

信息科学与工程学院

《软件工程》 实验报告九

**系 别 计算机系**

**专 业 计算机科学与技术**

**年 级 2020级**

**姓 名 刘子言**

**指导教师 阮 彤**

**2022-2023 学年 第 1 学期**

**实验九 性能测试**

**一、实验目的**

1. 了解Jmeter性能测试工具的用途和简单的操作；
2. 掌握Jmeter性能测试工具测试过程；
3. 能够使用Jmeter进行负载测试。

**二、实验环境**

个人PC机，架构Demo系统，Jmeter5.5

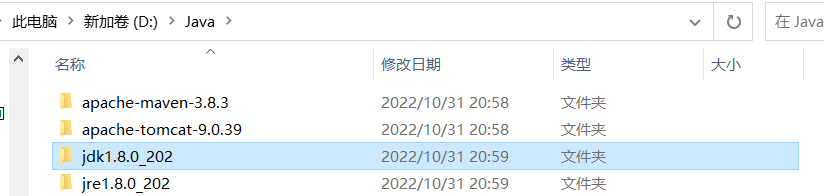
**三、实验内容**

下载并安装Jmeter，在架构Demo系统上进行测试脚本的编写，进行性能测试，并对测试结果进行分析。

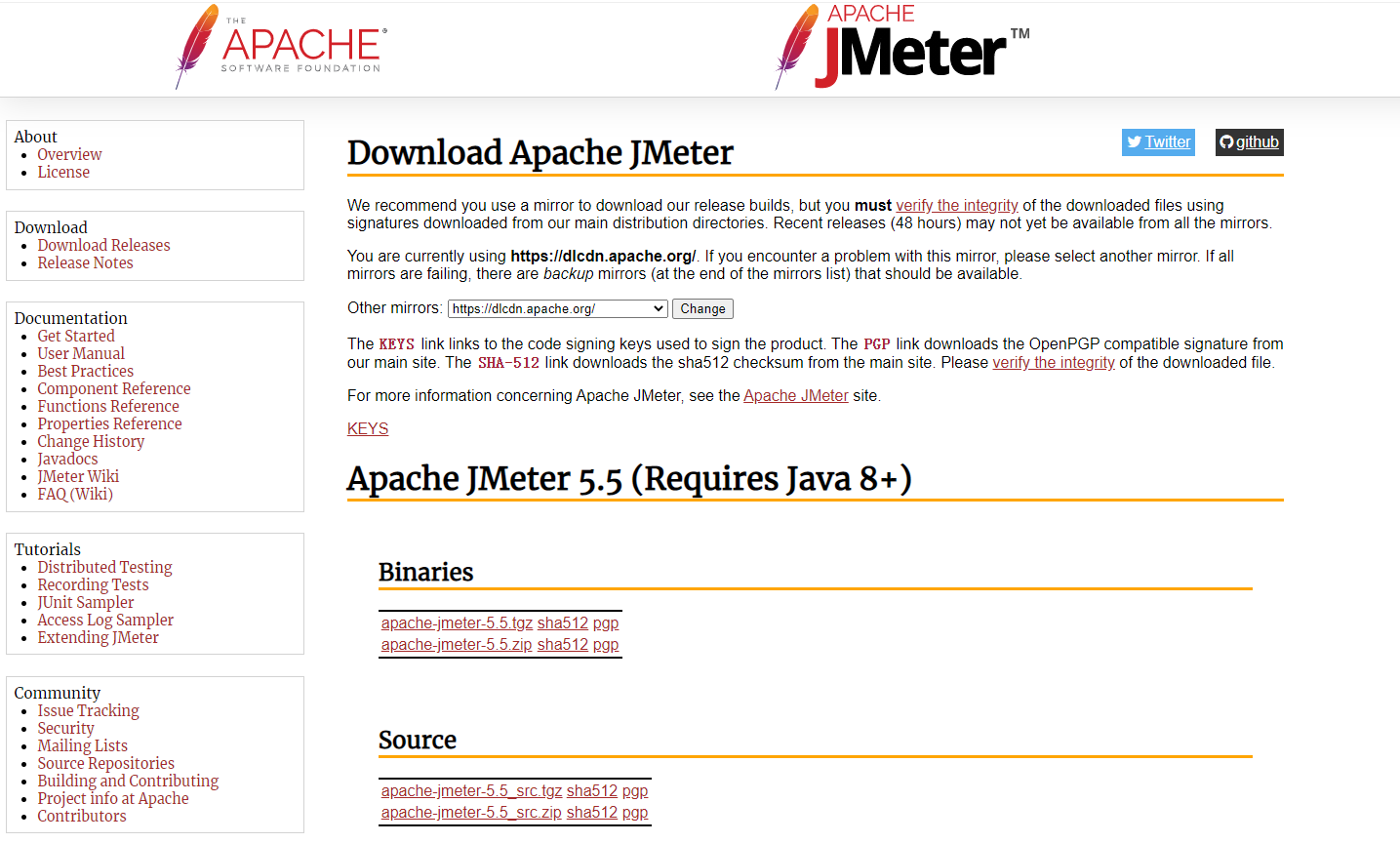
**四、实验步骤**

**1、Jmeter的安装和运行**

（1）安装jdk（Java8，即Jdk1.8）



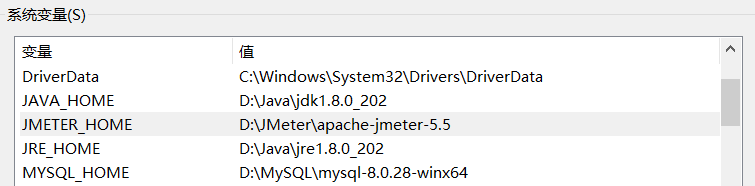
（2）前往官网进行下载<https://jmeter.apache.org/download_jmeter.cgi>，并解压压缩包到英文目录下：D:\JMeter\apache-jmeter-5.5



