

# 实验三：Software Unit Testing

## 实验指导书

### 一、实验目的

对 Java 单元测试有初步的了解，学会使用 Junit 编写自己的测试用例，掌握对被测试类进行测试的实现方法；了解轻量级的自动化工具 Selenium，学会使用 Selenium 进行网站自动化测试（选做）。

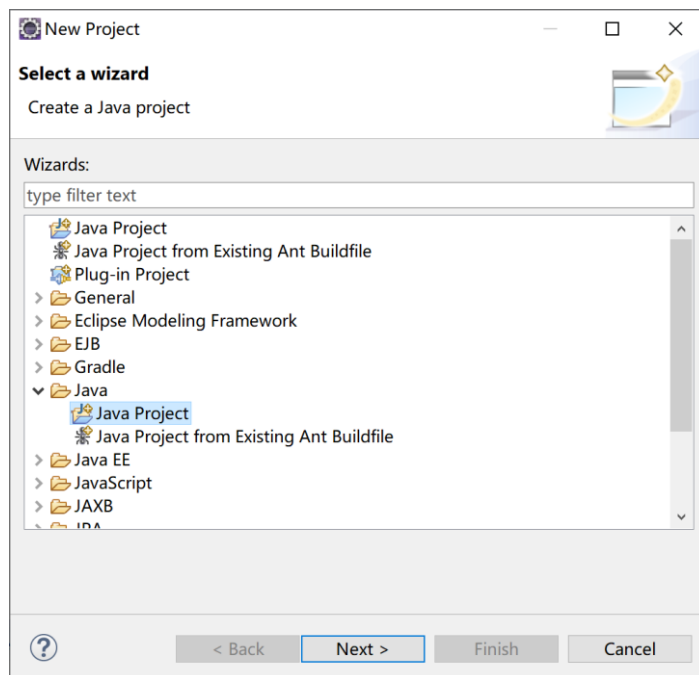
### 二、实验内容

1. 阅读理解 Junit 的工作机制。
2. Eclipse 环境下的单元测试。
3. 利用 Junit 编写测试用例，并对被测试类进行相关功能测试。
  - a) 验证 Book 类的 equals 方法；
  - b) 设计 Triangle 类的测试用例，并进行测试。
4. 利用 Selenium 进行 Web 应用程序的自动化测试。（选做）

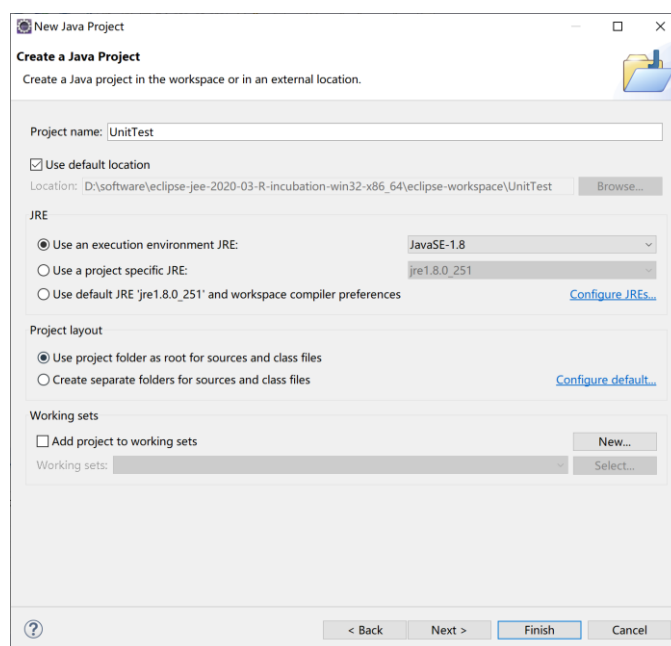
### 三、实验步骤

实验采用 Eclipse IDE for Enterprise Java Developers

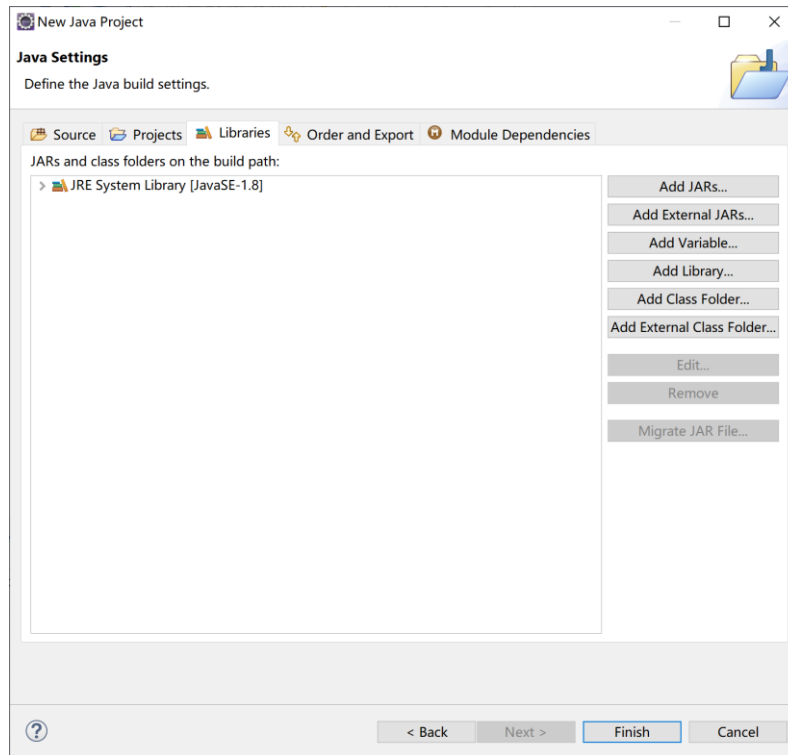
1. 开始—> 程序—> Eclipse IDE for Enterprise Java Developers, 启动 Eclipse 工作环境。
2. File—> New—> Project 创建一个新的工程, 选择 Java Project, 如图所示, 点击 Next。



