



软件工程IV课程实验指导书

软件工程IV课题组编著

上海海洋大学海洋智能信息实验教学示范中心

实验一 设计基于语句/判定覆盖及等价类划分的测试用例

一、实验目的

1. 巩固所学的语句覆盖和判定覆盖测试方法。
2. 提高运用语句覆盖和判定覆盖测试方法的能力。
3. 巩固所学的等价类划分的测试方法。
4. 提高运用等价类划分的测试方法能力。

二、实验环境

1. Windows 操作系统
2. Eclipse 应用程序

三、实验内容

- 1.选择一段程序语言，熟悉该代码的逻辑结构与基础知识。**
- 2. 根据代码画出程序流程图，并分析各判定节点。**
- 3. 根据代码流程图分析出判定条件与真假取值。**

四、实验步骤

1.测试分析

- 1.1 根据代码描述，分析主流程、分支流程以及正常流程、异常流程。
- 1.2 模拟不同流程场景：触发主流程、分支分支流程以及正常流程、异常流程。

1.3 数据项检查：数据项的计算规则，数据项后台判定逻辑。

2. 测试设计

根据代码，设计程序流程图，并对程序流程图做节点标记，分析判定。

3. 测试用例设计

根据业务场景与流程逻辑判定，运用语句覆盖法进行用例设计。

语句覆盖是一个比较弱的逻辑覆盖标准，通过选择足够多的测试用例，使得被测程序中的每个语句至少被执行一次，如表 1-1 所示。

表 1-1 语句覆盖测试用例设计		
测试用例名称	测试用例描述	测试路径
CASE1	投保成功：年龄 20，男性，健康体，有医疗保险	ABC
CASE2	投保成功；年龄 20，男性，非健康体且没有医疗保险	ABE

接下来我们运用判定覆盖法来进行用例设计。判定覆盖又称为分支覆盖，判定覆盖语句覆盖的标准稍微强一些，它是指通过设计足够多的测试用例，使得被测试程序中的每个判定都获得一次“真”“假”值，如表 1-2 所示。

表 1-2 判定覆盖测试用例设计		
测试用例名称	测试用例描述	覆盖判定
CASE3	投保成功：年龄 20，男性，健康体，有医疗保险	判定 A="真" 判定 B="真"
CASE4	投保不成功；年龄 15，男性	判定 A="假"
CASE5	投保成功：年龄 20，男性，健康体，没有医疗保险	判定 A="真" 判定 B="假"

4. 测试结果分析

从本实验可看出，语句覆盖实际上是很弱的，CASE1,CASE2 可以满足语句覆盖，但如果第 2 各条件语句中 OR 写成了 AND, CASE1, CASE2 都不能发现它。

“判定覆盖”比“语句覆盖”严格，因为如果每个分支都执行过了，则每个语句也就执行过了。但是，“判定覆盖”还是不够的，例如，CASE3-CASE5 未能检查 AB 分支中女性被保人的承担情况。

五、实验报告要求：

实验报告参考模板如下：



实验报告

题目：设计基于语句/判定覆盖的测试用例

学院：信息学院

专业：

班级：

学号：

姓名：

年 月 日

一、实验目的

1. 巩固所学的语句覆盖和判定覆盖测试方法。
2. 提高运用语句覆盖和判定覆盖测试方法的能力。

二、实验环境

1. Windows 操作系统
2. Eclipse 应用程序

三、实验内容

五、实验结果及分析（遇到的问题与解决）

六、实验体会

实验二 基于 Junit 的单元测试

一、实验目的

1. 通过动手实际操作，巩固所学的单元测试相关知识；
2. 初步了解 Junit 工具的使用方法，加深对单元测试的认识。
3. 熟悉 Eclipse 工具的基本操作；
4. 掌握基本 Eclipse 工具的 JAVA 编程。

二、实验环境

1. Windows 操作系统的客户端。
2. JAVA 环境：Eclipse 应用程序，Junit 工具，EclEmma 工具。

三、实验内容

- 1. 选择一个被测试的 Web 应用系统，能够正常编译部署单元测试对象。**
- 2. 下载并安装 Junit 工具；**
- 3. 在 Junit 单元测试环境下，完成对被测对象的单元测试；**
- 4. 使用 EclEmma 工具，根据单元测试成功与否以及单元测试覆盖率进行分析。**

四、实验步骤

- 1. 确定单元测试方案**

本实验由学生自选 web 应用系统的部分源码作为 java 单元测试的对象，选用 Eclipse 作为 java 开发工具，下载并安装 Junit 和 EclEmma 工具，使用 Junit 进行单元测试，使用 EclEmma 进行覆盖率分析来辅助进行单元测试。

2. JUnit 的下载与安装

JUnit 是一个开源的 Java 测试框架，是单元测试框架体系 xUnit 的一个实例，目前已成为在 Java 单元测试的事实标准。JUnit 软件包可以从网站 <https://www.junit.org> 中下载，实验中使用的版本是 Junit4。

无需解压 Junit 压缩包，选中 Project 项目，在 Eclipse 菜单 Project 的子项弹出的属性窗口 (Properties) 中，首先在左边选择 “Java Build Path”，然后到右上选择 “Libraries” 标签，之后在最右边点击 “Add Library...” 按钮，如下图所示：

