

天津大学

软件测试技术第二次实验报告



学 院 智能与计算学部
专 业 软件工程专业
年 级 2016 级
姓 名 侯钦瀚

2019 年 3 月 28 日

软件测试技术第二次实验报告

一、需求分析

1. 安装 Selenium、SeleniumIDE 插件
2. 使用 SeleniumIDE 记录并且导出脚本
3. “软件测试名单.xlsx” 包含学生信息，<http://121.193.130.195:8800> 可以在登录后看到个人信息。设计程序检查 excel 中每条信息和网站对应信息是否一致。

二、概要设计

1. 设计主类 TestWeb 进行测试，设计工具类 GetDataFromExcel 帮助主类进行 excel 文件的信息读取。
2. 通过 GetDataFromExcel 读取的 excel 三种信息分别存储在三个 static ArrayList 中，在主类 @Before setUp() 方法中先行进行缓存，避免测试时加载，加快正式测试时的速度。
3. 正式测试时只需要不断读取已经存储的信息，与页面当前信息比较即可。

三、详细设计

1. 工具类设计 GetDataFromExcel

- 1) readExcel() - 负责获得通过文件名获取 Workbook 对象

```
@Contract("null -> null")
```

```
public static Workbook readExcel(String fileName) {
```

```
    Workbook wb = null;
```

```
    if(fileName == null) {
```

```

        return null;
    }

    String extString = fileName.substring(fileName.lastIndexOf("."));
    InputStream is = null;
    try {
        is = new FileInputStream(fileName);
        if(".xls".equals(extString)) {
            return wb = new HSSFWorkbook(is);
        } else if(".xlsx".equals(extString)) {
            return wb = new XSSFWorkbook(is);
        }

    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    return wb;
}

```

2) analyzeExcel() - 主方法，负责通过 workbook 对象读取信息并且存储在 ArrayList 中。

```

public void analyzeExcel(String fileName, ArrayList<String> number,
ArrayList<String> name, ArrayList<String> gitAddress) {
    try {
        Workbook wb = readExcel(fileName);
        Sheet s = wb.getSheetAt(0);

        for(int i = 2; i < s.getPhysicalNumberOfRows(); i++) {
            Row row = s.getRow(i);
            number.add(getCellFormatValue(row.getCell(1)));
            name.add(getCellFormatValue(row.getCell(2)));
            gitAddress.add(getCellFormatValue(row.getCell(3)));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}

```

3) getCellFormatValue() - 辅助类，负责格式化 Excel 信息

```

public static String getCellFormatValue(Cell cell){
    String cellValue = "";
    if(cell!=null){
        //判断 cell 类型
        switch(cell.getCellType()){

```

```

        case NUMERIC:{
//            cellValue = cell.getNumericCellValue().toString();

            Double dou = cell.getNumericCellValue();

            NumberFormat nf = NumberFormat.getInstance();

            nf.setGroupingUsed(false);

            cellValue = nf.format(dou);

            break;

        }

        case STRING:{

            cellValue = cell.getRichStringCellValue().getString();

            break;

        }

        default:

            cellValue = "";

    }

}

return cellValue;

}

```

2. 测试类设计

1) @Before setup() – 负责初始化 webdriver 和三个 ArrayList

```
private static ArrayList<String> number = new ArrayList<>();
```

```
private static ArrayList<String> name = new ArrayList<>();
```

```
private static ArrayList<String> gitAddress = new ArrayList<>();
```

```
@Before
```

```
public void setUp() throws Exception {
```

```
    driver = new FirefoxDriver();
```

```
    baseUrl = "http://121.193.130.195:8800";
```

```
    driver.manage().timeouts().implicitlyWait(4, TimeUnit.HOURS);
```

```
    GetDataFromExcel getter = new GetDataFromExcel();
```

```
    getter.analyzeExcel("/Users/houqinhan/Documents/Class/3rd/
```

```
软件测试/SoftwareTest/src/Lab2/软件测试名单.xlsx", number, name,  
gitAddress);
```

```
}
```

2) testWeb() – 测试主方法，循环判断每条信息

```
@Test
```

```
public void testWeb() throws Exception {
```

```
    driver.get(baseUrl + "/login");
```

```
    for(int i = 0; i < number.size(); i++) {
```

```
        WebElement typeName = driver.findElement(By.name("id"));
```

```
        WebElement typePassword =
```

```
driver.findElement(By.name("password"));
```

```
        typeName.sendKeys(number.get(i));
```

```
        typePassword.sendKeys(number.get(i).substring(4));
```

```

        driver.findElement(By.id("btn_login")).click();

        assertEquals(number.get(i),

driver.findElement(By.id("student-id")).getText());

        assertEquals(name.get(i), driver.findElement(By.id("student-
name")).getText());

        assertEquals(gitAddress.get(i),

driver.findElement(By.id("student-git")).getText());

        driver.findElement(By.id("btn_logout")).click();

        driver.findElement(By.id("btn_return")).click();

    }

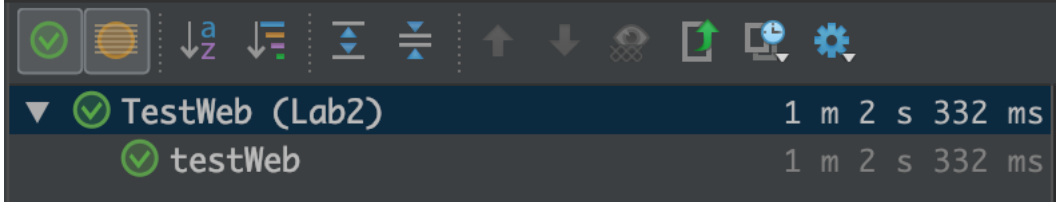
}

```

四、 调试分析

1. fileName 无法通过 reference path 传递，必须通过完整的绝对路径作为参数传递。
2. excel 中存在后接空格的 git 地址，可以通过 format 格式化表格信息去掉所有的空格。

五、 测试结果



Element	Class, %	Method, %	Line, %
Lab2	100% (3/3)	100% (7/7)	84% (56/66)

测试总共花费了 1 分钟左右，覆盖到了所有的方法，某些涉及 .xls 文件处理和

异常处理的语句没有执行到，因此行覆盖率是 84%。

六、 总结

本次实验使用 Java 里中 selenium 的 WebDriver 工具，编写测试程序进行了自动化 web 测试，认识到了自动化测试的效率。本次编写程序涉及到了文件 IO，POI 包的使用以及 Selenium 工具的学习，受益匪浅。