**中国矿业大学**

**计算机科学与技术学院**

**2019级本科生实验报告**

课程名称 软件工程实践

实验题目 实验四 测试

开课学期 2021学年第一学期

报告时间 2021年12月27日

学生姓名 胡钧耀

班 级 2019-4班

专 业 计算机科学与技术

任课教师 刘迎春

# 实验目的与要求

## 实验目的

学习 IBM Rational Robot、IBM Purify、WinRunner、NUnit、Junit、CPPUnit、Webstress 等各类软件测试工具，以及 Test Manager 测试管理工具，针对所选系统实现情况，编写测试计划、设计测试用例，掌握软件自动测试方法。熟悉华为云测，学习覆盖测试管理、接口测试、性能测试等方法和技术，完成用例设计、测试执行、缺陷提交、生成报告各环节的任务。在开发环境实验中，要求学习Docker，掌握基于Docker搭建开发环境，应用Docker的方法。

## 基本要求

1. 下载、安装 IBM Rational Robot、IBM Purify、WinRunner、NUnit、JUnit、CPPUnit 、Webstress、QTP、LoadRunner 和 Test Manager 等工具软件，学习软件使用；
2. 掌握 IBM Rational Robot 中测试脚本录制、测试代码编写方法，针对实验代码进行自动测试；
3. 了解程序运行错误分析软件 IBM Purify 的工作原理，使用 IBM Purify 进行代码错误检测，分析检结果；
4. 采用 WinRunner 等黑盒测试工具进行所选系统的黑盒测试；
5. 运用 NUnit、JUnit、CPPUnit 工具软件进行白盒测试用例设计及自动测试；
6. 运用 Webstress 工具软件进行性能自动测试；
7. 了解 Test Manager 测试管理工具的使用方法。
8. 学习 .Net、JAVA 集成开发工具中的软件测试、调试。；

# Webstress 性能测试工具

## 简介

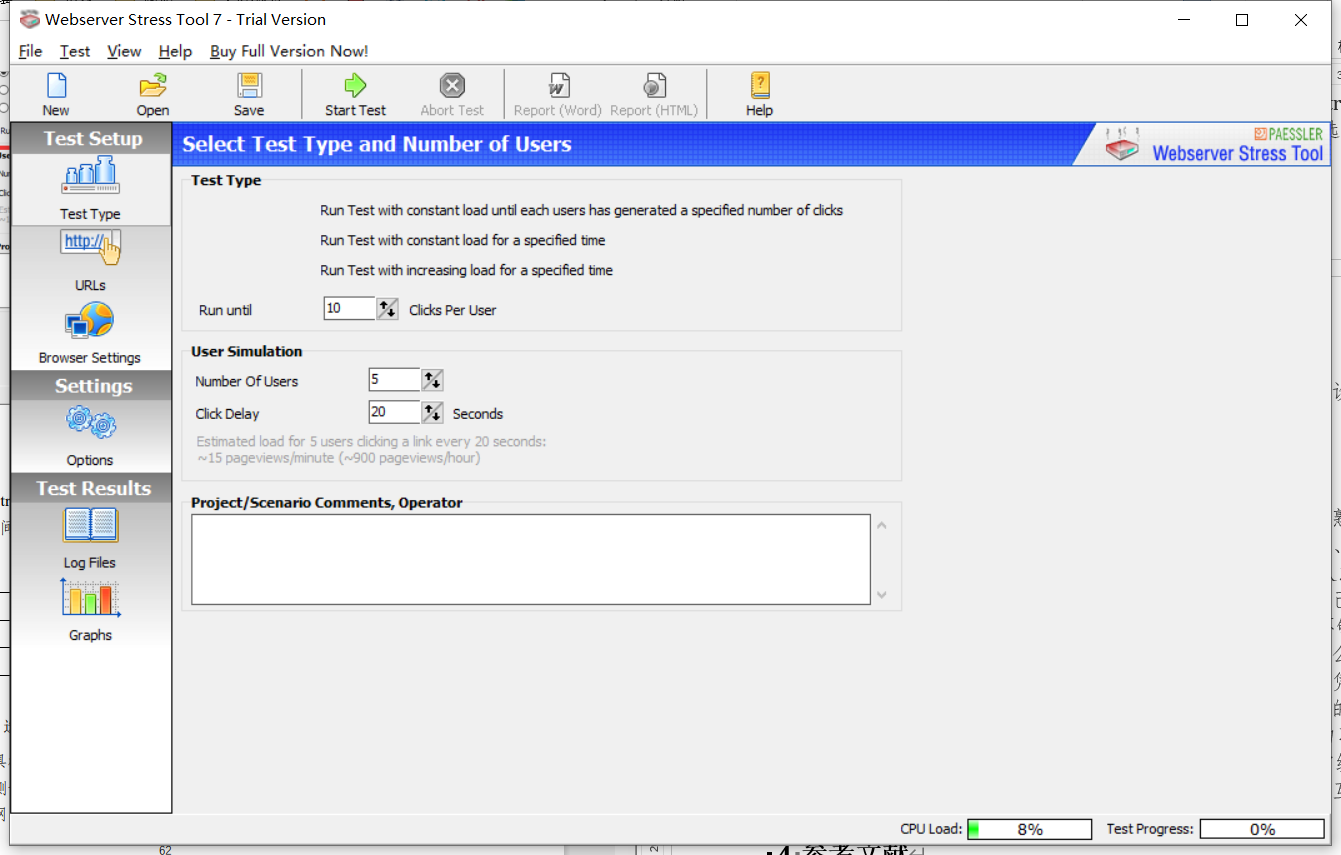
Webserver Stress Tool 能够模拟大量用户通过 HTTP 协议访问网站。根据用户的设置，模拟用户不仅可以下载 HTML 网页文件，也可以下载图片、动画等，跟普通用户进入网站的行为一模一样。

