# 实验六

**要求**：

1. 学习gradle 部署工具；
2. 学习静态代码质量检查方法和工具；
3. 应用定义的静态代码质量检查方法检查和改进工程代码；

**目的**：

1. 了解Gradle部署工具，编写gradle静态检查任务staticAnalysys；
2. 配置代码规范检查工具checkstyle、pmd和spotbugs；
3. 扩展构建工具，应用阿里巴巴java代码规范检查工程文件；

实验前提：

如果不了解grade，则可以参考grade安装实验指南：https://www.cnblogs.com/IcanFixIt/p/6909338.html

checkstyle 插件：

<https://docs.gradle.org/current/dsl/org.gradle.api.plugins.quality.Checkstyle.html>

pmd插件：

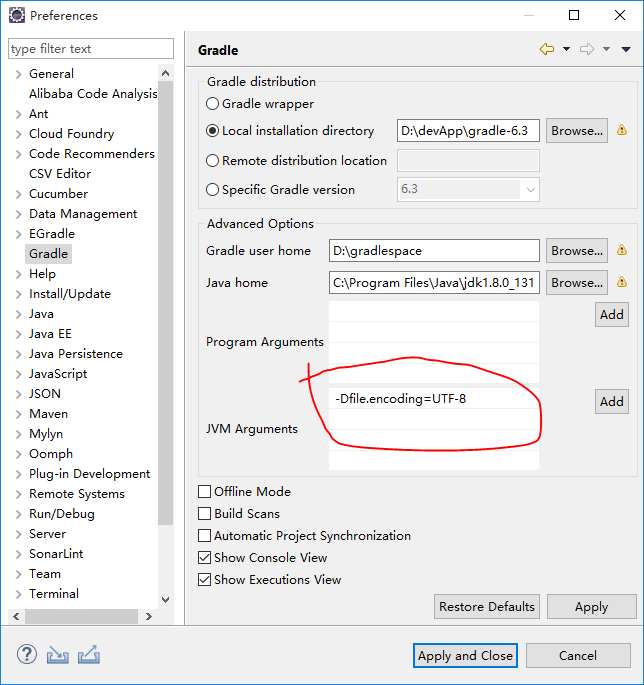
<https://docs.gradle.org/current/userguide/pmd_plugin.html>

