Enum(

소스 : https://github.com/hyomee/JAVABASIC

Content

1. 기본
2. Enum 원리
3. 코드 단순화
4. 서비스 호출

1. 기본

관련이 있는 상수들의 집합으로 완전한 Class로 Enum 사용시 1. 코드가 단순해지고, 가독성이 좋아짐 2. 상수값의 타입 안정성이 보장(컴파일 시점에 체크, 인스턴스 생성과 상속 방지) 3. 구도 의도가 열거형 임을 명확이 함

01. 사용방법

1. Class 처럼 사용 : enum
2. Class 내부에 enum 사용 : 내부 class처럼
3. enum의 생성자는 private
4. 관련 Method
   - valueOf(String) : String값을 enum에서 가져옴
   - valueOf(Class String) : 넘겨받은 class에서 String찾아, enum에 가져옴
   - values() : enum의 요소들을 순서대로 enum타입의 배열로 리턴

02. 선언 및 사용

1. Class Type

```
public enum ClassTypeEnum {
    NAC, SUS, RSP, CAN
}
```

2. Inner Class Type

```
public class InnerClassTypeEnum {
    public enum EventCode {
        NAC, SUS, RSP, CAN
    }
}
```

```
public static void main(String[] args) {
    // 열거형에 있는 요소 전체 참조
    ClassTypeEnum[] enumValues = ClassTypeEnum.values();
    Arrays.stream(enumValues).forEach(classTypeEnum -> System.out.println(classTypeEnum));

    // 요소 검사
    String compValue = "NAC";
    if (Arrays.stream(enumValues).anyMatch(classTypeEnum -> compValue.equals(String.valueOf(classTypeEnum)))) {
      System.out.print(compValue + " 는 존재 합니다.");
    } else {
      System.out.print(compValue + " 는 존재 하지 않습니다..");
    };.
  }
```

결과 :
NAC
SUS
RSP
CAN
NAC 는 존재 합니다.

```
public static void main(String[] args) {
    InnerClassTypeEnum.EventCode[] eventCode = InnerClassTypeEnum.EventCode.values();
    Arrays.stream(eventCode).forEach(code -> System.out.println(code));
}
```

결과 :
NAC
SUS
RSP
CAN

2. Enum 원리

01. 소스

```
public static void main(String[] args) {
    InnerClassTypeEnum.EventNm eventNm = InnerClassTypeEnum.EventNm.valueOf("SUS");
    System.out.print(eventNm.getEventNm());
}
```

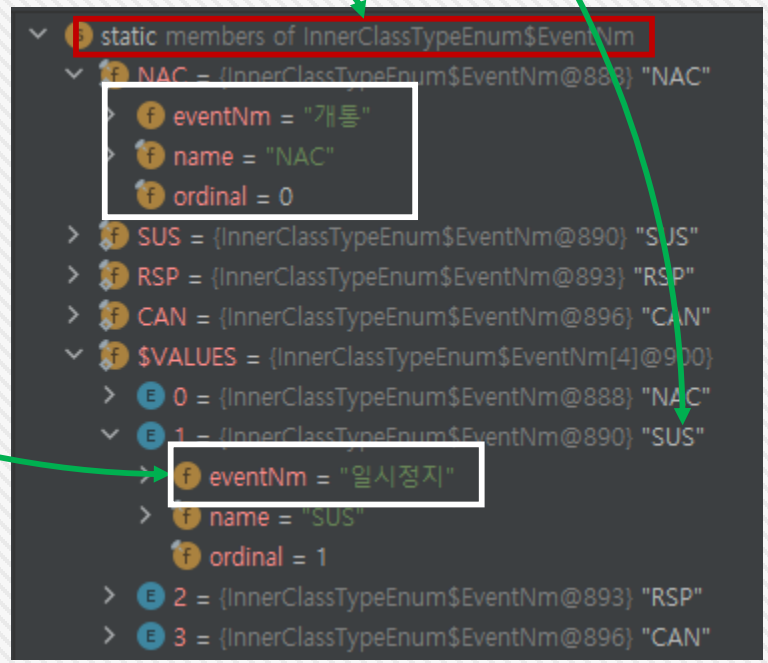
일시정지

```
@Getter
public enum EventNm {
  NAC("개통"),
  SUS("일시정지"),
  RSP("일시정지해제"),
  CAN("해지");

  private String eventNm;

  EventNm(String eventNm) {
    this.eventNm = eventNm;
  }
}
```

1. 생성시점 : static 으로 생성 함

2. 값 취득

3. 실제 Value 얻음



3. 코드 단순, 가독성

여러 조건이 만족 하는 IF문과 같은 곳에서 코드가 단순해 지며, 가독성이 좋아 지며, 코드의 추가 시 업무 로직의 변경 없이 Enum 객체만 수정 하면 되므로 유지보수 입장에서 보면 효과적임

01. 기존 IF 문

```
public static final String SVC_PRC_TYPE_CYB = "CYB"; // Mylgt
public static final String SVC_PRC_TYPE_MOB = "MOB"; // 모바일고객센터
public static final String SVC_PRC_TYPE_FCG = "FCG"; // FCG
public static final String SVC_PRC_TYPE_ARS = "ARS"; // ARS
public static final String SVC_PRC_TYPE_CAS = "CAS"; // CAS
public static final String SVC_PRC_TYPE_VPM = "VPM"; // 음성비화 등록화면
public static final String SVC_PRC_TYPE_DEV = "DEV"; // 기기변경
public static final String SVC_PRC_TYPE_DOC = "DOC"; // 개통서류
```

```
If (SVC_PRC_TYPE_CYB.equals(prcType) ||
    SVC_PRC_TYPE_ CAS.equals(prcType) ||
    SVC_PRC_TYPE_ ARS.equals(prcType) ||
    SVC_PRC_TYPE_ PPS.equals(prcType) ||
    SVC_PRC_TYPE_ FCG.equals(prcType) ||
    SVC_PRC_TYPE_ MNT.equals(prcType)
    // 업무 로직
}
```

02. Enum 사용

```
public static enum ONLINE_RUNDTTM_PRCTYPE {
  CYB, CAS, ARS, PPS, FCG, MNT;
}
```

```
If (Arrays.stream(SvcCodeVal.ONLINE_RUNDTTM_PRCTYPE.values())
   .anyMatch(items -> String.valueOf(items).equals(prcType))) {
    // 업무 로직
}
```

4. 서비스 호출

Enum 에서 서비스 호출

01. 서비스 업무 로직

```
public interface EnumService {

  String service1(String code);
  String service2(String code);
}

@Service
class EnumServiceImpl implements EnumService {

  @Override
  public String service1(String code) {
    return "\nservice1:: service1";
  }

  @Override
  public String service2(String code) {
    return "\nservice2:: service2";
  }
}
```

02. Enum 사용

```
EnumService service = new EnumServiceImpl();
InnerClassTypeEnum.EventServcie eventServcie = InnerClassTypeEnum.EventServcie.valueOf("NAC");
String vv = eventServcie.handleTask(service, "NAC");
System.out.print(vv);
```

```
public enum EventServcie {

  NAC {
    String handleTask(EnumService service, String code) {
      return service.service1(code);
    }
  },
  SUS {
    String handleTask(EnumService service, String code) {
      return service.service1(code);
    }
  };

  abstract String handleTask(EnumService service, String code);

}
```

THANKS

ABACUS

www.iabacus.co.kr

Tel. 82-2-2109-5400

Fax. 82-2-6442-5409