

Khoá học: Java Developer

Bài 05: Kiểu liệt kê (enumeration) và tĩnh (static)



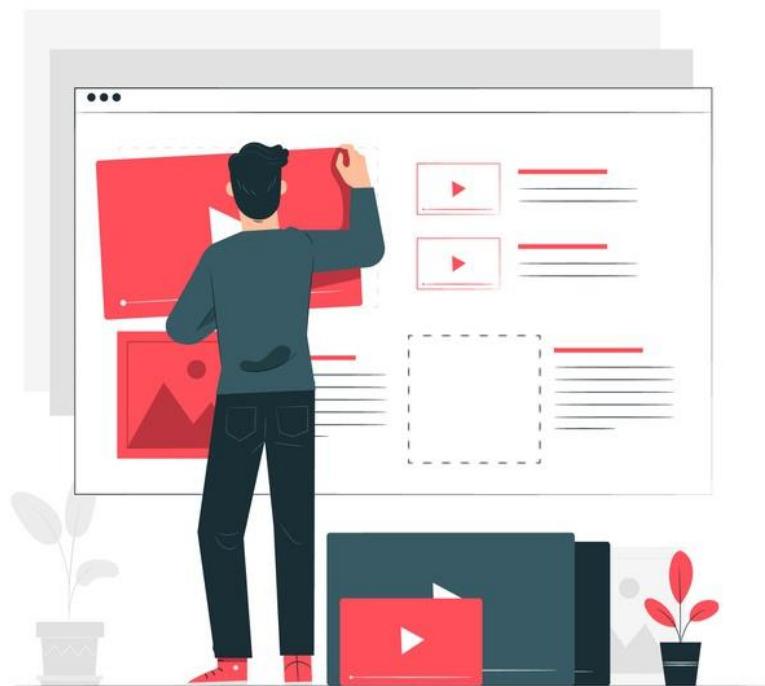
Mục tiêu bài học

Kết thúc bài học này, bạn có thể

- Hiểu khái niệm liệt kê (Enumeration)
- Triển khai kiểu enum
- Hiểu khái niệm tĩnh (Static)
- Triển khai lớp có chứa các thành viên tĩnh.



Nội dung



1. Khái niệm Enumeration
2. Khởi tạo và thao tác với kiểu enum
3. Khái niệm Static
4. Static fields và static methods

Tham khảo: <https://docs.oracle.com/javase/tutorial/java/javaOO/classvars.html>

Static

Tĩnh

Dòng code sau thể hiện phép tính căn bậc hai của 9.
Với **Math** là tên class chứa trong gói **java.lang**

x = Math.sqrt(9) ;

Class

Method



Vì sao method có thể được gọi thông qua tên Class mà không phải thông qua tên của đối tượng?!

Error: Cannot create an instance of the static class 'Math'.

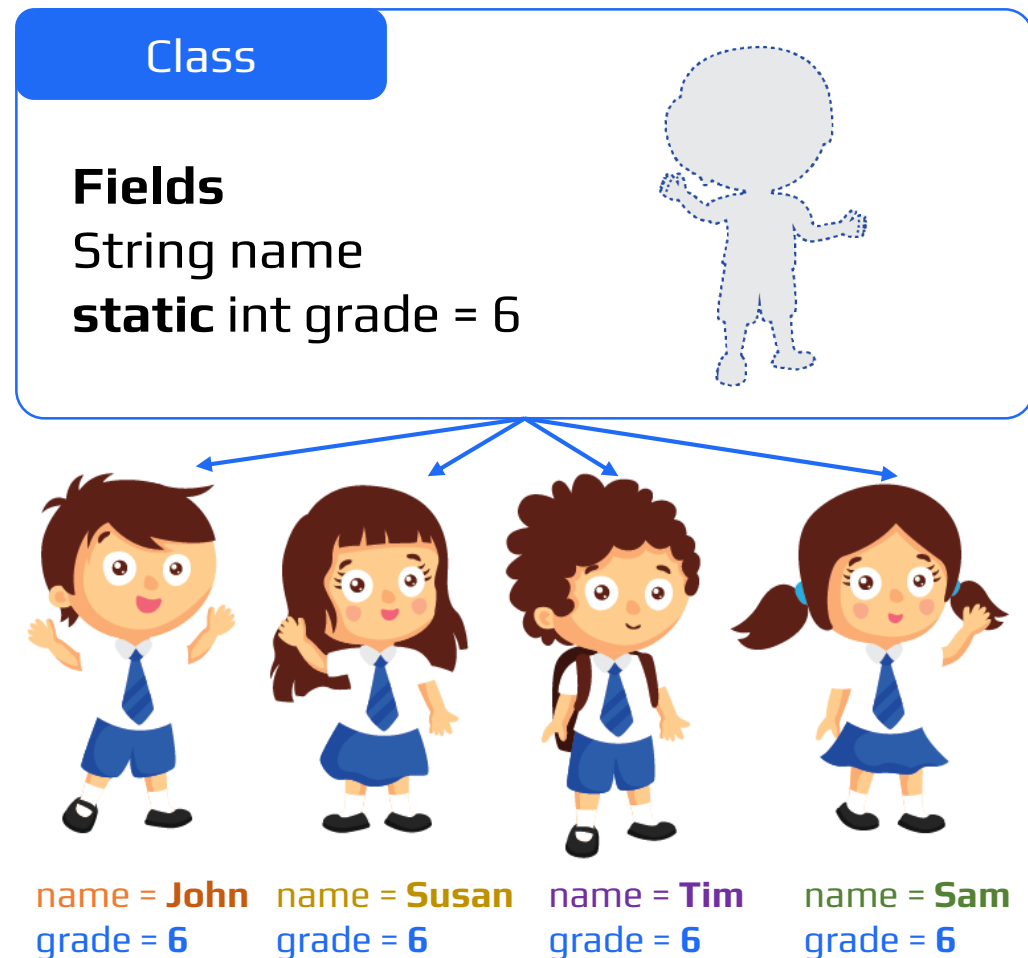
```
Math m = new Math();  
double x = m.sqrt(9);
```

!!! Impossibility !!!

