

제 22 강 : 열거형

※ 학습목표

- ✓ 열거형 특징을 설명할 수 있다.
- ✓ 열거형을 작성할 수 있다.

1. 열거형

① 열거형의 정의

- ✓ C언어에서 정수를 자동적으로 증가하여 상수들로 만들어 쓰이는 열거체(enum)와 비슷하다
- ✓ 자바에서 얘기하는 열거형은 상수를 하나의 객체로 인식하고 여러 개의 상수 객체들을 한 곳에 모아둔 하나의 묶음(객체)이라 할 수 있다.

② 열거형의 구성

```
[접근제한] enum [열거형_명]{  
    상수1, 상수2, ..., 상수n  
}
```

[실습]

```
1 package tommy.java.exam01;  
2  
3 public class EnumEx1 {  
4     public enum Lesson {  
5         JAVA, XML, EJB  
6     }  
7  
8     public static void main(String[] args) {  
9         Lesson le = Lesson.JAVA;  
10        System.out.println("Lesson : " + le);  
11        System.out.println("XML : " + Lesson.XML);  
12    }  
13 }
```

2. 열거형의 실체

- ✓ 열거형은 내부에서 순차적으로 정의되는 값들을 JAVA, XML, EJB라는 이름으로 그냥 일반적인 상수라는 개념만으로 저장되는 것이 아니다.
- ✓ 정확히 얘기를 하자면 java.lang이라는 패키지에 Enum이라는 추상 클래스를 상속 받는 이너 클래스가 정의되는 것이다.
- ✓ 앞의 예제에서 Lesson이라는 enumerated types(열거형)를 가지고 예를 들어보면 다음과 같다.

```
public static final class EnumEx1$Lesson extends Enum
```

[실습]

```
1 package tommy.java.exam02;
2
3 public class EnumEx2 {
4     public enum Item {
5         Add, Del, Search, Cancel
6     }
7
8     public static void main(String[] args) {
9         Item a1 = Item.Search;
10        if (a1 instanceof Object) { // 열거형이 객체인지 아닌지 비교
11            System.out.println(a1.toString() + "^^");
12            System.out.println("OK! instanceof Object");
13            System.out.println("저장된 실제 정수값 : " + a1.ordinal());
14        }
15        Item[] items = Item.values();
16        System.out.println("items.length : " + items.length);
17        for (Item n : Item.values())
18            System.out.println(n + ":" + n.ordinal());
19    }
20 }
```