Home Java Struts2 Hibernate Spring MyBatis Java WS Python Excel Web SQL Misc Recent Upda + Java Cơ Bản → Java là gì? LinkedList tron Lich sử Java Nâng Tầm Game Thủ ArrayList trong Các tính năng của Java Bài tập java có Cài đặt môi trường Java (cho → Sử dụng Javas Siêu Nhanh, Siêu bền - RAM Kingst người mới bắt đầu) Toán tử dấu 2 Bô nhớ lên đến 256GB và tốc đỗ tơ → Thiết lập PATH cho java 5333MHz! → Lambda Expres > Chương trình Hello World trong java 8 → JDK, JRE và JVM → Bài tân Java - ! > Biến trong java Sort) trong Jav → Các kiểu dữ liệu trong java Bài tâp Java - ! Kingston Technology → Ép kiểu (Type casting) Sort) trong Jav Các toán tử trong Java → Bài tập Java - 🖰 Package trong java → Hệ thống Unicode trong java Sort) trong Jav Bài tập quản lý + Câu Lênh Điều Khiển HashSet trong → Mênh đề If-else ⊕ Lớp và đối tương trong java HashMap trong Constructor trong java

O → Mệnh đề Switch-case Vòng lặp For → Vòng lặp While Một package (gói) trong java là một nhóm các kiểu tương tự của các lớp, giao diên và các package con . → Vòng lặp Do-While > Break trong java Package trong java có thể được phân loại theo hai hình thức, package được dựng sẵn và package do Continue trong java người dùng định nghĩa. > Chương trình java kinh điển Có rất nhiều package được dựng sẵn như java, lang, AWT, javax, swing, net, io, util, sql, ... + Các Tính Chất Của Java Trong bài này, chúng ta sẽ học: > Tính đóng gói trong java > Tính kế thừa trong java • Chi tiết về việc tạo và sử dụng các package người dùng định nghĩa. > Tính đa hình trong java Ví dụ cách tổ chức package chuyên nghiệp trong java. Nạp chồng phương thức VietTuts on fa Ghi đè phương thức Overloading vs Overriding Nội dung chính Hoc L + Các Khái Niệm Java Lơi thế của việc sử dụng package trong java **OOPs** Ví dụ về java package → Tổng quan java OOPs Biên dịch java package Theo dõi Tranç Lớp và đối tượng Run java package → Package trong java Truy cập package từ package khác Constructor trong java 1. Sử dụng packagename.* Từ khóa this trong java 2. Sử dụng packagename.classname Từ khóa super trong java 3. Sử dụng tên đầy đủ > Từ khóa final trong java Package con trong java → Từ khóa static trong java Ví dụ về Subpackage Access Modifier trong java Ví dụ cách tổ chức package chuyên nghiệp trong java → Abstract class trong java Interface trong java → Abstract vs Interface trong java

Lợi thế của việc sử dụng package trong java

- 1. Package được sử dụng để phân loại lớp và interface giúp dễ dàng bảo trì.
- 2. Package cung cấp bảo về truy cập

- 3. Package khắc phục được việc đặt trùng tên.

> Toán tử instanceof trong java

> Sử dụng regex trong java

trong java

> Truyền giá trị và tham chiếu

→ Lớp Object trong java → Object Cloning trong java

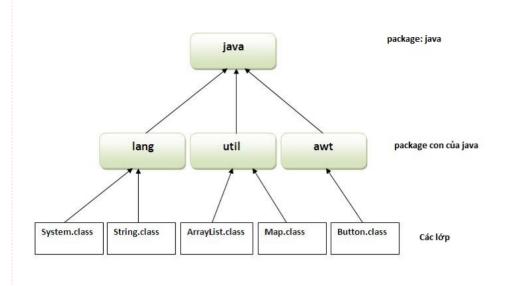
equals() và hashCode()

Array (mång) trong java

> Khai báo mảng trong java Duyệt mảng trong java Lóp Wrapper trong java → Đệ quy trong java

♣ 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 Đối 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 <u>tiết</u>



Sale Lớn iPhone Chính Hãng

Ví dụ về java package

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

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

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 xách 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 ban 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 (.) biển 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:

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

```
package mypack;
2
    import pack.*;
4
    class B {
5
        public static void main(String args[]) {
            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 du:

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

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

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

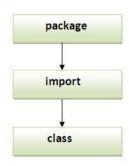
```
package mypack;
2
    class B{
        public static void main(String args[]){
4
            pack.A obj = new pack.A(); //Sử dụng tên đầy đủ
5
            obj.msg();
6
7
    }
```

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 Microsystem 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

```
package vn.viettuts.core;

public class Simple {
    public static void main(String args[]) {
        System.out.println("Hello subpackage");
    }
}
```

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

