

 Cập nhật tháng 8 năm 2024

[Bài đọc] Lớp trong Java và từ khóa this

1. Thành phần của Lớp trong Java.

Một lớp trong Java bao gồm ba thành phần chính: **thuộc tính**, **phương thức**, và **constructor**.

a. Thuộc tính (Attributes).

- Là các biến được khai báo trong lớp, đại diện cho trạng thái hoặc dữ liệu của một đối tượng.
- **Cú pháp:**

```
class ClassName {  
    // Thuộc tính  
    private String name;  
    private int age;  
}
```

- Thường sử dụng các phạm vi truy cập như private để đảm bảo tính đóng gói.
- Truy cập thông qua các phương thức getter và setter.

b. Phương thức (Methods).

- Là các hàm được định nghĩa trong lớp để thực hiện hành vi của đối tượng.
- **Cú pháp:**

```
class ClassName {  
    // Phương thức  
    public void display() {  
        System.out.println("This is a method.");  
    }  
}
```

- Phương thức có thể trả về giá trị (return type) hoặc không (void).
- Các loại phương thức:
 - **Getter/Setter:** Truy cập và chỉnh sửa giá trị thuộc tính.

- **Static Method:** Được gọi mà không cần tạo đối tượng.
- **Instance Method:** Liên kết với đối tượng cụ thể.

c. Constructor.

- Là phương thức đặc biệt dùng để khởi tạo đối tượng.
- Có tên giống tên lớp, không có kiểu trả về.
- **Cú pháp:**

```
class ClassName {
    // Constructor
    public ClassName(String name, int age) {
        this.name = name;
        this.age = age;
    }
}
```

- Có thể tạo nhiều constructor khác nhau (constructor overloading) với tham số khác nhau.

2. Định nghĩa Lớp trong Java.

Lớp là bản thiết kế (blueprint) cho các đối tượng, bao gồm định nghĩa thuộc tính, phương thức và constructor.

- **Cú pháp chung:**

```
public class ClassName {
    // Thuộc tính
    private String name;
    private int age;

    // Constructor
    public ClassName(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Phương thức
    public void display() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
```

- Từ khóa **this**:
 - Dùng để tham chiếu đến đối tượng hiện tại.
 - Thường được dùng để phân biệt giữa thuộc tính và tham số có cùng tên.

- Ví dụ:

```
public void setName(String name) {  
    this.name = name; // 'this.name' là thuộc tính, 'name' là tham số.  
}
```

3. Ví dụ minh họa đầy đủ.

```
public class Student new *  
{  
    // Thuộc tính  
    private String name; 4 usages  
    private int age; 2 usages  
  
    // Constructor  
    public Student(String name, int age) 1usage new *  
    {  
        this.name = name; // Sử dụng từ khóa this  
        this.age = age;  
    }  
  
    // Getter  
    public String getName() { return name; }  
  
    // Setter  
    public void setName(String name) { this.name = name; }  
  
    // Phương thức hiển thị thông tin  
    public void displayInfo() { System.out.println("Name: " + name + ", Age: " + age); }  
  
    // Main method để kiểm tra  
    public static void main(String[] args) new *  
    {  
        Student student = new Student( name: "John", age: 20); // Sử dụng constructor  
        student.displayInfo(); // Gọi phương thức  
    }  
}
```

Kết quả:

```
Name: John, Age: 20
```

Danh sách các bài học

