



상속과생성자

자바 상속과 생성자

이번 포스트에서는 자바의 **상속**과 **생성자**에 대해 살펴보겠습니다. 이 두 가지 개념은 객체지향 프로그래밍에서 매우 중요한 역할을 하며, 특히 상속은 자식 클래스가 부모 클래스의 속성과 메서드를 물려받아 재사용하고 확장할 수 있는 기능을 제공합니다. 생성자는 객체가 생성될 때 호출되며, 객체의 초기 상태를 설정하는 데 중요한 역할을 합니다.

1. 상속(Inheritance)

상속은 객체지향 프로그래밍에서 **부모 클래스**의 속성과 메서드를 **자식 클래스**가 물려받는 것을 의미합니다. 이를 통해 코드의 중복을 줄이고, 유지보수를 쉽게 할 수 있습니다.

상속의 기본 개념

- **부모 클래스**: 속성과 메서드를 정의한 클래스.
- **자식 클래스**: 부모 클래스의 속성과 메서드를 물려받아 사용할 수 있는 클래스.

상속의 문법

```
class 부모클래스 {  
    // 속성 및 메서드 정의  
}  
  
class 자식클래스 extends 부모클래스 {  
    // 자식클래스의 속성 및 메서드 정의  
}
```

자식 클래스는 `extends` 키워드를 사용하여 부모 클래스를 상속받습니다.

상속의 특징

1. **코드 재사용성**: 부모 클래스의 속성과 메서드를 재사용함으로써 코드를 효율적으로 작성할 수 있습니다.

2. **유연한 확장**: 자식 클래스에서 부모 클래스의 메서드를 ****오버라이딩(Overriding)****하여 수정할 수 있습니다. 오버라이딩은 자식 클래스가 부모 클래스의 메서드를 새롭게 정의하는 기능입니다.
3. **다형성(Polymorphism)**: 부모 클래스 타입의 참조 변수가 자식 클래스의 객체를 참조할 수 있어 다양한 객체를 하나의 타입으로 처리할 수 있습니다.

상속 예시

```
class 동물 {
    String 이름;

    void 소리내기() {
        System.out.println("동물이 소리를 냅니다.");
    }
}

class 강아지 extends 동물 {
    @Override
    void 소리내기() {
        System.out.println("강아지가 멍멍 짭니다.");
    }
}
```

이 예시에서는 **강아지** 클래스가 **동물** 클래스를 상속받아 **소리내기** 메서드를 오버라이딩하고 있습니다.

2. 생성자(Constructor)

생성자는 클래스의 객체가 생성될 때 호출되어 객체의 초기 상태를 설정하는 특별한 메서드입니다. 생성자는 클래스와 같은 이름을 가지며, 반환 타입이 없습니다.

생성자의 종류

1. **기본 생성자**: 매개변수가 없는 생성자.

```
class 자동차 {
    String 모델명;

    자동차() {
```

```

        모델명 = "기본 모델";
    }
}

```

객체가 생성될 때 자동으로 호출되며, 명시적으로 정의하지 않으면 자바에서 기본 생성자를 자동으로 제공합니다.

2. **매개변수가 있는 생성자:** 객체를 생성할 때 값을 전달하여 초기 상태를 설정하는 생성자.

```

class 자동차 {
    String 모델명;

    자동차(String 모델명) {
        this.모델명 = 모델명;
    }
}

```

3. **생성자 오버로딩:** 여러 개의 생성자를 정의하여 다양한 방법으로 객체를 생성할 수 있습니다.

```

class 자동차 {
    String 모델명;
    int 속도;

    자동차() {
        모델명 = "기본 모델";
        속도 = 0;
    }

    자동차(String 모델명, int 속도) {
        this.모델명 = 모델명;
        this.속도 = 속도;
    }
}

```

super()와 this()

- **super():** 자식 클래스의 생성자에서 부모 클래스의 생성자를 호출할 때 사용합니다.
- **this():** 같은 클래스 내에서 다른 생성자를 호출할 때 사용합니다.

생성자 예시

```
class 동물 {
    String 이름;

    동물(String 이름) {
        this.이름 = 이름;
    }
}

class 강아지 extends 동물 {
    String 종류;

    강아지(String 이름, String 종류) {
        super(이름); // 부모 클래스의 생성자 호출
        this.종류 = 종류;
    }
}
```

이 예시에서 `super(이름)` 을 통해 부모 클래스의 생성자를 호출하여 `이름` 을 초기화합니다.

결론

상속과 생성자는 자바의 객체지향 프로그래밍에서 핵심적인 역할을 합니다. 상속을 통해 코드의 재사용성과 유지보수성을 높일 수 있으며, 생성자를 통해 객체의 초기 상태를 설정할 수 있습니다. 이 두 가지 개념을 제대로 이해하면 자바 프로그래밍에서 더욱 효율적인 코드를 작성할 수 있을 것입니다.