

```
public enum LimitType {
    MAX , MIN
}
```

* enum, 사용자 지정 인터페이스 만들기
활용한 예제 만들고 활용하기

```
public enum WeekDay {
    //원하는 값으로 설정하려면 생성자를 이용해 줘야 한다 ( 멤버변수도 준비되어야 한다)
    MON("월") , THE("화") , WED("수") , THU("목") , FRI("금");
    private String name;

    WeekDay( String name){
        this.name = name;
    }

    public String getName() {
        return name;
    }
}
```

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

public @interface Blind {} //마커 애너테이션 , 값은 없고 어떤상태임을 나타낼 때 사용함

@Target(ElementType.FIELD)

@Retention(RetentionPolicy.RUNTIME)

```
public @interface MaxLength {
    int value() default 10;
}
```

@Target(ElementType.METHOD)

@Retention(RetentionPolicy.RUNTIME)

```
public @interface TestAndPrint {
    String before() default "";
    String after() default "";
}
```

@Retention(RetentionPolicy.RUNTIME)

```
public @interface WeekDaysAvail {
    WeekDay[] value();
}
```

```

public class Introduction {
    @Blind
    @MaxLength(4)
    private String name;

    private int age;

    @MaxLength
    private String job;

    @MaxLength(200)
    private String description;

    @WeekDaysAvail( { WeekDay.MON , WeekDay.WED , WeekDay.FRI})    //월수 있는 요일 목록
    private WeekDay day;

    public Introduction(String name, int age, String job, String description , WeekDay day) {
        this.name = name;
        this.age = age;
        this.job = job;
        this.day = day;
        this.description = description;
    }

    @TestAndPrint(before = "--<< 소개입니다.>>--")
    // @TestAndPrint(after = "-- 이상 간략 소개입니다. --")
    public void shortPrint () {
        System.out.printf(
            "%s (%d대 %s)%n",
            name, age, job
        );
    }

    public void fullPrint () {
        StringBuilder sb = new StringBuilder();
        sb.append("이름 : " + name + "\n");
        sb.append("나이 : " + age + "\n");
        sb.append("직업 : " + job + "\n");
        sb.append(description);
        System.out.println(sb.toString());
    }
}

```

```

public class Main {
    public static void main(String[] args) {
        Introduction[] objsToVerify = {
            new Introduction(
                "한효주", 36, "배우",
                "무빙 , 독전, 뷰티인사이드" , WeekDay.THU ),
            new Introduction(
                "울트라", 4, "책",
                "울트라러닝 ***** 세계 0.1%가 지식을 얻는 방법 2020년 출간" , WeekDay.MON),
            new Introduction(
                "피카소", 0, "화가",
                "마드리드왕립미술학교 , 이비농의 처녀들, 게르니카 ,파블로 루이스 피카소(스페인어: Pablo Ruiz Picasso, 1881년 10월
25일 ~ 1973년 4월 8일)는 스페인 말라가에서 태어나 주로 프랑스에서 미술활동을 한 20세기의 대표적 큐비즘 작가다. 대표작으로 《아비농의 처녀들》,
《게르니카》 등이 있다. " , WeekDay.WED),
            new Introduction(
                "조지", 90, "하버드 교수",
                " 조지베일런트 75년간 평생 연구결과 =>>1. 좋은배우자, 2. 어려움에 대처하는 자세 (정신건강) , 3. 금연 , 4. 절주, 5.적
당한체중 6. 규칙적인 운동습관 , 7.공부 (지적호기심, 앎을 확장해 나아가는 것 ) " , WeekDay.MON)
        };

        List<Introduction> objsVerified = new ArrayList<>();
        for (Introduction obj : objsToVerify) {
            try {
                objsVerified.add( 객체검증하기(obj) );
                System.out.println(" 통과");
            } catch (Exception e) {
                System.out.println("□ 반례 : " + e.getMessage());
                e.printStackTrace();
            } finally { //
                System.out.println("- - - -");
            }
        }

        System.out.println("통과개수=" + objsVerified.size());
    }

    public static Introduction 객체검증하기(Introduction obj) throws Exception {
        Class objClass = obj.getClass();

        for (Field f : objClass.getDeclaredFields()) {
            f.setAccessible(true);
            Object val = f.get(obj);
            for (Annotation a : f.getAnnotations()) {
                // 첫 글자 외 *으로
                if (a instanceof Blind) {
                    var s = (String) val;
                    f.set(obj, s.substring(0, 1) + "*".repeat(s.length() - 1));
                }
            }
        }
    }
}

```

```

// 최대 길이 검증
if (a instanceof MaxLength) {
    int maxLen = ((MaxLength) a).value();
    if (((String) val).length() > maxLen) {
        throw new Exception(
            "%s 최대 길이(%d) 초과" + f.getName() + maxLen);
    }
}

// 가용 요일 검증
if (a instanceof WeekDaysAvail) {
    WeekDaysAvail wda = (WeekDaysAvail) a;
    WeekDay wd = (WeekDay) f.get(obj);
    WeekDay[] availables = wda.value();
    var available = false;
    for (var inAvail : availables) {
        if (wd == inAvail) available = true;
    }
    if (!available) throw new Exception(
        "해당 요일(%s) 불가 (해당 요일은 실수 없어요 !) : " + wd.getName());
}
}

// 테스트해야 할 메소드 실행
for ( Method m : objClass.getDeclaredMethods()) {
    for (Annotation a : m.getAnnotations()) {
        if (a instanceof TestAndPrint) {
            TestAndPrint tp = (TestAndPrint) a;
            String printBefore = tp.before();
            String printAfter = tp.after();

            if (!printBefore.isBlank())
                System.out.println(printBefore);
            m.invoke(obj);
            if (!printAfter.isBlank())
                System.out.println(printAfter);
        }
    }
}
return obj;
}

}

```