spotbugs插件：

<https://plugins.gradle.org/plugin/com.github.spotbugs>

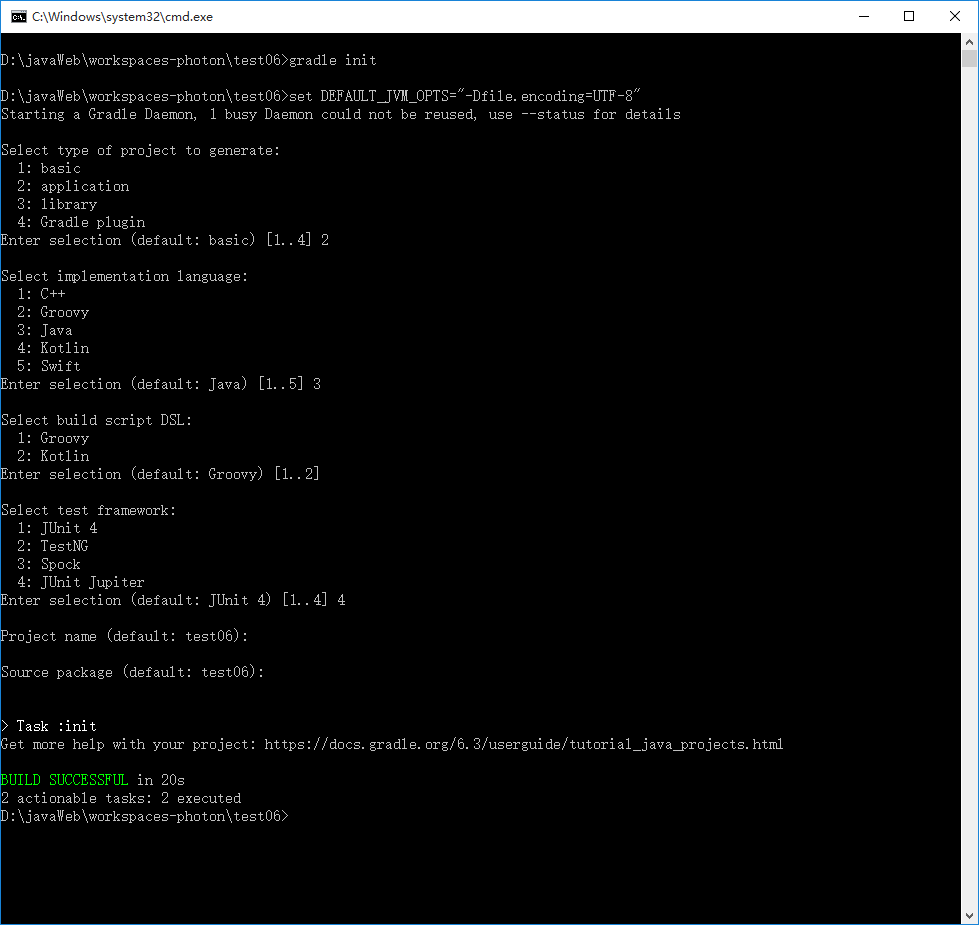
实验步骤：

1. 设置gradle参数



1. 创建文件夹如果test06，在该文件夹进入cmd命名窗口，输入gradle init

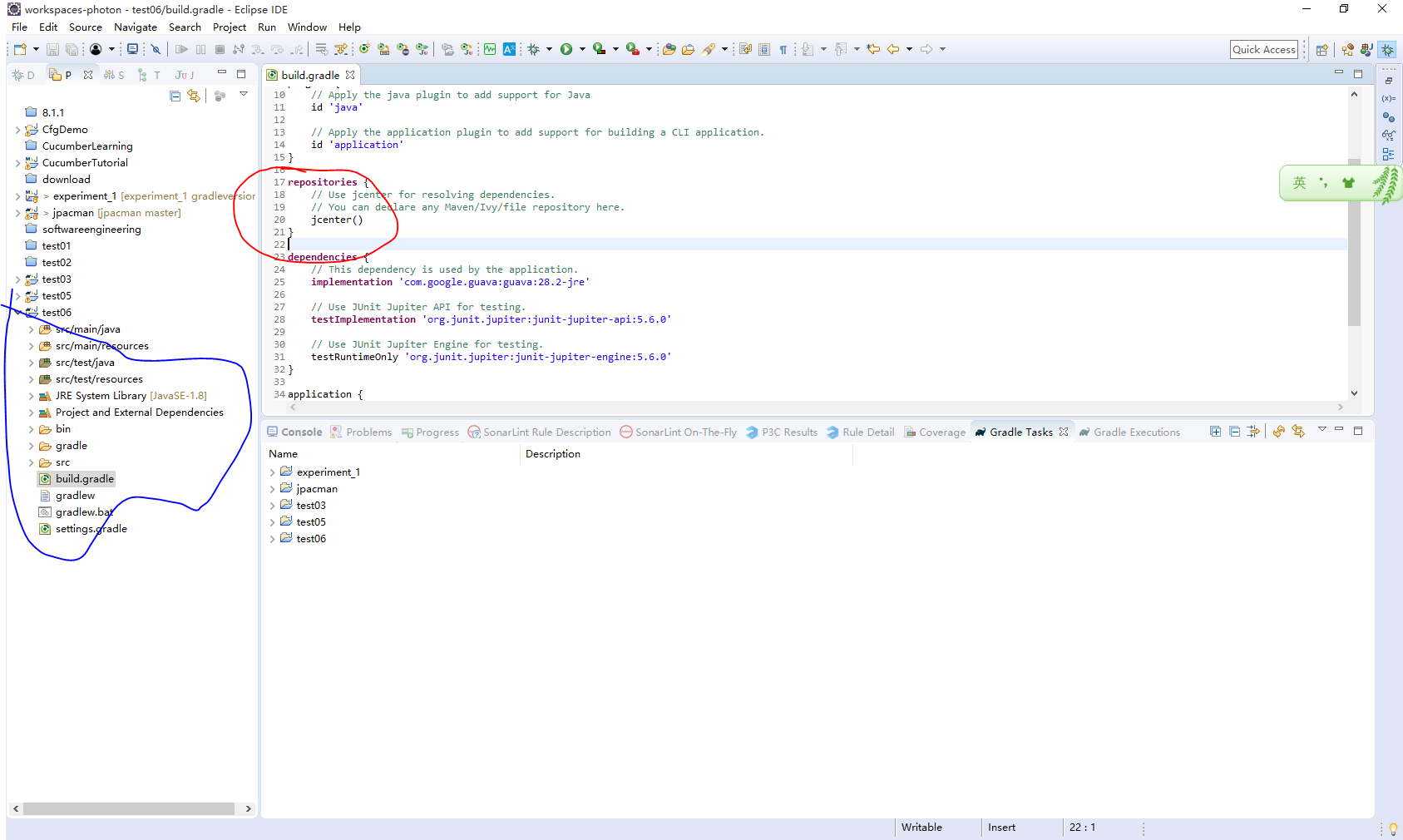
生成java application 工程框架，选择Junit5测试框架



1. 导入Eclipse或Intellj Idea；

Eclipse安装buildship插件（最新版），便于设置gradle执行参数

Eclipse：File>import>existing gradle project



1. 修改build.gradle构建脚本：

**repositories** {

// Use jcenter for resolving dependencies.

// You can declare any Maven/Ivy/file repository here.

