

# Bài 35: Các gói trong Java

---

- ✓ Tổng quan về package
- ✓ Các quy ước với package
- ✓ Các các gói có sẵn trong Java
- ✓ Static import
- ✓ Xử lý xung đột tên
- ✓ Ví dụ & bài tập thực hành

# Tổng quan về package

- Package(gói) trong Java là kỹ thuật cho phép đóng gói một nhóm các lớp, interface, enum, các gói con có liên quan lại với nhau
- Mục đích sử dụng của gói:
  - Tránh xung đột tên. Tên các lớp, enum, interface trong cùng gói là duy nhất. Nếu bạn muốn đặt tên nhiều lớp hay interface, enum giống nhau thì phải để chúng ở các gói khác nhau
  - Làm cho quá trình tìm kiếm, sử dụng các lớp, interface, enum trở nên đơn giản hơn
  - Cho phép ta điều khiển khả năng truy cập và sử dụng các thành phần của lớp.
  - Khi kết hợp với các access modifier như protected, default package access(bỏ trống) sẽ cho hiệu quả mong muốn

# Tổng quan về package

- Mục đích sử dụng của gói:
  - Các thành phần protected sẽ được truy cập và sử dụng trong lớp chứa nó và các lớp trong cùng gói hoặc trong chuỗi kế thừa
  - Các thành phần default package access(bỏ trống) có thể được truy cập và sử dụng trong lớp chứa nó và các lớp trong cùng gói
  - Góp phần thể hiện tính đóng gói dữ liệu của Java
- Vậy những lớp, interface hay enum liên quan đến nhau ta gom nhóm chúng bỏ vào một package
- Ở nơi sử dụng chỉ việc import kiểu cần sử dụng vào là được
- Có thể coi một gói như một thư mục trong đó chứa các file .java liên quan đến chức năng nào đó

# Quy ước với package

- Tên package viết thường hoàn toàn để tránh nhầm lẫn tên các kiểu(class, enum, interface) trong Java
- Nếu là một công ty, tổ chức thì thường sử dụng tên miền ngược để làm tên gói để tránh nhầm lẫn
- Ví dụ:
  - net.braniumacademy.core
  - net.braniumacademy.example
  - com.domainname.xyz
- Trong Java các gói mặc định của ngôn ngữ luôn bắt đầu bằng java. hoặc javax. ví dụ như java.util, java.lang...

# Quy ước với package

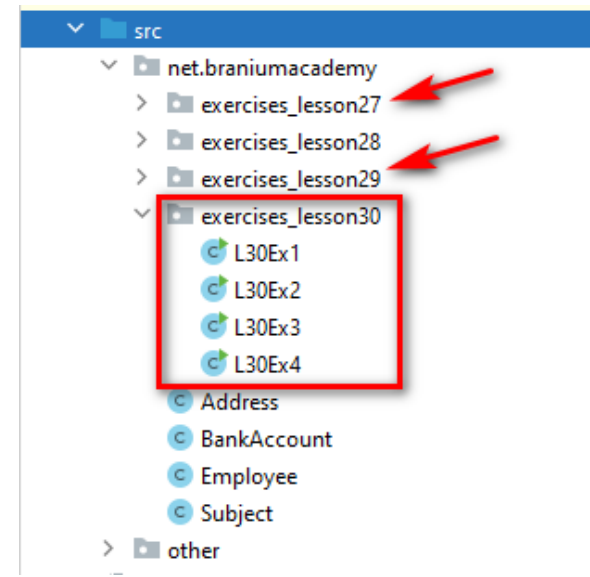
- Với các project nhỏ thì không cần tên của package hoặc tên package đặt đơn giản. Nhưng phải đảm bảo tên package là duy nhất
- Không sử dụng các keyword hoặc các gói gốc đã có trước đó để làm tên package mới.
- Nếu tên package bắt đầu bởi số hoặc keyword, tên phân tách bởi dấu - thì ta cho thêm dấu gạch dưới vào.
- Ví dụ:
  - com.example.\_123name
  - int\_.example
  - org.example.my\_package

# Quy ước với package

- Để thêm mới một lớp hay interface, enum vào một gói ta làm như sau:
  - Tạo lớp/interface/enum trong gói đã có hoặc đặt dòng package tên\_package; ở đầu file java và chuyển file này vào thư mục cùng tên package
  - Ta thường tạo lớp mới có access modifier là public hoặc thêm lớp cần thêm vào file.java đã có
  - Để việc thêm mới có hiệu lực ta phải compile lại project
- Gói con(subpackage) là các gói nằm trong một gói khác. Các gói này cũng là các gói độc lập và phải import chúng một cách tường minh để sử dụng



