**《软件测试》**

**实验报告五 ——基路径测试方法**

**姓 名： 黄文轩 学 号：2019176203**

**院 系： 计算机与信息学院 专 业： 数据科学与大数据技术**

**实 验 室： 实验日期：2022/4/24**

**总评成绩： 审阅教师：**

目录

[**《软件测试》** 1](#_Toc99813008)

[**实验报告四 ——决策表分析方法** 1](#_Toc99813009)

[一、实验目的： 3](#_Toc99813010)

[二、实验环境： 3](#_Toc99813011)

[三、实验要求： 3](#_Toc99813012)

[四、实验步骤与内容 3](#_Toc99813013)

[五、结论分析与体会 9](#_Toc99813014)

[六、仓库地址 9](#_Toc99813015)

### 一、实验目的：

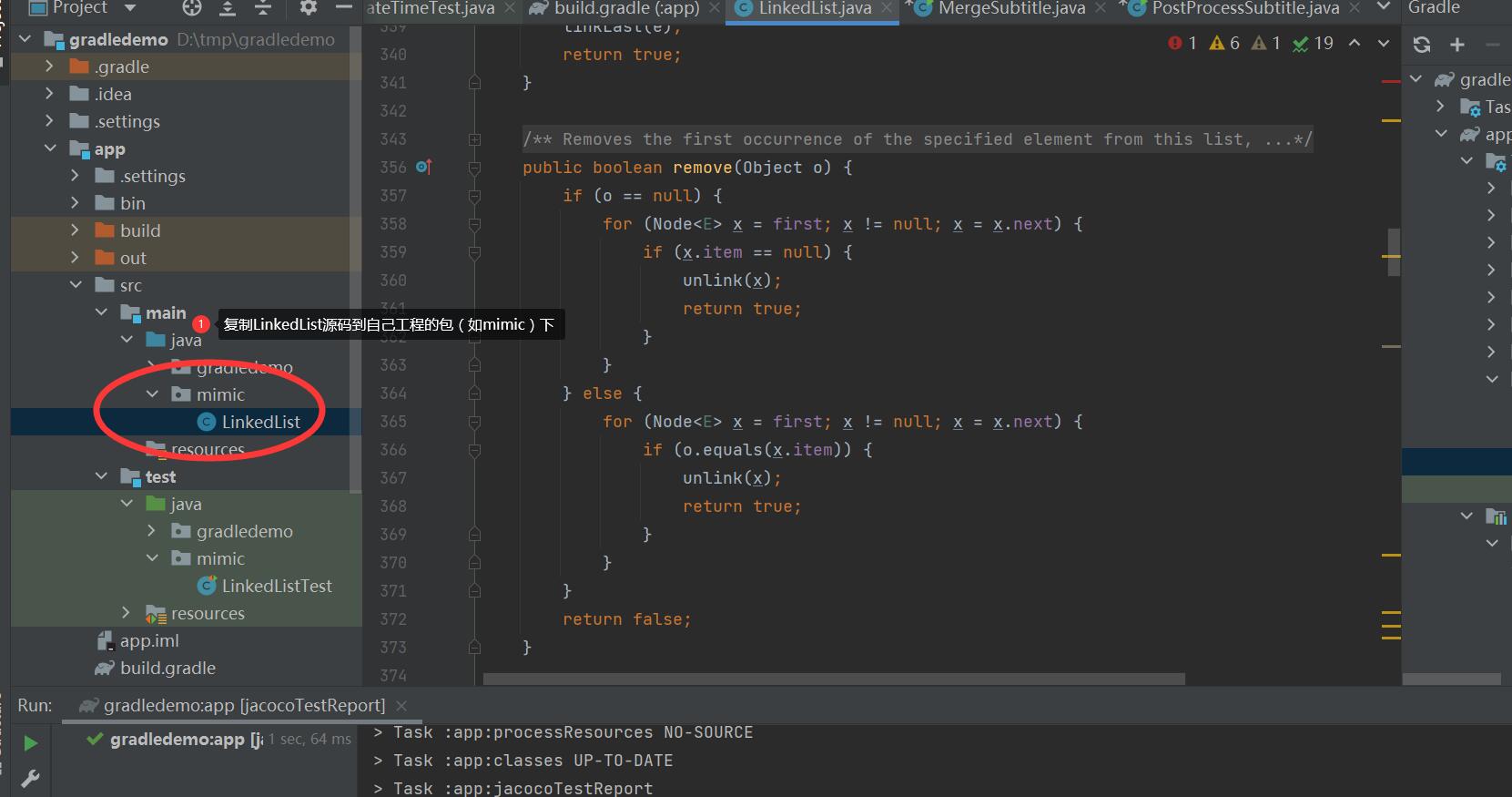
1. 学习逻辑覆盖准则
2. 学习绘制程序图
3. 学习基路径测试
4. 学习测试用例评估方法