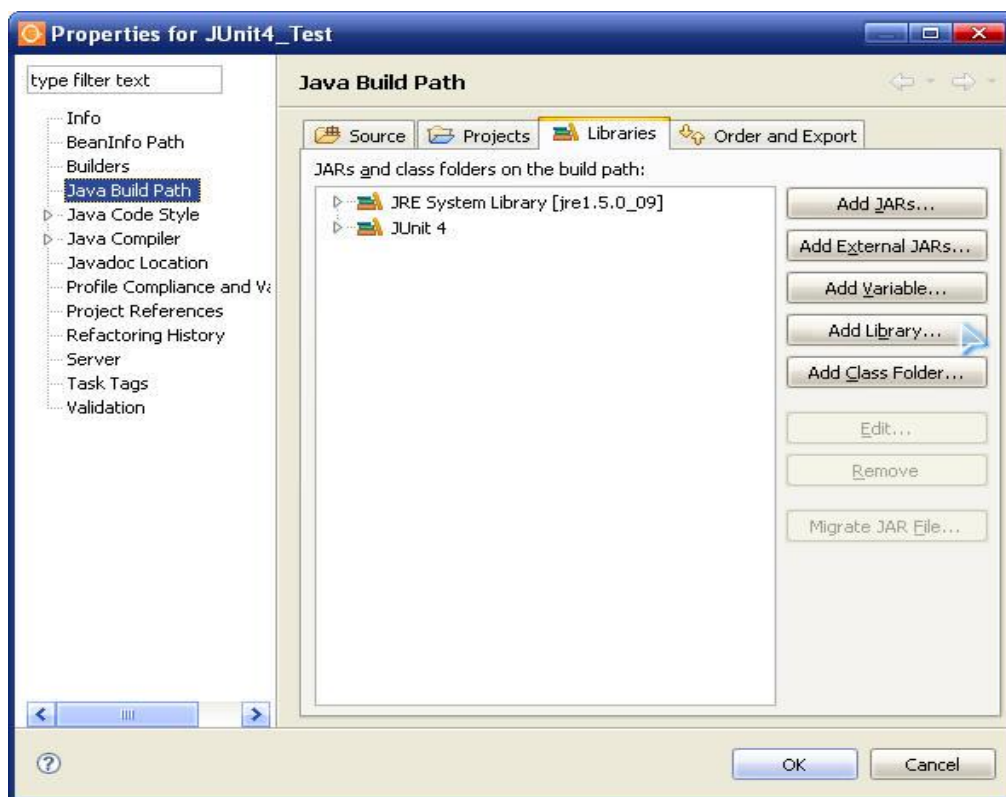


图 2-1 在 Eclipse 中使用 JUnit

2. EclEmma 的下载与安装

EclEmma 是一个开源的覆盖率分析工具，可以帮助大家在单元测试时分析代码覆盖情况。

它可从网站 <http://www.eclEmma.org/download.html> 中下载。

EclEmma 安装步骤：

(1) 启动 eclipse -> 点击 Help 菜单 -> Install New Software，在弹出的对话框中，点击 Add

(2) 输入 Name，例如 EclEmma。输入 Location: <http://update.eclEmma.org/>

(3) 安装后重新启动 eclipse，如果成功安装，工具栏上会出现一个新的按钮，如图 2-2 所示。



图 2-2 EclEmma 安装完成

3. 使用 JUnit 进行单元测试

在配置好 JUnit 工具后，就可以对 Java 类进行单元测试，具体步骤如下：

(1) 选择被 JUnit 测试的类。

(2) 第二步，将 JUnit4 单元测试包引入这个项目：在该项目上点右键，点“属性”，如图 2-3 所示。

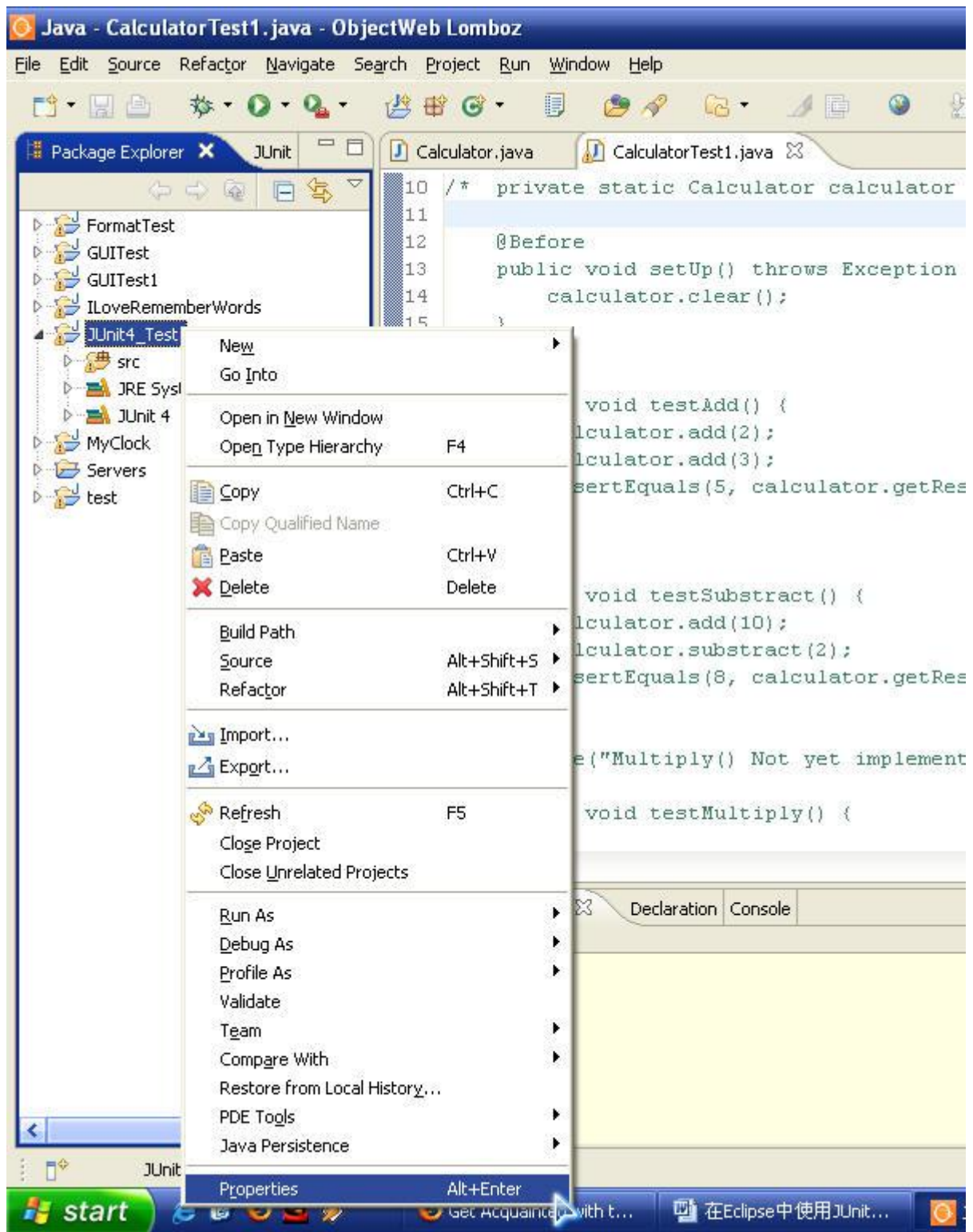


图 2-3 在 Eclipse 中使用 JUnit 4

(3) 第三步，生成 JUnit 测试框架：在 Eclipse 的 Package Explorer 中用右键点击该类弹出菜单，选择 “ New à JUnit Test Case ” 。如下图 2-4 所示：