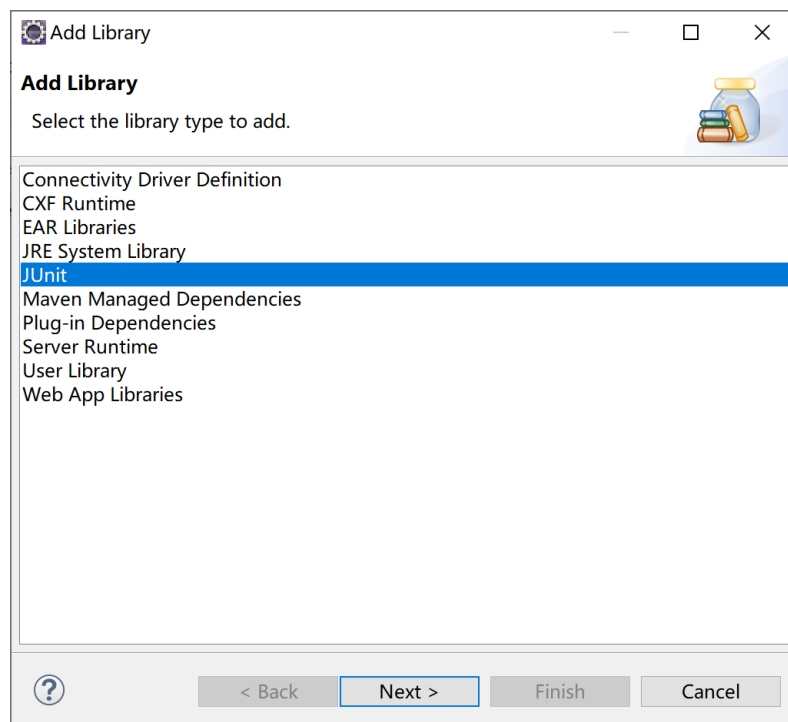
3. 项目名称 (Project name) 输入 “UnitTest”, 点击 Next。



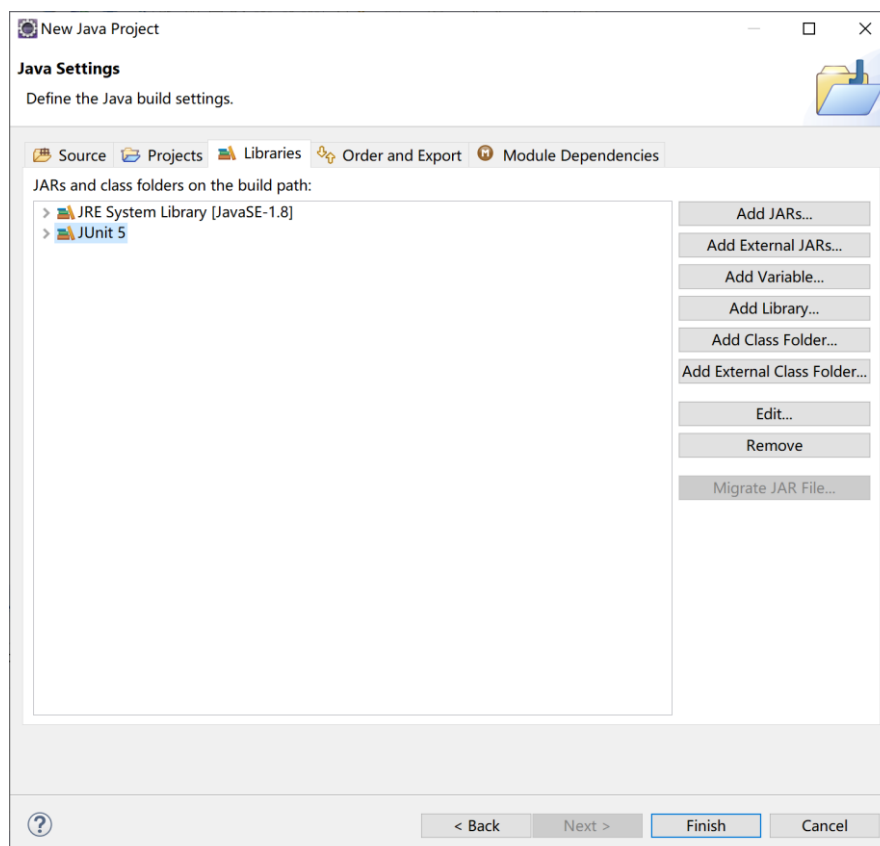
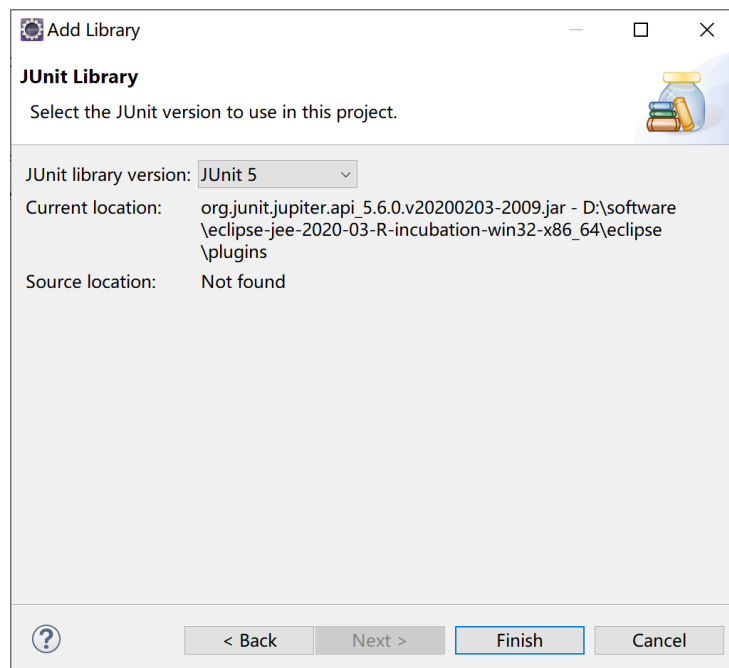
4. 选中 Libraries 标签, 点击 “add Library” 添加必要的 Junit 类库。



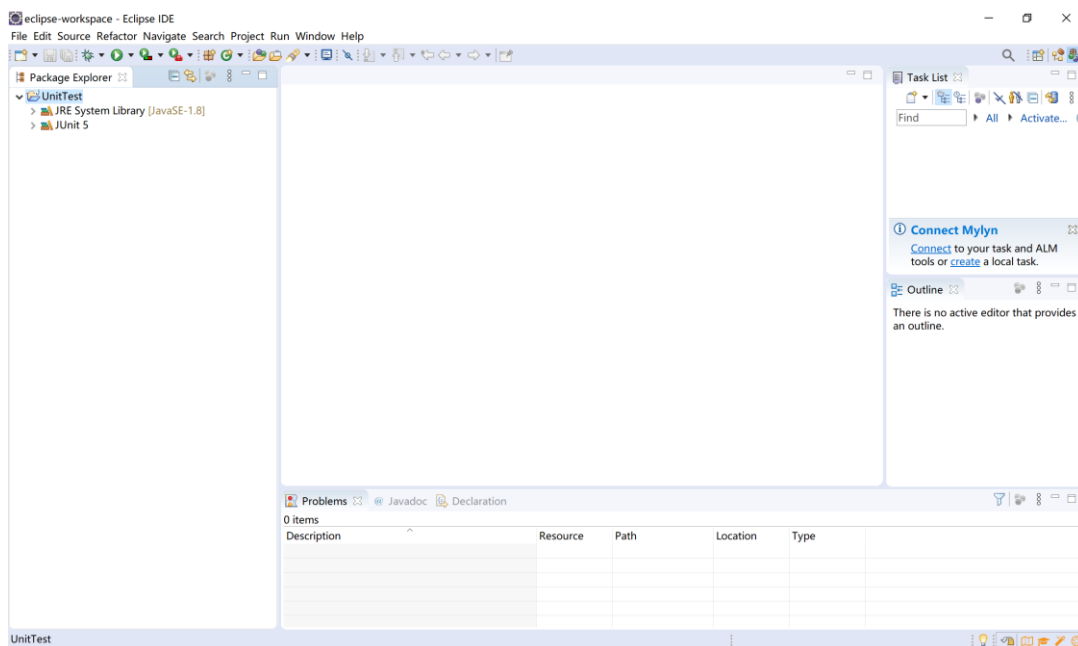
5. 选中 JUnit, 点击 Next;



