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

OOP, Classes

Nguyễn Anh Tuấn

K TECH COLLEGE

Nội dung bài giảng

1 OOP

2 Classes

OOP

Đặc tính của Java

- **4 đặc tính của Java:**
  - Hướng đối tượng
  - Độc lập nền tảng
  - Đa luồng
  - Bảo mật

Object-Oriented Programming (OOP)

- **Định nghĩa**: Object-Oriented Programming (OOP) là một mô hình lập trình dựa trên khái niệm về các đối tượng. Các đối tượng là các thực thể có trạng thái (dữ liệu) và hành vi (phương thức). OOP được sử dụng để thiết kế phần mềm có cấu trúc rõ ràng và dễ bảo trì.
- **4 nguyên tắc chính của OOP trong Java**:
  - **Encapsulation (Đóng gói)**: Giấu dữ liệu và chỉ cho phép truy cập qua các phương thức công khai.
  - **Inheritance (Kế thừa)**: Một lớp có thể kế thừa các thuộc tính và phương thức từ một lớp khác.
  - **Polymorphism (Đa hình)**: Một phương thức có thể có nhiều hình thức khác nhau.
  - **Abstraction (Trừu tượng)**: Ẩn các chi tiết thực thi và chỉ hiển thị chức năng cần thiết.

Encapsulation (Đóng gói)

❖ **Định nghĩa**: Là kỹ thuật giấu dữ liệu bằng cách ẩn các thuộc tính của đối tượng và chỉ cho phép truy cập hoặc thay đổi chúng thông qua các phương thức công khai (public methods).

- ➢ Các thuộc tính của một lớp được khai báo là private, ngăn chặn truy cập trực tiếp từ bên ngoài lớp.
- ➢ Phương thức getter và setter cho phép truy cập và thay đổi giá trị của các thuộc tính này.
- ➢ Nguyên tắc đóng gói giúp bảo vệ dữ liệu duy trì tính toàn vẹn của dữ liệu.

```
// Định nghĩa lớp Father với tính đóng gói
class Father {

        private String name;                                    // Thuộc tính private
        public String getName() { return name; }                // Getter cho name
        public void setName(String name) { this.name = name; }  // Setter cho name
        public Father(String name) { this.name = name; }        // Constructor
}
```

Inheritance (Kế thừa)

- **Định nghĩa**: Là cơ chế mà trong đó một lớp (lớp con) có thể kế thừa các thuộc tính và phương thức từ một lớp khác (lớp cha). Điều này cho phép tái sử dụng mã nguồn và tạo ra các mối quan hệ giữa các lớp.
  - Lớp con kế thừa tất cả các thuộc tính và phương thức của lớp cha nhưng có thể thêm các thuộc tính và phương thức mới hoặc ghi đè (override) các phương thức của lớp cha.
  - Inheritance giúp tái sử dụng mã nguồn và giảm thiểu sự lặp lại trong lập trình.
  - Kế thừa theo hướng "is-a", tức là lớp con là một kiểu của lớp cha.

```
// Định nghĩa lớp Son với tính kế thừa
class Son extends Father {
        public Son(String name) { super(name); }
}
```

Classes

Classes

❖ **Định nghĩa:** Lớp trong Java là một khuôn mẫu cho các đối tượng. Nó định nghĩa các thuộc tính và phương thức mà các đối tượng của lớp đó sẽ có.

- ➢ **Thuộc tính (Attributes/Fields)**: Các biến đại diện cho trạng thái của đối tượng.
- ➢ **Phương thức (Methods)**: Các hàm đại diện cho hành vi của đối tượng.
- ➢ **Constructor**: Phương thức đặc biệt được gọi khi tạo một đối tượng mới.

```
// Định nghĩa lớp Father
class Father {

        private String name;                                    // Thuộc tính
        public String getName() { return name; }                // Phương thức lấy tên
        public void setName(String name) { this.name = name; }  // Phương thức đặt tên
        public Father(String name) { this.name = name; }        // Constructor

}
```

Các loại Class

❖ **Concrete Class (Lớp Cụ Thể)**: Một lớp cụ thể chứa cả thuộc tính và phương thức, và có thể bao gồm cả constructor.

```
// Định nghĩa lớp Father
class Father {

    private String name;                                    // Thuộc tính
    public String getName() { return name; }                // Phương thức lấy tên
    public void setName(String name) { this.name = name; }  // Phương thức đặt tên
    public Father(String name) { this.name = name; }        // Constructor

}
```

Các loại Class

❖ **Abstract Class (Lớp Trừu Tượng)**: Là lớp không thể được tạo instance trực tiếp. Nó có thể chứa các phương thức trừu tượng (không có phần thân) và các phương thức cụ thể (có phần thân) mà các lớp con phải triển khai.

```
// Định nghĩa lớp trừu tượng People
public abstract class People {
        private String name;
        public People(String name) { this.name = name; }
        public String getName() { return name; }
        public abstract void makeNoise();                    // Phương thức trừu tượng
}

public class Father extends People {
        public Dog(String name) { super(name); }
        @Override
        public void Noise() { System.out.println("Tạo ra tiếng ồn"); }    // Kế thừa từ lớp People
}
```

Các loại Class

❖ **Interface Class (Lớp Giao Diện)**: Interface định nghĩa các phương thức mà một lớp phải triển khai. Interface không thể chứa bất kỳ phương thức có phần thân nào (ngoại trừ các phương thức Default và Static từ Java 8).

```
// Định nghĩa lớp giao diện People
public Interface People {
        void eat();
        void sleep();                    // Phương thức không có phần thân
}

public class Father implements People {
        // Triển khai chức năng eat() của lớp People
        @Override public void eat() { System.out.println("Father is eating"); }

        // Triển khai chức năng sleep() của lớp People
        @Override public void sleep() { System.out.println("Father is sleeping"); }
}
```

Các loại Class

- **Nested Class (Lớp Lồng Nhau)**: Là một lớp được định nghĩa bên trong một lớp khác. Có bốn loại lớp lồng nhau trong Java:
  - **Static Nested Class (Lớp Lồng Nhau Tĩnh)**: Được khai báo với từ khóa static.
  - **Non-static Nested Class (Inner Class)**: Không có từ khóa static.
  - **Local Class (Lớp Cục Bộ)**: Được định nghĩa bên trong một phương thức.
  - **Anonymous Class (Lớp Vô Danh)**: Không có tên và thường được sử dụng để triển khai các interface hoặc lớp trừu tượng.

- **Singleton Class (Lớp Đơn Nhất)**: Là lớp chỉ cho phép tạo một instance duy nhất. Điều này thường được thực hiện bằng cách sử dụng một constructor private và một phương thức static để trả về instance duy nhất.

ROAD TO KOREA

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

mail@mail.com hoặc Zalo 0xxx xxx xxx