**객체지향 프로그래밍** | 최윤정 교수님 | Summary #07 이서현 2276219  
[Chapter 05] inheritance-interface 클래스 상속/다형성/인터페이스

주요 내용:  상속 Review  생성자호출순서 : super(), this()  Up/Down Casting : 슬라이드 20  다형성 : 메소드 매개변수의 타입변환  오버라이딩 / 오버로딩 : 슬라이드 40  추상 클래스와 추상 메소드 : 슬라이드 45

 인터페이스 review : 슬라이드 54  추상클래스와 인터페이스의 비교  Supp. 중첩구조와 익명객체 75

**상속**

 객체지향의 상속: 부모클래스에 만들어진 필드(변수), 메소드를 자식클래스가 물려받음.

 상속을 통해 간결한 자식 클래스 작성. 동일한 특성을 재정의할 필요가 없어 중복 제거, 자식클래스가 간결해짐. 클래스 관리 용이(상위참조 가능),

부모 클래스 = 슈퍼 클래스 또는 상위 클래스

자식 클래스 = 서브 클래스 또는 하위 클래스

**클래스 상속과 객체: extends 키워드**

자식 클래스가 상속할 부모 클래스를 지정하는 키워드

자바는 단일 상속 – 부모 클래스 나열 불가 (다중 상속 지원X)

상속 횟수 무제한, but 2~4단계 상속이 적절하며 interface를 사용한다.

상속의 최상위 조상 클래스는 java.lang.Object 클래스 (모든 클래스는 자동으로 이 클래스를 상속받는다, 자바 컴파일러에 의해 자동으로 이루어진다.)

public class 사자 extends 동물

**상속과 접근 지정자**

자바의 접근 지정자 4가지: public, protected, 디폴트, private

슈퍼 클래스의 디폴트 멤버

슈퍼 클래스의 protected 멤버

같은 패키지 내의 모든 클래스 접근 허용

서브 클래스 다른 패키지에 있는 슈퍼 클래스의 protected 멤버 접근 가능

**슈퍼 클래스의 멤버에 대한 서브 클래스의 접근**

슈퍼 클래스와 서브 클래스가 동일한 패키지에: 슈퍼 클래스의 private 멤버 접근 안됨

다른 패키지에: 슈퍼 클래스의 private, 디폴트 멤버 접근 안 됨

생성자 호출순서

서브 클래스/슈퍼 클래스의 생성자 호출 및 실행

* 슈퍼클래스에 의해 생성자와 서브 클래스 생성자 모두 실행된다.
* 호출 순서는 서브 클래스의 생성자가 먼저 호출, but, 서브 클래스의 생성자는 실행 전 슈퍼 클래스 생성자 호출됨!

서브 클래스에서 슈퍼 클래스의 생성자를 선택하는 방법

* super() 이용
* 사용자가 명시하지 않으면 기본생성자가 자동 생성, 생성자에서 상위생성자가 자동 호출됨. 맨 윗줄에 super()가 생김

\* 슈퍼 클래스에 기본 생성자가 없으면 오류가 날 수 있으니 만들어 두자.

**super():** 서브 클래스에서 명시적으로 슈퍼 클래스 생성자 선택 호출

**super(parameter);** // 인자를 이용하여 슈퍼 클래스의 적당한 생성자 호출반드시 서브 클래스 생성자 코드의 제일 첫 라인에 와야 한다.

++

1. length: 배열의 길이를 알고자 할 때 사용된다. arrays(int[], double[], String[])

2. length(): String 클래스의 메소드. 문자열의 길이를 알고자 할 때 사용된다.

**캐스팅 : UpCasting / DownCasting**

업캐스팅?: 서브 클래스 객체를 슈퍼 클래스 타입으로 타입 변환. 업캐스팅된 레퍼런스는 객체 내에 슈퍼 클래스의 멤버만 접근 가능하다.

다운캐스팅?: 업캐스팅 시킨 슈퍼 클래스 객체를, 서브 클래스 타입으로 변환. 개발자의 명시적 타입 변환 필요.

자동타입변환: 바로 위의 부모가 아니더라도 상속 계층의 상위에 있다면 자동 타입 변환 가능.

**객체 연산자 : instanceof**

레퍼런스가 가리키는 객체의 타입을 식별한다. 상위 호환도 가능.

**다형성(Polymorphism)**

메소드에 전달된 매개변수의 타입을 이용한 다형성. 캐스팅과 instanceof