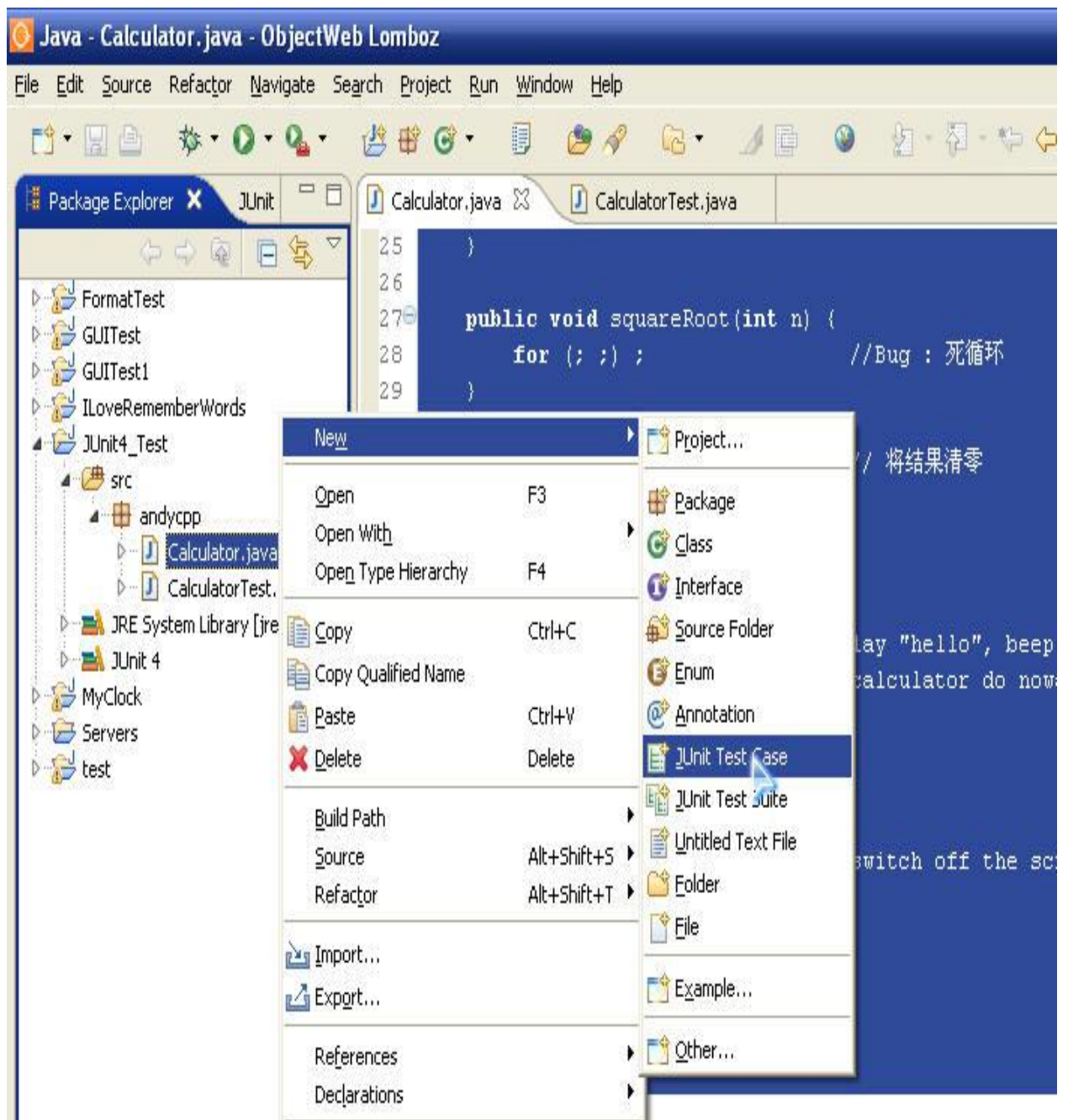


图 2-4 新建 JUnit Test Case

(4) 在弹出的对话框中，进行相应的选择，如下图 2-5 所示：

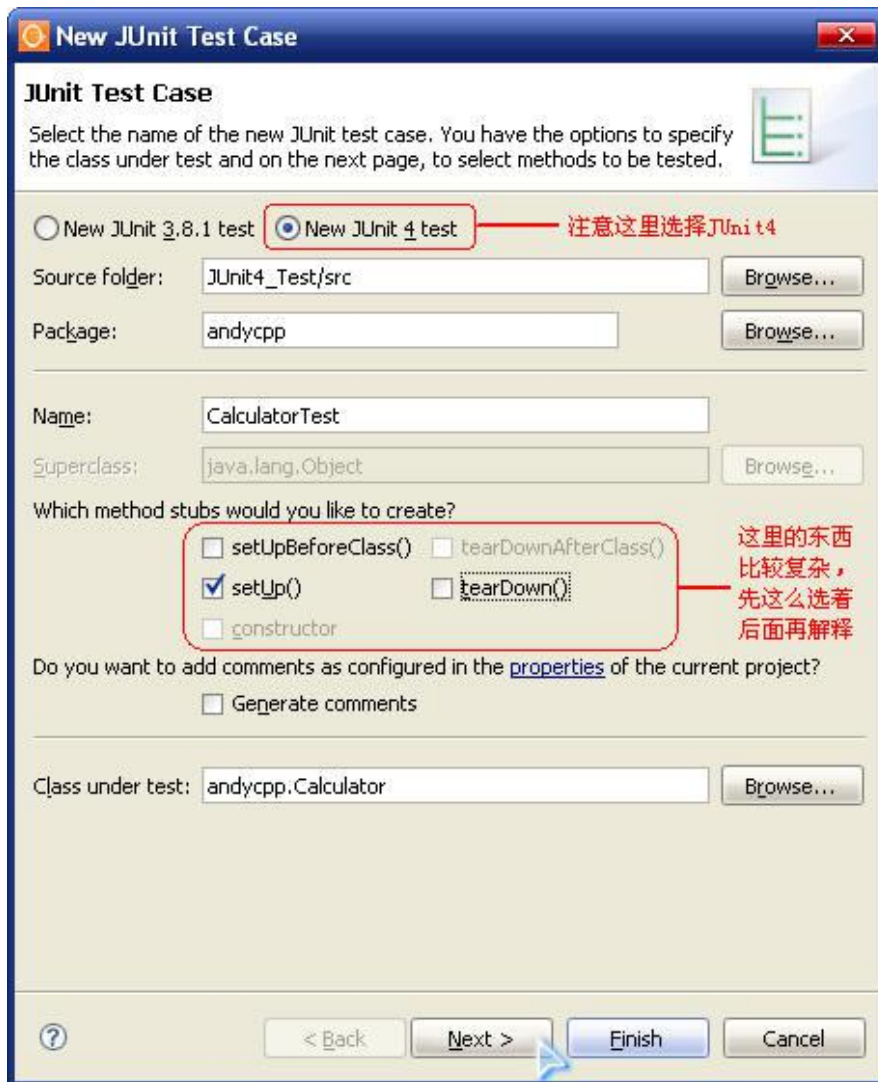


图 2-5 单元测试类信息填写

(5) 点击“下一步”后，系统会自动列出你这个类中包含的方法，选择你要进行测试的方法。此例中，我们仅对“加、减、乘、除”四个方法进行测试。如下图 2-6 所示：

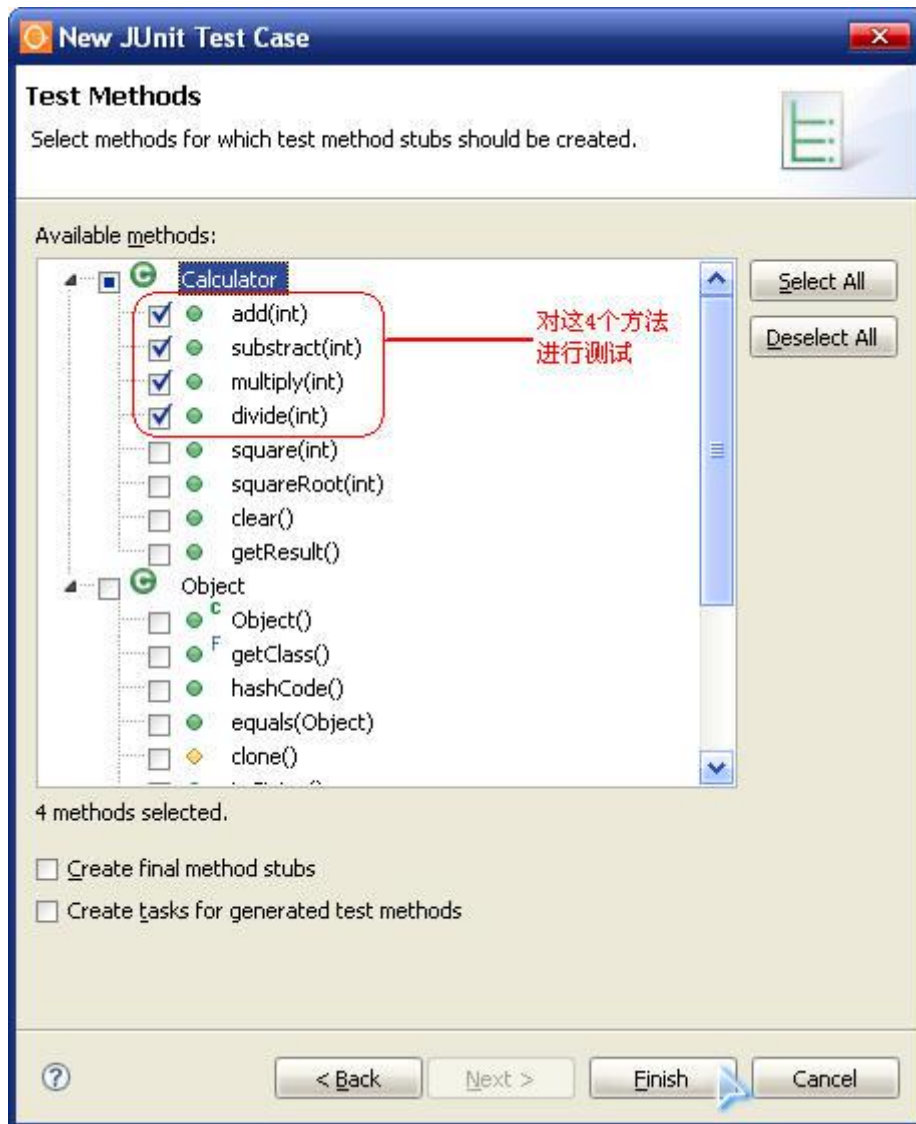


图 2-6 选择被测类的方法

之后系统会自动生成一个新类 CalculatorTest，里面包含一些空的测试用例。