（3）配置环境变量

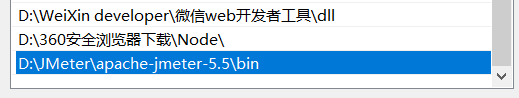
·设置jmeter解压目录的JMETER\_HOME环境变量

系统变量中新建变量，变量名为JMETER\_HOME，变量值为jmeter安装的路径



·设置jmeter中bin目录的path环境变量

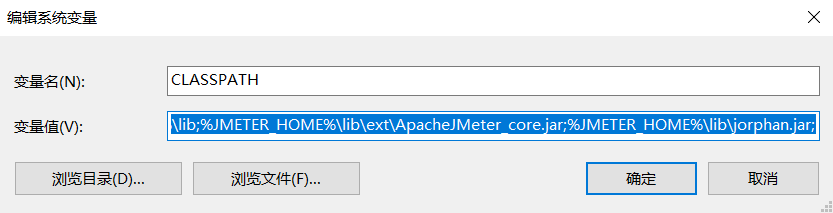
在系统的PATH变量中添加jmeter安装目录下bin目录的路径



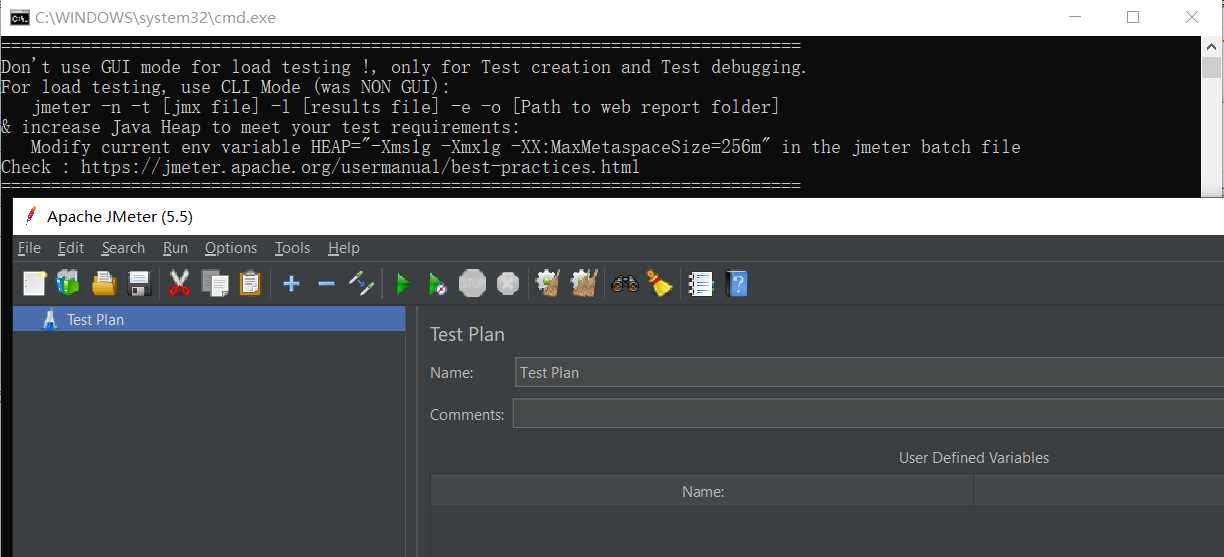
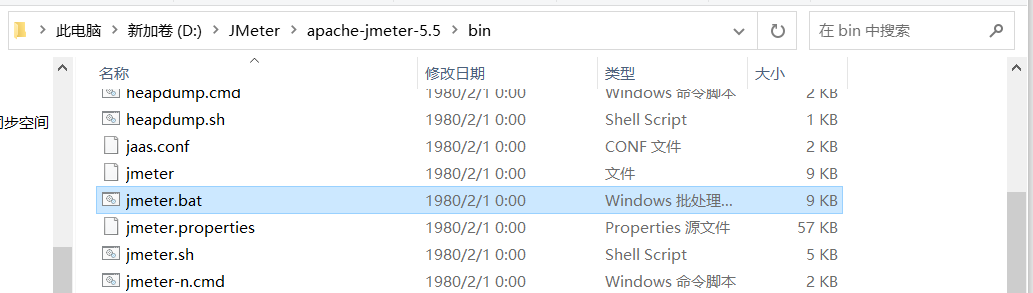
·设置jmeter的classpath变量

系统属性中找到一个变量名为CLASSPATH的系统变量，新增两条：

%JMETER\_HOME%\lib\ext\ApacheJMeter\_core.jar; %JMETER\_HOME%\lib\jorphan.jar;



