Ministry of SMEs and Startups × KOSME Korea SMEs and Startups Agency × LIKELION

Cú pháp cơ bản trong Java

Nguyễn Anh Tuấn

K TECH COLLEGE

Nội dung bài giảng

1 Cú pháp cơ bản trong Java

2 Biến và Kiểu dữ liệu

3 Toán tử

4 Lớp và Đối tượng

5 Kế thừa “extends"

6 Đa hình “override"

7 Biến và Phương thức tĩnh

Cú pháp cơ bản trong Java

Cú pháp cơ bản trong Java

1. Phương thức **main**

```
/*
định nghĩa 1 lớp công khai là main
phương thức main là phương thức khởi động của ứng dụng Java
*/
public static void main(String[] args) {
        // todo();
}
```

2. Cú pháp in kết quả ra màn hình console

```
public static void main(String[] args) {
        System.out.println("Hello World!");     // Cấu trúc in ra màn hình console
}
```

Biến và Kiểu dữ liệu

Kiểu dữ liệu

1. Kiểu dữ liệu nguyên thuỷ

- Kiểu số nguyên: int
  - *int number = 1;*
- Kiểu số nguyên: long
  - *long number = 1L;*
- Kiểu số thập phân: float
  - *float number = 1.2f;*
- Kiểu số thập phân: double
  - *double number = 12.3456789;*
- Kiểu ký tự: char
  - *char letter = 'a';*
- Kiểu điều kiện: boolean
  - *boolean flag = true;*

2. Kiểu dữ liệu tham chiếu

- Kiểu chuỗi ký tự: string
  - *String str = "chuỗi ký tự";*
- Kiểu mảng: array
  - *int[] array = {0, 1, 2, 3};*
- Kiểu đối tượng: object
  - *People people = new People();*

Kiểu điều kiện

- if - else
  - ```
    if (condition) {
        todo();
    } else {
        todo();
    }
    ```
  - ```
    if (condition) {
        todo();
    } else if (condition) {
        todo();
    } else {
        todo();
    }
    ```
- switch - case
  - ```
    int condition = 2;
    switch (condition) {
        case 1:
            todo();
            break;
        case 2:
            todo();
            break;
        default:
            todo();
            break;
    }
    ```
    ```

Kiểu vòng lặp

- for
  - ➢ *for (init; condition; incre/decre) {*
    todo();
    }
- foreach
  - ➢ *int[] index = {1, 2, 3, 4, 5};*
    for (type element: array) {
    todo();
    }
- while
  - ➢ *while (condition) {*
    todo();
    }
- do-while
  - ➢ *do {*
    todo();
    } while (condition);

Toán tử

Toán tử số học

Các toán tử để thực hiện các phép tính số học như cộng, trừ, nhân, chia, và lấy phần dư. Ví dụ:

- int sum = 5 + 3; // Phép cộng
- int difference = 5 - 3; // Phép trừ
- int product = 5 * 3; // Phép nhân
- int quotient = 5 / 3; // Phép chia
- int remainder = 5 % 3; // Lấy phần dư

Toán tử so sánh

Các toán tử để so sánh hai giá trị. Ví dụ:

- boolean isEqual = (5 == 3); // Kiểm tra bằng
- boolean isNotEqual = (5 != 3); // Kiểm tra không bằng
- boolean isGreater = (5 > 3); // Kiểm tra lớn hơn
- boolean isGreaterOrLesser = (5 >= 3); // Kiểm tra lớn hơn hoặc bằng
- boolean isLesser = (5 < 3); // Kiểm tra nhỏ hơn
- boolean isLesserOfGreater = (5 <= 3); // Kiểm tra nhỏ hơn hoặc bằng

Toán tử logic

Các toán tử để kết hợp các điều kiện logic. Ví dụ:

- boolean andCondition = (5 > 3 && 5 < 10); // Toán tử **AND**
- boolean orCondition = (5 > 3 || 5 > 10); // Toán tử **OR**
- boolean notCondition = !(5 > 3); // Toán tử **NOT**

Lớp và Đối tượng

Lớp và Đối tượng

Lớp là một tập hợp các thuộc tính và phương thức để tạo ra các đối tượng. Ví dụ:

- *public class Person { ... }* *// Định nghĩa lớp Person*
- *Person person = new Person("John", 30);* *// Tạo đối tượng Person*

Kế thừa

Kế thừa

Sử dụng từ khoá **extends**, cho phép một lớp kế thừa thuộc tính và phương thức của lớp khác. Ví dụ:

- *public class Dog **extends** Animal { ... }* *// Lớp con (subclass) **Dog** kế thừa từ lớp cha (superclass) **Animal***

Đa hình

Đa hình

Override là khái niệm ghi đè trong lập trình hướng đối tượng, cho phép một lớp con triển khai cụ thể cho một phương thức đã được định nghĩa trong lớp cha. Ví dụ:

- ```
  class Animal {
      public void makeSound() {
          System.out.println("Animal makes a sound");
      }
  }

  class Dog extends Animal {
      @Override
      public void makeSound() {
          System.out.println("Dog barks");
      }
  }
  ```
  ```

Biến và Phương thức tĩnh

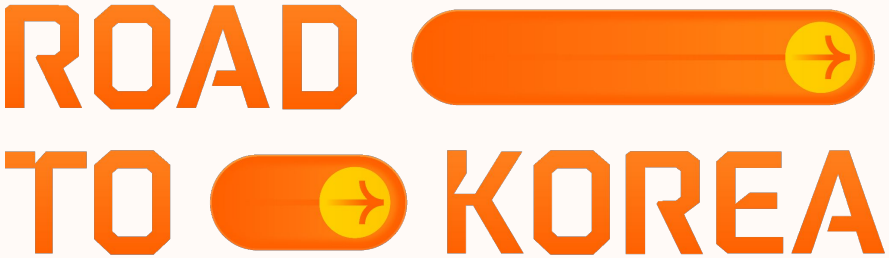
Biến và Phương thức tĩnh

Phương thức và biến tĩnh có thể được gọi mà không cần tạo đối tượng. Ví dụ:

```
public class A {

    public static functionA {
        todo();
    }

    public static void main(String[] args) {
        functionA.todo();
    }
}
```



Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx