

Bài 7

# **Biến tĩnh và phương thức tĩnh** **(Static Fields and Static Methods)**

# NỘI DUNG

- Biến tĩnh (Static variables)
- Phương thức tĩnh (Static methods)
- Demo

# Biến tĩnh

- Biến tĩnh (Static variables) được khai báo với từ khóa **static**
- Biến tĩnh được **dùng chung** cho tất cả các đối tượng được tạo ra từ lớp đó.
- Biến tĩnh được tham chiếu thông qua tên của lớp.  
*Ví dụ: **Student**.**Quantity***
  - *Student* là tên của lớp
  - *Quantity* là biến tĩnh
- Biến tĩnh cũng có thể được truy xuất từ một đối tượng của lớp (hạn chế sử dụng)  
*Ví dụ: **S1**.**Quantity***
  - **S1** là tên đối tượng
  - *Quantity* là biến tĩnh

# Biến tĩnh

```
public class Student {  
    // Instance  
    private String id, name;  
  
    // Static variable  
    private static int yob = 2000;  
  
    public static void main(String[] args) {  
        Student stud1 = new Student();  
        Student stud2 = new Student();  
  
        stud1.yob = 2001;  
        System.out.println(stud2.yob); // output???  
  
        Student.yob = 2002;  
        System.out.println(stud1.yob); // output???  
        System.out.println(stud2.yob); // output???  
    }  
}
```

2001

2002

# Biến tịnh

Khi nào sử dụng biến tịnh?

## Ví dụ:

Xây dựng lớp Student gồm các thông tin như ID, Name, Age, **Quantity**. Trong đó Quantity là biến chứa tổng số lượng sinh viên của lớp.

- **ID, Name, Age**: khai báo biến thường
- **Quantity**: khai báo biến tĩnh để dùng chung cho tất cả các đối tượng

# Biến tĩnh

```
3 public class Student {  
4     private String id;  
5     private String name;  
6     private int age;  
7     public static int Quantity = 0;  
8  
9     public Student(String id, String name, int age) {  
10         this.id = id;  
11         this.name = name;  
12         this.age = age;  
13         Quantity = Quantity + 1;  
14     }  
15 }
```

Quantity  
là biến  
tĩnh

Khi một đối tượng được tạo  
ra thì Quantity tăng lên 1

# Demo biến tĩnh

- Xem demo trực tiếp về biến tĩnh



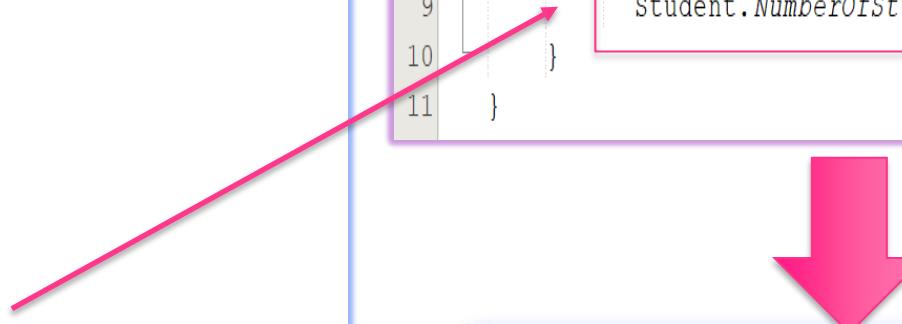
# Phương thức tĩnh

- Phương thức tĩnh (Static methods) được khai báo với từ khóa **static**
- Phương thức tĩnh được gọi trực tiếp thông qua tên lớp.
- Phương thức tĩnh chỉ truy cập đến các thành viên tĩnh khác trong lớp, **không được quyền** truy xuất đến các thành viên không tĩnh.

# Phương thức tĩnh

```
3 public class Student {  
4     private String id;  
5     private String name;  
6     private int age;  
7     public static int Quantity = 0;  
8  
9     public Student(String id, String name, int age) {  
10        this.id = id;  
11        this.name = name;  
12        this.age = age;  
13        Quantity = Quantity + 1;  
14    }  
15  
16    public static void NumberOfStudent()  
17    {  
18        System.out.println("So luong sinh vien: " + Student.Quantity);  
19    }  
20}
```

```
2  
3     public class HelloWorld {  
4  
5         public static void main(String[] args) {  
6             Student s1 = new Student("01", "Son", 20);  
7             Student s2 = new Student("02", "Minh", 18);  
8  
9             Student.NumberOfStudent();  
10        }  
11    }
```

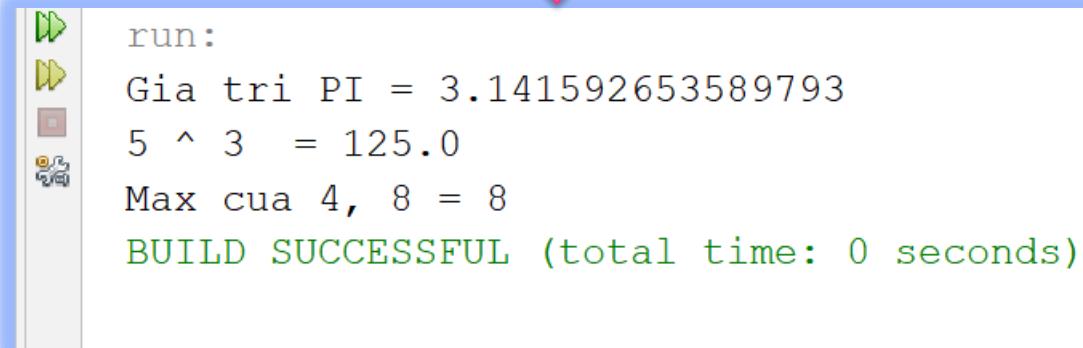


The diagram illustrates the execution flow. A pink arrow points from the `NumberOfStudent()` method in the `Student` class to the corresponding line in the `HelloWorld` main method. A large pink downward arrow indicates the flow of execution from the main method into the `main` function.

```
run:  
So luong sinh vien: 2  
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Phương thức tĩnh

```
3  public class HelloWorld {  
4  
5      public static void main(String[] args) {  
6          System.out.println("Gia tri PI = " + Math.PI);  
7  
8          System.out.println("5 ^ 3 = " + Math.pow(5, 3));  
9  
10         System.out.println("Max cua 4, 8 = " + Math.max(4, 8));  
11     }  
12 }
```



```
run:  
Gia tri PI = 3.141592653589793  
5 ^ 3 = 125.0  
Max cua 4, 8 = 8  
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Bài tập tại lớp

Viết một chương trình quản lý tài khoản ngân hàng. Mỗi tài khoản ngân hàng có số tài khoản, tên chủ tài khoản và số dư. Ngoài ra, bạn cần theo dõi tổng số lượng tài khoản đã tạo.

**Lớp BankAccount có các thành phần sau:**

- Biến accountNumber (số tài khoản)
- Biến accountHolder (tên chủ tài khoản)
- Biến balance (số dư tài khoản)
- Phương thức deposit() (nạp tiền vào tài khoản)
- Phương thức withdraw() (rút tiền từ tài khoản)
- Biến totalAccounts (tổng số tài khoản được tạo)
- Phương thức getTotalAccounts() (trả về tổng số tài khoản được tạo)

**Yêu cầu:**

- [1] Xác định biến nào nên làm biến tĩnh (static).
- [2] Xác định phương thức nào nên làm phương thức tĩnh (static).

# HỎI ĐÁP

