**Web개발\_Day09\_노트정리**

**오버로드, Overload**

ㆍ오버로드란 원칙적으로 하나의 클래스 안에 동일한 이름의 메소드가 두개 존재할 수 없지만, 이를 가능하게 하는 예외적인 처리기법

* 하나의 이름으로 여러 개의 메소드를 구현한다.

ㆍ오버로드 장점 : 하나의 메소드를 사용하여 여러 기능을 구현할 수 있다.

* 예) println()는 printlnBoolean(), printlnInt(), printlnChar(), … 등등의 기능을 전부 구현한다.

ㆍ오버로드의 조건 : 메소드간의 파라미터가 서로 달라야한다.

1. 파라미터의 데이터 타입이 다르다.
   1. Int fn(int a){ … }
   2. Int fn(long a){ … }
2. 파라미터의 개수가 다르다
   1. Int fn(int a){ … }
   2. Int fn(int a, long b){ … }
3. 서로 다른 데이터형을 갖는 파라미터들의 전달 순서가 다르다.
   1. Int fn(int a, long b) { … }
   2. Int fn(long b, int a) { … }

* 리턴형이 다른 경우는 위의 오버로드의 조건 성립에 아무런 영향을 주지 않는다.

ㆍ오버로드가 된 메소드들은 매개변수의 개수, 타입 차이로 자동적으로 그에 알맞은 메소드를 실행시킨다.

* 가변인자 사용시 예외의 경우가 발생한다.

**가변인자**

ㆍ하나의 자료형타입의 데이터를 매개변수로써 여러 개 받는 개념이다.

* 기존의 메소드는 매개변수가 고정적이었지만, JDK1.5 이후부터 동적으로 지정해 줄 수 있다.

ㆍ선언 : “타입…변수명”

ㆍ지정한 타입의 데이터가 매개변수로 몇 개가 들어올 지 모를 경우 사용한다.

* 한번의 선언으로 다수의 데이터가 들어올 수 있다.

ㆍ가변인자를 선언한 메소드를 오버로딩은 실행되지 않는 경우도 있다.

* 가변인자 사용시 오버로딩은 추천되지 않는다.
* 고정 매개변수와 가변인자를 사용하여 오버로드를 만들 경우, 넘어가는 매개변수가 가변인자중 하나인지 고정 매개변수인지를 파악하는 컴파일중에 에러가 발생되기 때문이다.