（4）进入jmeter下的bin目录，点击打开jmeter.bat文件，即可启动jmeter

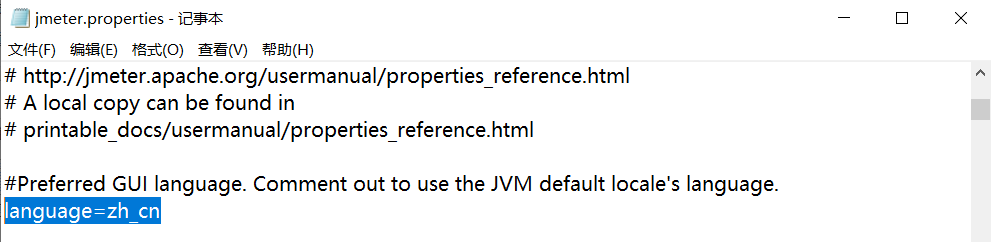


上图可见，jmeter启动成功！

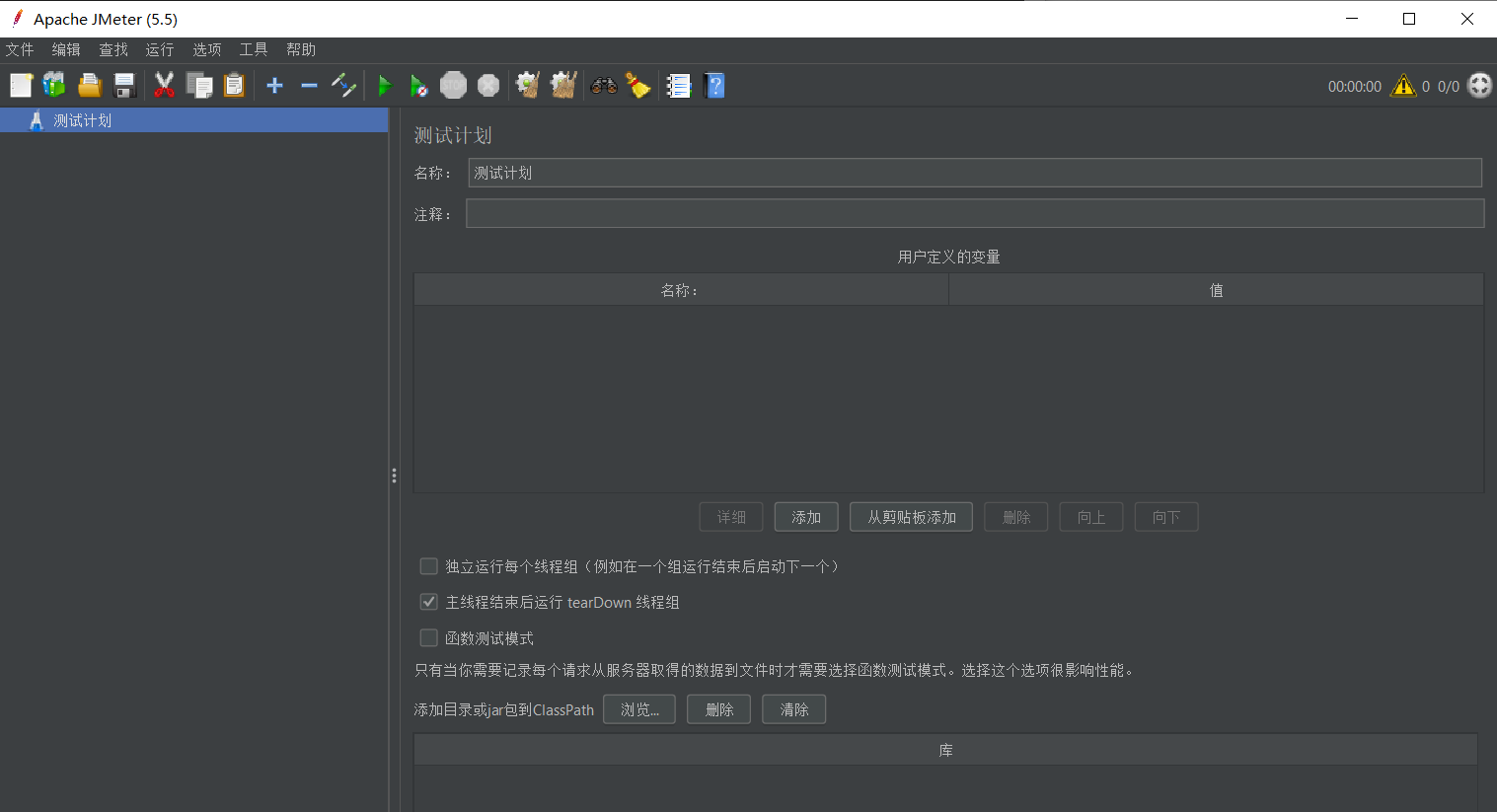
（5）将软件语言改为汉语

·修改配置文件 jmeter.properties

通过记事本打开该文件，找到第39行修改以下字段：language = zh\_cn



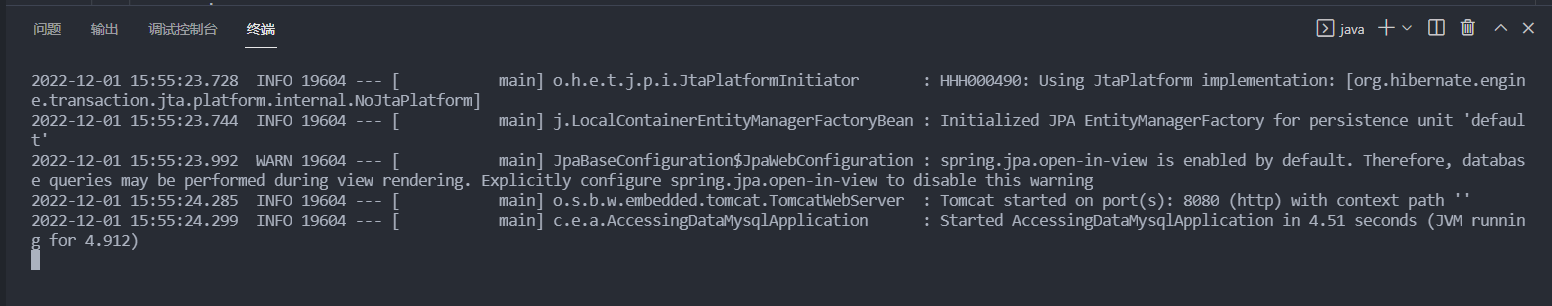
·随后重新启动，最终Jmeter主界面如下：



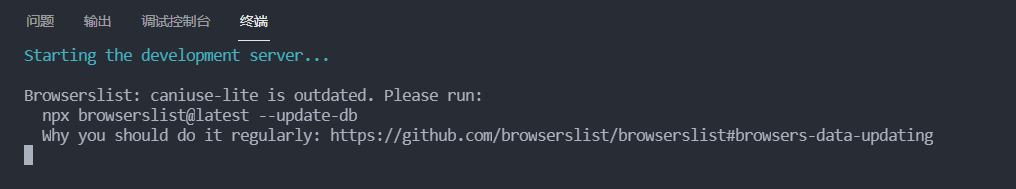
**2、对架构restfulDemo系统进行性能测试（批量添加数据）**

（1）运行架构restfulDemo系统

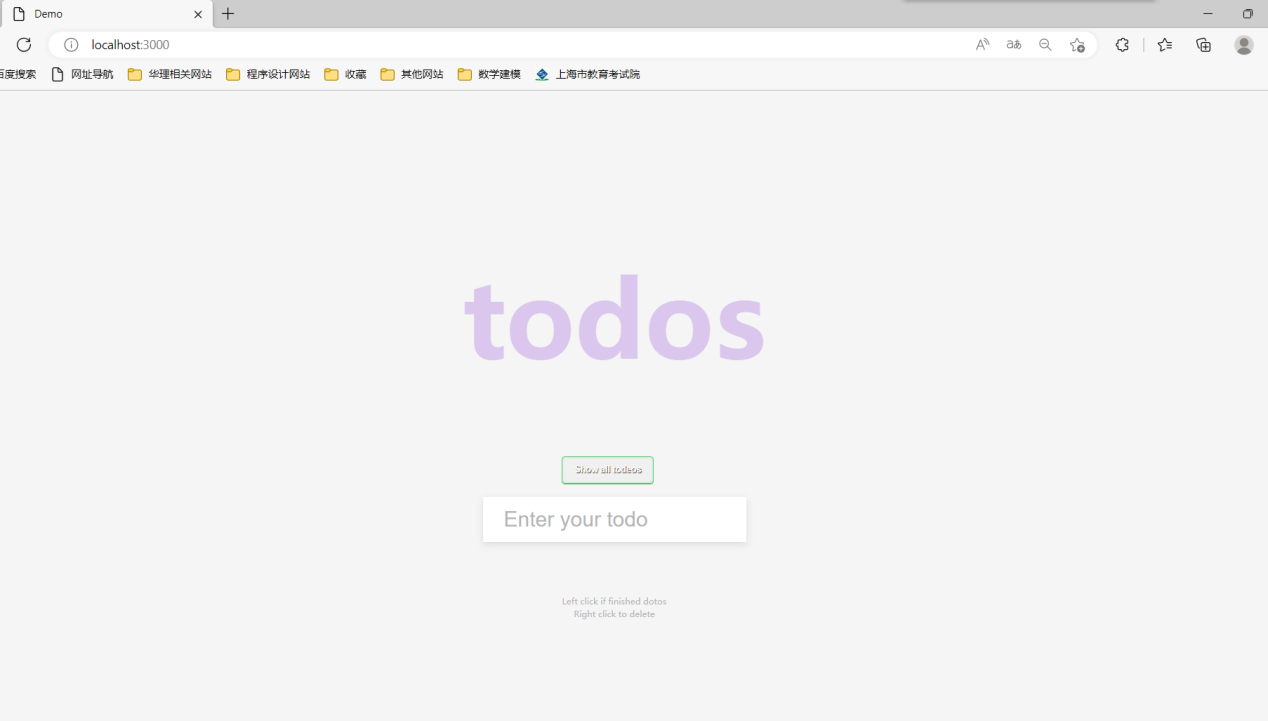
·启动后端，在...\restfulDemo\be-spring 目录下执行mvn spring-boot:run命令，启动成功：



·启动前端，在...\restfulDemo\fe\my-app 目录下执行npm start

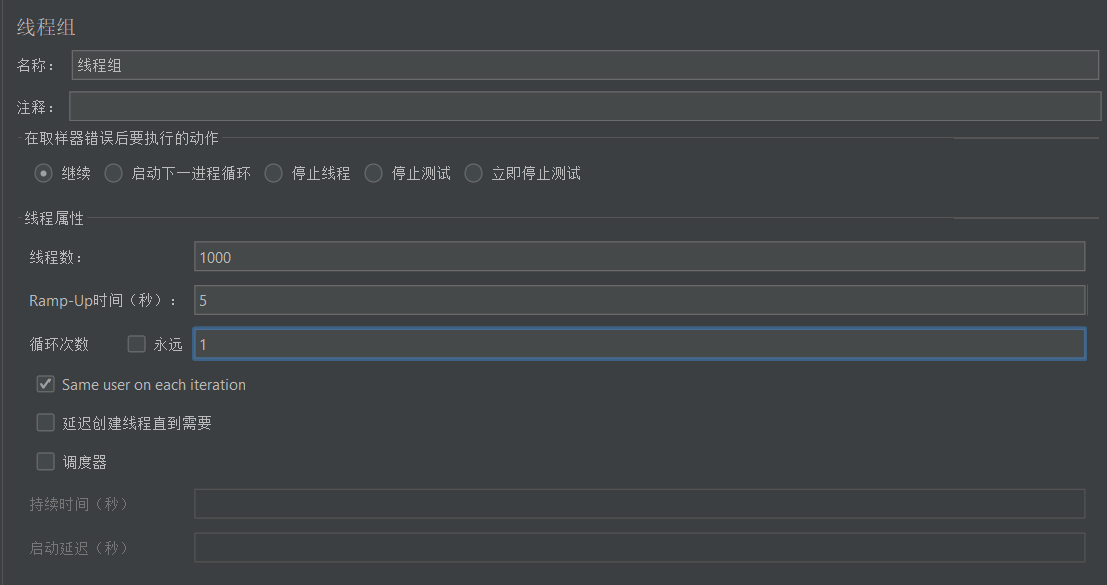


在浏览器中打开http://localhost:3000/，确认系统已经正常启动：



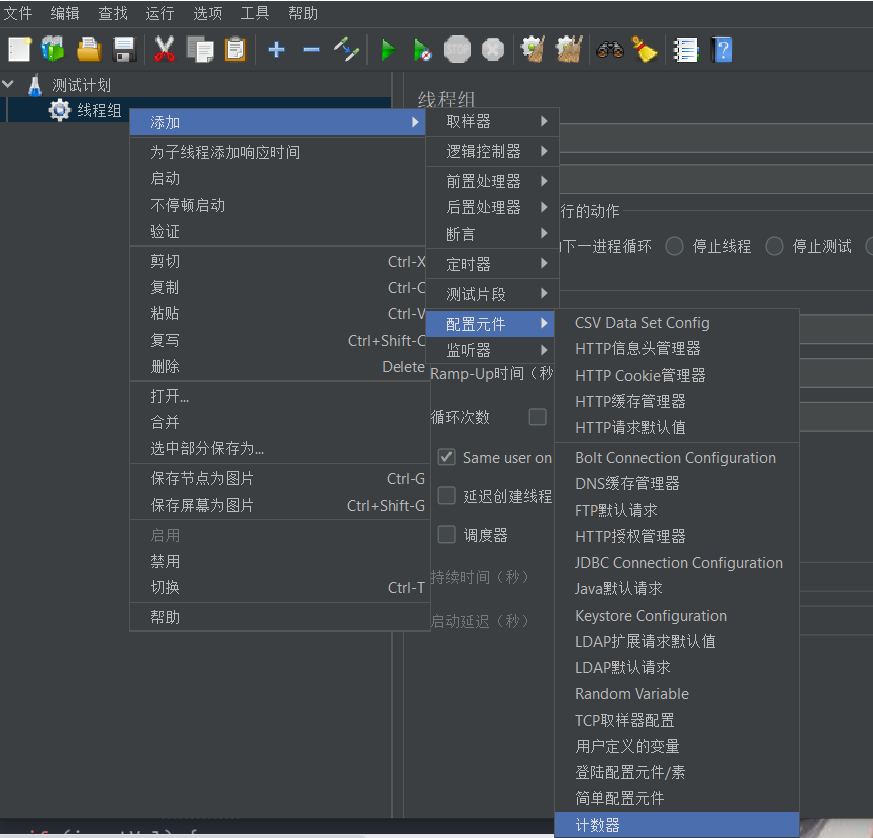
（2）在测试计划中添加取样器中的“线程组”

设置线程数为1000，Ramp-Up时间为5s：



（3）添加配置元件“计数器”

为了保证数据库主键的唯一性，并使生成的数据有规律、便于查看，在创建的线程组下添加配置元件中的“计数器”，初始值设置为0，递增为1，并设置引用名称为num，就可以在其他元件中通过${num}获取计数器的值：





（4）添加配置元件“用户已定义的变量”

在线程组下添加配置元件中的“用户已定义的变量”，对需要测试的数据进行封装。

添加一项，名称设置为id，值设置为test$num，此值可以作为添加数据的id值（但在本实验中没有用到）



（5）了解接口的相应参

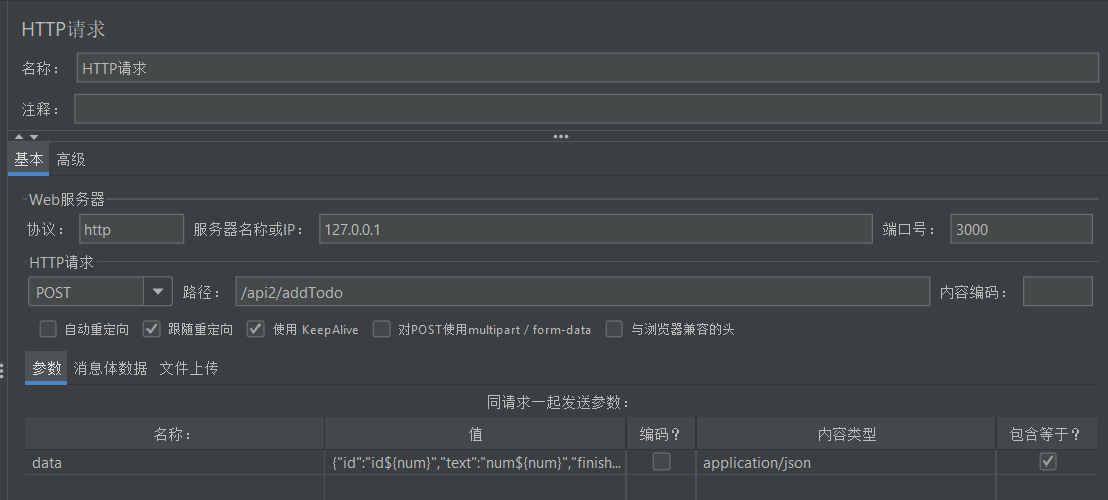
通过项目代码中的fe/my-app/src/App.jsx，可以看到：

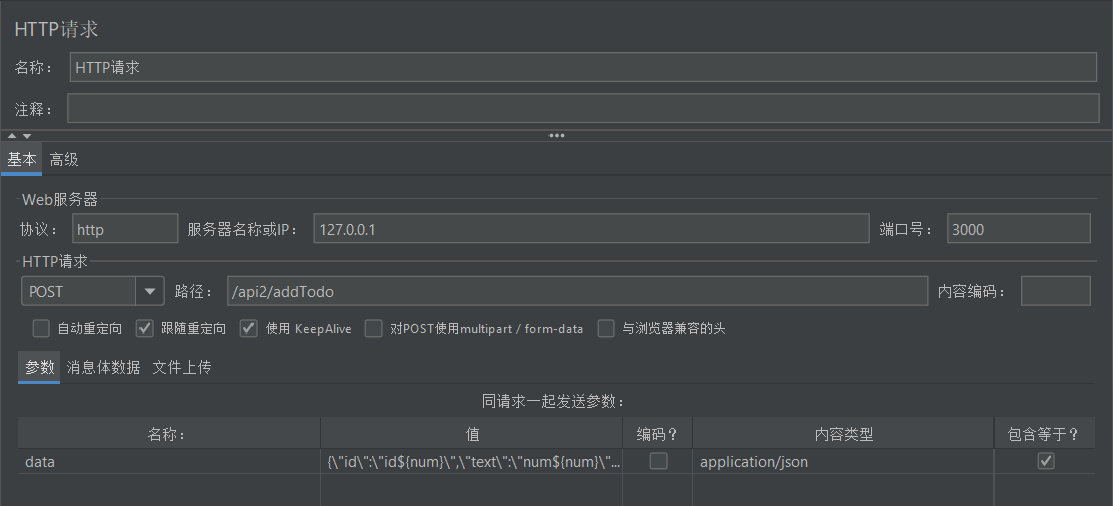
添加数据时使用的是post方法，调用了/${useApi}/addTodo接口；传递了数据data，数据格式为json，它有三个属性，分别为‘id’，‘text’’，‘finished’。

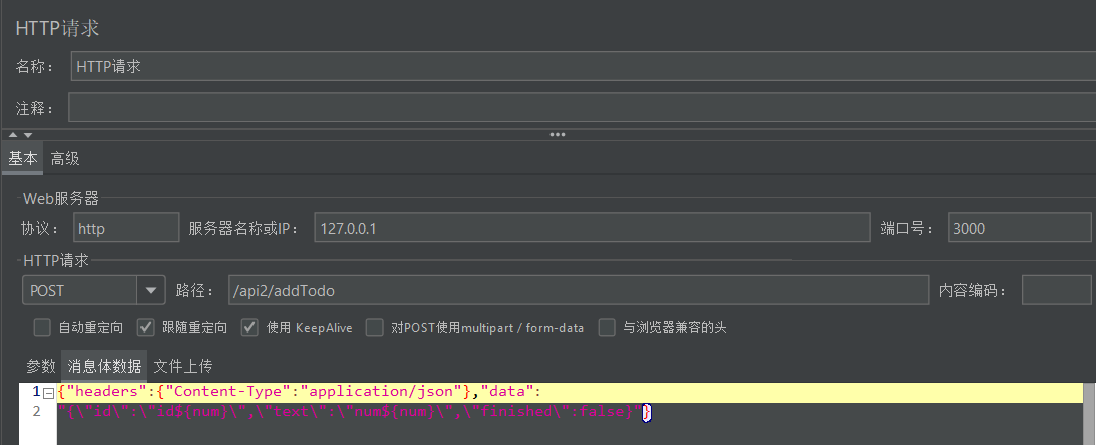


（6）添加取样器中的HTTP请求

在线程组下添加取样器中的HTTP请求，根据上述信息填写web服务器、HTTP请求以及发送的参数名称和值，如果需要调用测试计划中定义的变量，以${名称}格式调用（例如${num}，${id}）







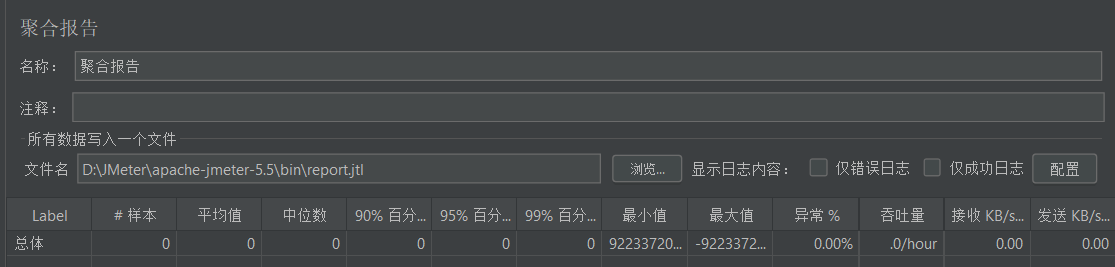
（7）添加定时器中的“同步定时器”

