

Like & Share & Subscribe  
thuydung ProIT4All



# Định nghĩa và tổ chức package cho dự án



Fan page: <http://facebook.com/Proit4All>

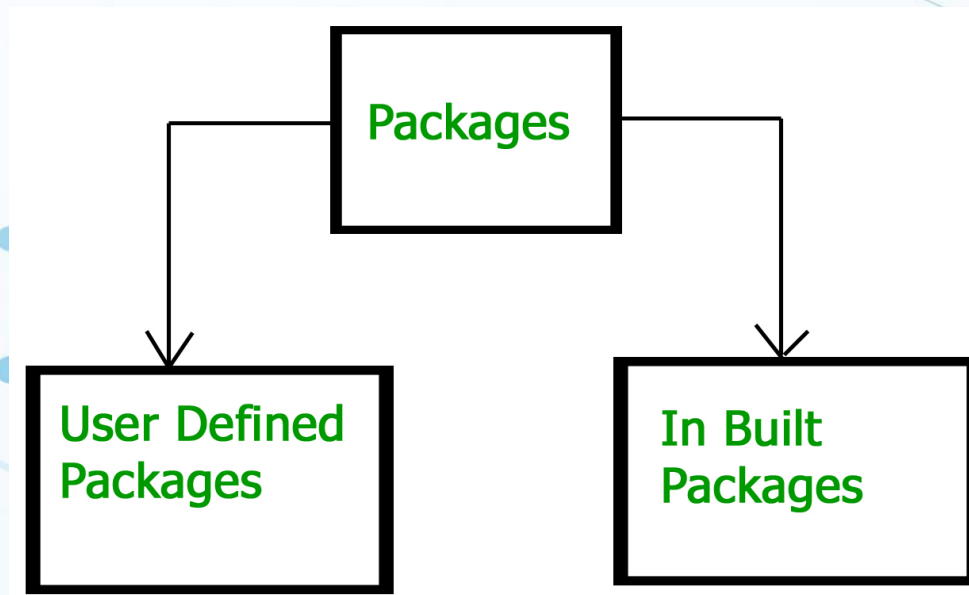


# Định nghĩa package

## □ Package (Gói)

- ❖ Là kỹ thuật nhóm các lớp, package con, và interface.
- ❖ Tư tưởng như các folder chứa các Folder con và file khác
- ❖ Giúp tránh xung đột tên và tạo ra các mã dễ bảo trì hơn

## □ Phân loại package:





# Các loại package

## ❑ Built-in package (Package dựng sẵn)

- ❖ Là các package được cung cấp sẵn trong Java hay các API
- ❖ `java.lang`: Chứa các lớp hỗ trợ ngôn ngữ Java.
  - *Package `java.lang` được import tự động.*
- ❖ `java.io`: Chứa các lớp, interface hỗ trợ nhập xuất.
- ❖ `java.util`: Chứa các lớp, interface tiện ích
- ❖ `java.net`: Chứa các lớp, interface hỗ trợ các hoạt động mạng.

## ❑ Package do người dùng định nghĩa

- ❖ Là những package định nghĩa bởi người lập trình





# Công dụng của package

## □ Package (Gói) được dùng để:

### ❖ Ngăn chặn xung đột đặt tên.

➤ Ví dụ: có thể có hai lớp có tên *Employee* ở trong hai package khác nhau

○ `com.udemy.ee.Employee` và `com.udemy.es.Employee`

### ❖ Làm cho việc tìm kiếm/định vị và sử dụng các lớp, interface, enum và chú thích dễ dàng hơn

### ❖ Cung cấp quyền truy cập được kiểm soát:

➤ *protected* và *default* có kiểm soát truy cập mức package.

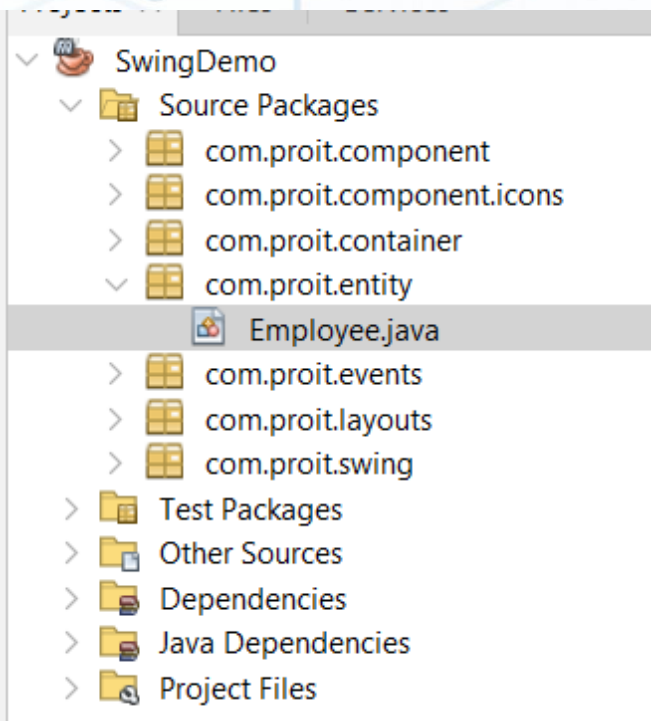
### ❖ Các package có thể được coi là đóng gói dữ liệu (hoặc ẩn dữ liệu).