Webserver Stress Tool 提供一些报告和日志来显示对网站服务器和网络平台有价值的数据。

Webserver Stress Tool 的功能从左边一栏可以看到，分为三大项，为别是建立测试、设置、测试结果，用分为 6 小项，分别是测试类型、网址、浏览器设置、选项、日志文件、图形。

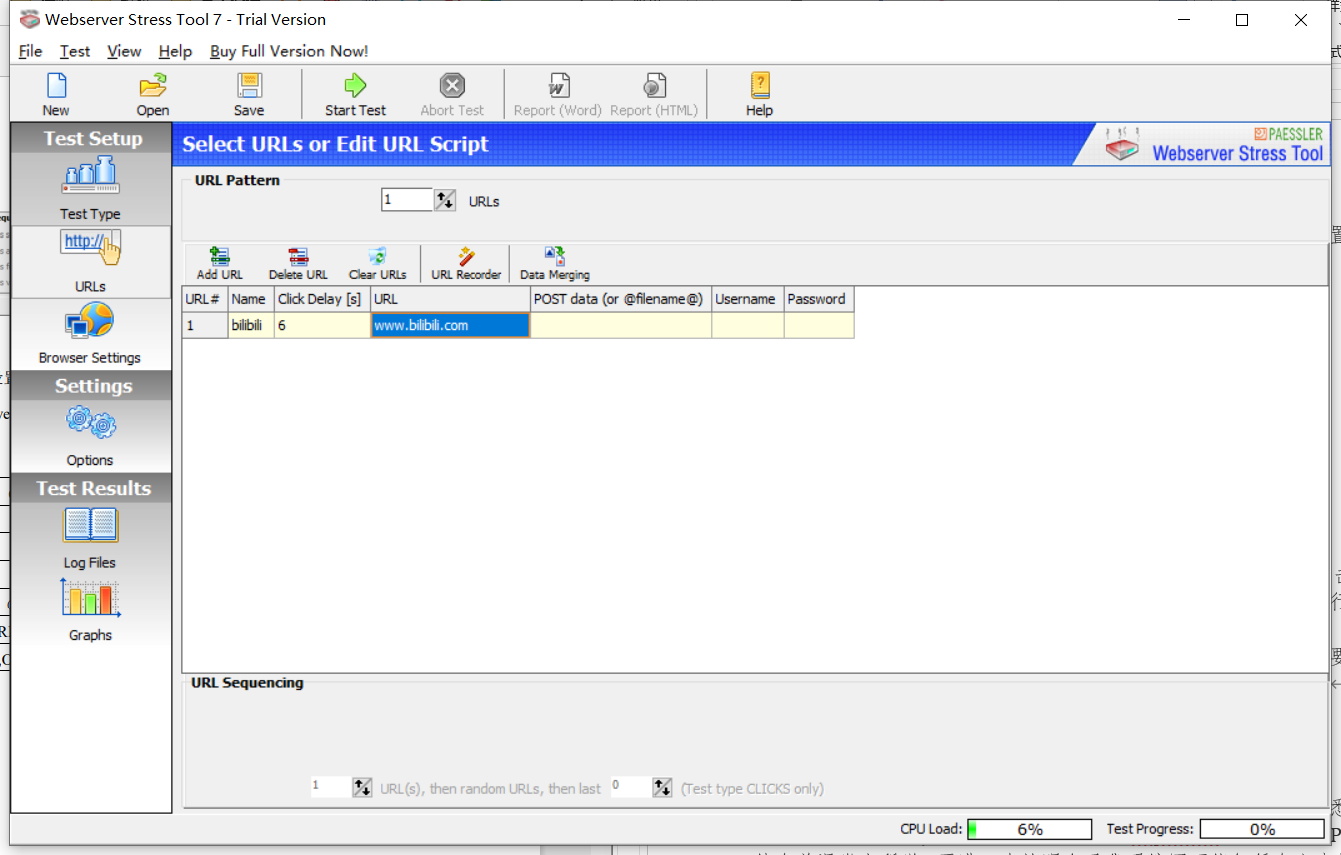
## 使用 Webstress 测试 Bilibili

选择测试类型和用户数量。当第一次打开 Webserver Stress Tool 时，会自动到选择用户和测试类型窗口，或者点击左边的工具栏选择 Test Type 进入此窗口。查看 User Simulation 项目，Number of Users 设置模拟访问者的人数，Click Delay 设置指定的时间间隔。