maven{ url 'http://maven.aliyun.com/nexus/content/groups/public/'}

}

设置为阿里云的maven提高依赖jar包下载速度；

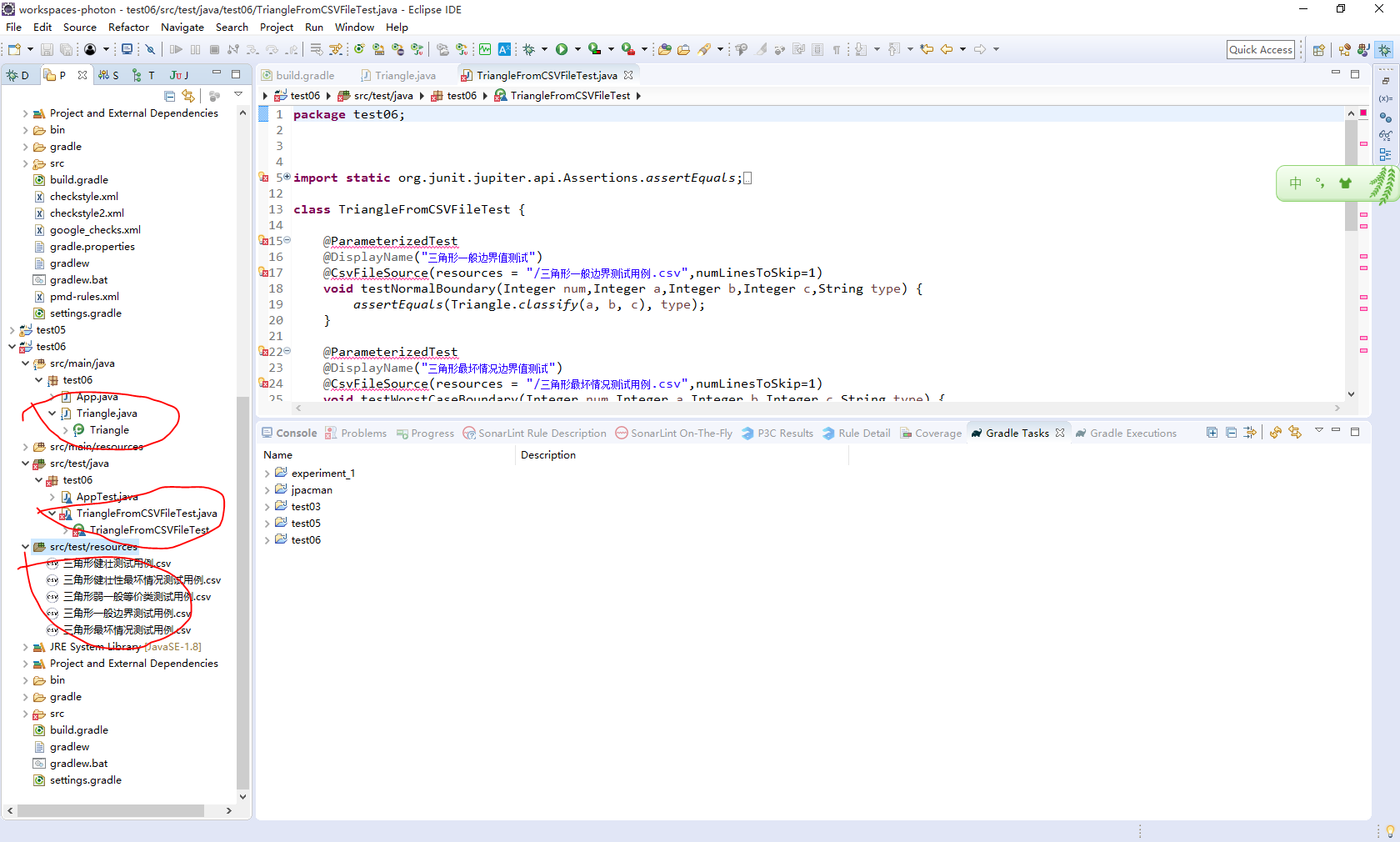
1. 加入三角形问题代码以及测试代码；

按照约定先于配置的原则，

三角形问题代码放置于src/main/java

测试用例代码放置于 src/test/java

测试资源放置于：src/test/resources



可以看到测试代码中对于参数化测试代码出错，找不到params包，这是因为JUnit5把功能进行了分拆，向平台化发展的原因。

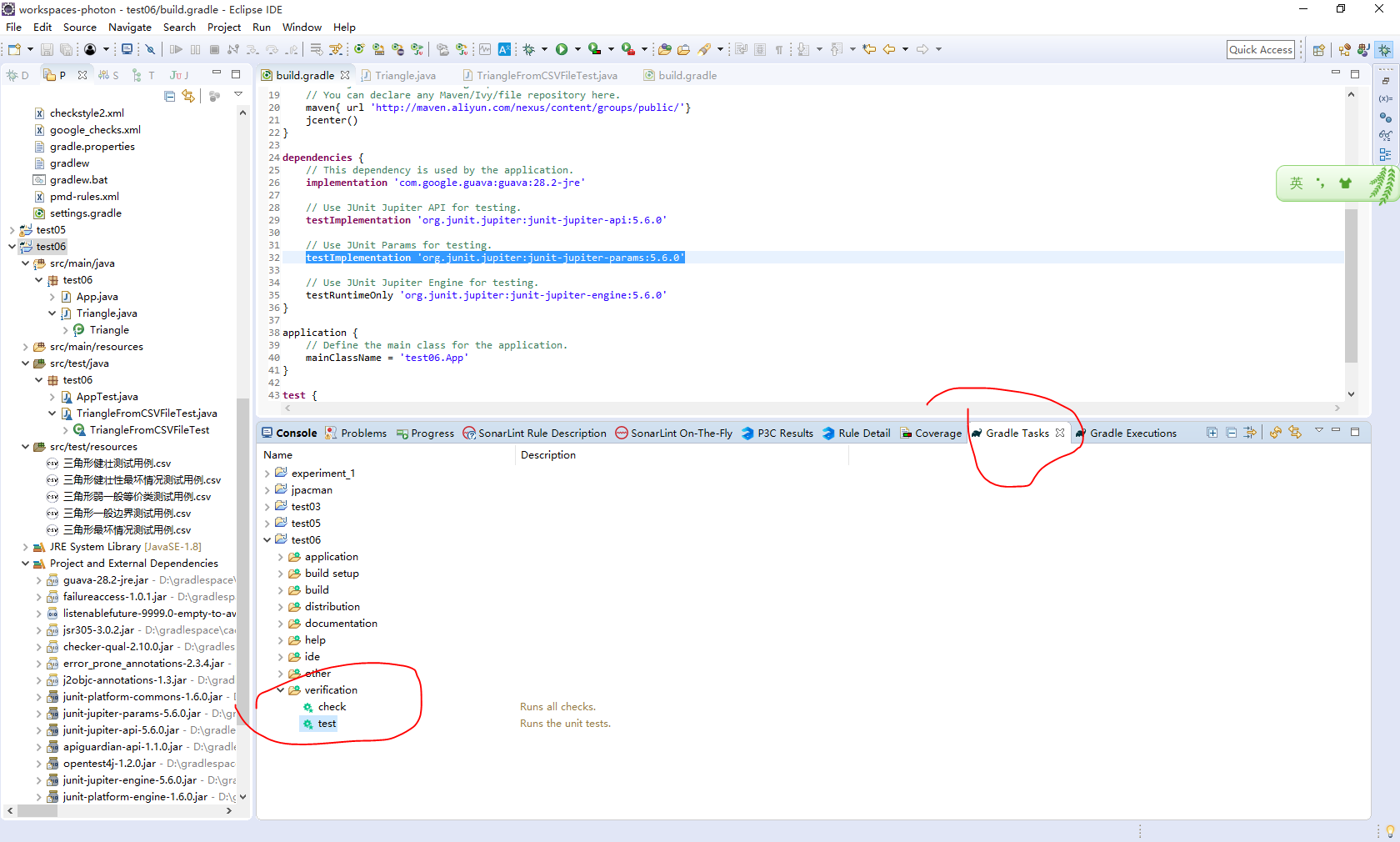
在build.gradle **dependencies块**中加入 params包：

