Chap08

인터페이스

작성자: 강태경



# 인터페이스

객체의 사용방법을 정의, 인터페이스는 추상클래스와 유사하며 모든 속성은 상수이며,

모든 함수는 추상함수임. 인터페이스는 부모로 사용되고, 구현된 클래스는 자식클래스임, ( 다형성을 적용할 수 있음)

* 인터페이스 : 구현 객체의 사용 방법을 정의한 타입이며, 다른 개발 코드와 구현 객체와 서로 통신하는 접점 역활을 함

다른 개발 코드가 인터페이스의 함수를 호출하면 인터페이스는 객체의 재정의된(구현된) 함수를 호출해줌

* 상수 속성 : *인터페이스는* 기본적으로 속성이 공유상수로(상수)로 이루어져 있음 - public static final 생략됨

관례적으로 모두 대문자로 작성되며, 선언 시 초기값 대입

* 추상 함수 : 인터페이스에 선언된 함수는 public abstract가 생략되고 함수 선언부만 있는 추상 함수임.

구현 클래스는 반드시 추상 함수를 재정의해야 함

* implements : 구현 클래스는 어떤 인터페이스로 사용 가능한 지를 기술하기 위해 클래스 선언 시 implements 키워드를 사용함
* 인터페이스 *사용* : 구현된 클래스(자식클래스)가 인터페이스로(부모) 선언된 변수에 저장될 수 있음( 다형성 이용)  
  ex) 부모인터페이스 a = new 구현클래스();

# 속성(필드) 다형성

기본 자료형에 자동타입변환/강제 타입변환이 있듯이, 인터페이스에도 자동타입변환/강제타입변환이 있음, 이것을 이용해 다형성을 구현함

* *자동 타입 변환 : 구현 객체가 인터페이스 변수에 대입되는 것(저장되는 것)*
* 다형성 *: 상속은 같은 종류의 하위 클래스를 만드는 기술이고, 인터페이스는 사용 방법이 동일한(함수가 동일한) 클래스를 만드는 기술.* 다형성을 *구현하는 방법(함수재정의 + 타입변환)은 똑같음*
* *강제 타입 변환 : 인터페이스에 저장된 구현 객체를 다시 원래 구현 클래스 타입으로 변환하는 것*
* instanceof *: 강제 타입 변환을 하기 전에 변환이 가능한지 조사할 때 사용*
* 인터페이스에*(부모클래스) 구현 객체가(자식클래스) 있는지 조사*