

Java 입문

상속

- 특정 클래스의 속성과 기능을 다른 클래스에 계승하는 방식으로 속성과 기능을 계승하는 클래스를 부모 클래스, 속성과 기능을 물려받는 클래스를 자식 클래스라 한다.
- `extends` 키워드를 통해 상속관계를 만든다.

```
public class Car {  
    public void move(){  
        System.out.println("차를 이동합니다.");  
    }  
}
```

```
public class ElectricCar extends Car {  
    public void charge(){  
        System.out.println("충전합니다.");  
    }  
}
```

- 위 코드에서 `Car`는 부모 클래스, `ElectricCar`는 자식 클래스로 `ElectricCar` 클래스에서는 부모 클래스인 `Car` 클래스의 `move()`메서드에 접근할 수 있다.

```
public class CarMain {  
    public static void main(String[] args) {  
        ElectricCar electricCar = new ElectricCar();  
        electricCar.move();  
        electricCar.charge();  
  
        GasCar gasCar = new GasCar();  
        gasCar.move();  
        gasCar.fillUp();  
    }  
}
```

- 공통으로 사용되는 기능은 부모 클래스에 정의하여 해당 클래스를 상속받는 자식 클래스들은 공통 기능을 자식 클래스 내에 정의하지 않아도 사용할 수 있어 코드의 중복을 제거할 수 있으며 유지보수와 확장이 편리해진다.
- 자바에서는 단일 상속만 가능하다.

다형성

- 하나의 객체가 여러 가지 타입을 가질 수 있는 성질.
- 부모 클래스에 있는 메서드를 자식 클래스에서 자식 클래스에 맞게 변형할 수 있는데 이를 메서드 오버라이딩이라고 한다.

```
public class Car {
    public void move(){
        System.out.println("차를 이동합니다.");
    }

    // 문열기 기능
    public void openDoor(){
        System.out.println("문을 엽니다.");
    }
}
```

```
public class ElectricCar extends Car {
    public void charge(){
        System.out.println("충전합니다.");
    }

    // 메서드 오버라이딩
    @Override
    public void move(){
        System.out.println("전기차를 빠르게 이동합니다.");
    }
}
```

```
public class Main{
    public static void main(String[] args){
        Car car1 = new Car();
    }
}
```

```
Car car2 = new ElectricCar();
```

```
car1.move(); // 결과 : 차를 이동합니다.
```

```
car2.move(); // 결과 : 전기차를 빠르게 이동합니다.
```

- move()라는 메서드를 자식 클래스인 ElectricCar에서 오버라이딩하여 클래스의 특성에 맞는 결과를 출력한다.

추상 클래스

- 객체의 공통 특징이나 속성을 추출하는 개념.
- 추상적인 개념을 제공하는 클래스로 인스턴스가 존재하지 않고, 상속을 목적으로 사용되며 부모 클래스의 역할을 한다.
- 클래스를 선언할 때 abstract 키워드를 붙이면 추상 클래스가 된다.
- 메서드 선언할 때 abstract 키워드를 붙이면 추상 메서드가 되는데 추상 메서드가 하나라도 있으면 해당 클래스는 추상 클래스가 되고, 추상 메서드는 반드시 추상 클래스를 상속받는 클래스에서 오버라이딩하여 사용해야 한다.

```
public abstract class AbstractAnimal {  
    public abstract void sound(); // 자식 클래스에서 반드시 오버라이딩 해야 한다.  
  
    public void move(){  
        System.out.println("동물이 움직입니다.");  
    }  
}
```

```
public class Cat extends AbstractAnimal {  
  
    @Override  
    public void sound() {  
        System.out.println("야옹");  
    }  
}
```

```
public class Dog extends AbstractAnimal{  
    @Override  
    public void sound() {  
        System.out.println("멍멍");  
    }  
}
```