// Use JUnit Params for testing.

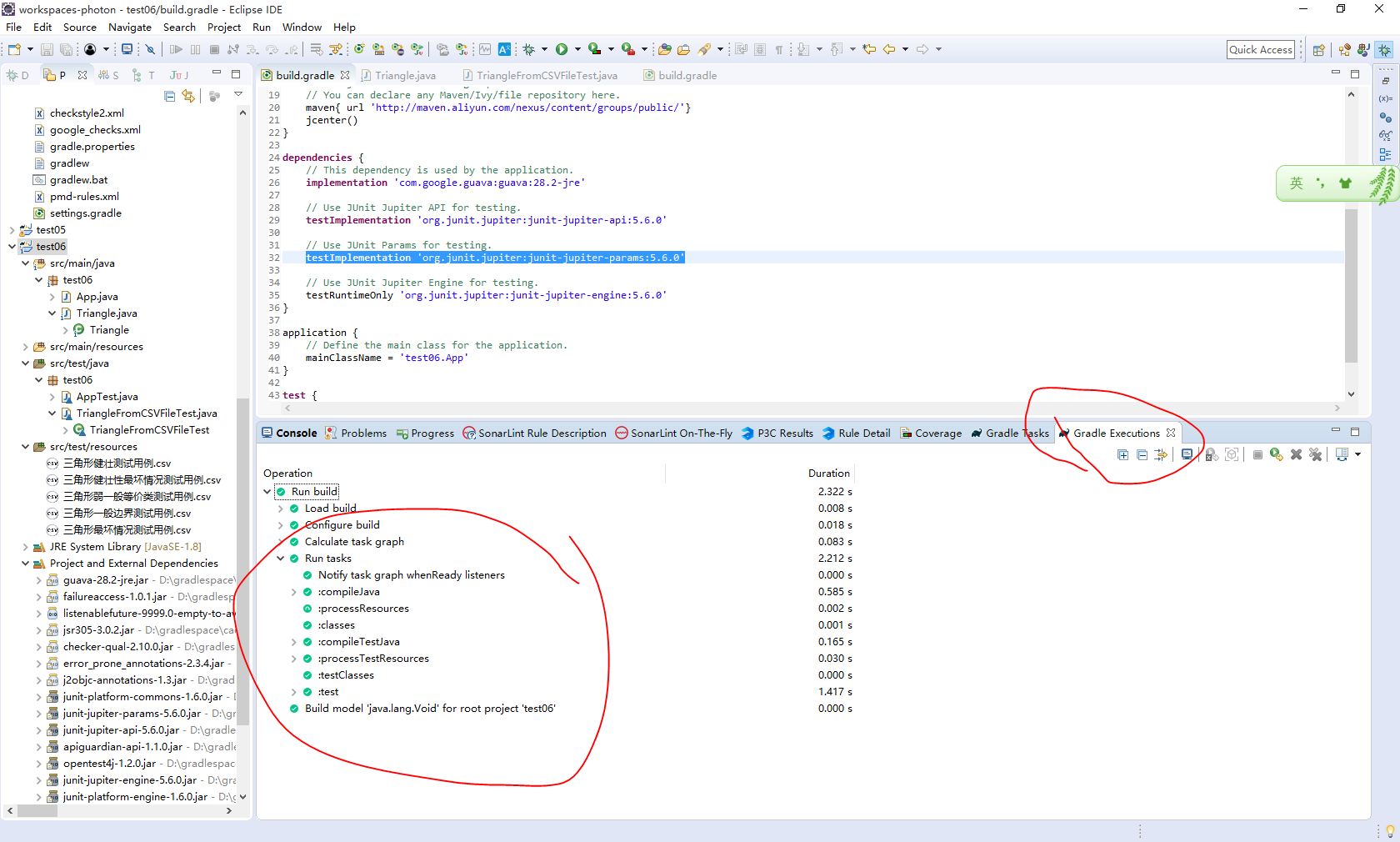
**testImplementation** 'org.junit.jupiter:junit-jupiter-params:5.6.0'

