Generics

Generics

Generics

Generics

JDK 1.5부터 제공되는 기능으로, 클래스 내부에서 사용할 데이터타입을 객체생성 시 지정하는 기법

Generics Type

클래스의 객체 생성 시 “<>”안에 내부에서 사용할 클래스명을 기입
→ 기본자료형을 사용 할 수 없음
클래스명<타입> 레퍼런스 = new 생성자<타입>();
ArrayList<Student> list = new ArrayList<Student>();

※ Generics를 이용하여 여러 개의 데이터타입을 지정하는경우 , 를사용하면됨.
→ GenericClass<T,E>

Generics

Generics 기본사용

```
public class GenericEx1<T>{
    private T data;
    public GenericEx1(){}
    public GenericEx1(T data){
        this.data = data;
    }
    public T getData(){ return data; }
    public void setData(T data){ this.data = data; }
}
```

Generics

멀티 Generics 사용

```
public class GenericEx2<T,E>{
    private T data1;
    private E data2;
    public GenericEx2(){}
    public GenericEx2(T data1, E data2){
        this.data1 = data1;
        this.data2 = data2;
    }
    public T getData1(){ return data1; }
    public void setData1(T data1){ this.data1 = data1; }
    public E getData2(){ return data2; }
    public void setData2(E data2){ this.data2 = data2; }
}
```

Generics

메소드에서 Generics 사용-1(매개변수)

```
public class GenericEx3{
    public GenericEx3(){}
    public void printMsg(ArrayList<Student> list){
        실행코드..
    }
}
```

제네릭이 설정된 레퍼런스를 매개변수로 넘겨주는 경우 매개변수에도 선언해 주어야 함

Generics

메소드에서 Generics 사용-2(매개변수)

```
public class GenericEx4{
    public GenericEx4(){}
    public ArrayList<Student> printMsg(){
        실행코드..
    }
}
```

제네릭이 설정된 레퍼런스를 리턴하는 경우에 도 반환값에 제네릭이 적용되어야 함

Generics

Generics 제한

- 제네릭은 해당 클래스의 객체가 만들어질 때 내부에서 데이터 타입 지정
- 이경우 의도치 않은 데이터타입이 들어 올 수 있음
- 제네릭 사용시의 데이터 타입을 제한할 수 있음

Generics Type 설정방법

1. <T> : 제한 없음 모든 타입 가능
2. <T extends A클래스> : A클래스와 그 자손클래스만 가능(상한제한)
3. <T super A클래스> : A 클래스와 그 조상들만 가능(하한제한)