



클래스의 상속 1

클래스의 상속 1

● 상속의 정의

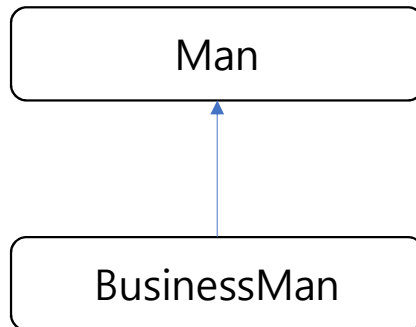
- 상위 클래스의 특성(멤버 변수, 메소드)을 그대로 이어 받아 확장하여 사용
- 상속의 예시 : 동물(포유류의 상위(부모)클래스) -> 포유류(고래의 상위(부모)클래스이자 동물의 하위(자식)클래스) -> 고래(포유류의 하위(자식) 클래스)
- 상속을 위해 extends 키워드 사용

```
public class Man {  
    protected String name;  
    public void tellYourName() {  
        System.out.println("My name is " + this.name);  
    }  
}
```

```
public class BusinessMan extends Man {  
    private String company;  
    private String position;  
    public void tellYourInfo() {  
        System.out.println("My company is " + this.company);  
        System.out.println("My position is " + this.position);  
        tellYourName(); // Man 클래스를 상속했기 때문에 호출 가능!  
    }  
}
```

클래스의 상속 1

- Man 클래스
 - 상속의 대상이 되는 클래스
 - 상위 클래스, 부모 클래스
- BusinessMan
 - 상속을 하는 클래스
 - 하위 클래스, 자식 클래스
- Man 클래스와 BusinessMan 클래스의 UML 상속 관계 표현



클래스의 상속 1

- 상속과 생성자

```
public class BusinessMan extends Man {
    private String company;
    private String position;
    public BusinessMan(String name, String company, String position) {
        this.name = name;    // 상위 클래스 Man의 멤버 변수 초기화
        this.company = company;
        this.position = position;
    }
    public void tellYourInfo() {
        System.out.println("My company is " + this.company);
        System.out.println("My position is " + this.position);
        tellYourName(); // Man 클래스를 상속했기 때문에 호출 가능!
    }
}
```

```
public class MyBusinessMan {
    public static void main(String[] args) {
        BusinessMan man = new BusinessMan("YOON", "kakao", "staff");
        man.tellYourInfo();
    }
}
```

클래스의 상속 1

- 상속 관계에서 생성자의 호출

```
public class SuperCLS {  
    public SuperCLS() {  
        System.out.println("I'm Super Class");  
    }  
}
```

```
public class SubCLS extends SuperCLS {  
    public SubCLS() {  
        System.out.println("I'm Sub Class");  
    }  
}
```

```
public class SuperSubCon {  
    public static void main(String[] args) {  
        new SubCLS();  
    }  
}
```

- 하위 클래스의 인스턴스 생성 시 상위 클래스, 하위 클래스의 생성자가 모두 호출된다
- 하위 클래스의 인스턴스 생성 시 상위 클래스의 생성자가 먼저 호출된다.

클래스의 상속 1

- 키워드 super를 사용한 상위클래스의 생성자 호출

```
public class SuperCLS {  
    public SuperCLS() {  
        System.out.println("Con : SuperCLS()");  
    }  
    public SuperCLS(int i) {  
        System.out.println("Con : SuperCLS(int i)");  
    }  
    public SuperCLS(int i, int j) {  
        System.out.println("Con : SuperCLS(int i, int j)");  
    }  
}
```

```
public class SuperSubCon2 {  
    public static void main(String[] args) {  
        new SubCLS();  
        System.out.println();  
        new SubCLS(1);  
        System.out.println();  
        new SubCLS(1,2);  
        System.out.println();  
    }  
}
```

```
public class SubCLS extends SuperCLS {  
    public SubCLS() {  
        System.out.println("Con : SubCLS()");  
    }  
    public SubCLS(int i) {  
        super(i);  
        System.out.println("Con : SubCLS(int i)");  
    }  
    public SubCLS(int i, int j) {  
        super(i, j);  
        System.out.println("Con : SubCLS(int i, int j)");  
    }  
}
```

클래스의 상속 1

- 상속 관계에 있는 두 클래스의 적절한 생성자 정의

```
public class Man {  
    protected String name;  
    public Man(String name) {  
        this.name = name;  
    }  
    public void tellYourName() {  
        System.out.println("My name is " + name);  
    }  
}
```

```
public class MyBusinessMan2 {  
    public static void main(String[] args) {  
        BusinessMan man = new BusinessMan("YOON", "kakao", "staff");  
        man.tellYourInfo();  
    }  
}
```

클래스의 상속 1

```
public class BusinessMan extends Man {  
    private String company;  
    private String position;  
    public BusinessMan(String name, String company, String position) {  
        super(name);    // 상위 클래스 생성자 호출  
        this.company = company;  
        this.position = position;  
    }  
    public void tellYourInfo() {  
        System.out.println("My company is " + this.company);  
        System.out.println("My position is " + this.position);  
        tellYourName();  
    }  
}
```


클래스의 상속 1

- 단일 상속 만을 지원하는 자바
 - 자바는 프로그램이 과도하게 복잡해지는 것을 막기 위해 단일 상속 만을 지원
 - 즉, 하나의 클래스가 상속할 수 있는 클래스의 수가 최대 하나