在HTTP请求下添加定时器中的“同步定时器（synchronized timer）”，设置模拟用户组的数量为200：

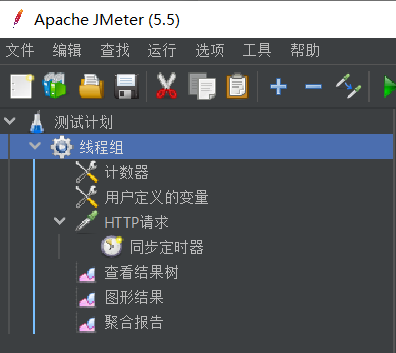
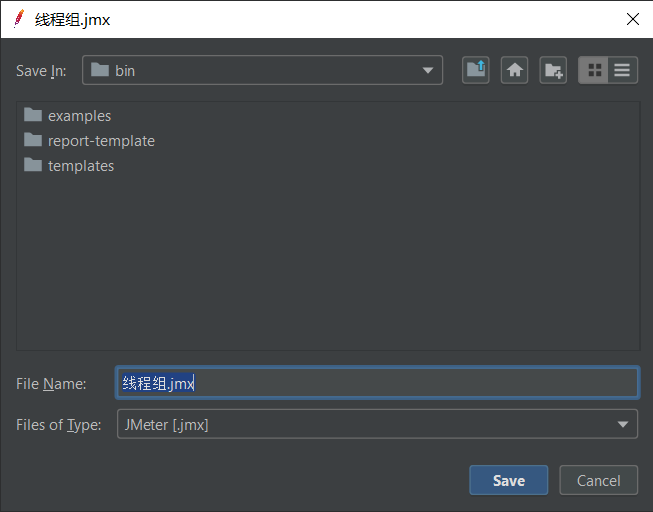


（8）添加监听器中的察看结果树、图形结果以及聚合报告

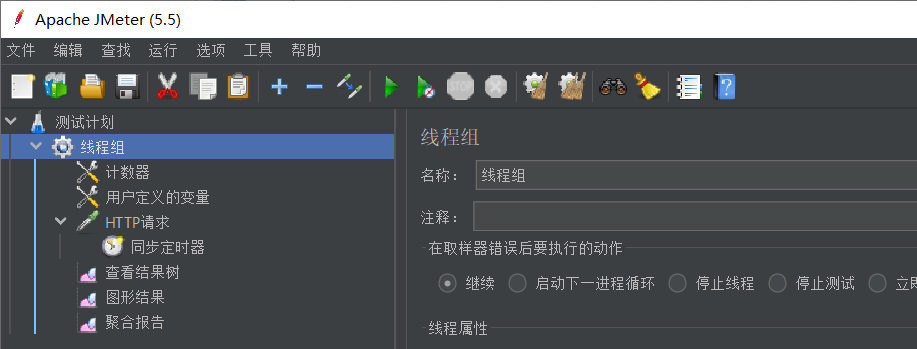
在线程组下分别添加监听器中的察看结果树、图形结果以及聚合报告，在聚合报告中填写一个文件名，使测试结果输出到文件中

