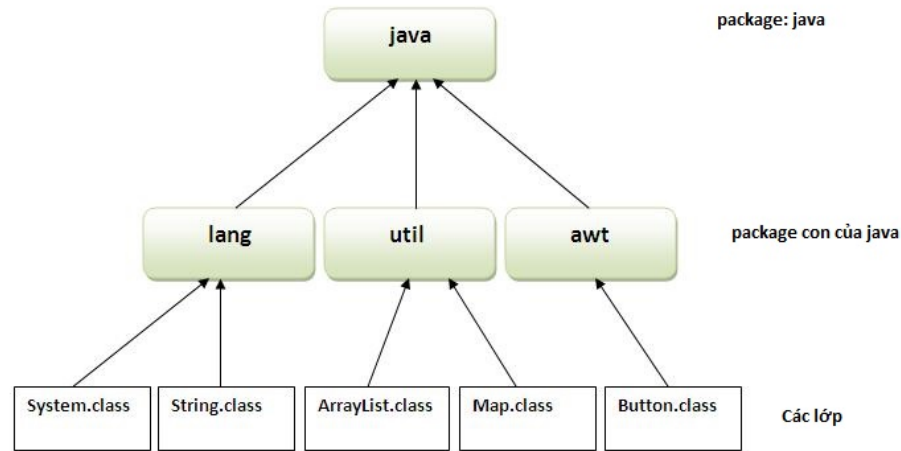


+ Mô Hình MVC

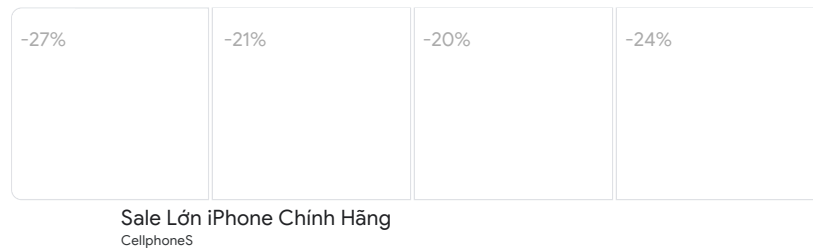
→ Mô hình MVC là gì? - Ví dụ login sử dụng mô hình MVC trong Java

+ Java Programs

- Tính giai thừa trong java
- Check số nguyên tố trong java
- Dãy số Fibonacci trong java
- Chuyển đổi hệ cơ số trong java
- Liệt kê n số nguyên tố đầu tiên trong Java

+ Java String**+ Xử Lý Ngoại Lệ****+ Các Lớp Lồng Nhau****+ Đa Luồng (Multithreading)****+ Java AWT****+ Java Swing****+ Java I/O****+ Ví Dụ Java I/O****+ Lập Trình Mạng Với Java****+ Java Date****+ Chuyển Đổi Kiểu Dữ Liệu****+ Java Collections****+ Java JDBC****+ Các Tính Năng Mới Trong Java****+ Java Math****+ Bài Tập Java Có Lời Giải****+ Câu Hỏi Phỏng Vấn Java**

Đã xóa quảng
tiết



Ví dụ về java package

Từ khóa **package** được sử dụng để tạo một package trong java

```

1 //Save file Simple.java
2 package mypack;
3 public class Simple {
4     public static void main(String args[]) {
5         System.out.println("Learn java package");
6     }
7 }
  
```

Biên dịch java package

Nếu bạn không sử dụng IDE, bạn cần thực hiện theo cú pháp dưới đây:

```
1 javac -d directory javafilename
```

Ví dụ:

```
1 javac -d . Simple.java
```

Lệnh **-d** được sử dụng để xác định nơi lưu trữ file .class sau khi biên dịch. Bạn có thể sử dụng bất kỳ tên thư mục nào như /home (Trong Linux OS), D:/temp (Trong Window OS). Nếu bạn muốn giữ các package này trong thư mục hiện tại bạn sử dụng dấu chấm (.).