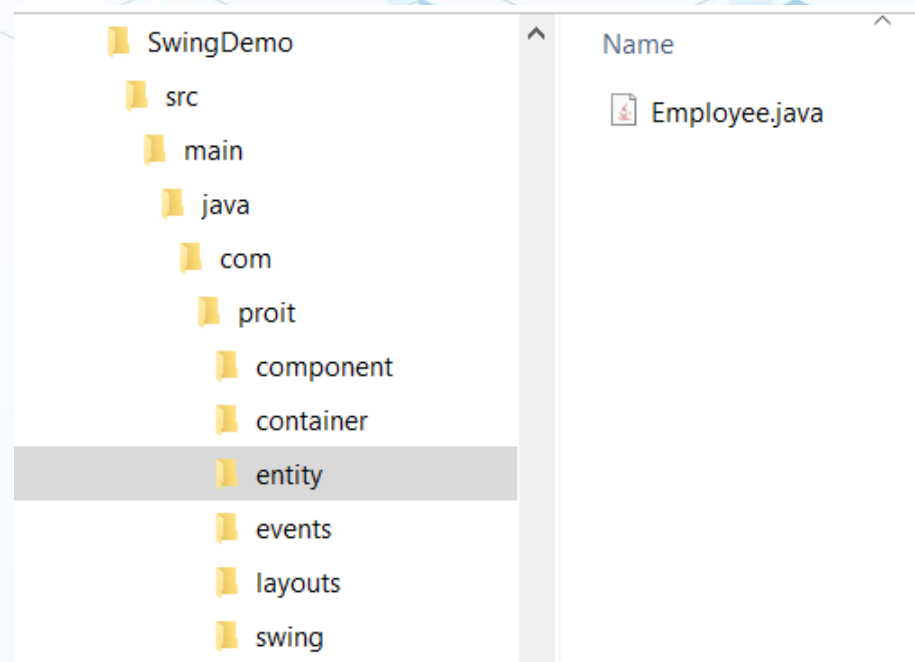
# Khai báo Package

## ❑ Sử dụng từ khoá package

❖ Ví dụ:



```
Employee.java x
Source History
1 package com.proit.entity;
2
3 public class Employee {
4
5 }
6
```

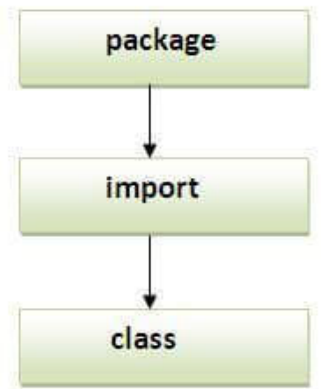




# Sử dụng package

- ❑ Các thành phần trong package (các lớp, interface, enum) có thể được sử dụng bằng cách:
  - ❖ Import package trước khi sử dụng
  - ❖ Sử dụng tên đầy đủ (fully qualified name):
    - *Package-name.class-name*



