

Constructor trong Java

Constructor trong Java là một kiểu phương thức đặc biệt mà được sử dụng để khởi tạo đối tượng. Constructor được triệu hồi tại thời gian tạo đối tượng. Nó xây dựng giá trị, cung cấp dữ liệu cho đối tượng, đó là lý do nó được gọi là Constructor.

Khi bàn luận về các lớp, một trong những chủ đề quan trọng là các constructor. Mỗi lớp có ít nhất một constructor. Nếu chúng ta không viết một constructor một cách rõ ràng cho một lớp thì bộ biên dịch Java xây dựng một constructor mặc định cho lớp đó.

Mỗi khi một đối tượng mới được tạo ra, ít nhất một constructor sẽ được gọi. Quy tắc chính của các constructor là chúng có cùng tên như lớp đó. Một lớp có thể có nhiều hơn một constructor.

Sau đây là ví dụ về một constructor:

```
public class Xecon{
    public Xecon(){
    }

    public Xecon(String ten){
        // Constructor nay co mot tham so la ten.
    }
}
```

Quy tắc để tạo Constructor trong Java

Có hai loại Constructor: Loại thứ nhất là Constructor mặc định, loại này không có tham số. Và loại thứ hai là Constructor được tham số hóa.

Constructor mặc định trong Java

Đây là loại Constructor mà không có tham số. Chúng có cú pháp như sau:

```
ten_lop(){}
```

Trong ví dụ này, chúng ta tạo Constructor không có tham số trong lớp Bike. Nó sẽ được triệu hồi tại thời gian tạo đối tượng.

```
class Bike1{
```

```
Bike1(){System.out.println("Bike duoc tao");}
public static void main(String args[]){
    Bike1 b=new Bike1();
}
}
```

Qui tắc: Nếu không có Constructor nào được xác định trong một lớp, thì Compiler tự động tạo một Constructor mặc định.

Câu hỏi: Mục đích của Constructor mặc định?

Constructor mặc định cung cấp các giá trị mặc định cho đối tượng như 0, null, ... tùy thuộc vào kiểu dữ liệu.

Ví dụ Constructor mặc định mà hiển thị các giá trị mặc định

```
class Student3{
    int id;
    String name;

    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
        Student3 s1=new Student3();
        Student3 s2=new Student3();
        s1.display();
        s2.display();
    }
}
```

Giải thích: Trong lớp trên, bạn không tạo bất cứ Constructor nào, vì thế Compiler cung cấp cho bạn một Constructor mặc định. Ở đây các giá trị 0 và null được cung cấp bởi Constructor mặc định.

Constructor được tham số hóa trong Java

Một Constructor mà có các tham số được gọi là constructor được tham số hóa.

Câu hỏi: Tại sao sử dụng Constructor được tham số hóa?

Constructor được tham số hóa được sử dụng để cung cấp các giá trị khác nhau cho các đối tượng riêng biệt.

Ví dụ về Constructor được tham số hóa:

Trong ví dụ này, chúng ta tạo Constructor của lớp Student mà có hai tham số. Chúng ta có thể có bất cứ số tham số nào trong Constructor.

```
class Student4{
    int id;
    String name;

    Student4(int i,String n){
        id = i;
        name = n;
    }
    void display(){System.out.println(id+" "+name);}

    public static void main(String args[]){
        Student4 s1 = new Student4(111,"Karan");
        Student4 s2 = new Student4(222,"Aryan");
        s1.display();
        s2.display();
    }
}
```

Nạp chồng Constructor trong Java

Nạp chồng Constructor là một kỹ thuật trong Java mà trong đó một lớp có thể có bất cứ số lượng Constructor nào mà khác nhau về danh sách tham số. Trình biên dịch Compiler phân biệt sự khác nhau giữa các Constructor này bằng việc phân tích số tham số trong danh sách và kiểu của chúng.

Ví dụ về nạp chồng Constructor

```
class Student5{
    int id;
    String name;
```

```
int age;
Student5(int i,String n){
    id = i;
    name = n;
}
Student5(int i,String n,int a){
    id = i;
    name = n;
    age=a;
}
void display(){System.out.println(id+" "+name+" "+age);}

public static void main(String args[]){
    Student5 s1 = new Student5(111,"Hoang");
    Student5 s2 = new Student5(222,"Thanh",25);
    s1.display();
    s2.display();
}
}
```

Điểm khác nhau giữa Constructor và Phương thức trong Java

Bảng dưới đây liệt kê một số điểm khác nhau giữa Constructor và phương thức trong Java:

Constructor	Phương thức
Constructor được sử dụng để khởi tạo trạng thái của một đối tượng	Phương thức được sử dụng để trưng bày hành vi của một đối tượng
Constructor phải không có kiểu trả	Phương thức phải có kiểu trả về

về	
Constructor được triệu hồi một cách ngầm định	Phương thức phải được triệu hồi một cách tường minh
Compiler cung cấp một Constructor mặc định nếu bạn không có bất cứ Constructor nào	Phương thức không được cung cấp bởi Compiler trong bất cứ trường hợp nào
Tên Constructor phải giống tên lớp	Tên phương thức có thể hoặc không giống như tên lớp

Copy Constructor trong Java

Trong Java không có Copy Constructor như trong C++. Tuy nhiên, bạn có thể sao chép các giá trị của một đối tượng tới đối tượng khác. Có nhiều cách để thực hiện việc sao chép các giá trị, đó là:

- Bởi Constructor
- Bởi gán các giá trị của một đối tượng vào trong đối tượng khác.
- Bởi phương thức clone() của lớp Object.

Trong ví dụ này, chúng ta sao chép các giá trị của một đối tượng vào trong đối tượng khác bởi sử dụng Constructor trong Java.

```
class Student6{  
    int id;  
    String name;
```

```
Student6(int i,String n){
    id = i;
    name = n;
}

Student6(Student6 s){
    id = s.id;
    name =s.name;
}

void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
    Student6 s1 = new Student6(111,"Hoang");
    Student6 s2 = new Student6(s1);
    s1.display();
    s2.display();
}
}
```

Sao chép các giá trị mà không sử dụng Constructor

Chúng ta có thể sao chép các giá trị của một đối tượng vào trong đối tượng khác bởi gán các giá trị của đối tượng đó vào trong đối tượng khác. Trong trường hợp này, chúng ta không cần tạo Constructor.

```
class Student7{
    int id;
    String name;
    Student7(int i,String n){
        id = i;
        name = n;
    }
    Student7(){}
```

```
void display(){System.out.println(id+" "+name);}
```

```
public static void main(String args[]){
    Student7 s1 = new Student7(111,"Hoang");
    Student7 s2 = new Student7();
    s2.id=s1.id;
    s2.name=s1.name;
    s1.display();
    s2.display();
}
}
```

Câu hỏi: Có phải Constructor trả về giá trị?

Có, đó là instance (sự thể hiện) của lớp hiện tại. (bạn không thể sử dụng kiểu trả về, tuy vậy nó trả về một giá trị).

Câu hỏi: Constructor có thể thực hiện các tác vụ khác ngoài khởi tạo không?

Có, giống như quá trình tạo đối tượng, bắt đầu một Thread, gọi phương thức, ... Bạn có thể thực hiện bất cứ hoạt động nào trong Constructor như khi bạn thực hiện trong phương thức.