(6) 运行测试代码：按照上述代码修改完毕后，我们在 CalculatorTest 类上点右键，选择

“Run as a JUnit Test” 来运行我们的测试，如下图 2-7 所示：

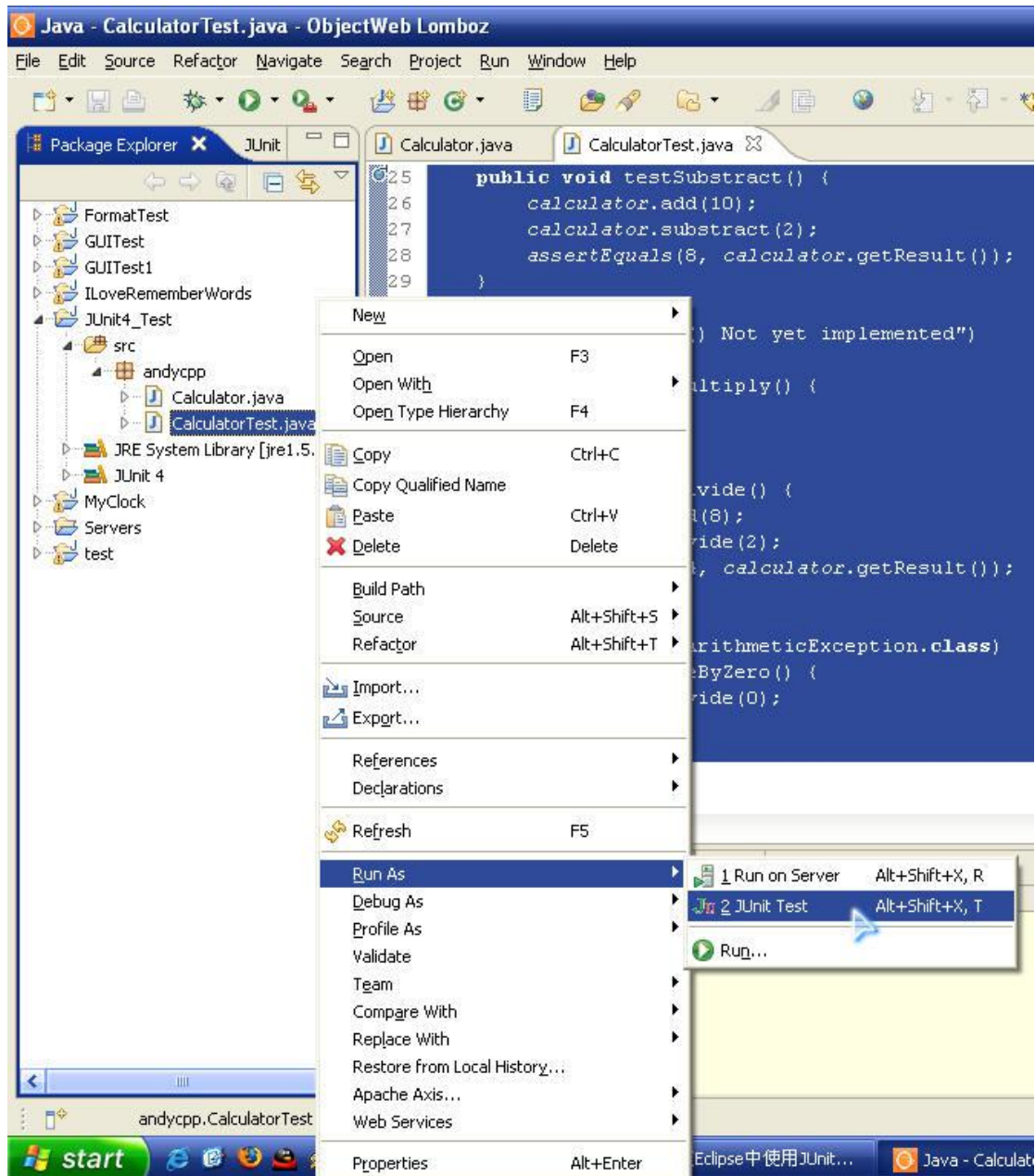


图 2-7 JUnit Test Case 测试运行

4. 使用 EclEmma 查看测试覆盖率

Eclipse 中已安装 EclEmma，在单元测试的基础上进行覆盖率的实验。

- (1) 项目中使用 TestNG 测试框架，包含被测类'Calculator'和测试类'CalculatorTest'。

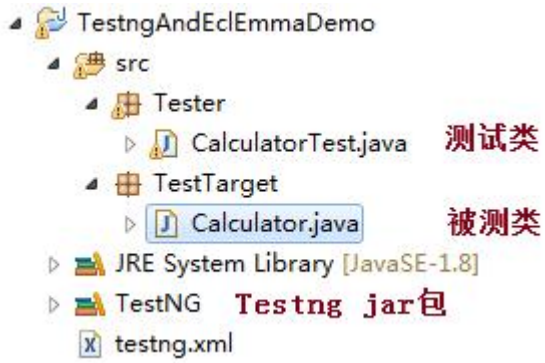