然后选择test06工程，右键菜单中选择gradle>refresh gradle project

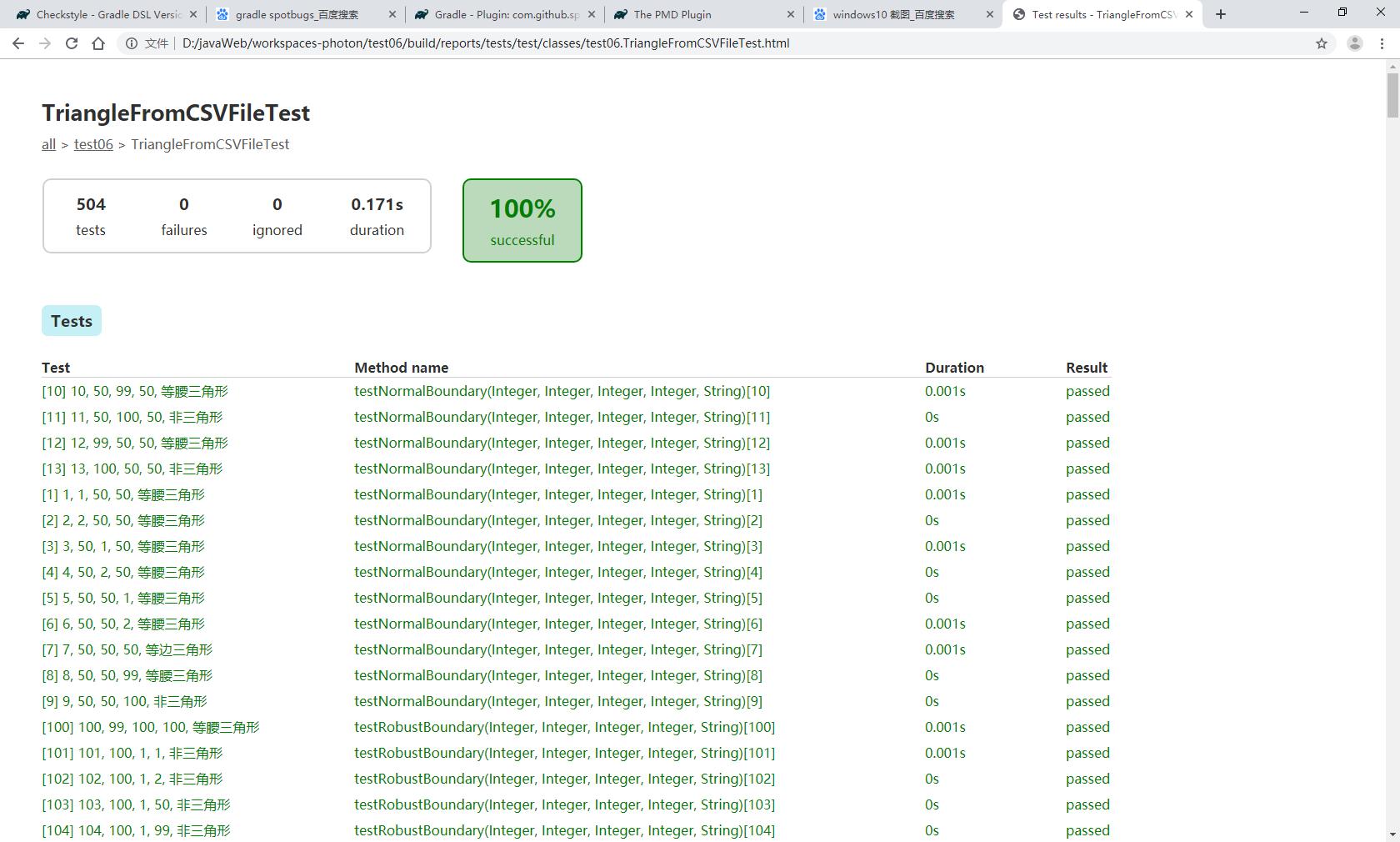
1. 运行gradle check 任务：



选择check任务，右键菜单选择run gradle task



查看test report：



1. 配置checkstyle插件

注：checkstyle、pmd在gradle安装目录下的/lib/plugins/gradle-code-quality-6.3.jar 包中。

修改build.gradle

plugins {

// Apply the java plugin to add support for Java

id 'java'

// Apply the application plugin to add support for building a CLI application.

id 'application'

id 'checkstyle'