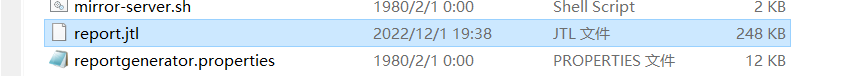


到目前为止，测试计划中线程组目录如下：

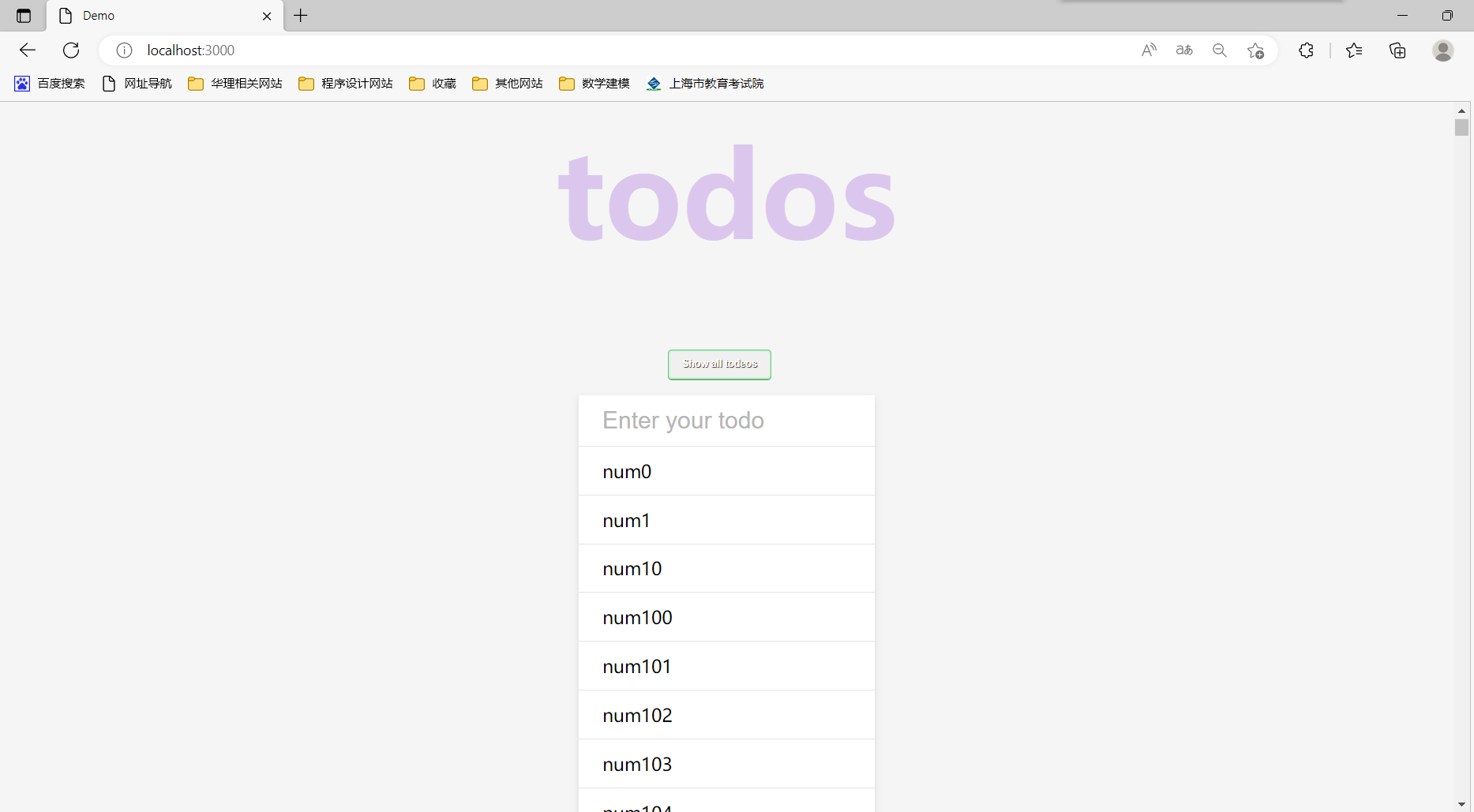
（9）启动测试计划，生成report,jtl文件

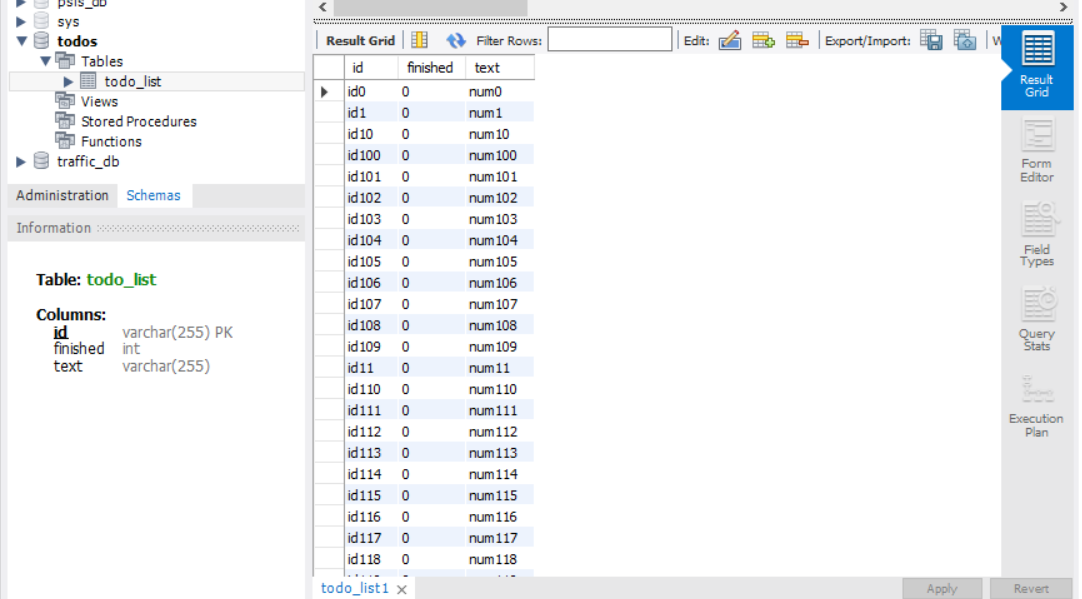




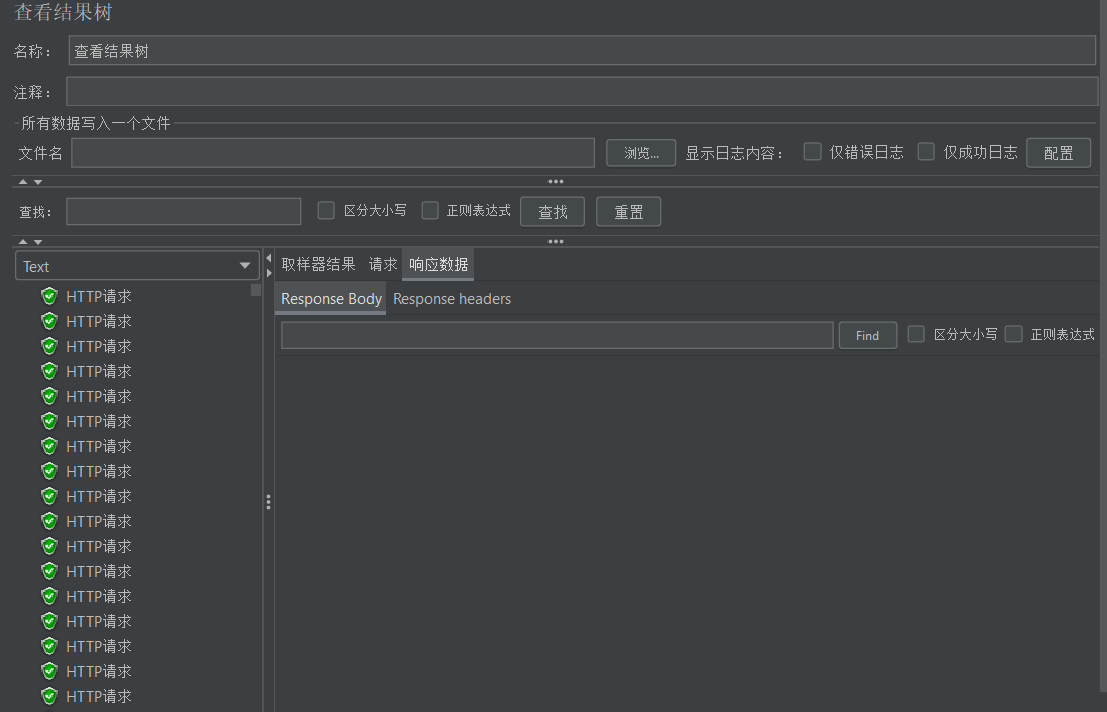
（10）在网页、查看结果树、图形结果、聚合报告中查看测试结果

·网页获取后台数据的结果以及数据库中数据的结果

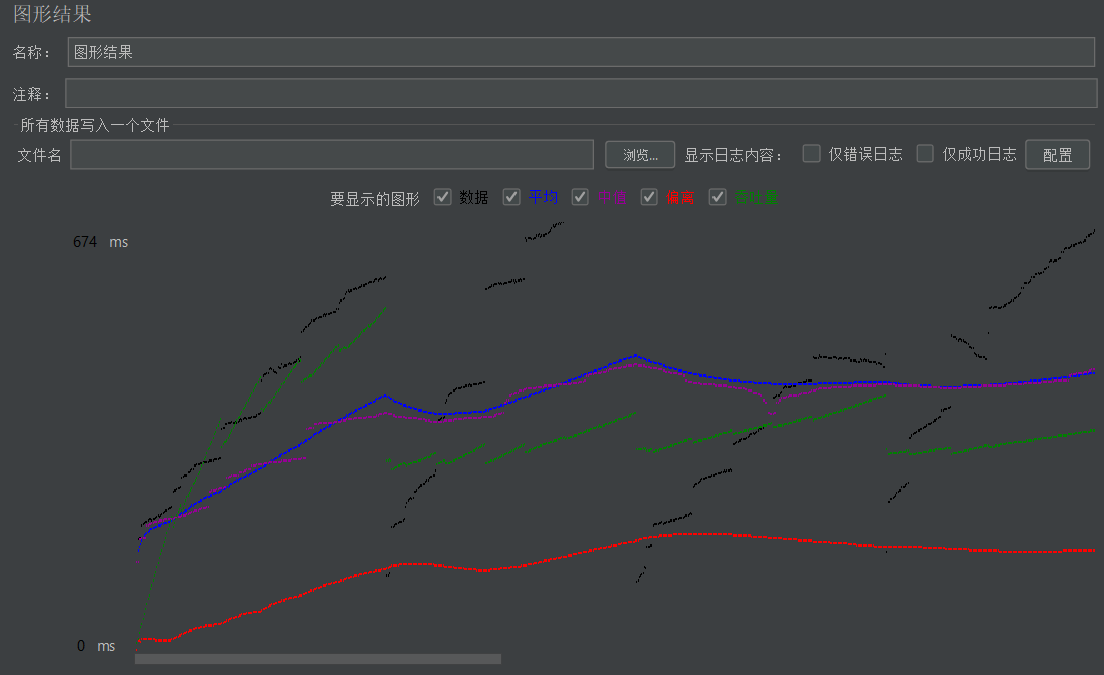