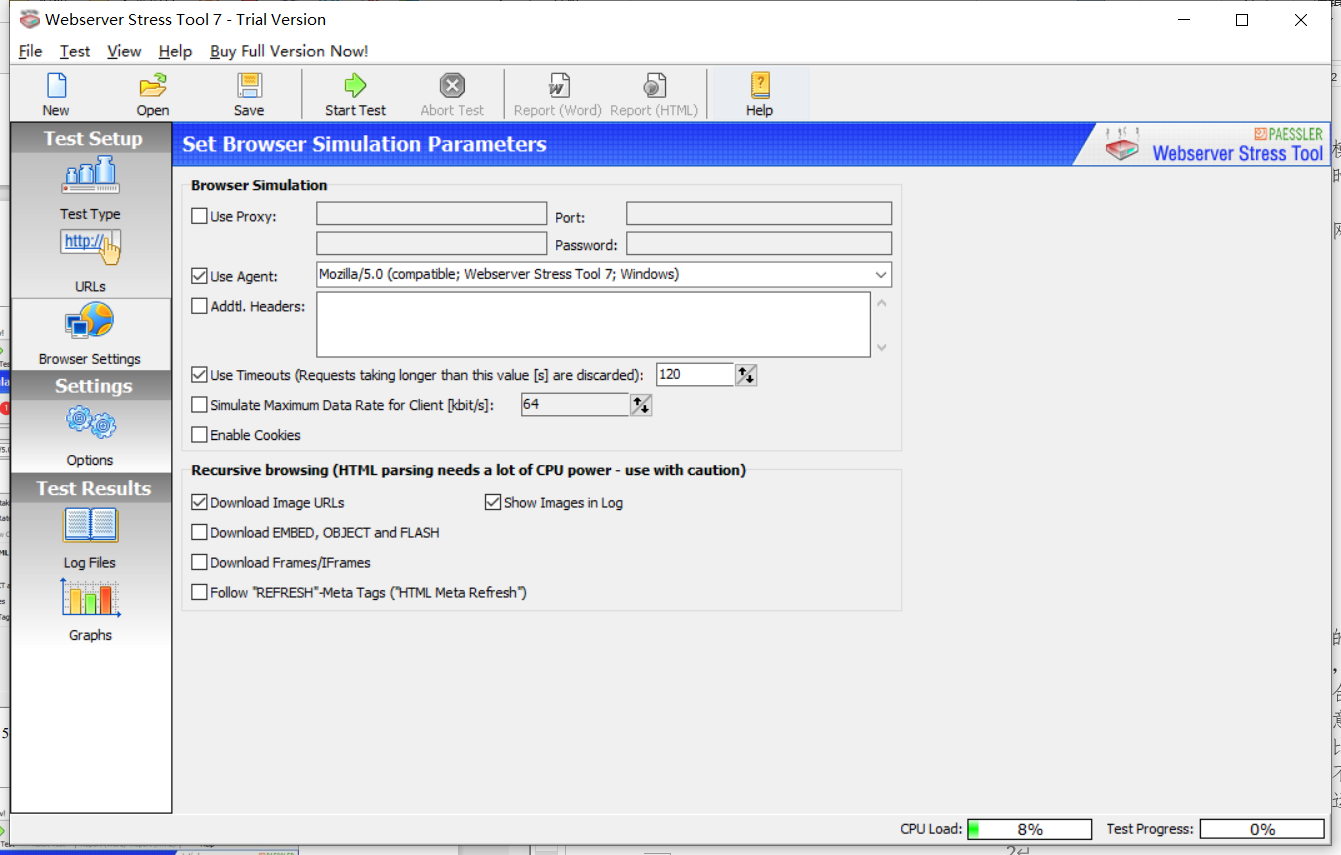


Webserver Stress Tool 提供了三种主要测试类型，分别为点击测试（模拟用户点击）、时间测试（测试运行指定的时间）、梯度测试（运行时定的时间，梯度增长）。

