**Day08 (\*￣3￣)**

접근 제어자: private, protected, public

기타 제어자 : 변수, 메소드, 클래스에 사용 가능하다.

기타 제어자 : final

변수: 초기 값 변경 불가능

함수: 함수의 재정의(오버라이드) 불가능

클래스: **상속** 불가능 즉, 서브클래스 생성 불가

has-a관계

집합과 공유관계

집합관계 (Aggregation) : 부분객체(프린터)는 여러 전체 객체(컴퓨터)들에 의해 공유될 수 있다. 각각의 라이프사이클이 다름

포함관계 (Composite) : 키보드는 컴퓨터에 의해 포함되어 같은 라이프사이클을 갖고있다. 즉, 컴퓨터가 꺼지면 키보드도 꺼진다.

is-a관계 (상속관계)

형 변환

1. 기본형의 형 변환
2. 참조형의 형 변환

서로다른 클래스가 호환이 가능한가?

Student student = new Member(); // ??? 가능한가?

원칙적으로 (예외있음) 참조형끼리의 형 변환 안된다. 구성요소가 다르기 때문이다. 하지만 상속 관계의 클래스(참조형) 간에 형 변환을 지원한다. 슈퍼클래스의 구성요소를 모두 가지고 있기 때문에 **업캐스팅**에 한해서 형변환이 가능하다. 별도의 명령어 없이 암묵적(자동)으로 지원이 된다. Duck duck = ~~(Duck)~~ new MallardDuck();

업캐스팅이 발생하면, 서브클래스에서 확장한 변수, 메소드 등을 사용할 수 없다. 하지만 오버라이드한 메소드는 서브클래스의 메소드로 사용할 수 있다. 왜냐하면 오버라이드가 발생하면 슈퍼 클래스의 원 메소드는 not visible 상태로 바껴서 서브 클래스의 메소드가 호출된다.

하지만 예외적으로, 다운 캐스팅을 지원한다. 전제조건은 이전에 업캐스팅을 한 경우이다. 단 명시적 형변환을 해야한다. MallardDuck mduck = (MallardDuck) duck;

* 1. 업캐스팅

서브 클래스의 인스턴스를 슈퍼 클래스의 참조변수(duck)로 가리키도록 형을 변환한 것이다.

Duck duck = new MallardDuck();

* 1. 다운캐스팅 – 자바에서 지원하지 않음

MallardDuck mduck = new Duck();

슈퍼 클래스의 인스턴스를 서브 클래스의 참조 변수(mduck)로 가리키도록 형을 변환한 것이다.

인터페이스

1. 코드의 재사용 목적이 아닌 제어를 위한 것이다. 다중 상속용으로 정의한 클래스이다.
2. 변수 정의 불가, 상수만 선언할 수 있음
3. 일반 메소드 선언 불가, 추상메소드만 선언 가능