图 2-8 测试结构

(2) 用 Coverage as TestNG Suite 运行测试用例

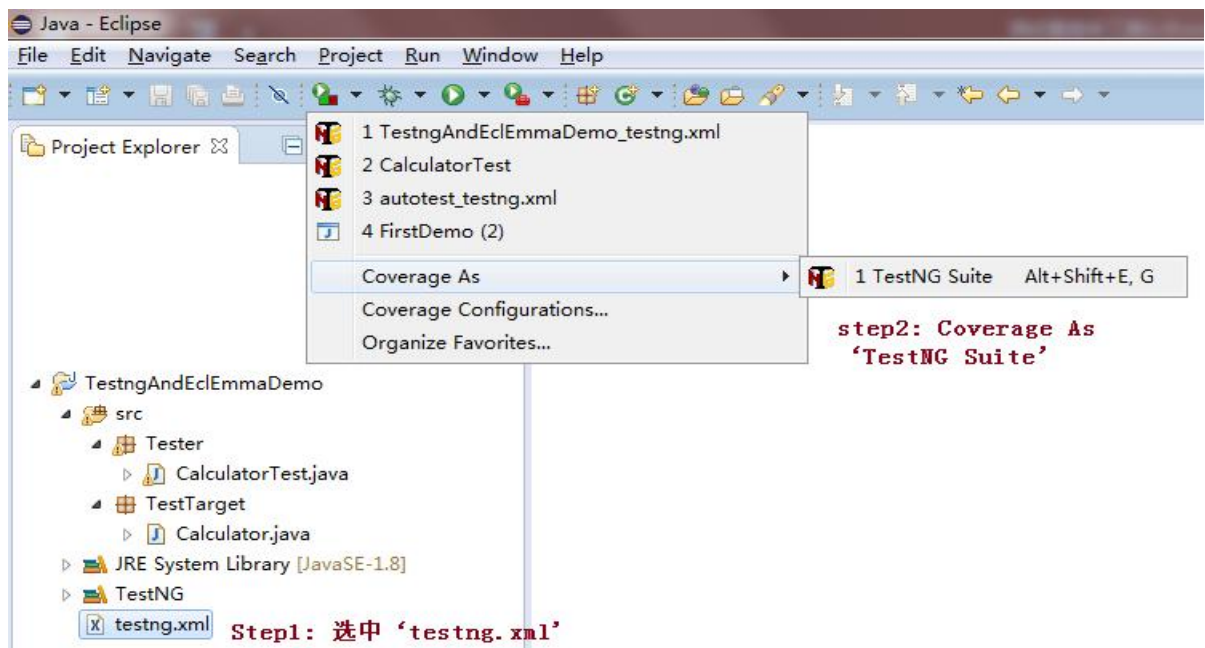


图 2-9 Coverage as TestNG Suite

(3) 运行结果，如图 2-10 所示。

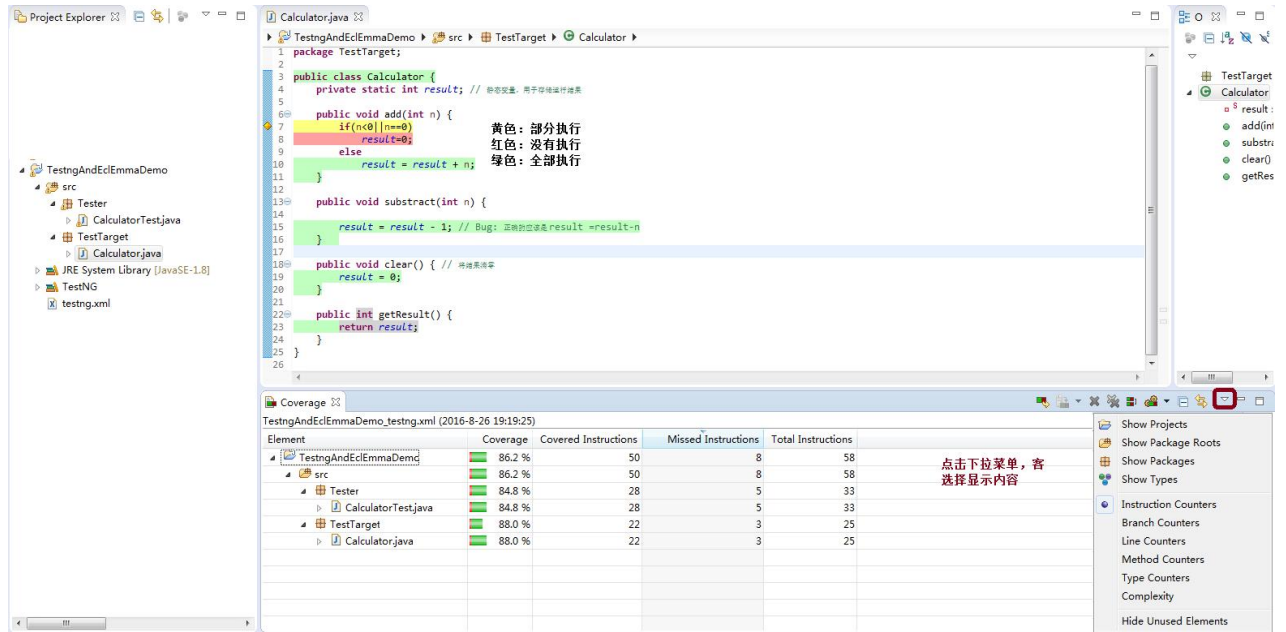


图 2-10 标注源代码和 Coverage 视图

5. 实验结果分析

本实验主要是通过添加补充方法或类的单元测试代码进行单元测试和覆盖率检查, 分析单元测试是否正常进行以及被测代码中的条件分支是否被全部覆盖。本实验的重难点在于对 JUnit、EclEmma 工具的使用, 以及对单元测试的正确理解。