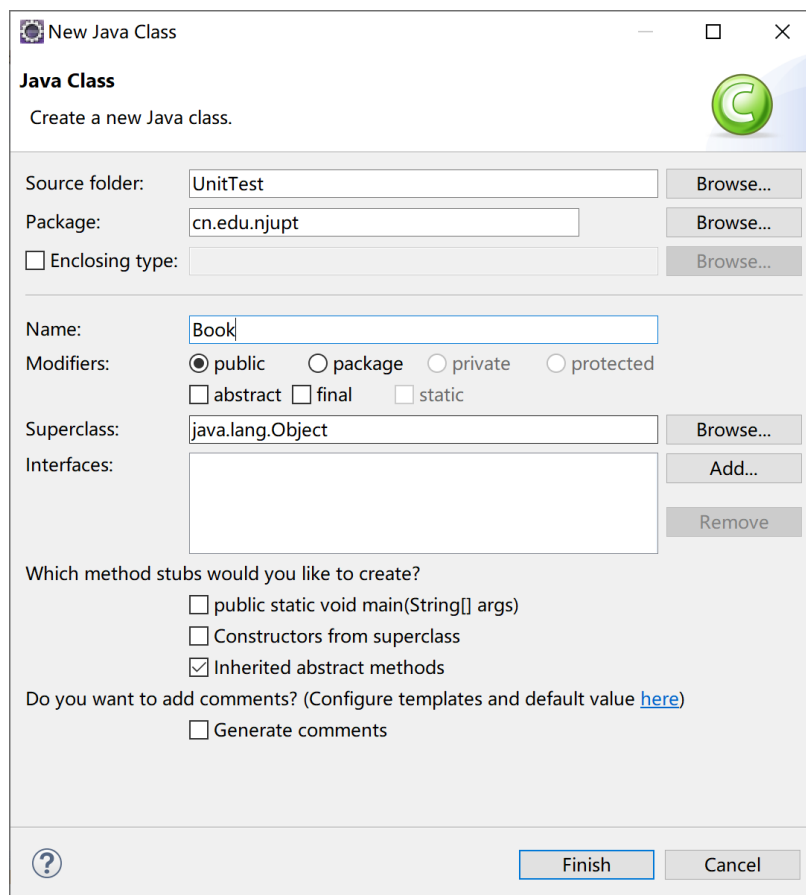
6. 选择默认的 JUnit5，点击 Finish，完成类库的添加，如图所示



7. 点击 **Finish**，完成 **UnitTest** 项目的设置，进入项目工作界面。



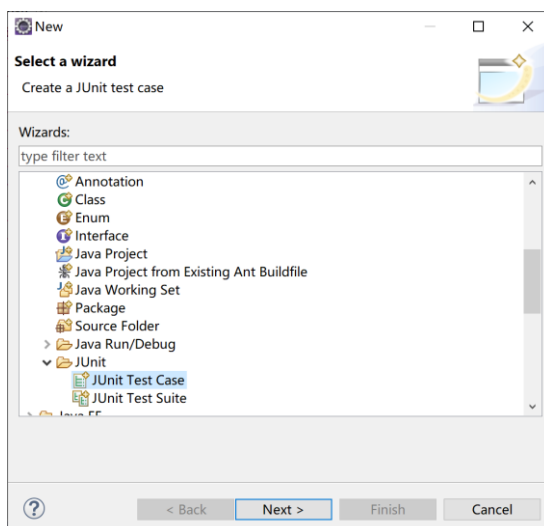
8. 在 **UnitTest** 上右键，——> **New**——> **Class**，弹出添加类窗口。Package 中填写 **cn.edu.njupt**，Name 中填写 **Book**。（注意大小写）



9. 在 Book 类中填充内容如下，完成 Book 类的编写。

```
private String title;
private double price;
/**
 * Constructor
 * @param title
 * @param price
 */
public Book(String title, double price) {
    this.title = title;
    this.price = price;
}
/**
 * Check if an object is an instance of book
 * and the values of title and price are equal
 * then return true, otherwise return false
 */
public boolean equals(Object object) {
    if (object instanceof Book) {
        Book book = (Book) object;
        return getTitle().equals(book.getTitle())
            && getPrice() == book.getPrice();
    }
    return false;
}
public double getPrice() {
    return price;
}
public void setPrice(double price) {
    this.price = price;
}
public String getTitle() {
    return title;
}
public void setTitle(String title) {
    this.title = title;
}
```

10. 右键点击 UnitTest 项目，——> New——> Other，选择 Java 下的 JUnit 下的 JUnit Test Case，点击 Next;



11. 在 Package 中填写 “cn.edu.njupt”，在 Name 中填写 “BookTest”，在 Class under test 中填写 “cn.edu.njupt.Book”（注意大小写），选中 setUp(), tearDown()复选框，点击 Next；在弹出的窗口中选中 equals 成员函数，所完成的 BookTest 类如图所示。

**New JUnit Test Case**

Select the name of the new JUnit test case. You have the options to specify

☐ New JUnit 3 test ☐ New JUnit 4 test ☒ New JUnit Jupiter test

Source folder:  

Package:  

Name:

Superclass:  

Which method stubs would you like to create?

☐ setUpBeforeClass() ☐ tearDownAfterClass()  
☒ setUp() ☒ tearDown()  
☐ constructor

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

Class under test:

**New JUnit Test Case**

**Test Methods**

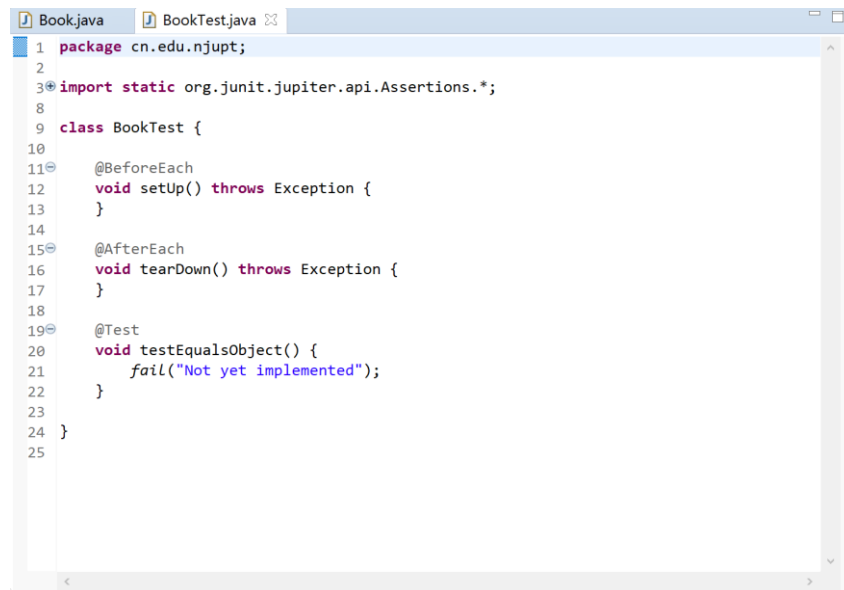
Select methods for which test method stubs should be created.

Available methods:

- ☒ **Book**
  - ☐ Book(String, double)
  - ☒ equals(Object)
  - ☐ getPrice()
  - ☐ setPrice(double)
  - ☐ getTitle()
  - ☐ setTitle(String)
- ☐ **Object**
  - ☐ Object()
  - ☐ getClass()
  - ☐ hashCode()
  - ☐ equals(Object)
  - ☐ clone()
  - ☐ toString()

1 method selected.

