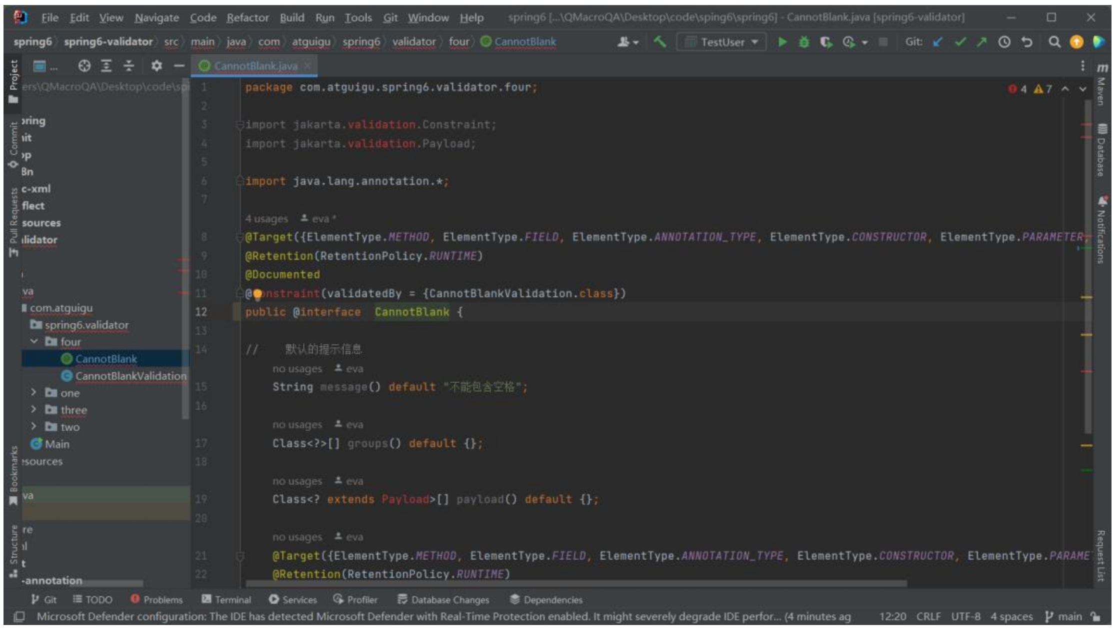
1、

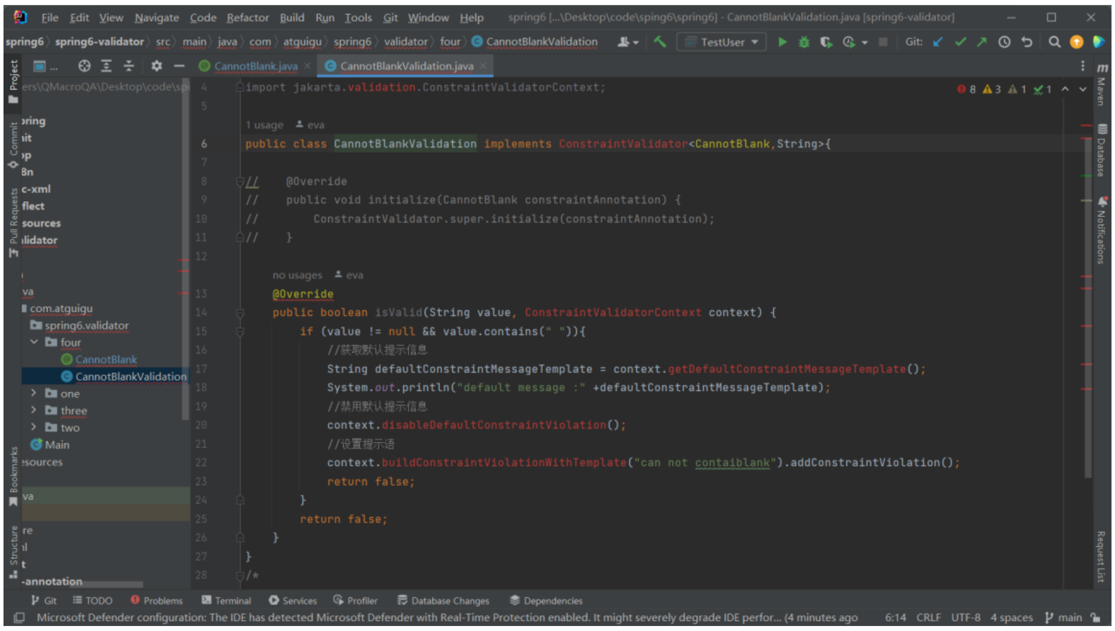


spring6 [...\QMacroQA\Desktop\code\sping6\spring6] - CannotBlank.java [spring6-validator]

```java
package com.atguigu.spring6.validator.four;

import jakarta.validation.Constraint;
import jakarta.validation.Payload;

import java.lang.annotation.*;

@Target({ElementType.METHOD, ElementType.FIELD, ElementType.ANNOTATION_TYPE, ElementType.CONSTRUCTOR, ElementType.PARAMETER,
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Constraint(validatedBy = {CannotBlankValidation.class})
public @interface  CannotBlank {

//    默认的提示信息
    String message() default "不能包含空格";

    Class<?>[] groups() default {};

    Class<? extends Payload>[] payload() default {};

    @Target({ElementType.METHOD, ElementType.FIELD, ElementType.ANNOTATION_TYPE, ElementType.CONSTRUCTOR, ElementType.PARAME
    @Retention(RetentionPolicy.RUNTIME)
```

Microsoft Defender configuration: The IDE has detected Microsoft Defender with Real-Time Protection enabled. It might severely degrade IDE perfor... (4 minutes ag   12:20 CRLF UTF-8 4 spaces main

2、



spring6 [...\Desktop\code\sping6\spring6] - CannotBlankValidation.java [spring6-validator]

```java
import jakarta.validation.ConstraintValidatorContext;

public class CannotBlankValidation implements ConstraintValidator<CannotBlank,String>{

//    @Override
//    public void initialize(CannotBlank constraintAnnotation) {
//        ConstraintValidator.super.initialize(constraintAnnotation);
//    }

    @Override
    public boolean isValid(String value, ConstraintValidatorContext context) {
        if (value != null && value.contains(" ")){
            //获取默认提示信息
            String defaultConstraintMessageTemplate = context.getDefaultConstraintMessageTemplate();
            System.out.println("default message :" +defaultConstraintMessageTemplate);
            //禁用默认提示信息
            context.disableDefaultConstraintViolation();
            //设置提示语
            context.buildConstraintViolationWithTemplate("can not contaiblank").addConstraintViolation();
            return false;