五、实验报告要求

同实验一

实验三 Web 应用的功能测试

一、实验目的

1. 通过动手实际操作，巩固所学的系统功能测试用例设计；
2. 掌握 Web 功能自动化测试方法，包括脚本编写、持续集成测试环境搭建。
3. 熟悉 Selenium IDE 工具的基本知识；

二、实验环境

1. Windows 操作系统的客户端。
2. Firefox 浏览器：Selenium IDE 工具。

三、实验内容

1. 选择一个被测试的 Web 应用系统，能够正常编译部署系统功能测试对象。
2. 下载并安装 Selenium IDE 工具；
3. 在 Selenium IDE 环境下，完成对被测对象的系统功能测试；

四、实验步骤

1. Selenium IDE 的下载与安装

在安装 Selenium IDE 之前，需要先安装 Firefox。Selenium IDE 是 Firefox 浏览器的一个插件，依附于 Firefox 浏览器，并允许您记录，编辑和调试测试，类似于 QTP 的自动录制功能，Selenium IDE 包括整个 Selenium Core，可以快速地在实际运行的环境中记录和回放

测试。

安装方法：

(1) 打开 firefox 浏览器，进入官方网址：<http://seleniumhq.org/download/>

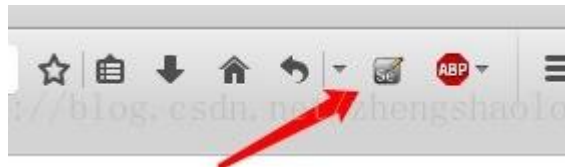
点击 selenium IDE 的下载链接，firefox 会有提示安装插件，点击安装即可。

(2) 打开 firefox 浏览器，进入火狐附加组件网址：

<https://addons.mozilla.org/zh-CN/firefox/>

然后搜索 selenium IDE ，选择点击下载安装重启即可。

安装完成后，在火狐浏览器右上角会出现一个 IDE 的小图标



点击打开就可以使用 IDE 了

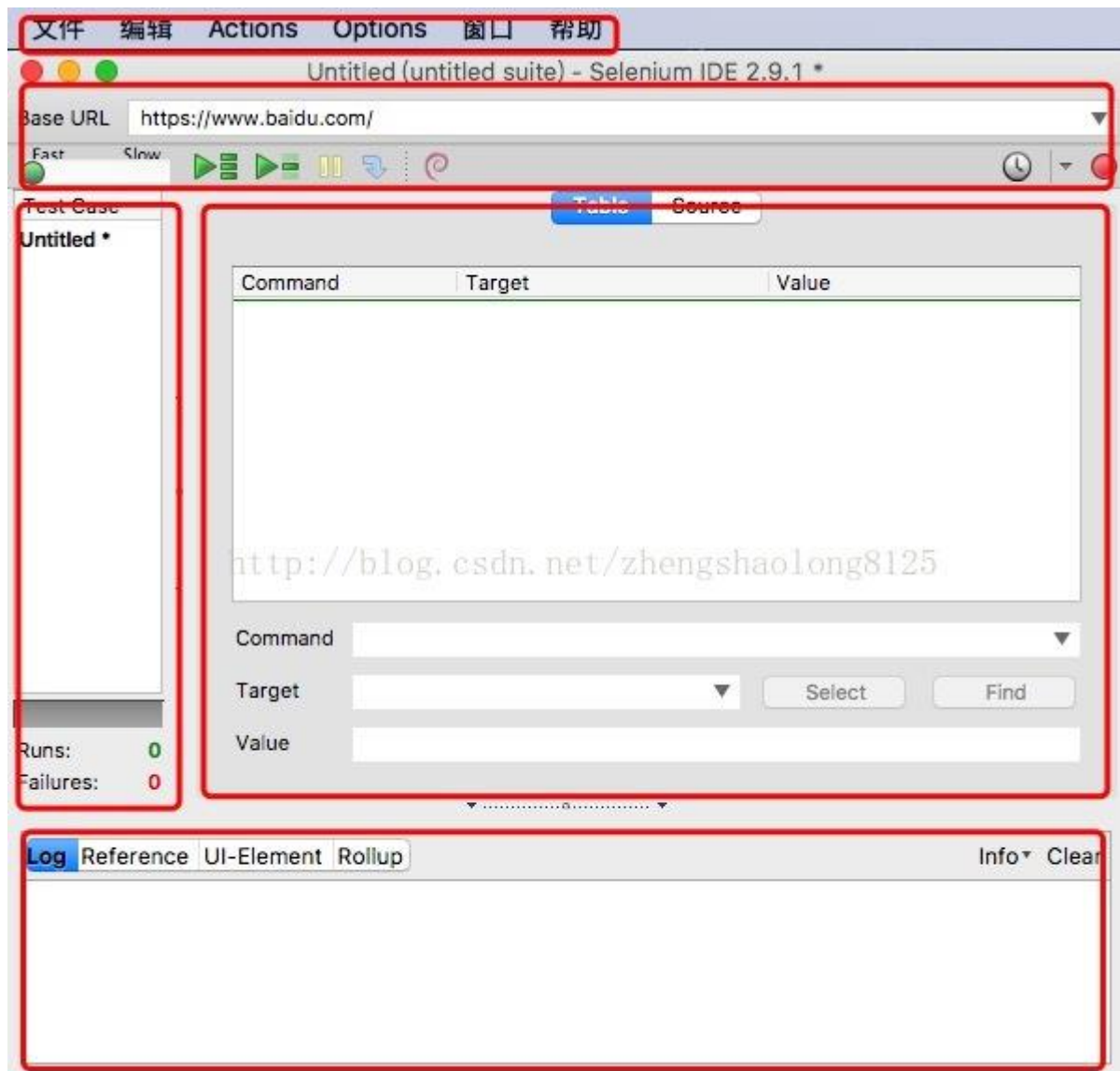


图 3-1 Selenium IDE 操作界面

3. 使用 Selenium IDE 进行操作录制

IDE 窗口主要分 5 大块：

一、菜单栏：

文件：创建、打开和保存测试案例和测试案例集

编辑：复制、粘贴、删除、撤销和选择测试案例中的所有命令

Actions：对 Actions 进行操作

Options：用于设置 seleniumm IDE

帮助：IDE 的使用帮助

二、Actions 操作快捷键：

Base URL：填写被测网站的地址

速度控制：控制 case 的运行速度。

运行所有：运行一个测试 suite 的 case。

运行当前 case：运行当前选定的 case。

暂停/恢复：暂停和恢复 case 执行。

单步：运行一个 case 中的一行命令。

应用汇总规则：此高级功能允许将一连串的 Selenium 命令组合为单个操作。有关汇总规则的详细文档，请参见“帮助”菜单上的“UI 元素文档”。

录制：点击之后，开始记录你对浏览器的操作。

三、case 列表和脚本运行通过/失败的个数

四、脚本编辑区

测试脚本展示：table 标签，用表格形式展现命令及参数。source 标签，用原始方式展现，默认是 HTML 语言格式，也可以用其他语言展示。

命令参数编辑：选择脚本中的一行，可以选择编辑命令、key 值、value 值

五、日志/参考/UI 元素/Rollup

日志：当你运行测试时，错误和信息将会自定显示

4. 实验结果分析

在系统测试中，常见的系统测试任务来自于系统功能、性能和安全性的需求。本实验主要是通过 Selenium IDE 进行系统功能测试的录制操作。本实验的重难点在于使用自动化测试工

