**Web개발\_Day14\_노트정리**

**제네릭스, generics**

ㆍArrayList의 복습

* ArrayList엔 어떤 타입이든 집어넣을 수 있다. 하지만…
* 데이터를 꺼낼 땐, 집어넣은 데이터의 타입을 알아야 한다.
* 자료형을 주지 않은 경우 에러가 난다.

ㆍArrayList<String> arrlist = new ArrayList<String>();

* <String> 이 부분이 제네릭스이다.
* 위와 같은 ArrayList에는 String 타입의 데이터만 올 수 있도록 제한되어 있다.

ㆍ제네릭스를 통해서 요소 타입을 고정했기 때문에, 사용시 형변환이 필요없다.

* 불필요한 코딩 등, 잘못된 형변환등의 오류를 피할 수 있다.

ㆍ제네릭스 장점

1. 타입 안정성 확보
2. 타입 체크와 형변환 생략가능(=코드 간결화)

**제네릭스 클래스 선언**

ㆍclass Box{

Object item;

Void setItem(Object item){return this.item = item;}

Object getItem(){return item;}

}

ㆍ위의 기본적인 클래스가 제네릭스 클래스가 된다면…

ㆍ class Box<T>{

T item;

Void setItem(T tem){return this.item = item;}

Object getItem(){return item;}

}

* T는 타입변수, Type의 첫글자에서 따온것으로써, 다른 이름을 사용해도 된다 (대체로 T를 사용)

ㆍ객체생성시, Box<String> b = new Box<String>();

* String말고 다른 실제 타입을 지정할 수 있다.
* 타입을 지정하지 않는다면 제네릭스가 아닌 일반 Box클래스를 선언하려 한다.
* Box클래스 내의 item은 String타입만 받을 수 있게 된다.

ㆍ타입 문자로 사용할 타입을 명시하면, 한 종류의 타입만 저장할 수 있도록 제한 할 수 있지만, 그래도 여전히 모든 종류의 타입을 지정할 수 있다는 것은 변함이 없다.

ㆍ제네릭스에 들어갈 타입을 제한하려면?

* 제네릭스 타입에서 extends를 사용하면 특정 타입의 객체들만 대입할 수 있게 제한할 수 있다.
* Class FruitBox < T extends Fruit > { … }
* Fruit 클래스의 자식클래스 타입들만 사용 가능하게 할 수 있다.