}

配置checkstyle规则

checkstyle {

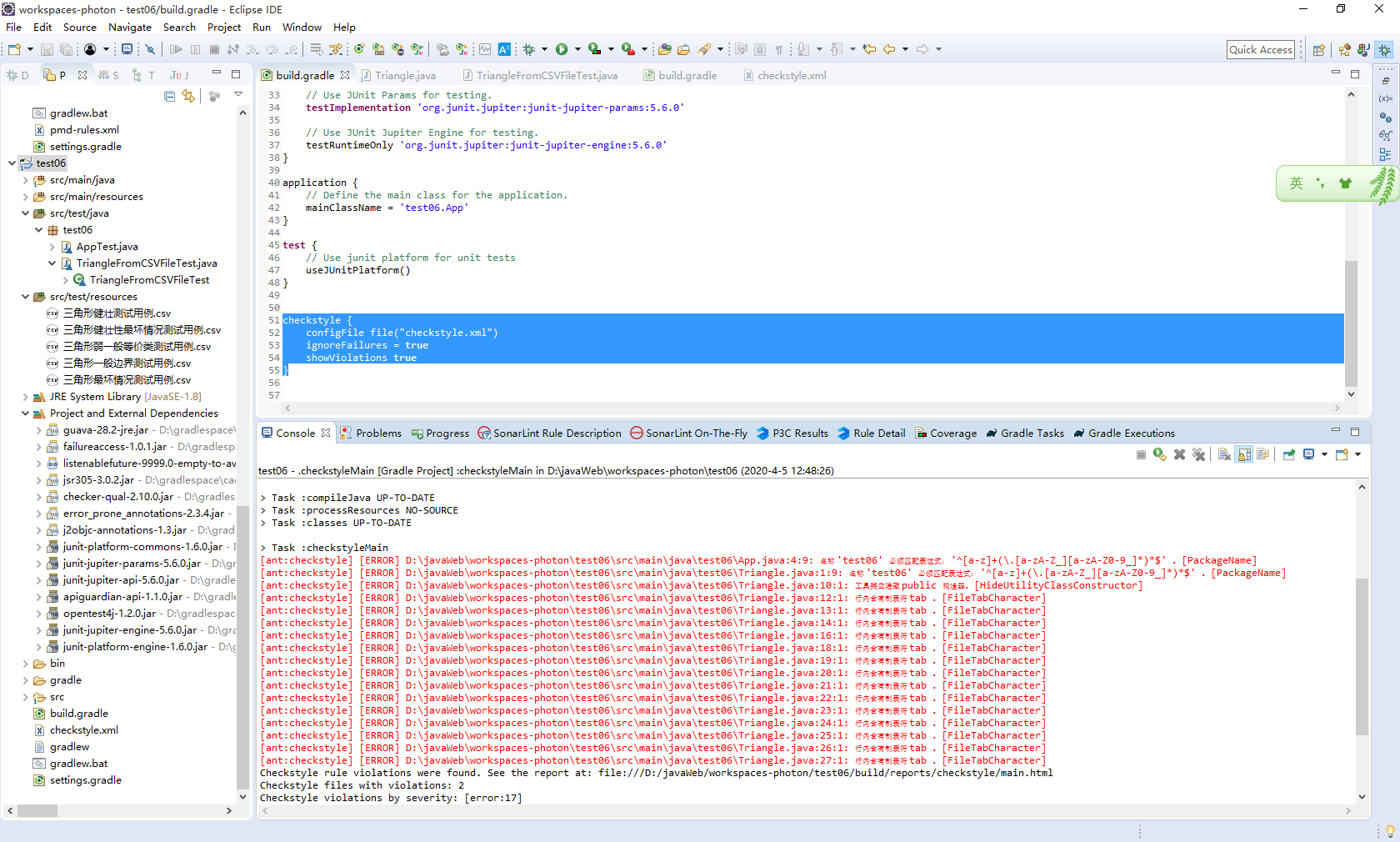
configFile file("checkstyle.xml")