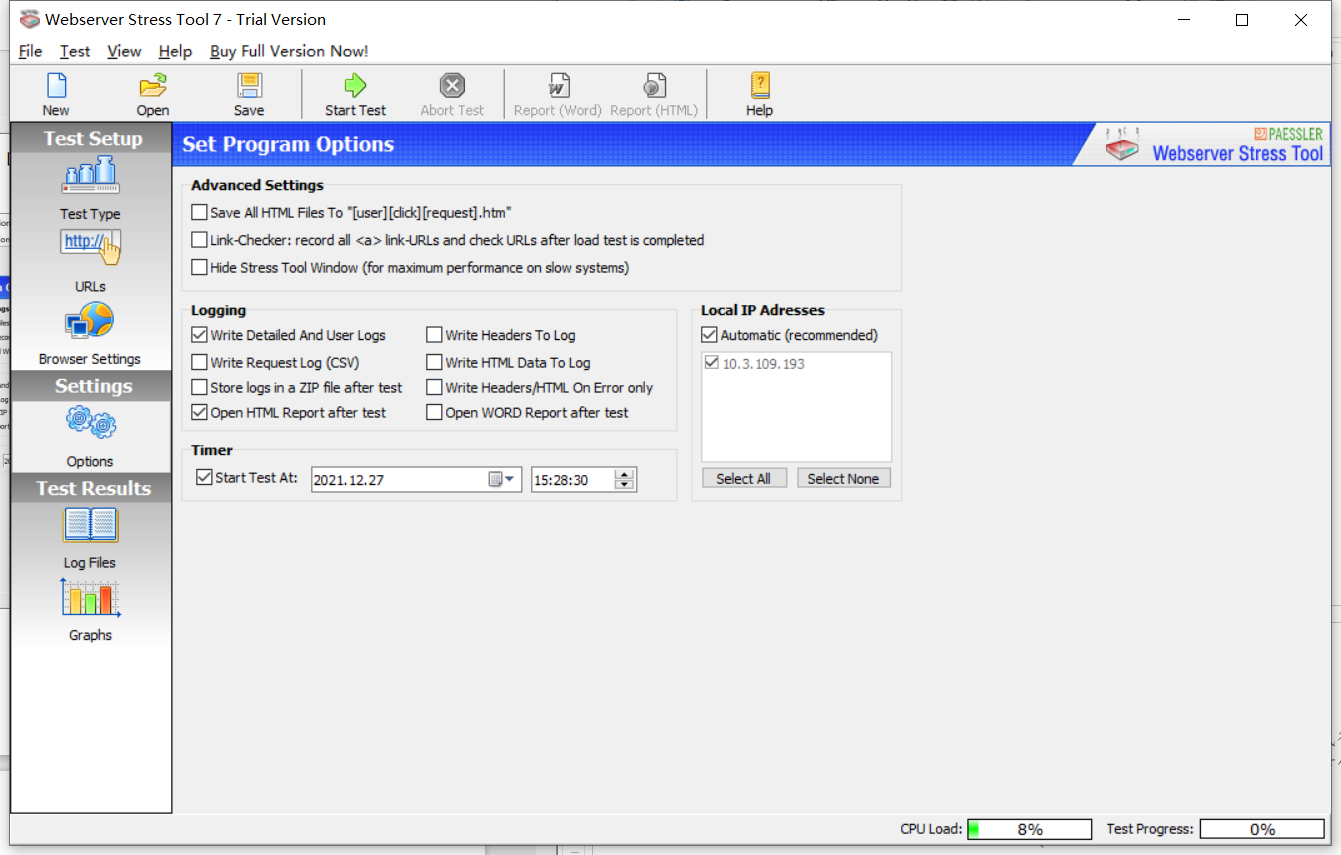
选取 URL 选项。点击工具栏选择 URLs 进入此窗口。新增要测试的网站网址。以淘宝网（www.bilibili.com）为例添加相应的 URL。