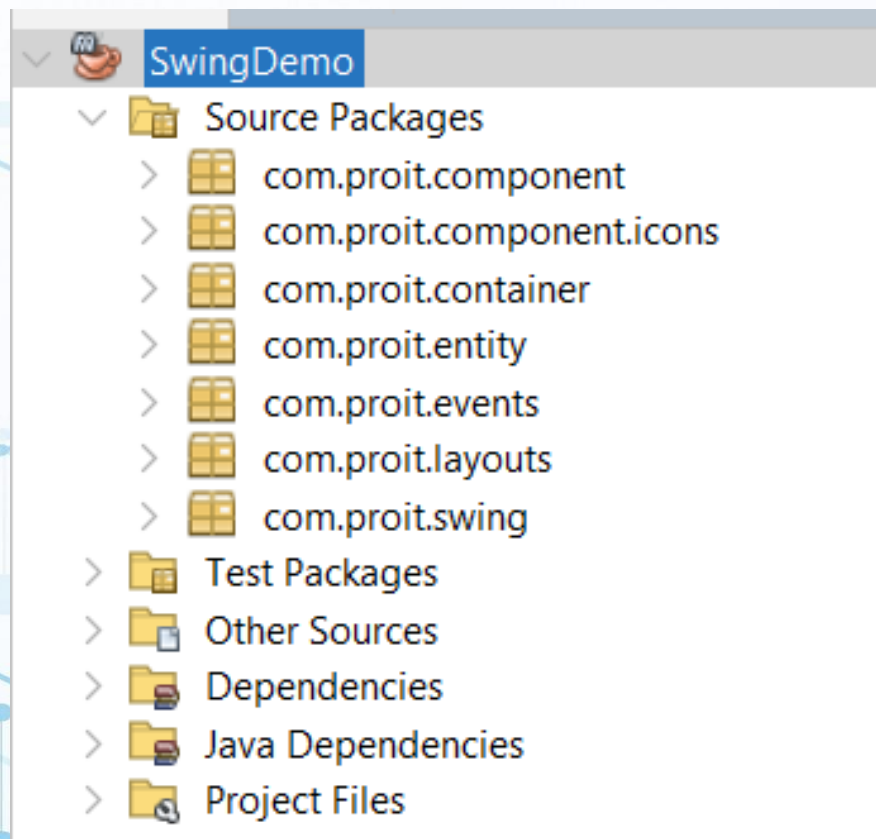


- ☐ Lệnh package phải là lệnh đầu tiên trong File Java
- ☐ Các package cần phải import khi sử dụng ở package khác
- ☐ Mỗi lớp là một phần của một số package.
- ☐ Nếu không có package nào được xác định, lớp sẽ thuộc package ngầm định.
- ☐ Tất cả các lớp/giao diện trong một File đều là một phần của cùng một package. Nhiều File có thể thuộc cùng package.
- ☐ Nếu tên package được xác định thì File phải nằm trong thư mục con cùng tên package.
- ☐ Các lớp public trong package có thể được truy cập bằng: `package-name.class-name`