☐ Create final method stubs  
☐ Create tasks for generated test methods



12. 为 BookTest 添加两个私有成员变量

```
private Book book1;
private Book book2;
```

13. 分别完成 setUp（）和 tearDown（）两个成员函数

```
protected void setUp() throws Exception {
    book1 = new Book("ES", 12.99);
    book2 = new Book("The Gate", 11.99);
}

protected void tearDown() throws Exception {
    book1 = null;
    book2 = null;
}
```

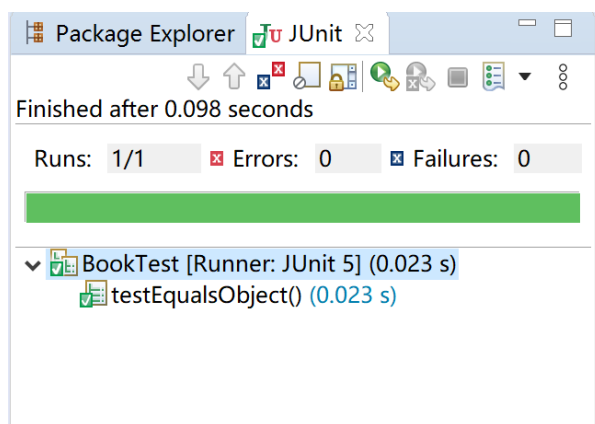
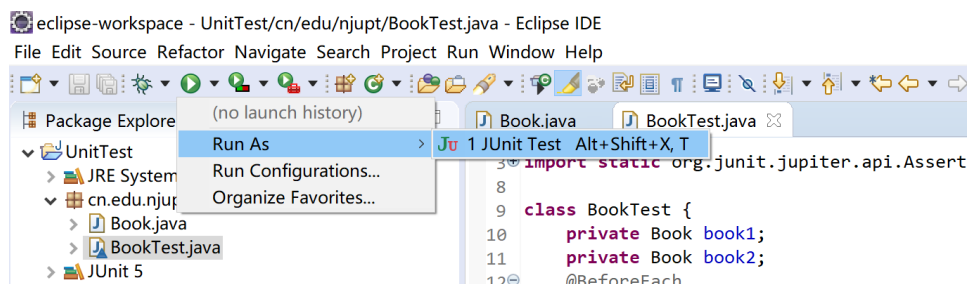
14. 编写测试用例函数 testEqualsObject（），使用下列两条语句替换

```
fail("Not yet implemented");
```

```
assertFalse(book2.equals(book1));
assertTrue(book1.equals(book1));
```



15. 点击 Run As JUnit Test，观察测试运行结果。



补充任务：为 Triangle 类编写测试用例。

```
/**
 * This is Triangle class
 */
public class Triangle {
    // 定义三角形的三边
    protected long lborderA = 0;
    protected long lborderB = 0;
    protected long lborderC = 0;

    // 构造函数
    public Triangle(long lborderA, long lborderB, long lborderC) {
        this.lborderA = lborderA;
        this.lborderB = lborderB;
        this.lborderC = lborderC;
    }

    /**
     * 判断是否是三角形
     * 是返回 true; 不是返回 false
     */
    public static boolean isTriangle(Triangle triangle) {
        boolean isTriangle = false;

        // 判断边界，大于 0 小于或等于 Long.MAX_VALUE，出界返回 false
        if ((triangle.lborderA > 0 && triangle.lborderA <= Long.MAX_VALUE / 2)
            && (triangle.lborderB > 0 && triangle.lborderB <= Long.MAX_VALUE / 2)
            && (triangle.lborderC > 0 && triangle.lborderC <= Long.MAX_VALUE / 2)) {

            // 判断两边之和大于第三边
            if ((triangle.lborderA < (triangle.lborderB + triangle.lborderC))
                && (triangle.lborderB < (triangle.lborderA + triangle.lborderC))
                && (triangle.lborderC < (triangle.lborderA + triangle.lborderB))) {
                isTriangle = true;
            }
        }
        return isTriangle;
    }

    /**
     * 判断三角形类型
     *
     * 只有两条边相等的三角形返回字符串“等腰三角形”;
     * 三边均相等的三角形返回字符串“等边三角形”;
```

```
* 三边均不等的三角形返回字符串“不等边三角形”；
*/
public static String getType(Triangle triangle) {
    String strType = "不是三角形 ";

    // 判断是否是三角形
    if (isTriangle(triangle)) {
        // 判断是否是等边三角形
        if (triangle.lborderA == triangle.lborderB
            && triangle.lborderB == triangle.lborderC) {
            strType = "等边三角形";
        }
        // 判断是否是不等边三角形
        else if ((triangle.lborderA != triangle.lborderB)
            && (triangle.lborderB != triangle.lborderC)
            && (triangle.lborderA != triangle.lborderC)) {
            strType = "不等边三角形";
        }
        // 三角形既非三边全部相等，又非全部不等，只能是部分相等即等腰三角形
        else {
            strType = "等腰三角形";
        }
    }

    return strType;
}
}
```

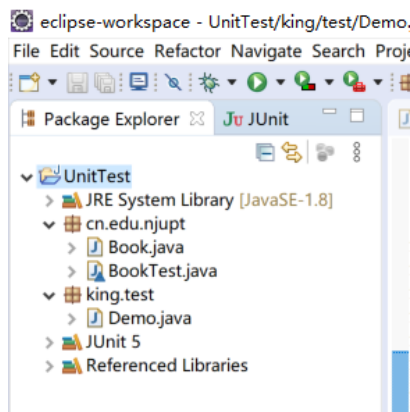
## 四、参考资料

Junit 官方网站:

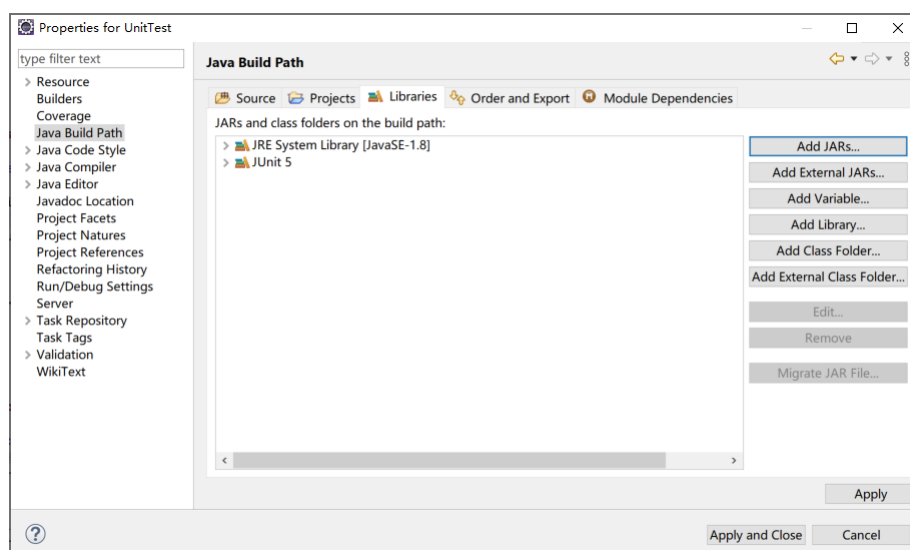
<http://www.junit.org>

## （选做）利用 Selenium 进行 Web 应用程序的自动化测试

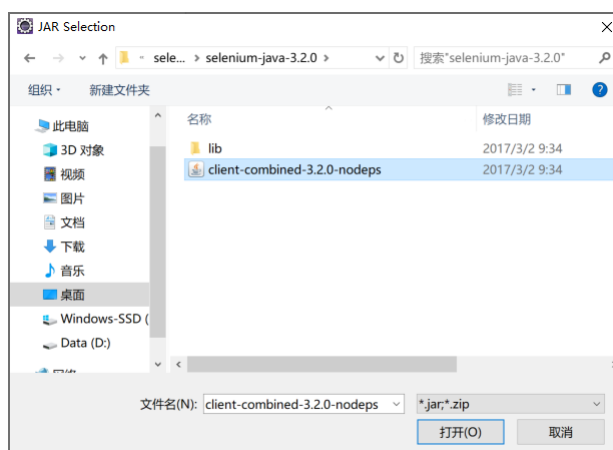
1. 在新建项目 UnitTest 下，新建一个包 king.test，包下新建一个类 Demo。然后，把 Selenium 的相关 jar 包引进来，把下载的 Selenium 的 zip 包解压。



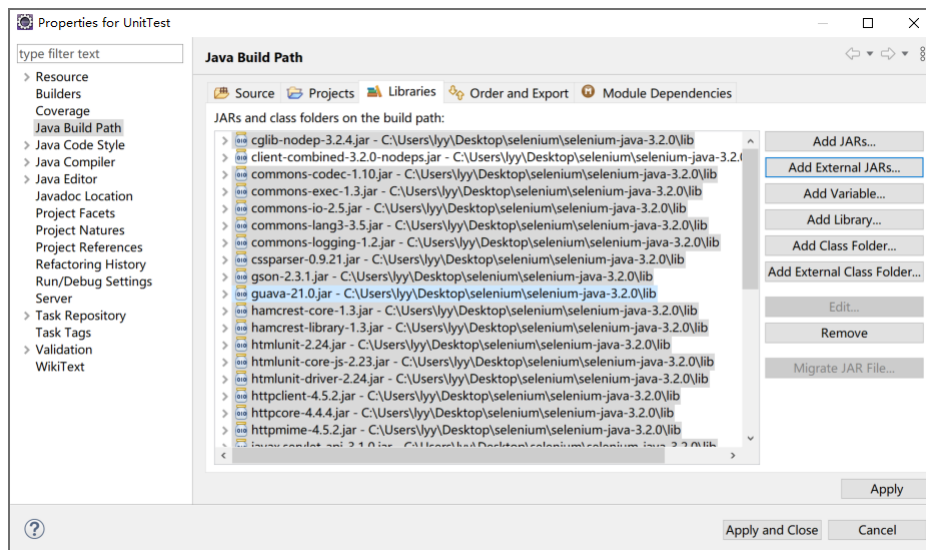
2. 右击 UnitTest，点击最后一项 Properties，左边点 Java Build Path，右边点 Add External JARs...。然后文件路径中选择到 Selenium 的包解压的路径。



3. 选择 client 开头的 jar 文件，点确定



4. 然后再点 Add External JARs..., 这次, 进入 lib, 把 lib 里的 jar 文件全部引入项目。进入 lib, 按 Ctrl+A 全选, 再点确定, 就全部引进来了