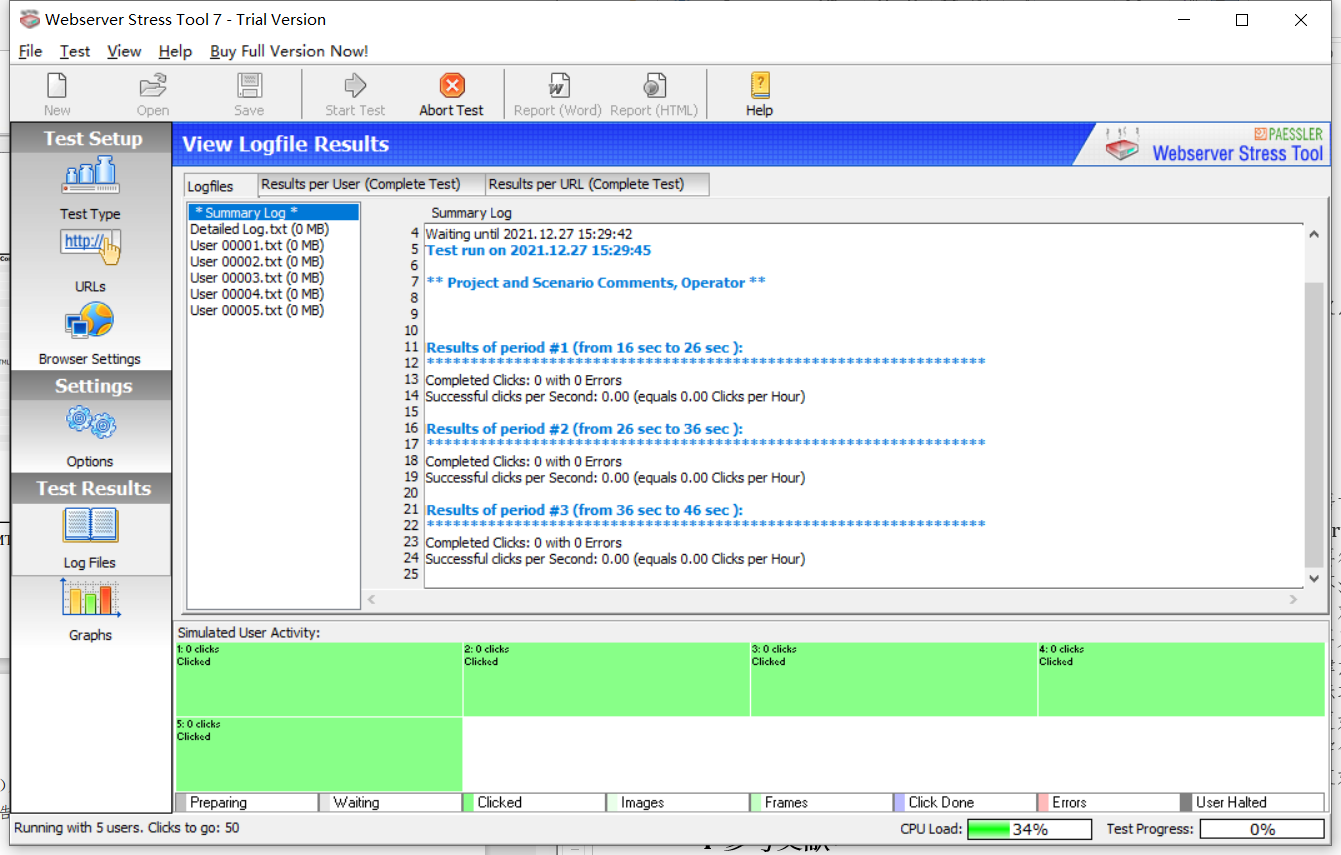
点击左侧 Brower Settings 按钮，进入模拟浏览器设置。



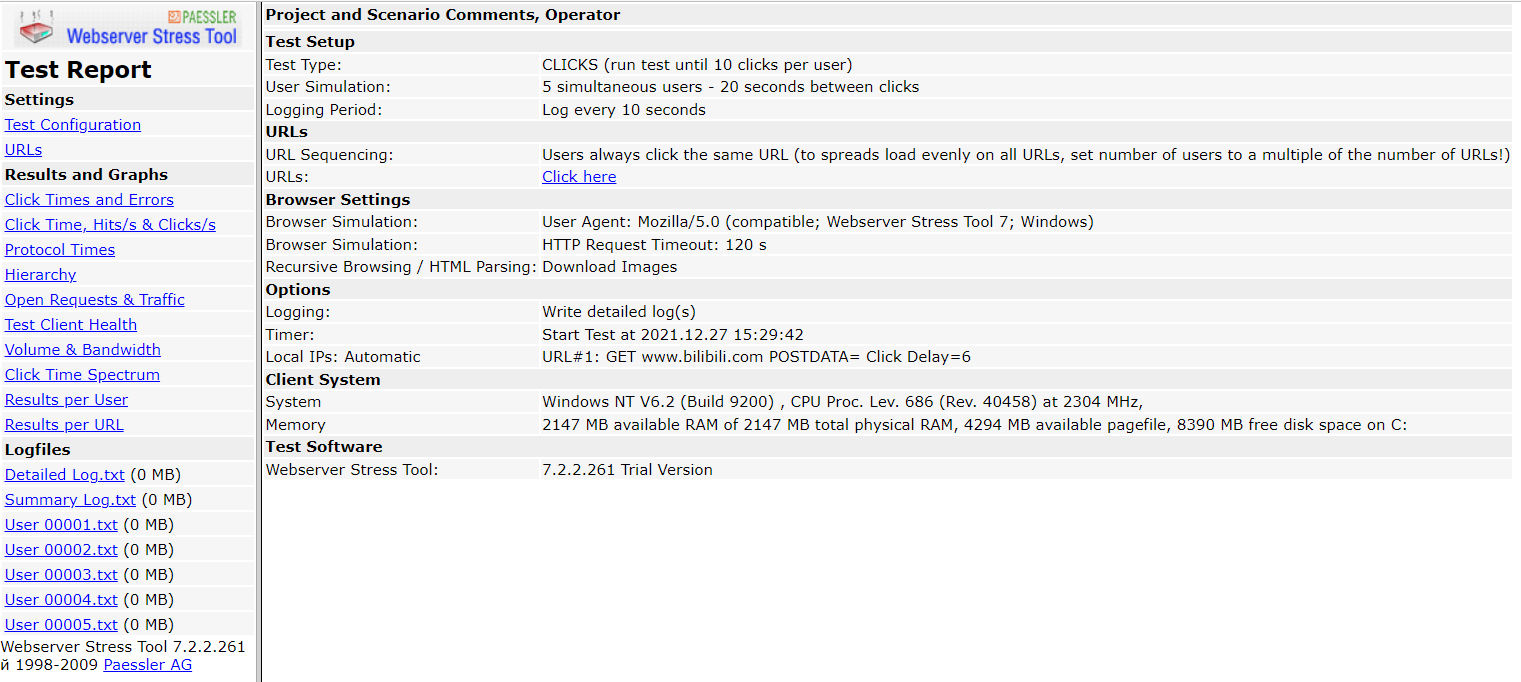
在左侧点击 Options 按钮，进入浏览器选项设置。如希望定时执行测试操作，则在其中的 Timer 项目中勾选 Start Test At 复选框，在其右侧设置预定的日期和世界，设置日志记录的内容。