ignoreFailures = **true**

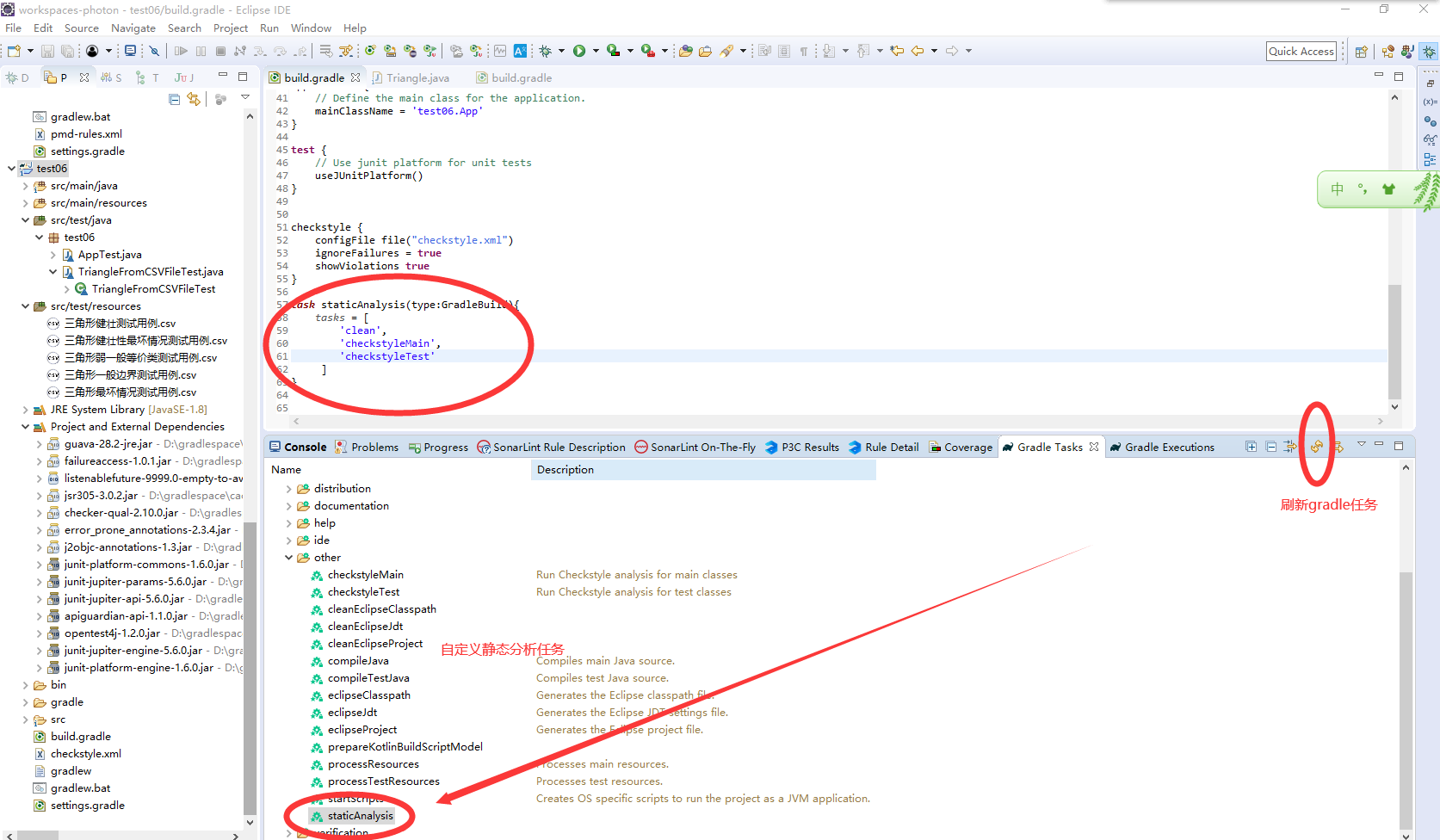
showViolations **true**

}

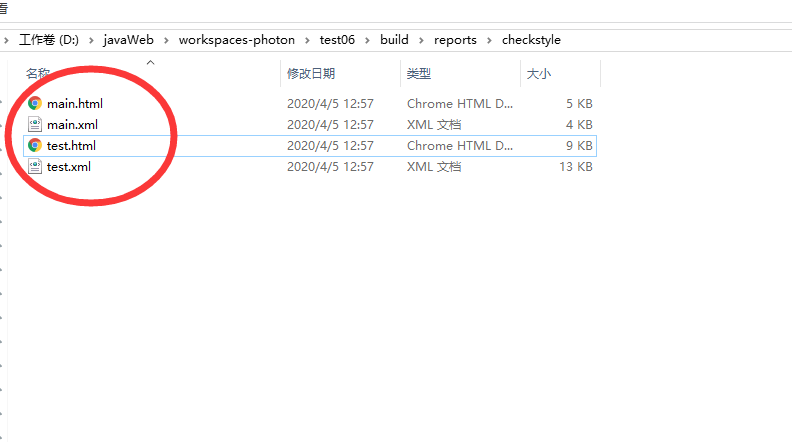
运行checkstyleMain，checkstyleTest任务



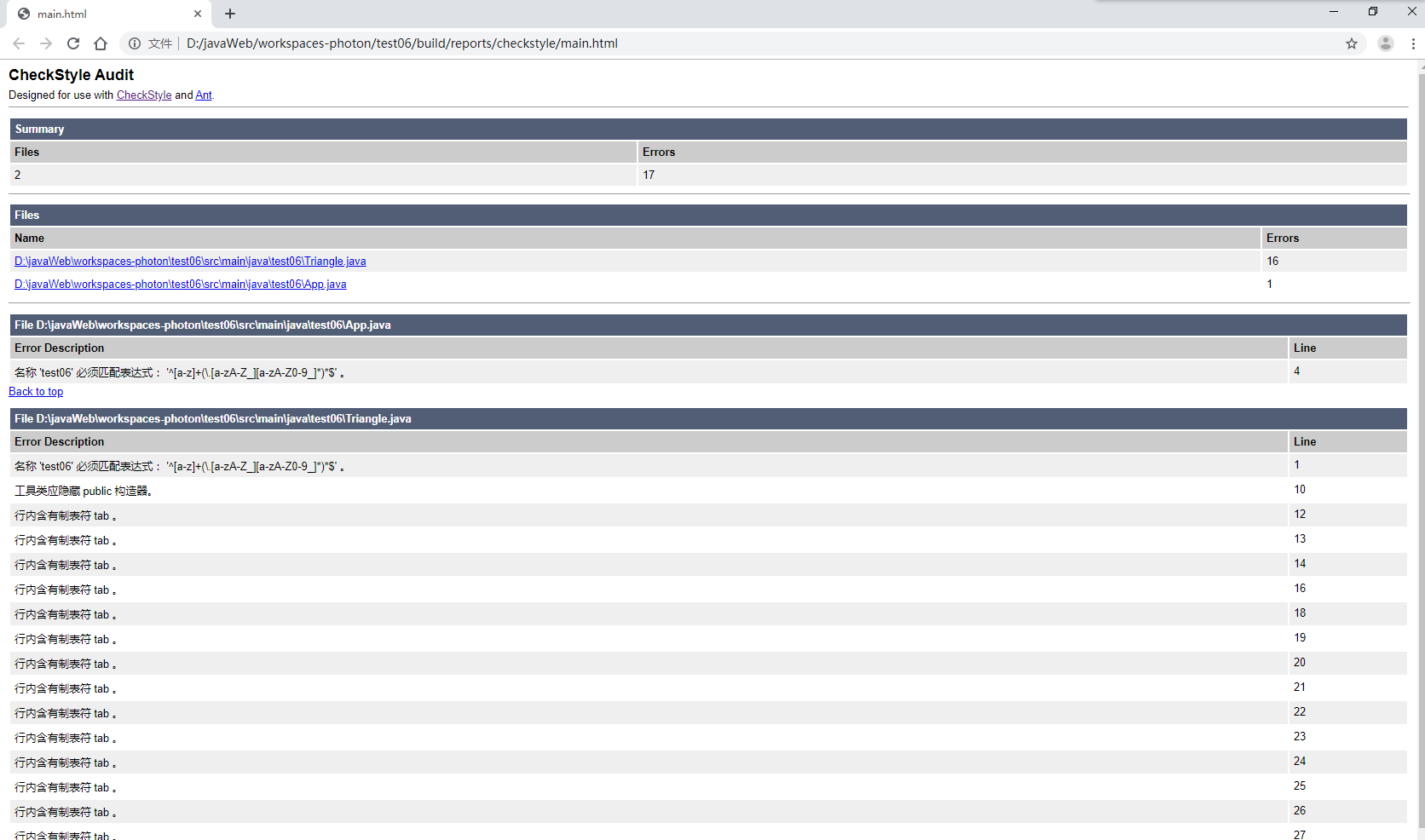
添加一个自定义任务，结合两个任务checksyleMain,checkstyleTest,定义之后，刷新gradle任务，other类别下会出现自定义的任务，右键菜单选择run gradle task