5. 点 Apply and Close, jar 包就引用好了。然后, 就可以在 Demo 里编写代码。还需要先下载谷歌浏览器的驱动, 使用 selenium 启动谷歌 Chrome 浏览器的时候, 是需要用到 chromedriver 的, 两者之间的版本需要匹配。Demo 的相关代码如下:

```
package king.test;

import java.util.List;
import java.util.concurrent.TimeUnit;

import org.junit.After;
import org.junit.Before;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

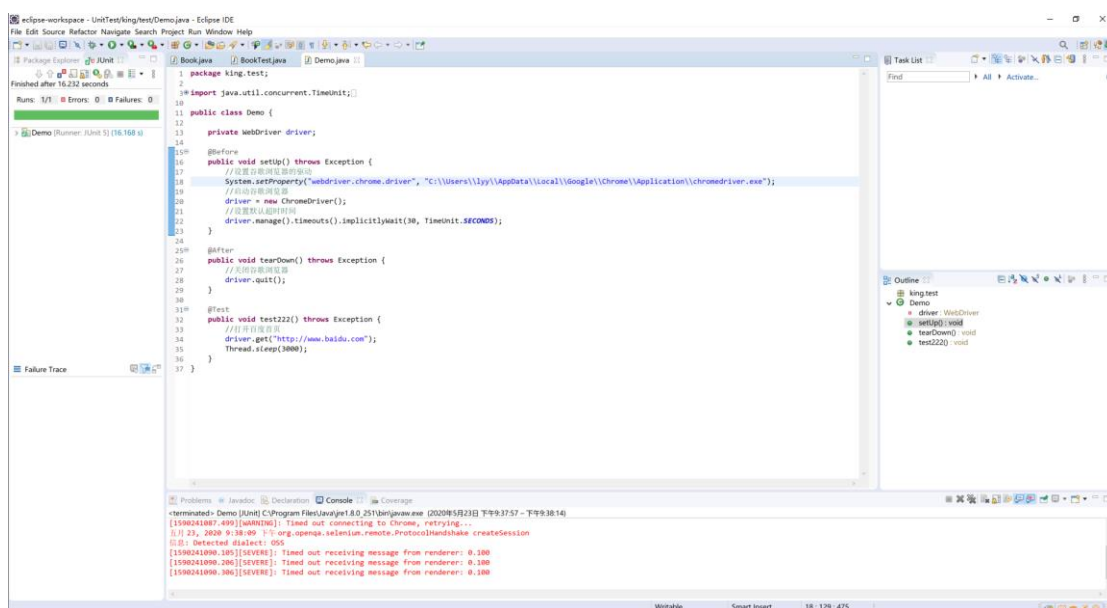
public class Demo {

    private WebDriver driver;

    @Before
```

```
public void setUp() throws Exception {  
    //设置谷歌浏览器的驱动  
    System.setProperty("webdriver.chrome.driver",  
"C:\\Users\\lly\\AppData\\Local\\Google\\Chrome\\Application\\chromedriver.exe");  
    //启动谷歌浏览器  
    driver = new ChromeDriver();  
    //设置默认超时时间  
    driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);  
}  
  
@After  
public void tearDown() throws Exception {  
    //关闭谷歌浏览器  
    driver.quit();  
}  
  
@Test  
public void test222() throws Exception {  
    //打开百度首页  
    driver.get("http://www.baidu.com");  
    Thread.sleep(3000);  
}  
}
```

6. 设置谷歌浏览器的驱动，后面的路径，换成在你电脑上驱动存放的路径即可，然后，运行脚本，就会启动谷歌浏览器了，然后浏览器就会打开百度首页，停留三秒钟，关闭浏览器。



7. 接下来进行元素的查找及操作，首先，准备一个简单的 html 页面，新建一个记事本，输入以下代码，更改后缀为 html，名称为 test.html。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="Generator" content="EditPlus@">
  <meta name="Author" content="">
```

```
<meta name="Keywords" content="">
<meta name="Description" content="">
<title>Document</title>
</head>
<body>
  <input id="un" class="css1" name="username" />
  <input id="pw" class="css2 lala" name="password" />
  <input id="co" class="css3" name="code" /><br />
  <input type="button" id="btn" class="css4" name="submit" value="按钮" />
  <script>
    document.getElementById('btn').onclick = function(){
      alert("haha");
    }
  </script>
</body>
</html>
```

8. 双击它，就会用浏览器打开，打开以后，把地址栏里的地址复制一下，贴粘到 java 代码里，替换掉原先写的百度首页的网址。

