**Mục đích sử dụng**

* Đôi khi một biến chỉ nên chứa những giá trị đã được giới hạn trong một tập xác định nào đó. Enum giúp ta tạo ra tập hợp của các giá trị như vậy. Cụ thể hơn, enum giúp ta tạo ra tập hợp của các giá trị là hằng số.
* Enum là một lớp đặc biệt dùng để tập hợp và vận hành dựa trên các thành phần hằng số.
* Ví dụ: một tuần chỉ có 7 ngày từ thứ 2 đến chủ nhật. Điều này là không đổi, ta có thể dùng enum để tạo ta tập hợp của ngày trong tuần cho thuận tiện sử dụng.
* Ví dụ khác: kích thước quần áo có các size siêu nhỏ, nhỏ, vừa, lớn và siêu lớn. Ta hoàn toàn có thể dùng enum để lưu trữ các hằng số trên để tiện sử dụng.
* Một trong những ưu điểm của enum so với việc sử dụng các giá trị thông thường là tránh được việc bị gán nhầm giá trị vì tên các thành phần hằng số của nó rất tường minh.

**Cú pháp tổng quát**

**access enum name {**

***// các hằng số***

***// thuộc tính***

***// phương thức khởi tạo***

***// phương thức khác***

**}**

Trong đó:

* Phần access là access modifier. Có thể là public, protected, private hoặc bỏ trống. Ý nghĩa sử dụng của access modifier đã biết.
* Mặc định enum là final nên ta không thêm keyword này vào. Ngoài ra không sử dụng từ khóa abstract trong khai báo enum.
* Enum bắt buộc phải có từ khóa enum trong khai báo.
* Phần name là tên của enum. Do tính chất tập hợp của enum nên ta đặt tên enum sao cho biểu đạt được ý nghĩa của tập hợp ở số ít. Ví dụ Size với tập hợp kích thước quần áo. Season với tập hợp mùa của 1 năm. Rank với học lực, các mức giải thưởng chẳng hạn.
* Phần kế tiếp là các hằng số. Hằng số sẽ đặt theo quy ước hằng số đã biết. Chúng nên liên quan đến ý nghĩa chung của tập hợp đang mô tả.
* Các hằng số phân biệt nhau bằng dấu phẩy. Các hằng số trong enum không được phép trùng nhau nhưng có thể trùng giá trị.
* Các thuộc tính đặt theo quy ước tính đóng gói dữ liệu. Thường dùng các thuộc tính mô tả giá trị hoặc biểu hiện của các hằng số.
* Tiếp theo là phần phương thức khởi tạo. Phương thức khởi tạo mặc định của enum là private. Nếu ta khai báo phương thức khởi tạo mà không xác định access modifier cho nó thì nó sẽ có access modifier là private.
* Phần phương thức có thể chứa các phương thức cung cấp các hành động nào đó liên quan đến tập hợp đang mô tả. Ví dụ trả về giá trị tương ứng từng hằng số trong enum.
* Ví dụ:

*// enum mô tả tập các mùa trong năm*

public enum Season {

*// các hằng số*

*SPRING*("MÙA XUÂN"), *SUMMER*("MÙA HẠ"), *FALL*("MÙA THU"), *WINTER*("MÙA ĐÔNG");

*// phương thức khởi tạo*

Season(String name) {

        seasonName = name;

    }

*// tên mùa trong tiếng Việt*

private String seasonName;

*// phương thức trả về tên mùa trong tiếng việt*

public String getSeasonName() {

        return seasonName;

    }

}

**Các thao tác với enum**

* Có thể dùng phép so sánh tương đương để so sánh giá trị của các hằng số trong enum.
* Dùng phương thức values() để lấy ra mảng các hằng số theo thứ tự khai báo trong enum hiện thời.
* Dùng valueOf(name) để lấy ra hằng số trong enum với tên tương ứng.
* Dùng tên enum.tên\_hằng để truy cập đến hằng số nào đó trong enum.

**Ví dụ minh họa**

Ví dụ 1: minh họa sử dụng enum mô tả mùa trong năm.

public class ExampleEnum1 {

*// enum mô tả tập các mùa trong năm*

enum Season {

*// các hằng số*

*SPRING*("MÙA XUÂN"), *SUMMER*("MÙA HẠ"),

*FALL*("MÙA THU"), *WINTER*("MÙA ĐÔNG");

*// phương thức khởi tạo*

Season(String name) {

            seasonName = name;

        }

*// tên mùa trong tiếng Việt*

private String seasonName;

*// phương thức trả về tên mùa trong tiếng việt*

public String getSeasonName() {

            return seasonName;

        }

    }

    public static void main(String[] args) {

        Season winter = Season.*WINTER*;

        System.*out*.println("Tên mùa trong tiếng Anh: " + winter);

        System.*out*.println("Tên mùa trong tiếng Việt: " + winter.getSeasonName());

        System.*out*.println(Season.*valueOf*("WINTER"));

    }

}

Kết quả:

Tên mùa trong tiếng Anh: WINTER

Tên mùa trong tiếng Việt: MÙA ĐÔNG

WINTER

Ví dụ 2: dùng enum mô tả kích thước quần áo.

public class ExampleEnum2 {

*// enum mô tả size quần áo*

public enum Size{

*EXTRA\_SMALL*, *SMALL*, *MEDIUM*, *LARGE*, *EXTRA\_LARGE*

}

    public static void main(String[] args) {

*// minh họa sử dụng*

Size mySize = Size.*EXTRA\_LARGE*;

        Size myBrotherSize = Size.*LARGE*;

        if(myBrotherSize == mySize) {

            System.*out*.println("We can share our clothes with each other.");

       } else {

System.out.println("We cannot share our clothes with each other.");

}

   }

}

Kết quả:

We cannot share our clothes with each other.