点击上方工具栏上的 Start Test，开始进行指定网站的测试。再次点击 Abort Test 即可结束测试。

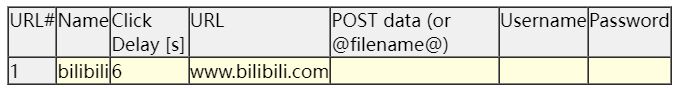


## 结果分析

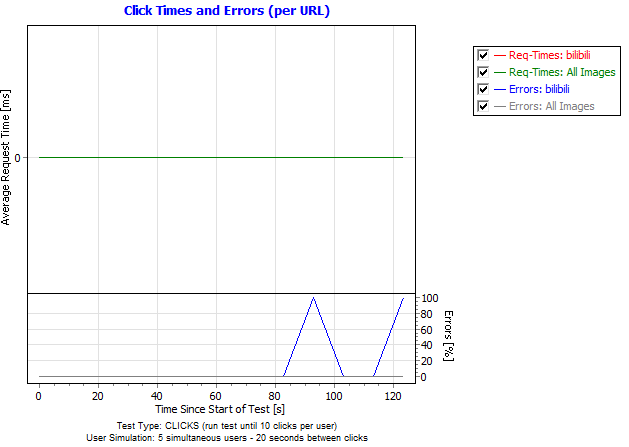
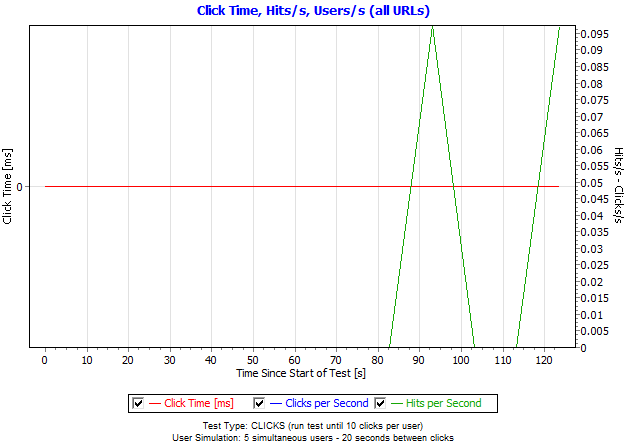


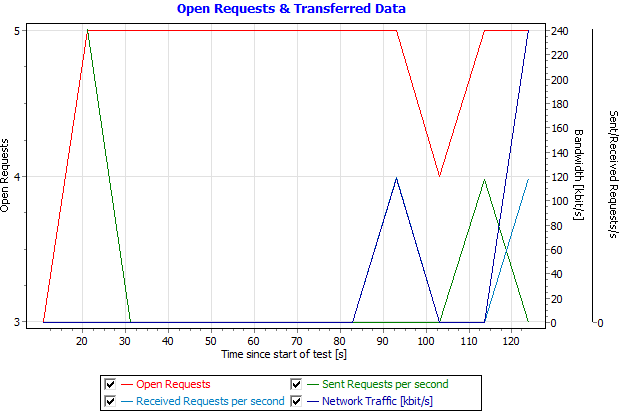
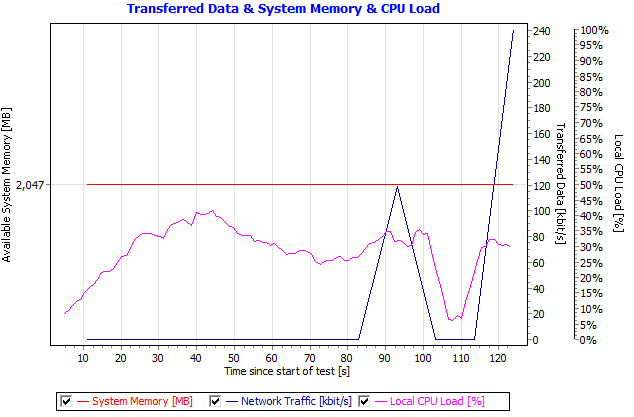
测试完成后可以打开相应的 HTML 格式报告或是 Word 格式报告。在这里选取 HTML 格式报告加以分析。

Setting–URLs 可以查看 URLs 的设置。



此外，还可以查看点击次数和错误、点击时间速度、请求和数据传输、转移数据和系统存储等，截图如下。

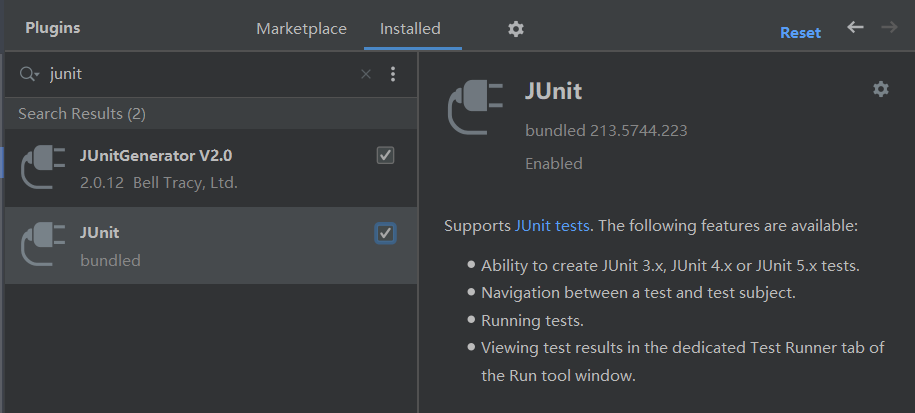
# Junit 的使用

## 简介

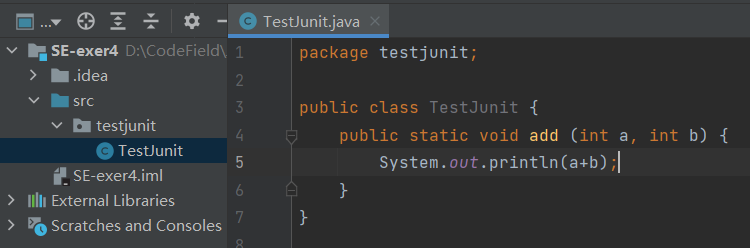
Junit 是一个 Java 语言的单元测试框架。它由 Kent Beck 和 Erich Gamma 建立，逐渐成为源于 Kent Beck 的 sUnit 的 xUnit 家族中最为成功的一个。 Junit 有它自己的 Junit 扩展生态圈。多数 Java 的开发环境都已经集成了 Junit 作为单元测试的工具。