```
30
31 @Test
32 public void test222() throws Exception {
33     //打开百度首页
34     driver.get("file:///C:/Users/lyy/Desktop/test.html");
35     Thread.sleep(3000);
36 }
37 }
```

9. 然后运行脚本，就会使用谷歌浏览器打开测试页面了。那现在，这个页面有三个输入框和一个按钮，我们现在要操作这几个元素。

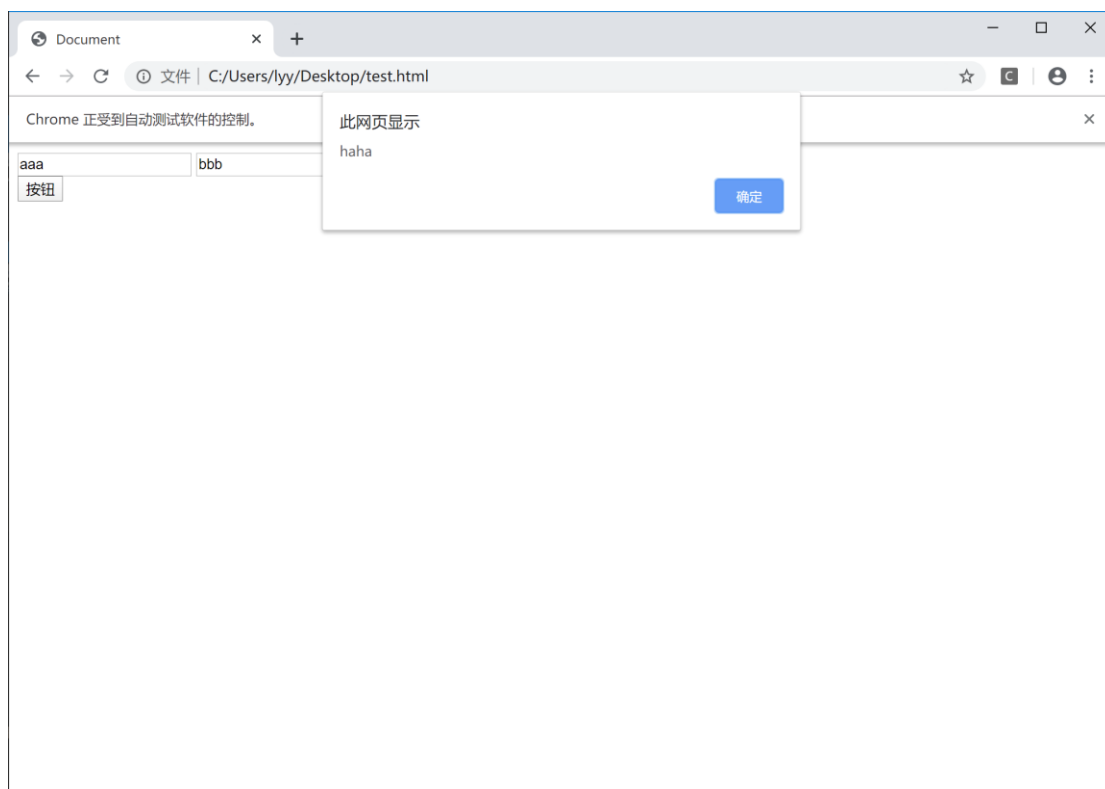
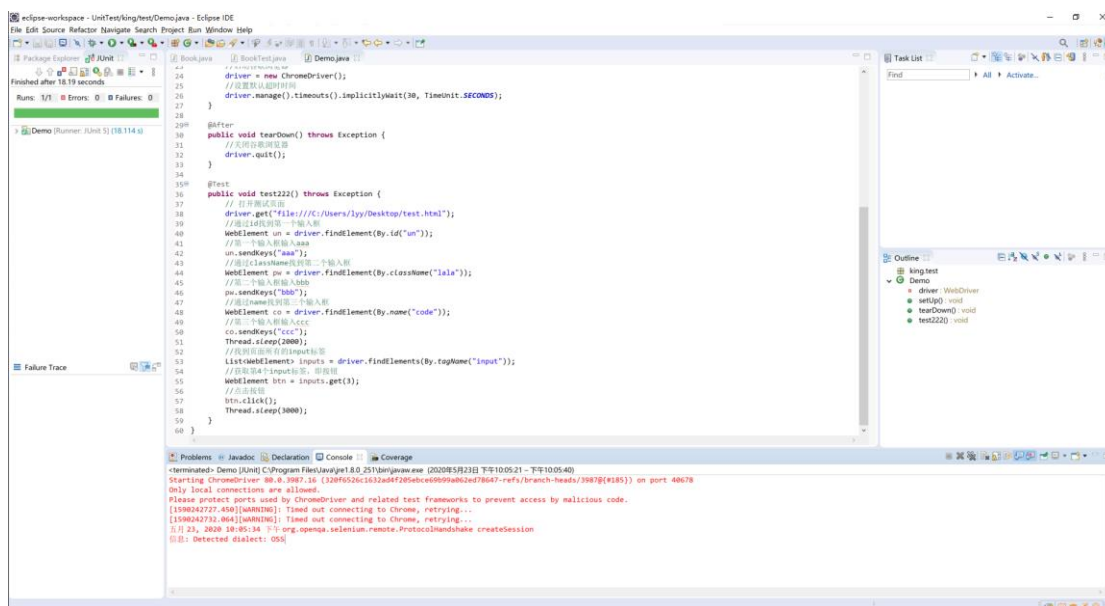


10. 替换 `public void test222() throws Exception` 中的代码为



```
@Test
public void test222() throws Exception {
    // 打开测试页面
    driver.get("file:///C:/Users/Administrator/Desktop/test.html");
    //通过 id 找到第一个输入框
    WebElement un = driver.findElement(By.id("un"));
    //第一个输入框输入 aaa
    un.sendKeys("aaa");
    //通过 className 找到第二个输入框
    WebElement pw = driver.findElement(By.className("lala"));
    //第二个输入框输入 bbb
    pw.sendKeys("bbb");
    //通过 name 找到第三个输入框
    WebElement co = driver.findElement(By.name("code"));
    //第三个输入框输入 ccc
    co.sendKeys("ccc");
    Thread.sleep(2000);
    //找到页面所有的 input 标签
    List<WebElement> inputs = driver.findElements(By.tagName("input"));
    //获取第 4 个 input 标签，即按钮
    WebElement btn = inputs.get(3);
    //点击按钮
    btn.click();
    Thread.sleep(3000);
}
```

11. 得到的结果如下：



## 参考资料

<http://www.selenium.org.cn/>

<http://www.51testing.com>