具、开发自动化脚本来完成相应的测试任务。

五、实验报告要求：

同实验一

实验四 Web 应用的性能测试

一、实验目的

1. 通过动手实际操作，巩固所学的系统性能测试用例设计；
2. 熟悉系统性能测试过程和工具使用的基本知识；
3. 掌握 Web 性能自动化测试方法，包括脚本编写、持续集成测试环境搭建。
4. 熟悉 JMeter 工具的基本知识；

二、实验环境

1. Windows 操作系统的客户端。
2. Java 运行环境：JMeter 工具。

三、实验内容

1. 选择一个被测试的 Web 应用系统，能够正常编译部署系统性能测试对象。
2. 下载并安装 JMeter 工具；
3. 在 JAVA 环境下，使用 JMeter 完成对被测对象的系统性能测试；

四、实验步骤

1. 测试方案

针对 Web 服务器的性能测试，可以直接通过发送 HTTP 数据包来施加负载，根据所学的知识，选定关键业务来进行负载模拟，完成不同的负载、负载模式的性能测试，获得主要的性能指标数据，包括系统相应时间、数据吞吐量等。

Web 性能测试工具有很多，以 JMeter、Gatling、nGrider 等为代表，本实验选择大家熟悉的、开源的 JMeter 作为本次实验的性能测试工具。

2. JMeter 的下载与安装

Apache JMeter 是一款纯 java 编写负载功能测试和性能测试开源工具软件。相比 Loadrunner 而言，JMeter 小巧轻便且免费，逐渐成为了主流的性能测试工具，是每个测试人员都必须要掌握的工具之一。从 Apache 官网

(http://jmeter.apache.org/download_jmeter.cgi) 下载 JMeter，然后直接解压到相应的目录下，就基本完成其安装。下载完成后解压 zip 包，启动 JMeter，双击 JMeter 解压路径 (apache-jmeter-3.3\bin) bin 下面的 jmeter.bat 即可，如图 4-1 所示：

电脑 > 工作 (D:) > SW > apache-jmeter-3.3 > bin >

名称	修改日期	类型	大小
templates	2017/9/17 19:10	文件夹	
ApacheJMeter.jar	2017/9/17 19:26	Executable Jar File	14 KB
BeanShellAssertion.bshrc	2017/9/17 19:34	BSHRC 文件	2 KB
BeanShellFunction.bshrc	2017/9/17 19:34	BSHRC 文件	3 KB
BeanShellListeners.bshrc	2017/9/17 19:34	BSHRC 文件	2 KB
BeanShellSampler.bshrc	2017/9/17 19:34	BSHRC 文件	3 KB
hc.parameters	2017/9/17 19:34	PARAMETERS 文...	2 KB
heapdump.cmd	2017/9/17 19:10	Windows 命令脚本	2 KB
heapdump.sh	2017/9/17 19:10	Shell Script	2 KB
jaas.conf	2017/9/17 19:34	CONF 文件	2 KB
jmeter	2017/9/17 19:23	文件	4 KB
jmeter.bat	2017/9/17 19:23	Windows 批处理...	5 KB
jmeter.properties	2017/9/17 19:34	PROPERTIES 文件	52 KB
jmeter.sh	2017/9/17 19:10	Shell Script	3 KB
jmeter-n.cmd	2017/9/17 19:23	Windows 命令脚本	2 KB
jmeter-n-r.cmd	2017/9/17 19:10	Windows 命令脚本	2 KB
jmeter-server	2017/9/17 19:23	文件	2 KB

图 4-1 启动 JMeter

3. JMeter 脚本编写

(1) 右键点击“测试计划” -> “添加” -> “Threads(Users)” -> “线程组”



图 4-2 添加“线程组”

这里可以配置线程组名称, 线程数, 准备时长 (Ramp-Up Period(in seconds)) 循环次数, 调度器等参数, 如图 4-3 所示。

 A screenshot of the '线程组' (Thread Group) configuration dialog box. The dialog is titled '线程组'. It contains several sections:

- 名称:** 线程组
- 注释:** (empty field)
- 在取样器错误后要执行的动作:**
 - 继续
 - Start Next Thread Loop
 - 停止线程
 - 停止测试
 - Stop Test Now
- 线程属性:**
 - 线程数:** 1
 - Ramp-Up Period (in seconds):** 1
 - 循环次数:** 永远 1
 - Delay Thread creation until needed
 - 调度器
- 调度器配置:**
 - 持续时间 (秒):** (empty field)
 - 启动延迟 (秒):** (empty field)
 - 启动时间:** 2017/10/27 11:39:59
 - 结束时间:** 2017/10/27 11:39:59

4-3 设置“线程组”

(2) 添加 HTTP 请求

右键点击“线程组” -> “添加” -> “Sampler” -> “HTTP 请求”, 如图 4-4 所示。



图 4-4 添加“HTTP 请求”

Http 请求主要参数详解：

Web 服务器

协议：向目标服务器发送 HTTP 请求协议，可以是 HTTP 或 HTTPS，默认为 HTTP

服务器名称或 IP ： HTTP 请求发送的目标服务器名称或 IP

端口号：目标服务器的端口号，默认值为 80

Http 请求

方法：发送 HTTP 请求的方法，可用方法包括 GET、POST、HEAD、PUT、OPTIONS、TRACE、DELETE 等。

路径：目标 URL 路径（URL 中去掉服务器地址、端口及参数后剩余部分）

Content encoding ： 编码方式，默认为 ISO-8859-1 编码，这里配置为 utf-8，同请求一起发送参数

在请求中发送的 URL 参数，用户可以将 URL 中所有参数设置在本表中，表中每行为一个参数（对应 URL 中的 name=value），注意参数传入中文时需要勾选“编码”。

(3) 添加查看结果树

右键点击“线程组” -> “添加” -> “监听器” -> “察看结果树”，如图 4-5 所

示。



图 4-5 添加“查看结果树”

这时，我们运行 Http 请求，修改响应数据格式为“HTML Source Formatted”，可以看到本次搜索返回结果页面标题为“jmeter 性能测试_百度搜索”，如图 4-6 所示。