        }
        return false;
    }
}
/*
```

Microsoft Defender configuration: The IDE has detected Microsoft Defender with Real-Time Protection enabled. It might severely degrade IDE perfor... (4 minutes ago   6:14 CRLF UTF-8 4 spaces main

```
package com.atguigu.spring6.validator.four;

import jakarta.validation.Constraint;
import jakarta.validation.Payload;

import java.lang.annotation.*;

@Target({ElementType.METHOD, ElementType.FIELD, ElementType.ANNOTATION_TYPE,
ElementType.CONSTRUCTOR, ElementType.PARAMETER, ElementType.TYPE_USE})
@Retention(RetentionPolicy.RUNTIME)
@Documented
@Constraint(validatedBy = {CannotBlankValidation.class})
public @interface  CannotBlank {

//    默认的提示信息
    String message() default "不能包含空格";

    Class<?>[] groups() default {};

    Class<? extends Payload>[] payload() default {};

    @Target({ElementType.METHOD,                          ElementType.FIELD,
ElementType.ANNOTATION_TYPE,                     ElementType.CONSTRUCTOR,
ElementType.PARAMETER, ElementType.TYPE_USE})
    @Retention(RetentionPolicy.RUNTIME)
    @Documented
    public @interface List {
        CannotBlank[] value();
    }
}
```

```
package com.atguigu.spring6.validator.four;

import jakarta.validation.ConstraintValidator;
import jakarta.validation.ConstraintValidatorContext;

public class CannotBlankValidation implements ConstraintValidator<CannotBlank,String>{

//	@Override
//	public void initialize(CannotBlank constraintAnnotation) {
//		ConstraintValidator.super.initialize(constraintAnnotation);
//	}

	@Override
	public boolean isValid(String value, ConstraintValidatorContext context) {
		if (value != null && value.contains(" ")){
			//获取默认提示信息
			String defaultConstraintMessageTemplate = context.getDefaultConstraintMessageTemplate();
			System.out.println("default message :" +defaultConstraintMessageTemplate);
			//禁用默认提示信息
			context.disableDefaultConstraintViolation();
			//设置提示语
			context.buildConstraintViolationWithTemplate("can not containblank").addConstraintViolation();
			return false;
		}
		return false;
	}
}
/*

在 three 中做验证
自定义校验
 */
```