

# H/W 2 -상속

2016707079 하상천

1. 첫 번째 프린트 문장에서 weapon 레퍼런스 변수는 Weapon(슈퍼클래스) 객체를 가리키고 있으므로, fire 메소드가 오버라이딩 되었음에도 불구하고 슈퍼클래스의 fire()가 실행된다. 반면 두 번째 프린트 문장에서 weapon 레퍼런스 변수는 Cannon(서브클래스) 객체를 가리키고 있으므로, 오버라이딩된 fire 메소드가 실행된다. 레퍼런스 변수 타입은 상관이 없다.

2. 같은 이름의 함수를 여러 개 정의하고, 매개변수의 유형과 개수를 다르게 하여 다양한 유형으로 호출하는 것을 메소드 오버로딩이라고 한다. 예를 들어 계산기 같은 경우, ADDER를 만들 때 이름을 동일시하고, 매개변수 타입이나 개수가 다르면 되기 때문에 리턴타입이 int형인 경우이던 float형인 경우이던 상관이 없기 때문에 리턴타입은 무시된다.

서브클래스에서 부모클래스의 메소드를 재정의 하는 것을 오버라이딩이라고 한다. 오버라이딩은 부모클래스에 정해져있는 메소드와 동일한 메소드 이름과, 매개변수 개수, 타입이 같아야 한다. 즉, 메소드를 동일한 이름으로 재작성 하는 것이므로 동일한 리턴타입을 사용해야한다.

3. 인터페이스는 그 인터페이스를 구현한 클래스를 어떻게 조작할 것인가를 결정한다. 그렇기 때문에 외부에서 제어 할 수 있는 가장 개방적인 접근 제어자인 public만을 사용한다. public을 생략하면 접근 제어자는 default가 되는 것이 아니라 그대로 public이다. 왜냐하면 인터페이스의 멤버는 무조건 public이기 때문이다. 또한 default 메소드는 인터페이스에 메소드를 구현해놓을 수 있는 것이다. 또한 default 메소드도 오버라이딩 할 수 있다. void는 오버라이딩 했을 때 메소드 리턴타입이 없는 것이다.

4. default가 static으로 바뀌었다면 static 메소드 또한 인터페이스에 메소드를 구현해놓을 수있다. 하지만 오버라이딩으로 재정의 하는 것은 불가능하다. static은 객체 생성전부터 생성되기 때문이다. 인터페이스 이름.함수이름 으로 접근한다. default가 private로 바뀌었다면 인터페이스를 구현 하고 있는 서브클래스에서 오버라이딩으로 재정의 할 수 없다. 인터페이스 내에 있는 다른 메소드에 의해서만 호출이 가능하다.