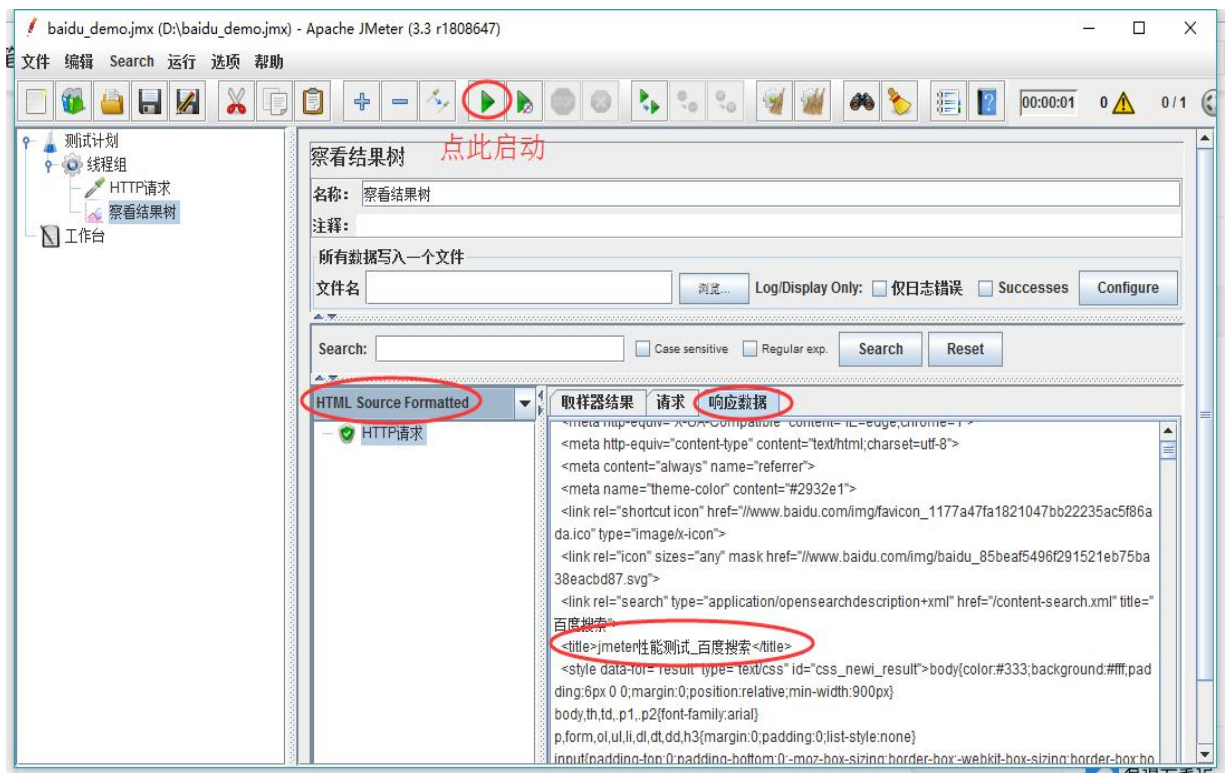


图 4-6 查看结果树

(4) 添加断言

右键点击“HTTP 请求” -> “添加” -> “断言” -> “响应断言”，如图 4-7 所示。



图 4-7 添加“断言”

(5) 添加断言结果

右键点击“HTTP 请求” -> “添加” -> “监听” -> “断言结果”，如图 4-8 所示。

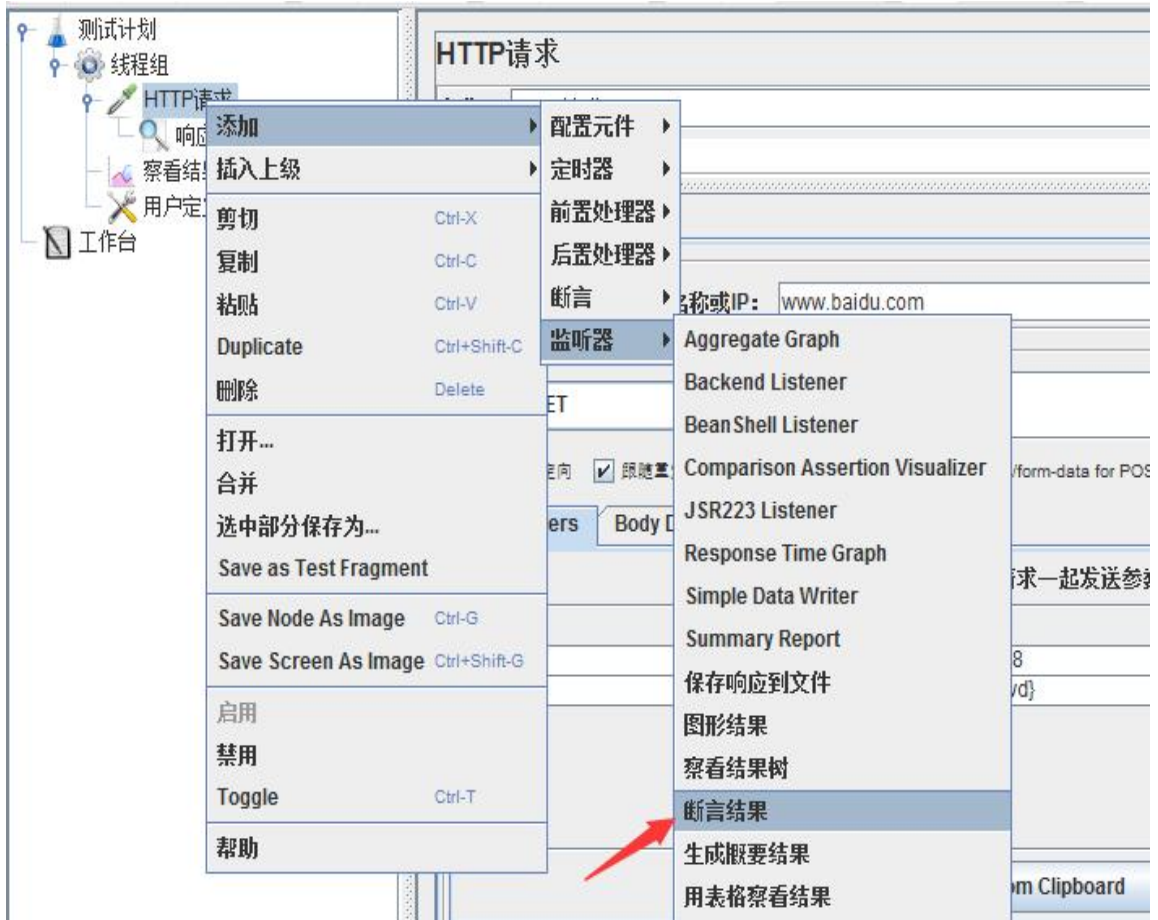


图 4-8 添加“断言结果”

这时,我们再运行一次就可以看到断言结果成功或失败了,我们就完成了一个完整 Http 接口的 JMeter 性能测试脚本编写。

4. 实验结果分析

在系统测试中,常见的系统测试任务来自于系统功能、性能和安全性的需求。本实验主要是通过 JMeter 进行系统性能测试的脚本编写。本实验的重难点在于使用自动化测试工具、开发自动化脚本来完成相应的测试任务。

五、实验报告要求:

同实验一