Kiến Thức Đầy Đủ về Từ Khóa this trong Java

Trong Java, từ khóa this là một biến tham chiếu đặc biệt, luôn tồn tại trong các phương thức và constructor của lớp. Nó trỏ tới đối tượng hiện tại (current object) – đối tượng đang gọi phương thức hoặc đang được khởi tạo. Dưới đây là 6 cách sử dụng phổ biến và tổng quát nhất của this.

1. Truy cập biến instance (this.variable)

Dùng để phân biệt giữa biến instance và biến cục bộ hoặc tham số bị trùng tên.

```
class Student {
  String name;
  int age;

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

  void display() {
    System.out.println("Name: " + this.name);
    System.out.println("Age: " + this.age);
  }
}

public class Main {
  public static void main(String[] args) {
    Student s1 = new Student("An", 20);
    s1.display();
  }
}
```

---

2. Gọi phương thức khác (this.method())

Dùng để gọi một phương thức khác trong cùng class, giúp code rõ ràng hơn.

```
class Demo {
  void method1() {
    System.out.println("Method 1 is called");
  }

  void method2() {
    System.out.println("Method 2 is calling method 1...");
```

```
        this.method1();
    }
}

public class Main {
    public static void main(String[] args) {
        Demo obj = new Demo();
        obj.method2();
    }
}
```

---

3. Gọi constructor khác (this(...))

Dùng để tái sử dụng code bằng cách gọi một constructor khác trong cùng class. Lệnh this(...) phải đứng đầu constructor.

```
class Book {
    String title;
    double price;

    Book() {
        this("Unknown", 0.0);
    }

    Book(String title, double price) {
        this.title = title;
        this.price = price;
    }

    void display() {
        System.out.println(title + " - " + price + "$");
    }
}

public class Main {
    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book("Java Basics", 15.5);
        b1.display();
        b2.display();
    }
}
```

---

4. Truyền object hiện tại (method(this))

Dùng để truyền chính đối tượng hiện tại làm tham số cho phương thức hoặc constructor khác.

```
class Test {
  void display(Test obj) {
    System.out.println("Display method is called with object: " +
obj);
  }

  void call() {
    display(this);
  }
}

public class Main {
  public static void main(String[] args) {
    Test t = new Test();
    t.call();
  }
}
```

---

5. Trả về object hiện tại (return this)

Dùng nhiều trong method chaining, cho phép gọi liên tiếp các phương thức trên cùng một object.

```
class Calculator {
  int value;

  Calculator add(int n) {
    this.value += n;
    return this;
  }

  Calculator multiply(int n) {
    this.value *= n;
    return this;
  }

  void display() {
    System.out.println("Result: " + value);
  }
}
```

```
public class Main {
  public static void main(String[] args) {
    Calculator c = new Calculator();
    c.add(5).multiply(3).display();
  }
}
```

---

6. Truy cập lớp bao ngoài (Outer.this)

Dùng trong inner class để tham chiếu tới object của outer class khi có biến hoặc phương thức bị trùng tên.

```
class Outer {
  int num = 10;

  class Inner {
    int num = 20;

    void printNums() {
      System.out.println("Inner num: " + this.num);
      System.out.println("Outer num: " + Outer.this.num);
    }
  }
}

public class Main {
  public static void main(String[] args) {
    Outer outer = new Outer();
    Outer.Inner inner = outer.new Inner();
    inner.printNums();
  }
}
```

Tóm tắt nhanh

Trường hợp	Cách dùng	Mục đích
1	this.variable	Truy cập biến instance bị che khuất bởi biến cục bộ
2	this.method()	Gọi phương thức khác trong cùng class
3	this(...)	Gọi constructor khác trong

		cùng class
4	method(this)	Truyền object hiện tại làm tham số
5	return this	Trả về object hiện tại để method chaining
6	Outer.this	Truy cập object của lớp bao ngoài từ inner class