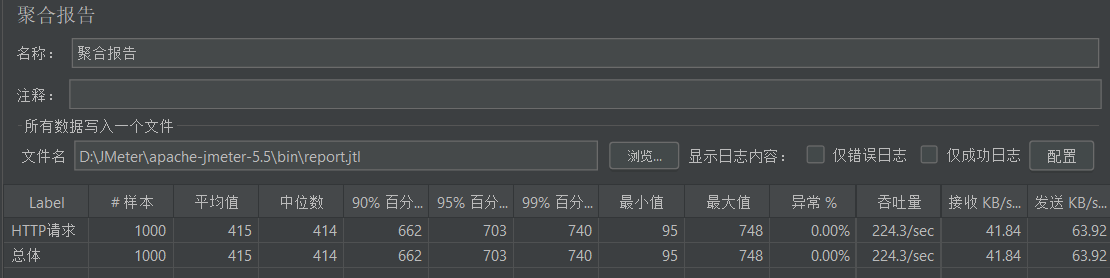
·查看结果树中的测试结果



·图形结果中的测试结果



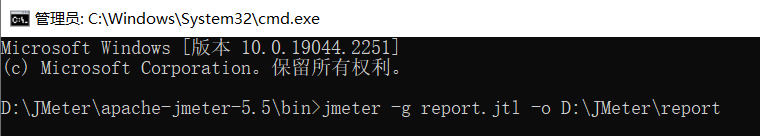
·聚合报告中的测试结果



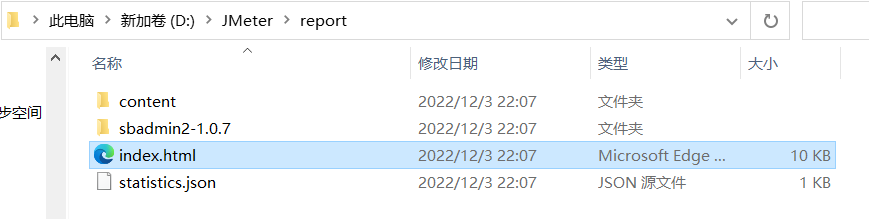
（11）将生成的jtl测试结果文件转换成html

·使用命令行前往jmeter安装路径下的bin文件夹中，执行以下命令

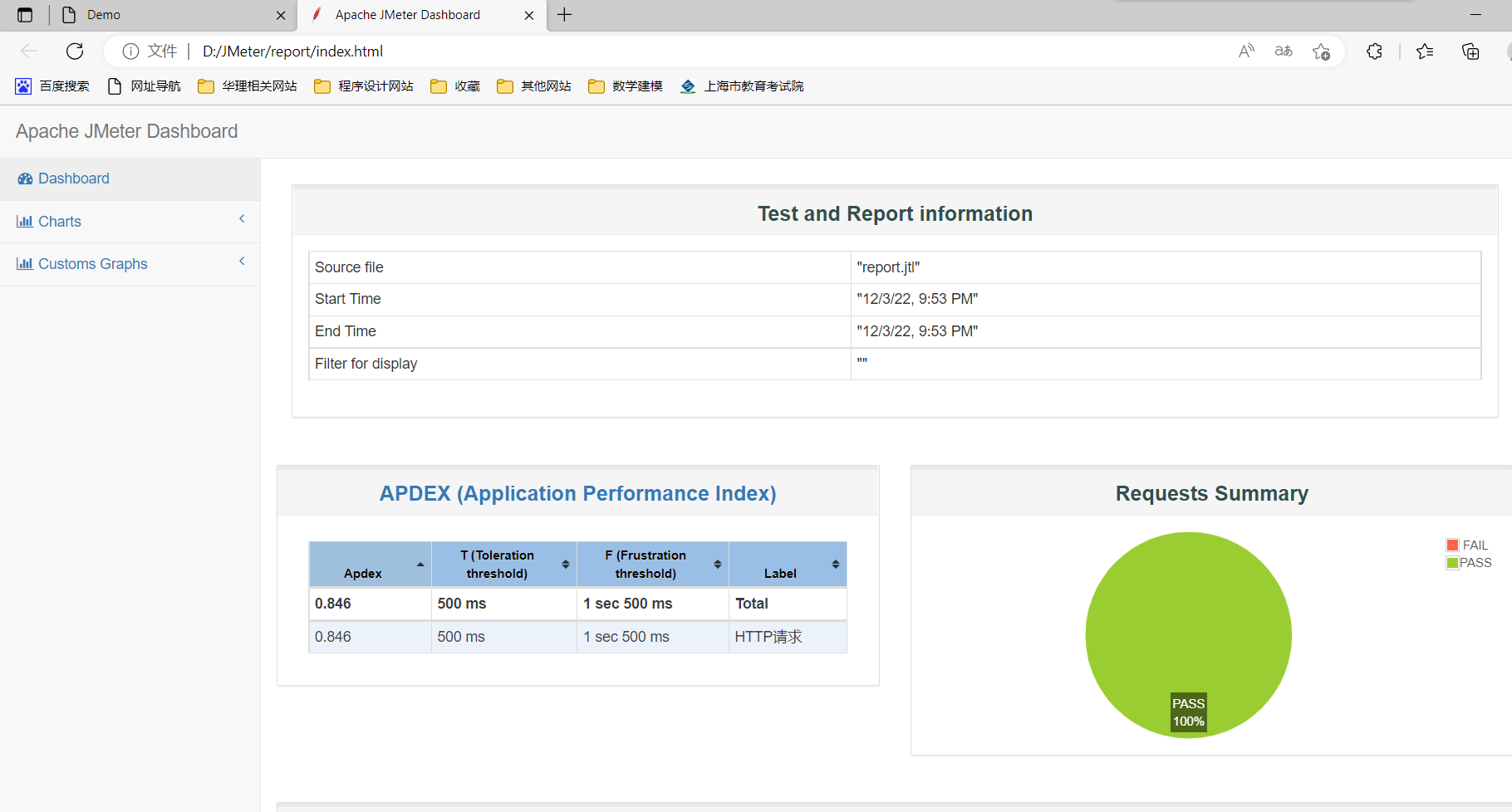
jmeter -g report.jtl -o D:\JMeter\report

