인터페이스와추상클래스

자바의 상속과 인터페이스 개념 정리

1. 상속(Inheritance)
상속은 부모 클래스의 속성과 메소드를 자식 클래스가 물려받는 것을 의미합니다. 부모 클래스는 공통적인 기능이나 속성을 가지고 있고, 자식 클래스는 이를 상속받아 더 구체적인 기능을 추가하거나 변경할 수 있습니다. 상속 관계는 1:1 또는 1:다 관계로 설정될 수 있지만, 부모 클래스가 여러 개일 수는 없습니다. 이를 다중 상속이 불가능하다고 말합니다.

예시:

```
class Animal {
    void sound() {
        System.out.println("Animal sound");
    }
}

class Dog extends Animal {
    @Override
    void sound() {
        System.out.println("Bark");
    }
}
```

2. 인터페이스(Interface)
인터페이스는 상속과 비슷하지만 중요한 차이가 있습니다. 상속에서는 부모 클래스가 하나만 가능하지만, 인터페이스는 여러 개의 부모 인터페이스를 가질 수 있습니다. 이는 다중 상속의 제약을 극복하기 위한 방법입니다. 인터페이스는 메소드의 형태(선언부)만 정의하며, 실제 구현은 이를 상속받은 클래스에서 합니다.

또한, 인터페이스는 결합도를 낮추고 응집도를 높이는 데 도움을 줍니다. 즉, 코드가 서로 강하게 결합되지 않아 하나의 부분에 문제가 생기더라도 다른 부분에 영향을 최소화할 수 있게 합니다.

인터페이스 예시:

```
interface Animal {
    void sound();
}

interface Human {
    void walk();
}

class Dog implements Animal, Human {
    @Override
    public void sound() {
        System.out.println("Bark");
    }

    @Override
    public void walk() {
        System.out.println("Walk like a dog");
    }
}
```

3. 추상 클래스(Abstract Class)

추상 클래스는 일반 클래스와 인터페이스의 중간 개념입니다. 추상 클래스는 일반 메소드와 추상 메소드(구현되지 않은 메소드)를 모두 가질 수 있습니다. 추상 메소드는 인터페이스와 마찬가지로 선언만 하고, 실제 구현은 상속받는 클래스에서 합니다. 하지만, 추상 클래스는 상속받을 때 부모 클래스가 하나만 가능하다는 점에서 다중 상속을 지원하지 않습니다.

추상 클래스 예시:

```
abstract class Animal {
    abstract void sound();

    void sleep() {
        System.out.println("Animal sleeps");
    }
}

class Dog extends Animal {
```

```
    @Override
    void sound() {
        System.out.println("Bark");
    }
}
```

4. 상속과 인터페이스 비교

- **상속**: 부모 클래스에서 자식 클래스가 속성과 메소드를 상속받으며, 다중 상속은 불가능.
- **인터페이스**: 메소드의 선언만 하고, 구현은 상속받는 클래스에서 하며, 다중 상속이 가능.
- **추상 클래스**: 일반 메소드와 추상 메소드를 모두 가질 수 있으며, 다중 상속은 불가능.

5. 결론
일반적으로 인터페이스는 코드의 일관성을 유지하고 결합도를 낮추는 데 유리하기 때문에 더 자주 사용됩니다. 추상 클래스는 필요한 경우에만 사용되며, 인터페이스보다 사용 빈도가 낮습니다. 인터페이스와 추상 클래스의 차이점을 명확히 이해하고 적절하게 사용하는 것이 중요합니다.
