Bài: Từ khóa static trong lập trình hướng đối tượng

Xem bài học trên website để ủng hộ Kteam: Từ khóa static trong lập trình hướng đối tượng

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Trong bài trước, Kteam đã giới thiệu cho các bạn xong về các PHẠM VI TRUY CẬP TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG. Ở bài này, nhẹ nhàng hơn ta sẽ tìm hiểu từ khóa **static**.

Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- CÁC BIẾN TRONG JAVA.
- CÁC KIỂU DỮ LIỆU TRONG JAVA.
- CÁC HẠNG TOÁN TỬ TRONG JAVA
- CẤU TRÚC RẼ NHÁNH TRONG JAVA
- VÒNG LẶP WHILE TRONG JAVA
- VÒNG LẶP FOR TRONG JAVA
- MẢNG TRONG JAVA
- VÒNG LẶP FOR-EACH TRONG JAVA
- VAI TRÒ BREAK, CONTINUE TRONG VÒNG LẶP JAVA
- SWITCH TRONG JAVA
- LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- CLASS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Từ khóa static làm gì?
- Cách sử dụng static

Từ khóa static làm gì?

Khi ta khai báo các thuộc tính, phương thức thì nó chỉ được sử dụng khi khởi tạo đối tượng, thông tin cũng thuộc đối tượng đó.

Có những lúc, ta cần những thông tin chung cho tất cả các đối tượng. Có nghĩa những thông tin đó lưu ở một vùng nhớ duy nhất. Từ khóa static sử dụng để quản lý bộ nhớ, khi những thành viên bên trong một lớp có từ khóa **static** thì nó thuộc về lớp, không phải thuộc về riêng một đối tượng nào đó.

Cách sử dụng static

Tạo biến tĩnh

Khi khai báo một biến tĩnh, biến đó có thể lưu thông tin chung cho tất cả các đối tượng.

Ví dụ: tạo một class Student của một trường 'Kteam Education', như vậy chỉ cần một bộ nhớ chung lưu thông tin tên trường, như vậy tiết kiệm bộ nhớ hơn. Ngoài ra, ta có thể tạo một biến đếm có bao nhiêu đối tượng Student đã được tạo ra:

Ta sẽ tạo một **class Student** như sau:

:

```
public class Student {

	public String name;
	public int age;
	public float height;

	public static String universityName = "Kteam Education";
	public static int total = 0;

	public Student(String name, int age, float height) {
		this.name = name;
		this.age = age;
		this.height = height;
		total += 1;
	}
}
```

Ta dùng 2 biến tĩnh là **universityNam** và **total**, mỗi khi tao khởi tạo một đối tượng của lớp Student, ta sẽ tăng giá trị **total** lên một đơn vị.

Tiếp theo, ta sẽ viết chương trình main:

:

```
public class HelloWorld {

	public static void main(String[] args) {
		Student a = new Student("Chau", 21, 1.7f);
		System.out.println("University (from class):" + Student.universityName);
		System.out.println("University (from instance):" + a.universityName);

		System.out.println("Total (from class):" + Student.total);
		Student b = new Student("Long", 24, 1.7f);
		System.out.println("Total (from instance):" + b.total);
	}
}
```

Trong đoạn chương trình, Kteam đã thử truy xuất biến tĩnh từ lớp hoặc từ đối tượng.

```
University (from class):Kteam Education
University (from instance):Kteam Education
Total (from class):1
Total (from instance):2
```

Tạo phương thức tĩnh

Phương thức tĩnh cũng giống như biến tĩnh, có thể gọi mà không cần khởi tạo đối tượng. Phương thức tĩnh rất thích hợp cho những class thư viện viết sẵn, không cần khởi tạo mà chỉ cần gọi ra để chạy chương trình.

Ví dụ: giới thiệu trường học từ class Student.

Tại class Student

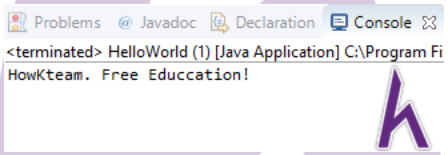
:

```
public class Student {

	public String name;
	public int age;
	public float height;

	public static String universityName = "Kteam Education";
	public static int total = 0;

	public Student(String name, int age, float height) {
		this.name = name;
		this.age = age;
		this.height = height;
		total += 1;
	}

	public static void getInfoUniversity() {
		System.out.println("HowKteam. Free Educcation!");
	}
}
```

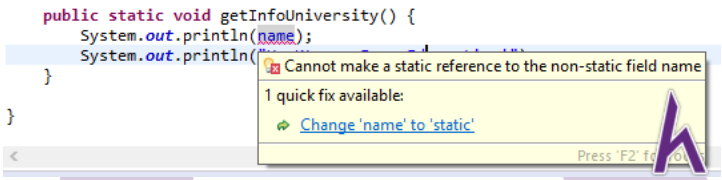
Tại chương trình main

:

```
public class HelloWorld {

	public static void main(String[] args) {
		Student.getInfoUniversity();
	}
}
```



Tuy nhiên, phương thức **static** không thể tác động đến thuộc tính và phương thức liên quan đối tượng (non-static).



Khối static

Khối **static** được sử dụng cho mục đích khởi tạo giá trị các biến **static**. Khối sẽ được thực hiện khi lớp chứa nó được **load** vào trong bộ nhớ.

Trong một lớp có thể nhiều khối tùy ý. Các khối này sẽ chạy cùng nhau, và chạy trước cả chương trình main của lớp đó.

Ví dụ: ta tạo khối static ở class HelloWorld

:

```
public class HelloWorld {
	static String course;

	static {
		System.out.println("HowKteam");
		course = "Java core";
	}

	public static void main(String[] args) {
		System.out.println("Free education");
		System.out.println("course:"+ HelloWorld.course);
	}
}
```

```
HowKteam
Free education
course:Java core
```

Kết

Như vậy chúng ta đã tìm hiểu từ khóa static trong lập trình hướng đối tượng

Ở bài sau, Kteam sẽ giới thiệu đến bạn về TỪ KHOÁ THIS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".