# Package con

□ Package được khai báo bên trong package khác gọi là package con



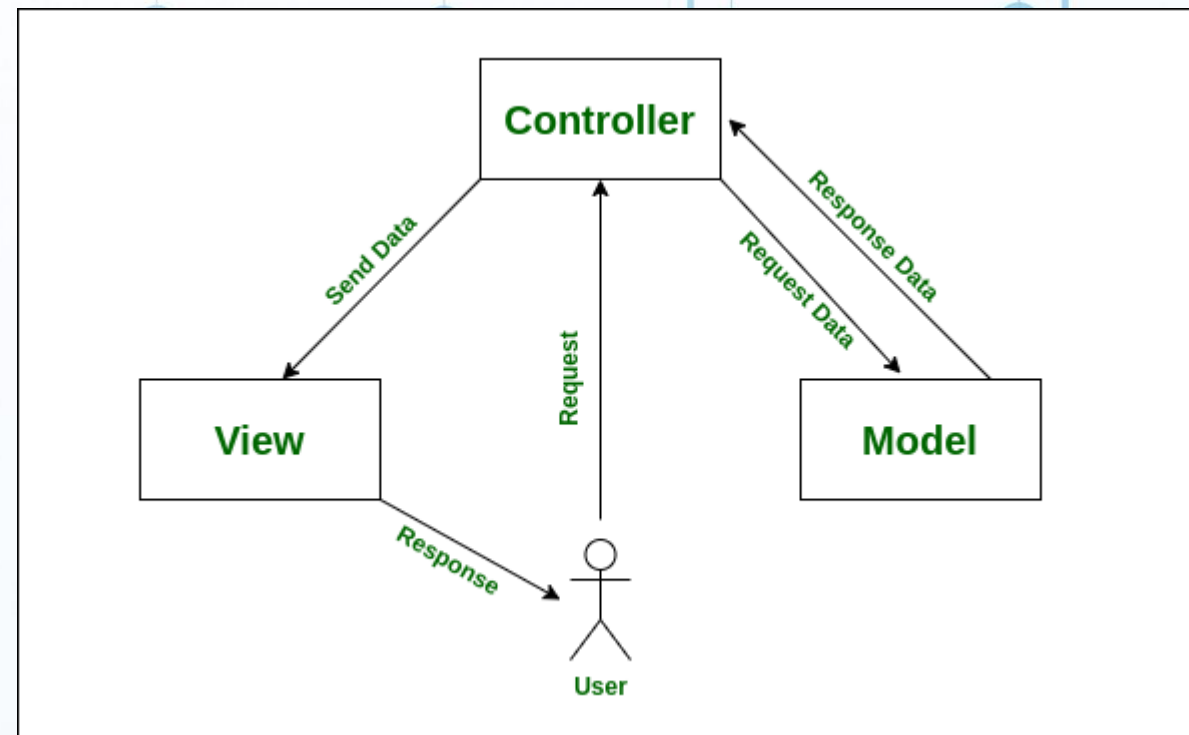




# Mô hình MVC (Model-View-Controller)

## □ MVC

- ❖ Là mẫu kiến trúc phần mềm phổ biến
- ❖ Chia ứng dụng thành ba thành phần riêng biệt:
  - *Model: Lưu trữ dữ liệu*
  - *View: Hiển thị giao diện người dùng*
  - *Controller: Xử lý các yêu cầu của người dùng và điều khiển sự tương tác giữa Model và View*





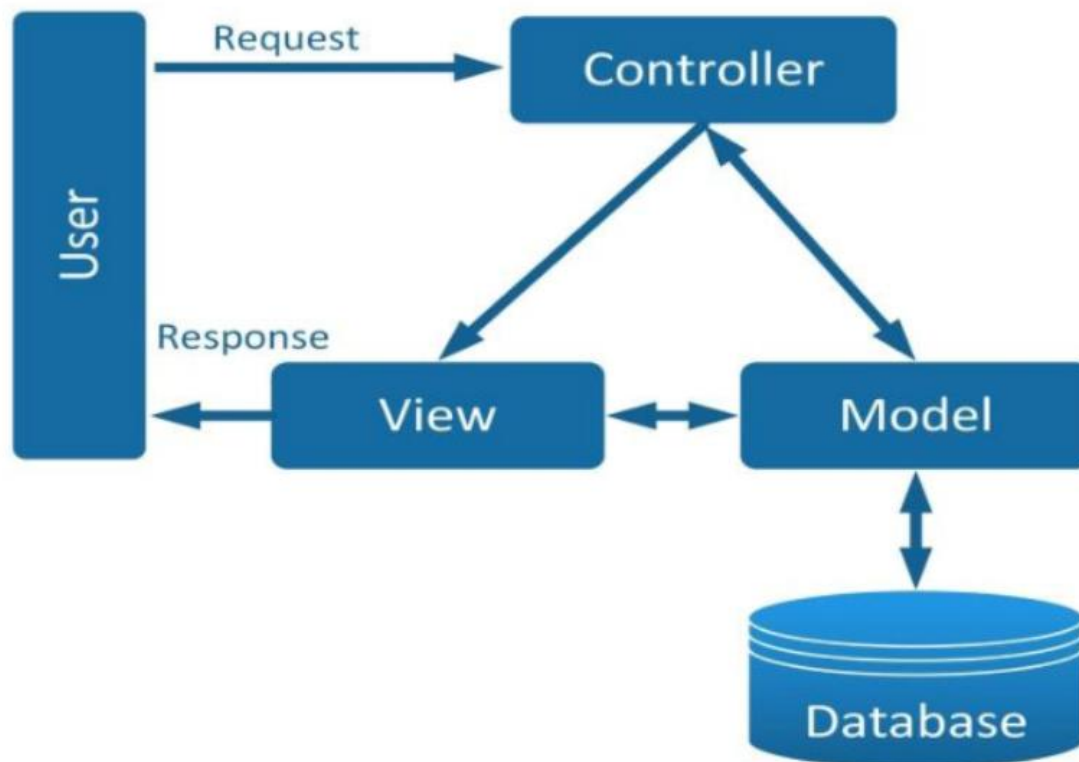
# Lợi ích của Mô hình MVC

- ❑ Dễ dàng bảo trì và mở rộng: Do các thành phần của MVC được tách biệt với nhau
- ❑ Tăng tính linh hoạt: dễ dàng thay đổi giao diện người dùng (UI) hoặc logic nghiệp vụ mà không cần thay đổi toàn bộ ứng dụng.
- ❑ Tăng khả năng tái sử dụng: Các thành phần của MVC có thể được tái sử dụng trong các ứng dụng khác nhau.



# Lợi ích của Mô hình MVC

- ❑ Mô hình MVC là một mẫu kiến trúc phần mềm mạnh mẽ và linh hoạt có thể được sử dụng để phát triển các ứng dụng web phức tạp.







Like & Share & Subscribe

Fan page: <http://facebook.com/Proit4All>

# DEMO



## Khai báo và import package



Java™