# Ví dụ



# Các phương thức static

- Sau đây là một phần nội dung của file java trong gói exercises\_lesson30:

```
package net.braniumacademy.exercises_lesson30;

import java.util.Scanner;

/**
 * @author Branium Academy
 * @address https://braniumacademy.net/
 */

public class L30Ex1 {
    enum Season {
        SPRING("MÙA XUÂN"), SUMMER("MÙA HẠ"),
        FALL("MÙA THU"), WINTER("MÙA ĐÔNG");
        private String value;

        Season(String value) {
            this.value = value;
        }

        public String getValue() {
            return value;
        }
    }
}
```



# Các gói có sẵn của Java

- Các gói trong Java chia làm hai loại: gói có sẵn và gói do người dùng tự tạo ra
- Một số gói có sẵn và ý nghĩa:
  - `java.lang`: chứa các lớp nền tảng của ngôn ngữ lập trình Java. Ví dụ các lớp định nghĩa các kiểu nguyên thủy, lớp `Math`... các lớp này tự động được import vào chương trình
  - `java.io`: chứa các lớp hỗ trợ các hành động thao tác vào ra dữ liệu
  - `java.util`: chứa các lớp tiện ích thực thi các cấu trúc dữ liệu như `LinkedList`... và các lớp quản lý thời gian như `Date`...
  - `java.applet`: chứa các lớp để tạo Applet

# Các gói có sẵn của Java

- Một số gói có sẵn và ý nghĩa:
  - java.awt: chứa các lớp hỗ trợ tạo giao diện đồ họa như các nút, bảng, combobox...
  - java.net: chứa các lớp hỗ trợ các thao tác với networking
- Các gói do người dùng tự định nghĩa:
  - Đầu tiên tạo gói trong project theo quy ước đã biết
  - Sau đó thêm các lớp, interface, enum... mới hoặc đã tồn tại trước đó vào package
  - Phần này sẽ được minh họa chi tiết trong phần ví dụ minh họa

# Các gói có sẵn của Java

- Để sử dụng gói và các lớp trong gói ta có hai cách:
  - Import tường minh: chỉ rõ tên gói đầy đủ và tên lớp cần dùng (khuyến dùng)
  - Import chung cả gói: chỉ cung cấp tên gói và .\* (không khuyến khích dùng)

## ➤ Ví dụ:

```
package net.braniumacademy.exercises_lesson29; // gói của lớp hiện thời
import net.braniumacademy.Subject; // import Lớp Subject vào để dùng
import java.util.Scanner; // import Lớp Scanner vào để sử dụng
```

```
public class L29Ex4 {
    public static void main(String[] args) {
        var input = new Scanner(System.in);
        System.out.println("Nhập số lượng môn học: ");
        var n = input.nextInt();
        input.nextLine();
        if (n > 0) {
            Subject[] subjects = new Subject[n];
        }
    }
}
```

# Static import

- Đây là một tính năng của Java cho phép các thành phần static được import vào và sử dụng trực tiếp để giảm bớt câu lệnh
- Trong cú pháp có từ khóa **static** sau import:

```
package other;
import static java.lang.System.*;
import static java.lang.Math.*;

public class Example {
    public static void main(String[] args) {
        System.out.println("Hello World");
        // tương đương:
        out.println("Hello World");

        double s = sin(toRadians(60)); // Math.sin(toRadians(60))
    }
}
```

# Xử lý xung đột tên

- Ta thường chỉ quan tâm đến các package khi xảy ra xung đột tên lớp/interface/enum
- Ví dụ ta có nhiều lớp giống tên nhau cần sử dụng và chúng lại nằm trong các gói khác nhau
- Ví dụ điển hình là lớp Date của gói java.util và Date của java.sql
- Nếu import chung vào sẽ bị lỗi:

```
package other;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
public class Example {  
    public static void main(String[] args) {  
        Date today; // error  
    }  
}
```

# Xử lý xung đột tên

- Nếu cần dùng cả hai lớp trên ta có thể giải quyết theo cách chỉ rõ tên đầy đủ của gói và của lớp như sau:

```
package other;

public class Example {
    public static void main(String[] args) {
        java.util.Date today = new java.util.Date();
        java.sql.Date sqlDate = new java.sql.Date(today.getTime());
        //...
    }
}
```

## Lưu ý quan trọng

- Mỗi lớp đều thuộc một package nào đó
- Nếu tên package không được chỉ rõ trong file .java thì lớp trong file đó tự động được bỏ vào một gói mặc định không có tên
- Các kiểu(lớp, interface, enum) trong một file đều thuộc về cùng một gói. Một gói có thể có nhiều file
- Nếu tên package được chỉ định trong file .java thì file đó phải nằm trong thư mục của gói. Tên gói và tên thư mục bắt buộc phải trùng nhau
- Ta có thể sử dụng các kiểu public từ các gói khác

# Nội dung tiếp theo

**Đọc file text với Scanner**