Run java package

Để run java package, bạn cần phải sử dụng tên đầy đủ. Ví dụ mypack.Simple.

Compile: javac -d . Simple.java



Run: java mypack.Simple

Kết quả:

Learn java package

Dấu chấm (.) biểu diễn thư mục hiện tại.

Truy cập package từ package khác

Có 3 cách để truy cập package từ package bên ngoài:

- Khai báo import package.*;
- Khai báo import package.classname;
- Sử dụng tên đầy đủ.

1. Sử dụng packagename.*

Nếu bạn sử dụng packagename.*. Thì tất cả các lớp và các interface của các gói này sẽ có thể truy cập, nhưng gói con của gói này thì không được truy cập.

Từ khóa import được sử dụng để truy cập các lớp và interface của gói khác từ gói hiện tại.

Ví dụ:

```
1 package pack;
2 public class A {
3     public void msg() {
4         System.out.println("Hello");
5     }
6 }
```

```
1 package mypack;
2 import pack.*;
3
4 class B {
5     public static void main(String args[]) {
6         A obj = new A();
7         obj.msg();
8     }
9 }
```

Kết quả:

Hello

2. Sử dụng packagename.classname

Nếu bạn khai báo import package.classname thì chỉ được truy cập tới lớp đã được khai báo của package này.

Ví dụ:

```
1 package pack;
2 public class A {
3     public void msg() {
4         System.out.println("Hello");
5     }
6 }
```

```
1 package mypack;
2 import pack.A;
3
4 class B {
5     public static void main(String args[]) {
```

```
6      A obj = new A();
7      obj.msg();
8  }
9  }
```

Kết quả:

Hello

3. Sử dụng tên đầy đủ

Nếu bạn sử dụng tên đầy đủ thì chỉ được truy cập tới lớp đã được khai báo của package này. Bạn không cần phải sử dụng đến từ khóa import. Nhưng bạn cần phải sử dụng tên đầy đủ mỗi khi bạn truy cập vào các lớp hoặc interface.

Cách này thường được sử dụng khi 2 package có tên lớp giống nhau. Ví dụ, 2 package java.util và java.sql chứa lớp có tên giống nhau là lớp Date

Ví dụ:

```
1 package pack;
2 public class A {
3     public void msg() {
4         System.out.println("Hello");
5     }
6 }
```

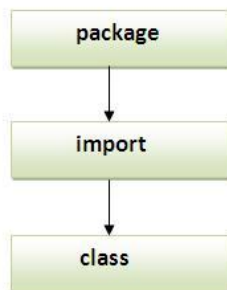
```
1 package mypack;
2 class B{
3     public static void main(String args[]){
4         pack.A obj = new pack.A(); //Sử dụng tên đầy đủ
5         obj.msg();
6     }
7 }
```

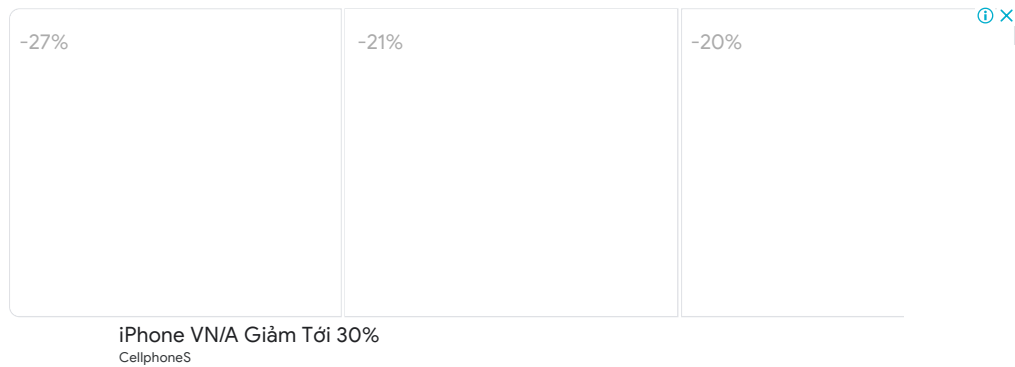
Kết quả:

Hello



Note: Nếu bạn import một package thì package con của package đó không được import.
Thứ tự của chương trình phải là package->import->class.