* 1. **IDEA 下 JUnit 对代码进行单元测试**

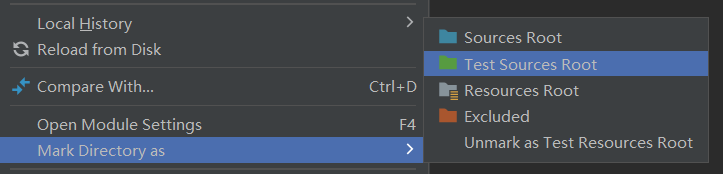
安装方法：file-setting-plugins-搜索 Junit-安装插件（一般已默认安装，无需手动安装）。



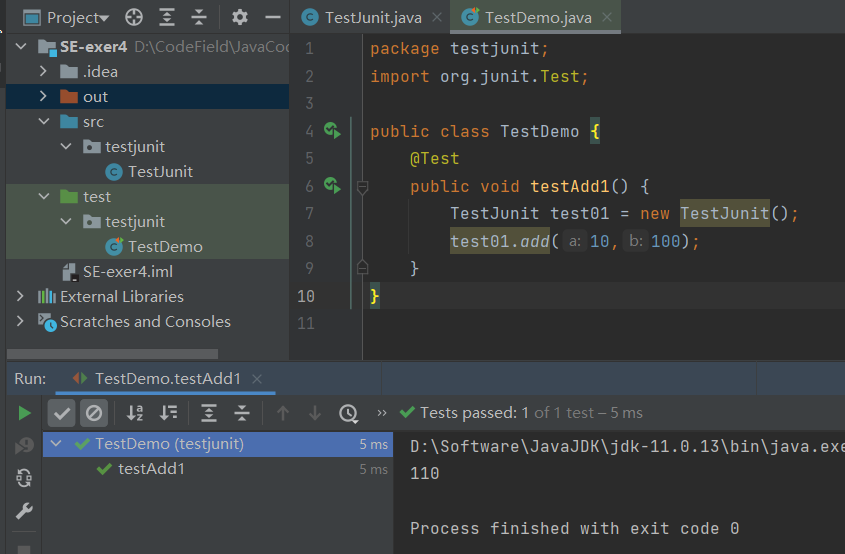
在 src 下创建一个包命名为 testjunit，包内创建类。



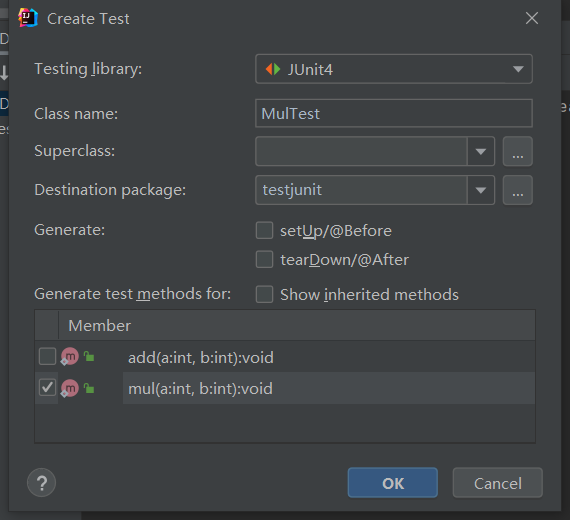
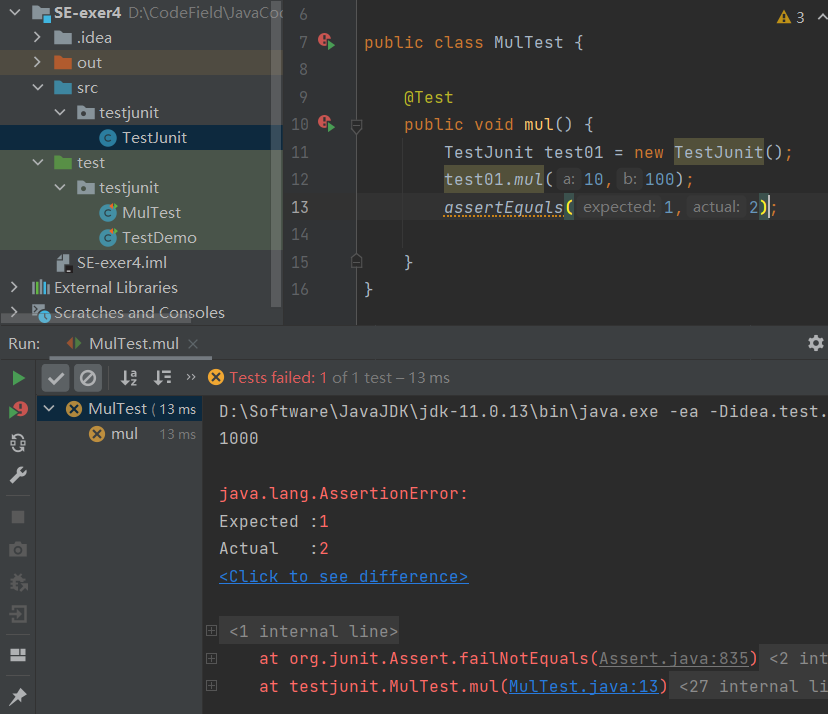
在src的同级目录下创建一个文件夹名为test，右键选择Test Sources Root，于文件夹中创建与testjunit包名称一致的包,并创建测试类。