查看checkstyle报告



查看checkstyleMain报告main.html



1. 配置pmd

build.gradle plugins块添加

id ‘pmd’

定义pmd插件属性，使用本地指定的规则文件

pmd {

ruleSetFiles "pmd-rules.xml"

ruleSets = []

ignoreFailures = **true**

consoleOutput = **true**

**}**

**类似checkstyle，运行pmdMain，pmdTest**

**添加到staticAnalysis任务中**

***task*** staticAnalysis(type:GradleBuild){

*tasks* = [

'clean',

'checkstyleMain',

'checkstyleTest',

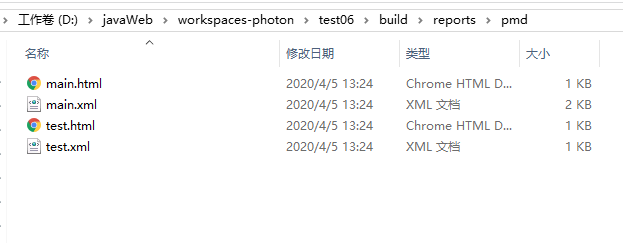
'pmdMain',

'pmdTest'

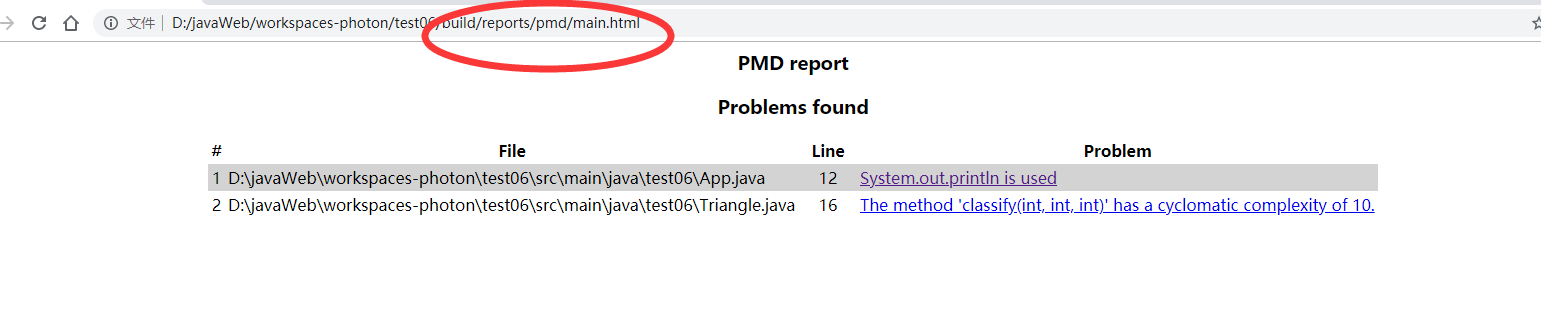
]

}

查看pmd报告



查看pmdMain.html



1. 配置spotbugs

添加插件：

plugins {

// Apply the java plugin to add support for Java

id 'java'

// Apply the application plugin to add support for building a CLI application.

id 'application'

id 'checkstyle'

id 'pmd'

id 'com.github.spotbugs' version '4.0.5'

}

类似运行添加的spotbugsMain，spotbugsTest任务，查看报告

添加到staticAnalysis任务中，作为静态分析代码质量任务的子任务。

***task*** staticAnalysis(type:GradleBuild){

*tasks* = [

'clean',

'checkstyleMain',

'checkstyleTest',

'pmdMain',

'pmdTest',

'spotbugsMain',

'spotbugsTest'

]

}

1. 添加阿里巴巴Java编程规则

阿里巴巴使用的是pmd规则，因此只需要添加阿里巴巴规范插件，并定义规则即可。

定义扩展属性

ext{

p3c = "1.3.0"

}

加入依赖

dependencies{

pmd "com.alibaba.p3c:p3c-pmd:${p3c}"

}

pmd指定阿里巴巴的Java编程规范

pmd-rules-alibaba.xml，因为内置的pmd规则与阿里巴巴的有冲突，这个是注释掉冲突的编程规范并集成阿里巴巴的编程规范的规则文件。

1. 把针对新冠肺炎病毒代码以及测试加入到本工程代码，运行静态代码检查，解决发现的不规范的代码，并提交到码云源代码管理器中。