Bài: Từ khóa this trong lập trình hướng đối tượng

Xem bài học trên website để ủng hộ Kteam: Từ khóa this 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 giải thích các bạn từ khóa STATIC. Tiếp tục ở bài này, chúng ta sẽ tìm hiểu lại từ khóa **this**.

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 this làm gì?
- Cách sử dụng this

Từ khóa this làm gì?

Từ khóa **this** dùng để ánh xạ đối tượng hiện tại. Giống như trong lớp **Student** có rất nhiều đối tượng như bạn Châu, Long, Thanh,... thì khi xử lý các thuộc tính và phương thức ta sẽ dùng từ 'bạn ấy' để ám chỉ đối tượng hiện tại cần thực hiện.

Cách sử dụng this

Ánh xạ đối tượng khi cần sử dụng

Như những ví dụ trước. Nếu như không sử dụng **this** trong phương thức khởi tạo:

:

Copyright © Howkteam.com

```
public class Person {
	public String name;
	public int age;
	public float height;

	public Person(String name, int age, float height) {
		name = name;
		age = age;
		height = height;
	}

}
```

Ta hãy thử khởi tạo đối tượng và in thông tin ra:

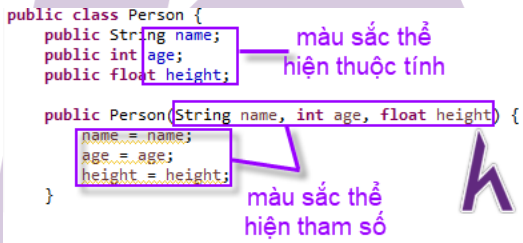
:

```
public class HelloWorld {

	public static void main(String[] args) {
		Person a = new Person("Chau", 21, 1.7f);
		System.out.println(a.name);
		System.out.println(a.age);
		System.out.println(a.height);
	}
}
```

```
null
0
0.0
```

Ta thấy xuất hiệt giá trị mặc định của thuộc tính khi khởi tạo, có nghĩa việc gán giá trị khi khởi tạo không thành công. Bây giờ ta hãy quan sát lại cách hiển thị trên Eclipse:



Như cách Eclipse thể hiện thì những biến màu xanh biển là chính là thuộc tính, còn biến mà nâu là tham số trong phương thức. Như vậy, nếu không có this, ta đang gán giá trị tham số cho chính nó. Không tác động đến thuộc tính của đối tượng.

```
public class Person {
    public String name;
    public int age;
    public float height;

    public Person(String name, int age, float height) {
        this.name = name;
        this.age = age;
        this.height = height;
    }
```

Chỉ cần thêm **this**, màu sắc đã thay đổi.

Copyright © Howkteam.com

Gọi phương thức từ lớp hiện tại

Ngoài gọi được thuộc tính, thì this có thể gọi đến phương thức từ lớp hiện tại.

Ví dụ: viết phương thức trả thông tin Person và gọi mỗi lần khởi tạo đối tượng.

:

```
public class Person {
	public String name;
	public int age;
	public float height;

	public Person(String name, int age, float height) {
		this.name = name;
		this.age = age;
		this.height = height;
		this.getInfo();
	}

	public void getInfo() {
		System.out.println("Name:"+this.name);
		System.out.println("Age:"+this.age);
		System.out.println("Height:"+this.height);
	}
}
```

Tại chương trình main

:

```
public class HelloWorld {

	public static void main(String[] args) {
		Person a = new Person("Chau", 21, 1.7f);
		Person b = new Person("Long", 24, 1.7f);
	}
}
```

```
Name:Chau
Age:21
Height:1.7
Name:Long
Age:24
Height:1.7
```

Gọi lại phương thức khởi tạo

Khi dùng **this()** thì sẽ triệu hồi phương thức khởi tạo **Constructor** của lớp hiện tại. Thường được sử dụng trong việc có nhiều phương thức khởi tạo và muốn tái sự dụng code nhiều lần:

Ví dụ: ta tạo 3 phương thức khởi tạo cho Person như sau

:

Copyright © Howkteam.com

```
public class Person {
	public String name;
	public int age;
	public float height;

	public Person(String name) {
		this.name = name;
	}

	public Person(String name, int age) {
		this(name);
		this.age = age;
	}

	public Person(String name, int age, float height) {
		this(name, age);
		this.height = height;
	}

	public void getInfo() {
		System.out.println("Name:"+this.name);
		System.out.println("Age:"+this.age);
		System.out.println("Height:"+this.height);
	}
}
```

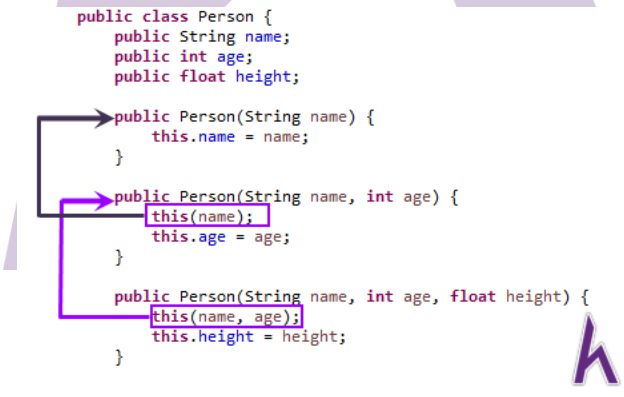
Như vậy ở phương thức main, ta có thể khởi tạo đối tượng Person theo nhiều cách khác nhau:

:

```
public class HelloWorld {

	public static void main(String[] args) {
		Person a = new Person("Chau");
		Person b = new Person("Chau", 21);
		Person c = new Person("Chau", 21, 1.7f);
	}
}
```

Việc chạy các phương thức khởi tạo có thể hiểu theo cách sau:



Trả về đối tượng (instance) của lớp hiện tại

Ta sẽ trả về **instance** của lớp hiện bằng từ khóa **this** như sau:

:

Copyright © Howkteam.com

```
public class Person {
	public String name;
	public int age;
	public float height;

	public Person(String name, int age, float height) {
		this.name = name;
		this.age = age;
		this.height = height;
	}

	public Person getInstance() {
		return this;
	}
}
```

:

```
public class HelloWorld {

	public static void main(String[] args) {
		Person a = new Person("Chau", 21, 1.7f);
		System.out.println(a);
		System.out.println(a.getInstance());
		Person b = a;
		Person c = a.getInstance();
		System.out.println(b);
		System.out.println(c);
	}
}
```

```
mypack.Person@15db9742
mypack.Person@15db9742
mypack.Person@15db9742
mypack.Person@15db9742
```

Ở kết quả đều trả về chung một giá trị, giá trị in ra theo kiểu quy tắc kiểu dữ liệu tham chiếm mà Kteam đã nói trước đó: **[tên lớp]@[vị trí lưu trữ]**

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 KẾ THỪA 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**".

Copyright © Howkteam.com