在TestDemo.class中创建测试方法：单元测试方法时候，方法命名规则：public void 方法名() {}。方法名不能有参数。使用注解方式运行测试方法，在方法的上面：@Test；如无注解，则不能运行。



再写一个 mul 函数，使用自动生成方法生成测试类，快捷键是Ctrl+Shift+T。编写测试代码，测试错误的显示情况如下：

## 注解说明

@Test：表示方法进行单元测试

@Ignore：表示方法不进行单元测试

@Before：在每个方法前执行运行

@After：在每个方法之后运行

# NUnit 单元测试

# CPPUnit 单元测试

# 华为云测

## 注解说明

# 实验感悟

这次实验，学习 PHP 、 Java、 Python 集成编程环境，熟悉各语言的编码规范。学习了 JetBrains 全家桶中的 PhpStrom 、 IDEA 、 PyCharm，结合着课堂上所学，更进一步的明白了代码编写不能仅凭个人志趣，要符合规范。此外，实验中在阅读代码规范的过程中，发现了许多自己以前不注意的点。其中，很大一部分的原因在于以往所编写的代码太小不够完整。对比规范，让我切身的感受到了编写易懂又便于维护的代码是多么的重要又不容易，让我不再想当然的觉得软件开发只是敲代码，只是一件凭灵感肆意运行的事情。未来将学以致用，将软件面向对象的分析方法真正的在实际项目中加以运用，同时在平常学习中，逐步培养起规范编写代码的习惯，更好的开发软件。此外，我觉得 JetBrains 系列软件的界面风格非常统一，学会了一个开发IDE就能很快迁移到另外的IDE上面，同时界面交互非常友好，这也是我们设计软件UI的时候需要注意的点。

# 参考文献

1. Introduction PHP编码规范（中文版） 看云https://www.kancloud.cn/thinkphp/php-fig-psr/3139
2. 阿里巴巴Java开发手册（嵩山版）https://github.com/alibaba/p3c/blob/master/Java%E5%BC%80%E5%8F%91%E6%89%8B%E5%86%8C%EF%BC%88%E5%B5%A9%E5%B1%B1%E7%89%88%EF%BC%89.pdf
3. PEP 8 -- Style Guide for Python Code https://www.python.org/dev/peps/pep-0008/

# 附件

附件1：

PHP面向过程代码：

键值对输出与九九乘法表的打印

<?php  
$x=array(1=>"Google", 2=>"Runoob", 3=>"Taobao");  
foreach ($x as $key => $value)  
{  
 echo "key 为 " . $key . "，对应的 value 为 ". $value . *PHP\_EOL*;  
}  
  
$j = 1;  
while($j<=9){  
 $i = 1;  
 while($i<=$j){  
 echo "{$i}x{$j}=".($i\*$j)." ";  
 $i++;  
 }  
 echo "<br />";  
 $j++;  
}

附件2：

Java 过独木桥多线程编程代码：

import java.util.HashSet;  
import java.util.Set;  
  
class People implements Runnable{  
  
 @Override  
 public void run() {  
 synchronized(this) {  
 System.out.println(Thread.currentThread().getName()+"正在过桥");  
 try {  
 Thread.sleep(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 System.out.println(Thread.currentThread().getName()+"已通过桥");  
 }  
 }  
  
}  
public class demoThread {  
  
 public static void main(String[] args) {  
 String[] name= {"一","二","三","四","五","六","七","八","九","十"};  
 Set<Integer> set=new HashSet<Integer>();  
 while(set.size()<10) {  
 set.add((int)(Math.random()\*10));  
 }  
 People people =new People();  
 for(int a:set) {  
 Thread thread =new Thread(people,name[a]);  
 thread.start();  
 }  
 }  
}

附件2：

Python二维码识别代码：检测图片中的编码点，并在图片上画出结果

# -\*- coding:utf-8 \_\*-  
import cv2  
import cv2.aruco as aruco  
import numpy as np  
import matplotlib.pyplot as plt  
  
'''检测图片中的编码点，并在图片上画出结果'''  
  
dict1 = aruco.getPredefinedDictionary(aruco.DICT\_7X7\_1000) # 编码点的类型,与生成的时候对应  
font = cv2.FONT\_HERSHEY\_SIMPLEX # 使用默认字体  
  
'''载入图片'''  
Img = cv2.imread('testBigImg\_new.png')  
h = Img.shape[0] # 图片的尺寸  
w = Img.shape[1]  
  
'''检测编码点，并在图片上显示检测结果'''  
corners, ids, rejectedImgPoints = aruco.detectMarkers(Img, dict1)  
c = np.array(corners) # 检测出来的marker  
id1 = np.array(ids).flatten() # 对应marker的编码信息  
ImgDetection = Img  
for i in range(0, c.shape[0]): # 遍历每一个marker  
 for j in range(0, c[i][0].shape[0]): # 遍历marker的4个角点  
 ImgDetection = cv2.circle(Img, (int(c[i][0][j][0]), int(c[i][0][j][1])), 30, (55, 255, 155), 10) # 圈出角点  
 ImgDetection = cv2.putText(ImgDetection, str(id1[i]),  
 (int(c[i][0][0][0]),  
 int(c[i][0][0][1])),  
 font, 3, (255, 0, 0), 10) # 写上id信息  
plt.imshow(ImgDetection, 'gray')  
plt.show()  
cv2.imwrite('BigImg\_detect.png', ImgDetection)