

상속(inherit)





상속



상속

상속

다른 클래스가 가지고있는 멤버(변수와 메소드)들을 새로 작성할 클래스에서 직접 만들지 않고, 상속받음으로써 새 클래스가 자신의 멤버처럼 사용할 수 있는 기능

상속의 목적

클래스의 재사용, 연관된 일련 클래스들에 대해 공통적인 규약을 정의



상속

상속의 장점

1. 보다 적은 양의 코드로 새로운 클래스를 작성 가능
2. 코드를 공통적으로 관리하기 때문에 코드의 추가 및 변경 용이
3. 코드 중복을 제거하여 프로그램의 생산성과 유지보수에 크게 기여



상속

상속방법

클래스간 상속 시 **extends** 키워드 사용

표현식

※ 이 경우 B클래스가 A클래스를 상속받아 새로 만들어지는 클래스

[접근제어지시자] class 클래스명B extends 클래스명A{
}

상속받는 클래스
후손클래스
자식클래스
파생클래스
서브(sub)클래스

상속하는 클래스
상위클래스
부모클래스
선조클래스
슈퍼클래스



상속

단일상속(Single inheritance)

- 클래스 작성 시 한 개의 클래스만 상속하는 것
- 명확하고 신뢰성 있는 코드 작성가능
- **자바에서는 단일상속만 가능(다중상속 지원X)**

다중상속(Multiple inheritance)

- 클래스 작성 시 여러 클래스로부터 상속을 받는 것
- 복합적인 기능을 가진 클래스를 쉽게 작성 가능
- 서로 다른 클래스로 상속받은 멤버 간의 이름이 같은 경우 문제 발생



상속

상속의 특징

1. 부모클래스의 생성자는 상속 안됨
 - 자식클래스 객체 생성 시, 부모클래스 생성자가 먼저 실행(자동)
 - 자식클래스 생성자 안에서 부모클래스 생성자 호출을 명시하기 위해서는 `super()` 활용
2. 부모의 `private` 멤버는 상속은 되지만 직접 접근 불가
 - 접근제어지시자 `private`는 해당 클래스에서만 접근이 가능하므로, 자식클래스에서도 변수에 접근이 불가능
 - `private`변수의 초기화는 `super()`를 이용하여 부모생성자를 통해 진행
 - `getter`, `setter` 메소드를 이용하여서 접근 가능
3. 모든 클래스는 `Object`클래스의 후손
 - 클래스 작성 시 아무 클래스도 상속하지 않으면 자동으로 `Object`클래스를 상속함
 - 다른 클래스를 상속했더라도 이미 부모클래스가 `Object`클래스를 상속했기 때문에 모든 클래스는 `Object`클래스의 후손



상속

super()

- 부모객체의 생성자를 호출하는 메소드
- 자식객체의 생성자에는 자동으로 부모객체 생성자를 호출하는 코드가 포함되어 있으며, 이를 명시적으로 표현할 때 사용
- 자식객체의 생성자보다 먼저 실행되어야 하기 때문에 자식객체 생성자의 가장 첫 줄에만 사용이 가능
- 매개변수가 있는 부모생성자를 호출하는 경우 super(매개변수)를 이용



상속

오버라이딩(overriding)

- 자식 클래스가 상속받은 부모 메소드를 재작성하는 것
- 부모가 제공하는 기능을 후손이 일부 고쳐서 사용하겠다는 의미
- 자식객체를 통한 실행 시 후손의 것이 우선권을 가짐(동적바인딩)

Annotation

Annotation : 자바 컴파일러에게 알리는 주석문

@Override

```
public void testMethod(){  
}
```

→ 컴파일러에 상속받은 부모 클래스의 메소드를 내가 수정해서 쓰겠다는 의미



상속

오버라이딩(overriding) 성립조건

부모클래스의 메소드와 자식클래스의 메소드 비교

1. 메소드의 이름이 동일해야 한다.
2. 매개변수의 개수와 타입이 동일해야 한다.
3. 리턴 타입이 동일해야 한다.
4. `private` 메소드의 오버라이딩이 불가능하다.
5. 접근제어 지시자는 수정이 가능하지만 부모의 것보다 같거나 넓은 범위로 가능하다.(`default` → `public`으로 가능, `public` → `private` 불가능)

