@AssertTrue 해당 값은 항상 ture 여야한다

@AssertFalse 해당 값은 항상 false여야한다

@NotNull 해당 값은 Null 을 허용하지 않는다

@Digits 해당 값은 실수3자리이상의 값을 가질 수 없다

@Pattern(regexp=””) 해당 값은 regexp(정규식)의 부여된 규칙을 따라야한다

@NotEmpty 해당 값은 null,”” 를 허용하지 않는다

@NotBlank 해당 값은 null,””,” “ 를 허용하지 않는다

@Positive 해당 값은 양수만 가질 수 있다

@PositiveOrZero 해당 값은 양수또는 0만 가질 수 있다

@Negative 해당 값은 음수만 가질 수 있다

@Negative 해당 값은 음수또는 0만 가질 수 있다

Interface Validator 를 상속받아서 직접 유효성 검사 조건을 만들 수 있다

public boolean supports(Class<?> clazz) {

return (검사할객체).class.isAssignableFrom(clazz);

}

ValidationUtils 의 rejectIfEmpty() 를 사용

값이 비어있는지 확인하겠다.

RejectIfEmptyOrWhitespace()

값이 비어있거나 공백이 있는지 검사한다

rejectValue: 유효성 조건을 직접 만드어 검사할 때 사용한다

완성된 커스텀 validator는

사용하려는 컨트롤러에 빈(bean)으로 등록해서 사용한다

@InitBinder

public void initBinder(WebDataBinder binder) {

V v = new V();

binder.setValidator(v);

}

추가할 validator가 한 개면 setValidator

두 개 이상이면 addValidators를 사용한다