Static

Tĩnh

- **Khái niệm tĩnh (Static)** là việc sử dụng chung vùng nhớ của một vài thành phần trong class cho tất cả các đối tượng tạo ra từ class đó.
- Chúng ta có thể dùng từ khóa **static** để chia sẻ vùng nhớ cho:
 - Thuộc tính của lớp (*field of class*)
 - Phương thức của lớp (*method of class*)
 - Lớp (class)



Thuộc tính tĩnh

Static fields

- Thuộc tính tĩnh là giống nhau cho tất cả đối tượng.
- Tất cả đối tượng được tạo ra từ class Student đều có cùng một dữ liệu grade = 6.

```
public class Student {  
    //Object fields  
    String name;  
    //Static fields  
    static int grade = 6;  
    //Constructor  
    public Student(String name) {  
        this.name = name;  
    }  
}
```

```
Student std1 = new Student("John");  
System.out.println(std1.name + "-" + std1.grade);  
Student std2 = new Student("Sam");  
System.out.println(std2.name + "-" + std2.grade);
```

Kết quả từ
đoạn code:
John - 6
Sam - 6

Thuộc tính tĩnh

Static fields

- Thuộc tính tĩnh nên được gọi thông qua tên class.

Student.grade = 6;

- Thuộc tính tĩnh không thể gọi thông qua tên của đối tượng.
- Khi dữ liệu của thuộc tính tĩnh bị thay đổi thông qua tên của đối tượng thì các đối tượng còn lại cũng bị ảnh hưởng theo.

```
Student std1 = new Student("John");
Student std2 = new Student("Sam");
//Gọi thuộc tính tĩnh thông qua tên class
Student.grade = 8;
System.out.println(std1.name + "-" + std1.grade);
System.out.println(std2.name + "-" + std2.grade);
```

Kết quả từ đoạn code:

John - 8

Sam - 8

Phương thức tĩnh

Static methods

- **Phương thức tĩnh** là phương thức có sử dụng đến các thuộc tính tĩnh.
- Các thuộc tính không khai báo static **KHÔNG THỂ** được sử dụng bên trong phương thức static.
- Phương thức tĩnh cũng được gọi thông qua tên Class.

`Student.sayHello();`

- Không thể sử dụng từ khoá **this** bên trong phương thức static.

~~this~~.numberOfStudents;

```
public class Student {  
    //Object fields  
    String name;  
    //Static fields  
    static int grade;  
    private static int numberOfStudents = 0;  
    //Constructor  
    public Student(String name) {  
        this.name = name;  
        numberOfStudents++;  
    }  
    //Static methods  
    public static int getNumberOfStudents() {  
        return this.numberOfStudents;  
    }  
    public static void sayHello() {  
        System.out.println("Hello" + name);  
    }  
}
```


Kiểu liệt kê

Enumeration

- **Enumeration** là kiểu dữ liệu chứa một tập các hằng số.
- Các giá trị của enumeration được gọi là **hằng số liệt kê**.
- Enumeration được khai báo với từ khoá **enum**. Các hằng số liệt kê cách nhau bằng dấu phẩy (,).
- Khi không khai báo giá trị đi kèm, enum sẽ mặc định **giá trị đầu tiên là 0** và tăng dần 1 đơn vị cho mỗi giá trị sau đó.

Cú pháp

```
enum TênTậpHợp {  
    HẰNG SỐ LIỆT KÊ 1 [= Giá trị 1],  
    HẰNG SỐ LIỆT KÊ 2 [= Giá trị 2],  
    HẰNG SỐ LIỆT KÊ N [= Giá trị N]  
}
```

Ví dụ

```
enum Gender {  
    MALE, FEMALE  
}  
  
enum Season {  
    SPRING = 1, SUMMER = 2, AUTUMN = 3, WINTER = 4  
}
```

Thao tác với Enum

- **Gender sex;**
Khai báo field sử dụng kiểu enumeration đã định nghĩa trước đó.
- **sex = Gender.MALE;**
Gán dữ liệu cho biến kiểu enum.
- **if(sex == Gender.FEMALE)**
Có thể sử dụng phép so sánh với kiểu enum.
- **switch(sex) {**
 case MALE: //do something 1
 case FEMALE: //do something 2
 }
Enum có thể được sử dụng trong phát biểu switch

```
namespace Bai_tap_1
{
    enum Season
    {
        SPRING, SUMMER, AUTUMN, WINTER
    }
    class Program
    {
        static void Main(string[] args)
        {
            Season s;
            s = Season.WINTER;
            switch (s)
            {
                case Season.SPRING: Console.WriteLine("Spring consists of Jan, Feb, Mar"); break;
                case Season.SUMMER: Console.WriteLine("Summer consists of Apr, May, Jun"); break;
                case Season.AUTUMN: Console.WriteLine("Autumn consists of Jul, Aug, Sep"); break;
                case Season.WINTER: Console.WriteLine("Winter consists of Oct, Nov, Dec"); break;
            }
        }
    }
}
```

Kết quả từ đoạn code trên: Winter consists of Oct, Nov, Dec