Package con trong java

Package bên trong một package khác được gọi là **subpackage** hay package con trong java.

Ví dụ, Sun Microsystems đã định nghĩa một gói có tên **java** chứa nhiều lớp như System, String, Reader, Writer, Socket, ... Các lớp này đại diện cho một nhóm cụ thể ví dụ như các lớp Reader và Writer cho các hoạt động Input/Output, Socket và ServerSocket các lớp xử lý mạng, ... Vì vậy, Sun đã phân loại lại gói java thành các gói phụ như lang, net, io, ... Và đặt các lớp liên quan đến Input/Output trong gói io, các lớp Server và ServerSocket trong các gói net.



Tiêu chuẩn để định nghĩa tên package trong java là **domain.company.package** ví dụ như: vn.viettuts.action hoặc org.sssit.dao.

Ví dụ về Subpackage

```
1 package vn.viettuts.core;
2
3 public class Simple {
4     public static void main(String args[]) {
5         System.out.println("Hello subpackage");
6     }
7 }
```

Compile: javac -d . Simple.java

Run: java vn.viettuts.core.Simple

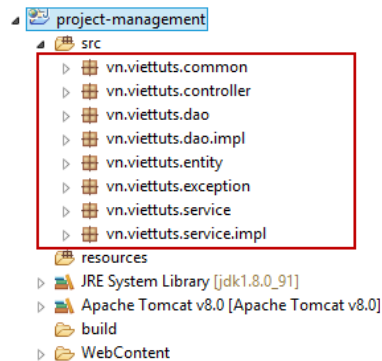
Kết quả:

```
Hello subpackage
```

Ví dụ cách tổ chức package chuyên nghiệp trong java

Các ví dụ ở trên cung cấp cho bạn cách để compile và run một lớp bên trong package do người dùng định nghĩa một cách thủ công. Khi đi làm về java trong môi trường thực tế, bạn sẽ phải học và sử dụng các IDE như eclipse, netbean, ... Những IDE này cung cấp cho bạn cách tạo và biên dịch cả một dự án đơn giản hơn rất nhiều. Khi đó bạn không cần phải nhớ các lệnh compile và run như ở trên.

Dưới đây là ví dụ cách tổ chức package chuyên nghiệp cho một dự án java web trên eclipse:


[🕒 Lớp và đối tượng trong java](#)
[Constructor trong java 🕒](#)

Bài viết liên quan:

- [Access Modifier trong Java](#)
- [Tính đóng gói trong java](#)

Correct All Grammar Errors

Check your grammar, spelling, and punctuation instantly with Grammarly

Learn

Danh Sách Bài Học

[Học Java | Hibernate | Spring](#)
[Học Excel | Excel VBA](#)
[Học Servlet | JSP | Struts2](#)
[Học C | C++ | C#](#)
[Học Python](#)
[Học SQL](#)

Bài Tập Có Lời Giải

[Bài tập Java](#)
[Bài tập C](#)
[Bài tập C++](#)
[Bài tập C#](#)
[Bài tập Python](#)
[Ví dụ Excel VBA](#)

Câu Hỏi Phỏng Vấn

[201 câu hỏi phỏng vấn java](#)
[25 câu hỏi phỏng vấn servlet](#)
[75 câu hỏi phỏng vấn jsp](#)
[52 câu hỏi phỏng vấn Hibernate](#)
[70 câu hỏi phỏng vấn Spring](#)
[57 câu hỏi phỏng vấn SQL](#)