public, protected, default, private) →

Apache Kafka

AWS

C/C++

CDI

Clean Code

Demo

Design Pattern

Docker

Eclipse

Elasticsearch

Excel

FAQ

Framework

Freemaker

FreeMarker

Gradle

Hibernate

HttpClient

HttpComponents

Install

Intellij IDEA

Java

Java Basic

Java Core

Java8

JavaScript

jooq

JSF

JSP-Servlet

JUnit

Library

Linux

Maven

MongoDB

MySQL

Network Programming

Node.js

OOP

PostgreSQL

PrimeFaces

Principle

Python

quartz

Redis

SDKMan

Security

SocketCluster

Spring

Spring Boot

Spring Core

Spring Data

Spring Hibernate

Spring JDBC

Spring MVC

Spring Security

Thymeleaf

Tomcat

Uncategorized

Web Service

WebSocket

Wordpress



stackjava.com