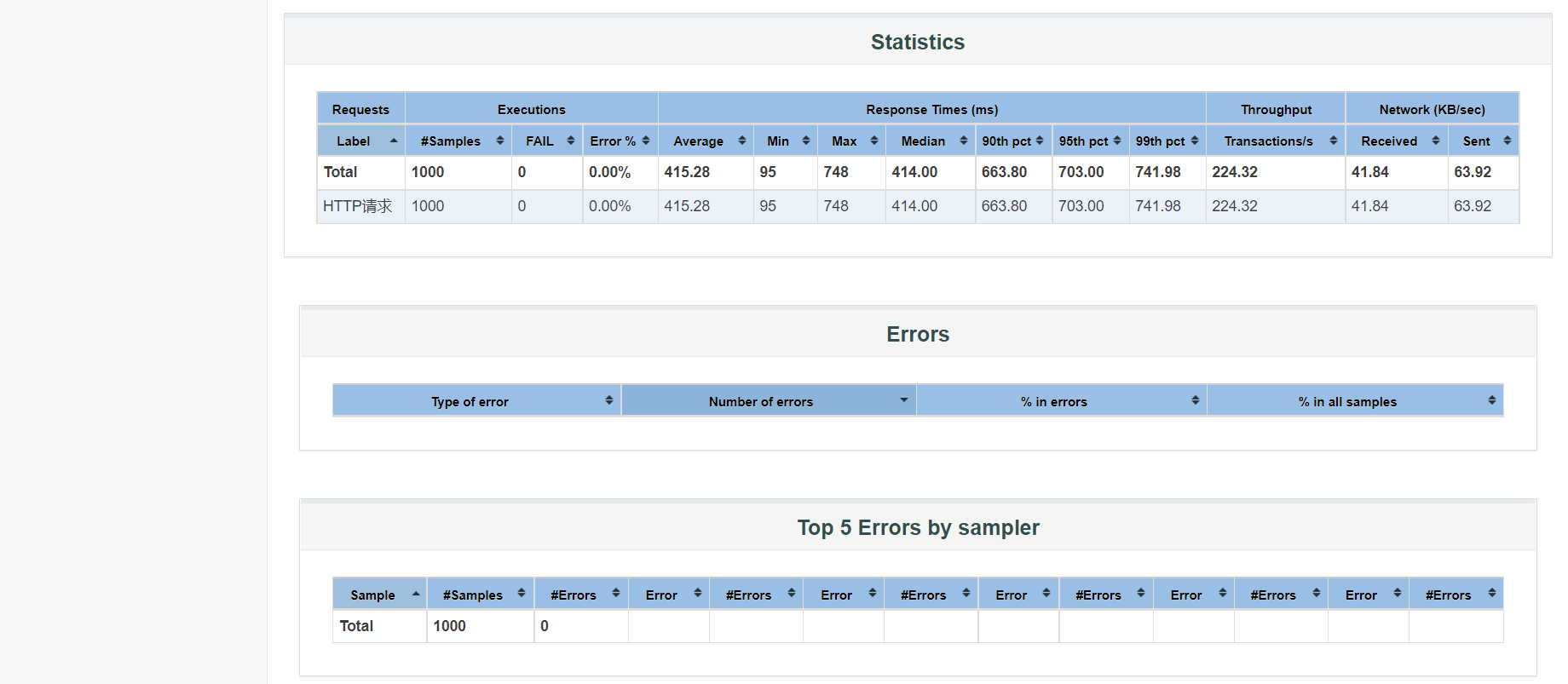


·执行命令后，在report文件夹中看到结果：



·打开index.html看到全面的测试报告：





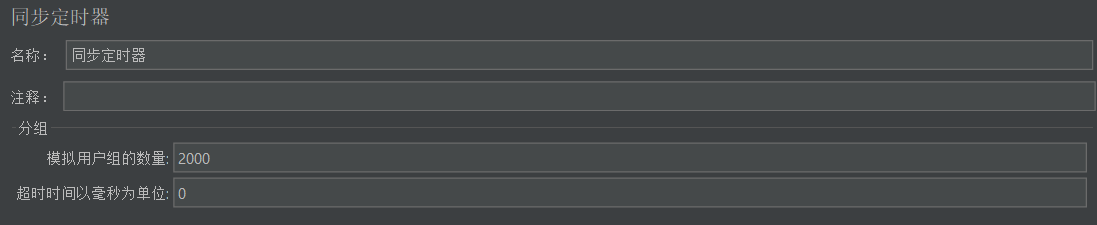
**3、对异常测试情况的判断**

（1）在数据库中对上一步骤生成的数据全部删除

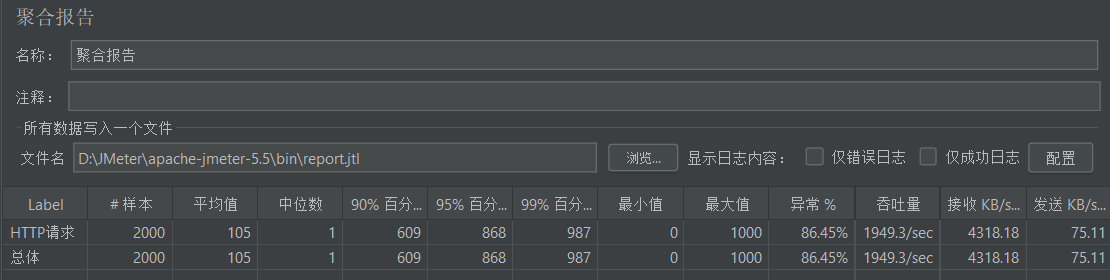
（2）对上一步骤的实验结果进行清除

（3）修改线程组的线程数为2000，同步定时器模拟用户组的数量为2000

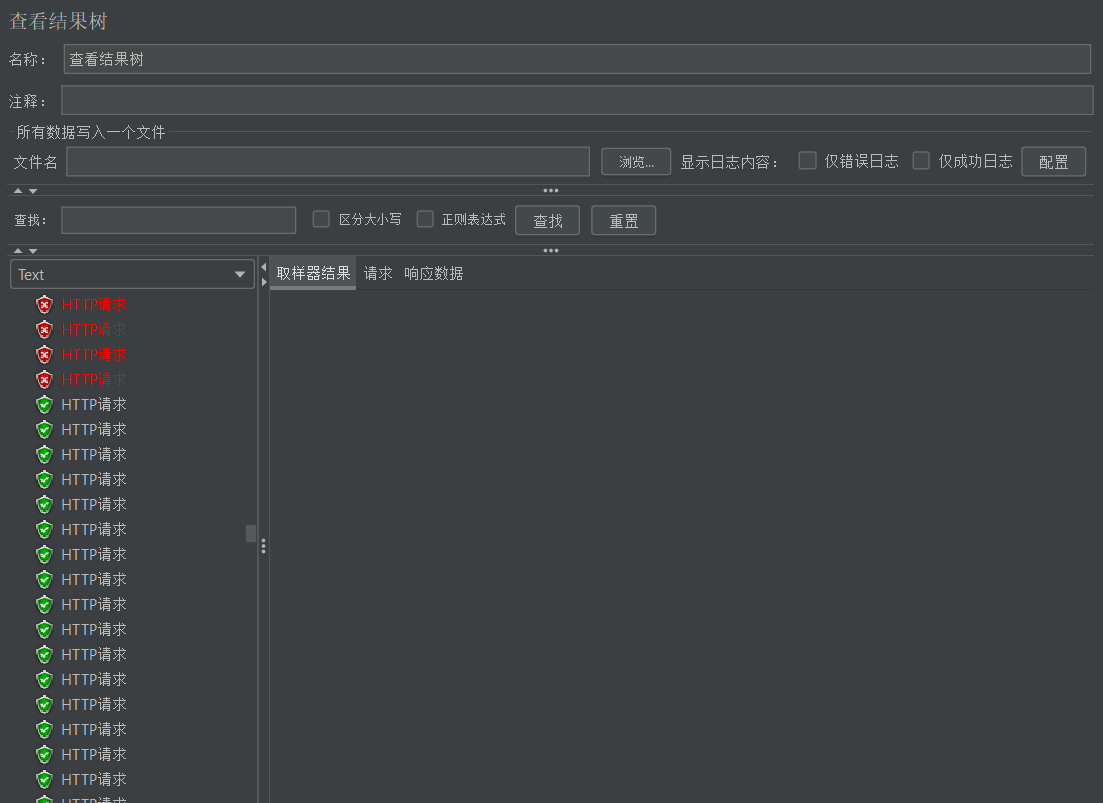




（4）启动测试，聚合报告结果如下，异常率为86.45%：



（5）查看察看结果树中，发现有失败请求，报错504，可能是由于并发量太大，使得部分请求超时，可见并发请求数已经达到服务器的瓶颈。

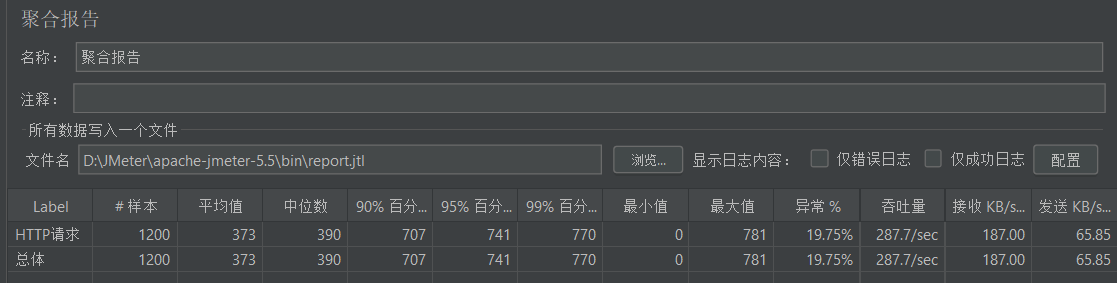


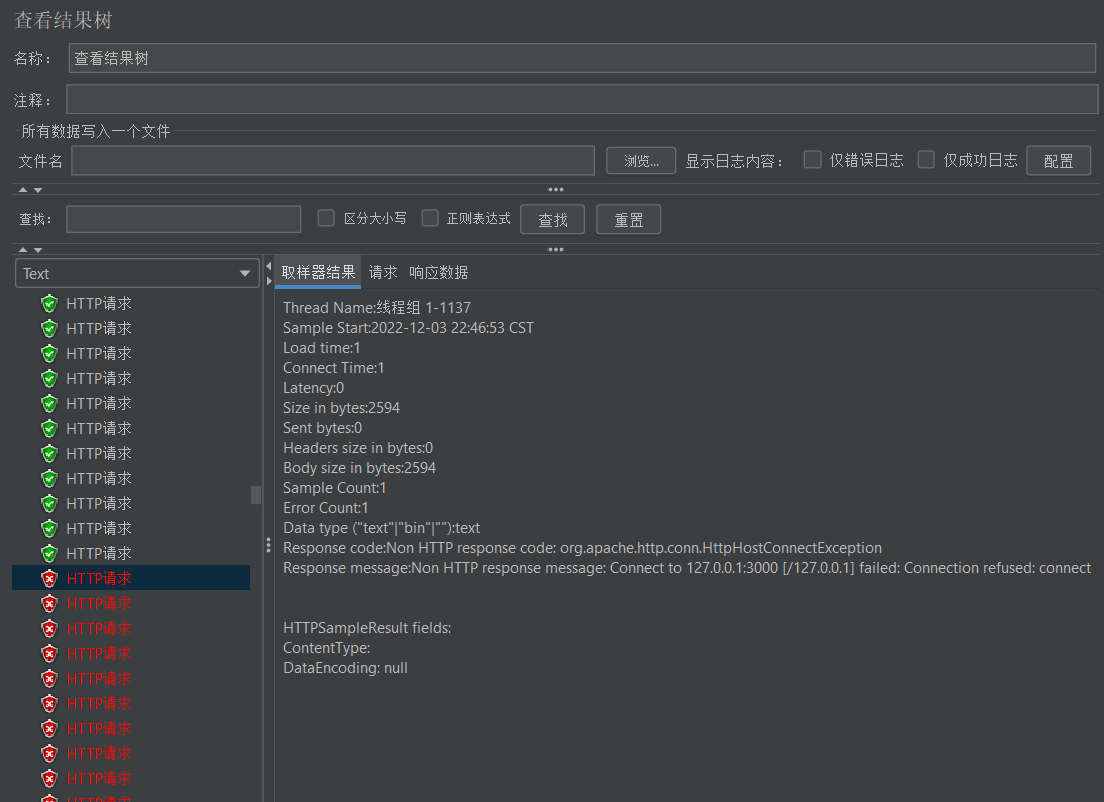
（6）再次清除之前的测试结果数据，修改线程组的线程数为1200，同步定时器模拟用户组的数量为300，再次测试。





由测试结果可知异常率19.75%，此时不再是504请求超时的报错，而是有部分数据异常http请求无响应，服务器返回的请求消息是拒绝连接：





**五、实验心得**

通过本次的实验，我了解了Jmeter性能测试工具的基本用途以及简单的操作，掌握了Jmeter性能测试工具的基本测试过程，并且能够使用Jmeter对项目进行负载测试。

在实验中，我在restfulDemo架构系统上进行了测试脚本的编写以及性能测试，对测试结果进行了分析，并尝试通过修改Jmeter线程组中的部分参数（线程数、模拟用户组的数量等）来加深对异常测试情况的了解、判断，发现其原因。

此次实验我的收获颇丰，同时这也为我在之后的项目推进及性能测试方面打下了实践基础。

**实验成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 教师签名：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**