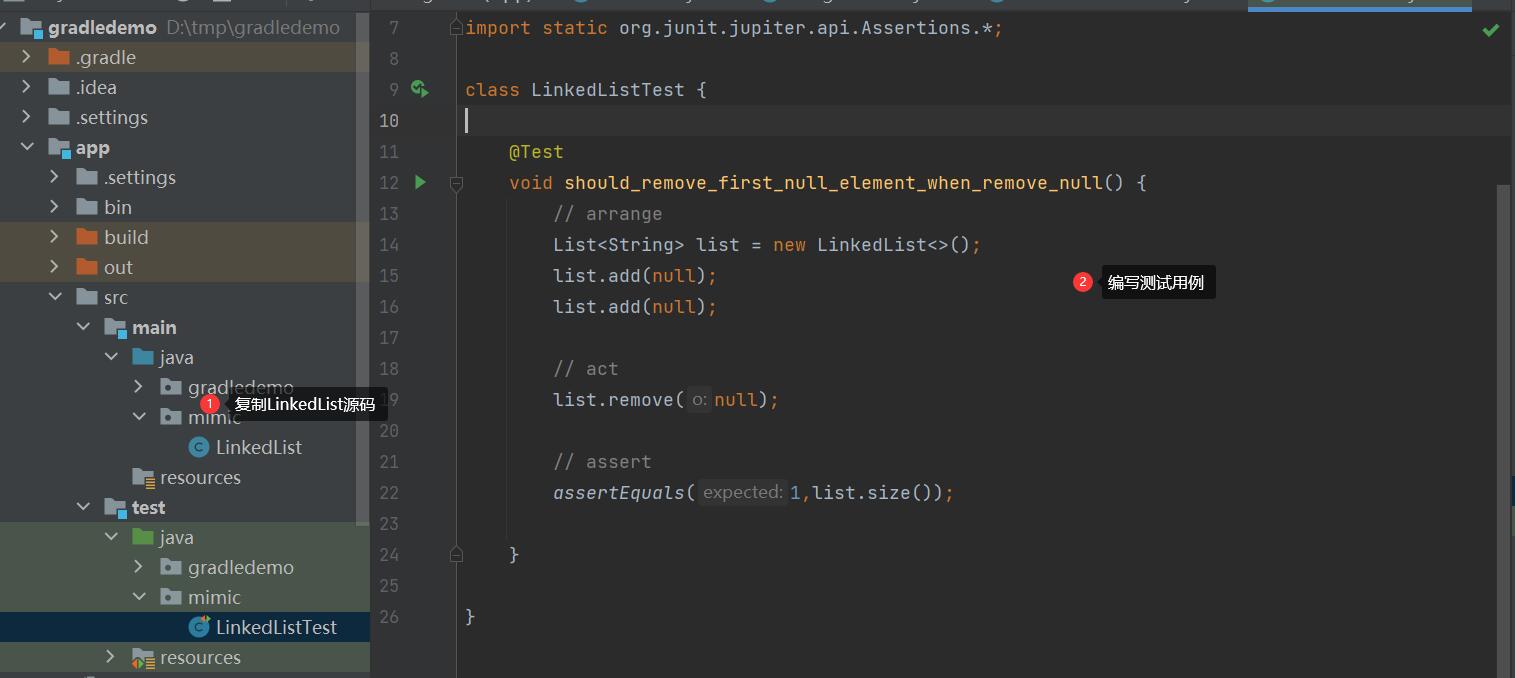
### 二、实验环境：

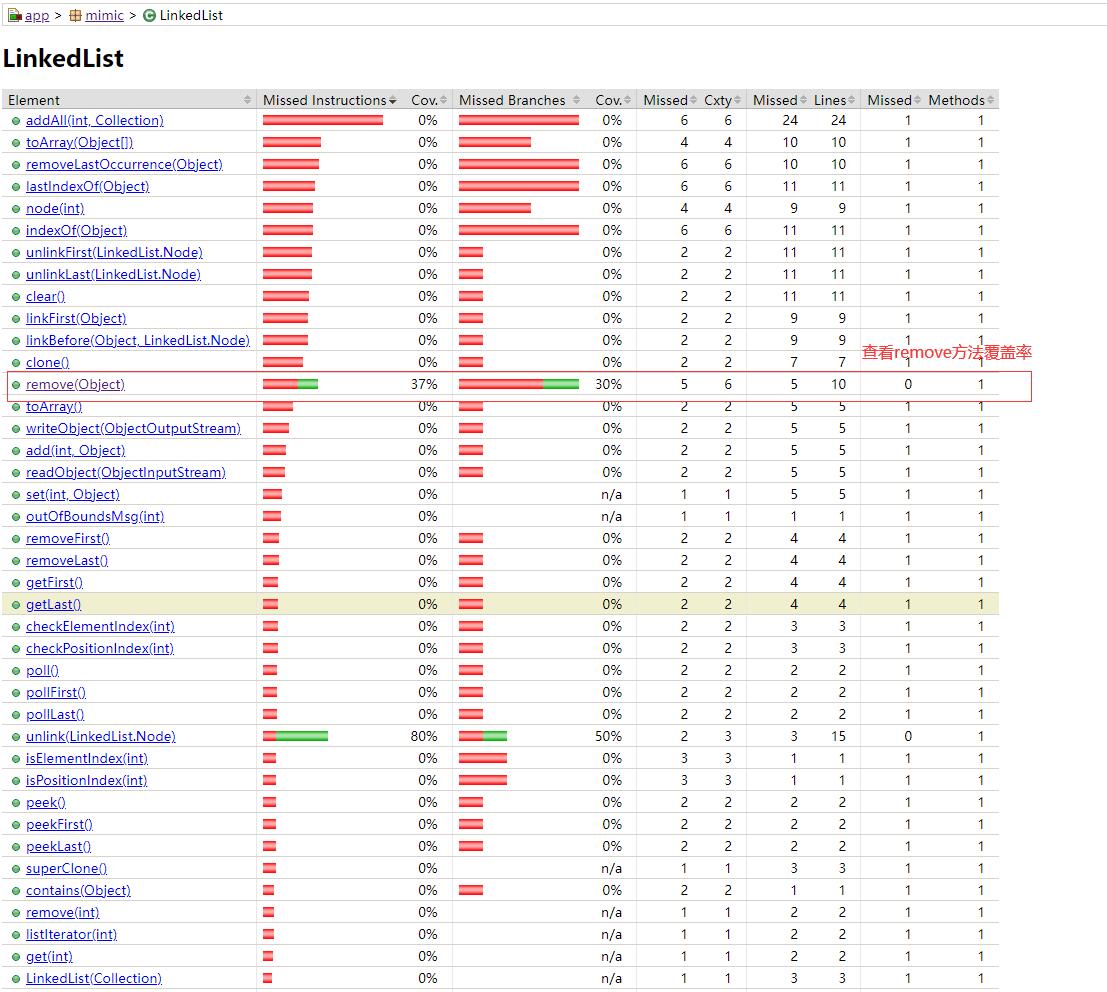
Eclipse2020,JUnit

### 三、实验要求：

1. 针对上面的remove方法，设计尽可能少的测试用例，要求达到 100% 语句覆盖率，参考remove方法的文档说明
2. 注：为了便于Jacoco 计算覆盖率，可以把linkedlist源码复制一份到自己工程中的一个包下（如mimic），然后针对该源码测试。可参考







1. 创建remove方法的程序图；
2. 根据创建的程序图，计算圈复杂度
3. 采用基路径测试方法，构造基路径集
4. 根据基路径方法，构造测试数据
5. 使用Junit5框架实现自动化测试，提交到码云仓库
6. 配置插件Pitest，评估测试用例有效性
7. 针对实验中的经验和教训，编写实验总结
8. 按照实验报告模板 编写实验报告，以“学号-姓名-软件测试实验五”命名，提交到雨课堂“软件测试实验五”

### 四、实验步骤与内容

1. **题基路径测试步骤**

**1.根据程序代码创建程序图**

**2.计算圈复杂度**

**3.获取基路径集**

**4.构造测试数据，覆盖基路径集**

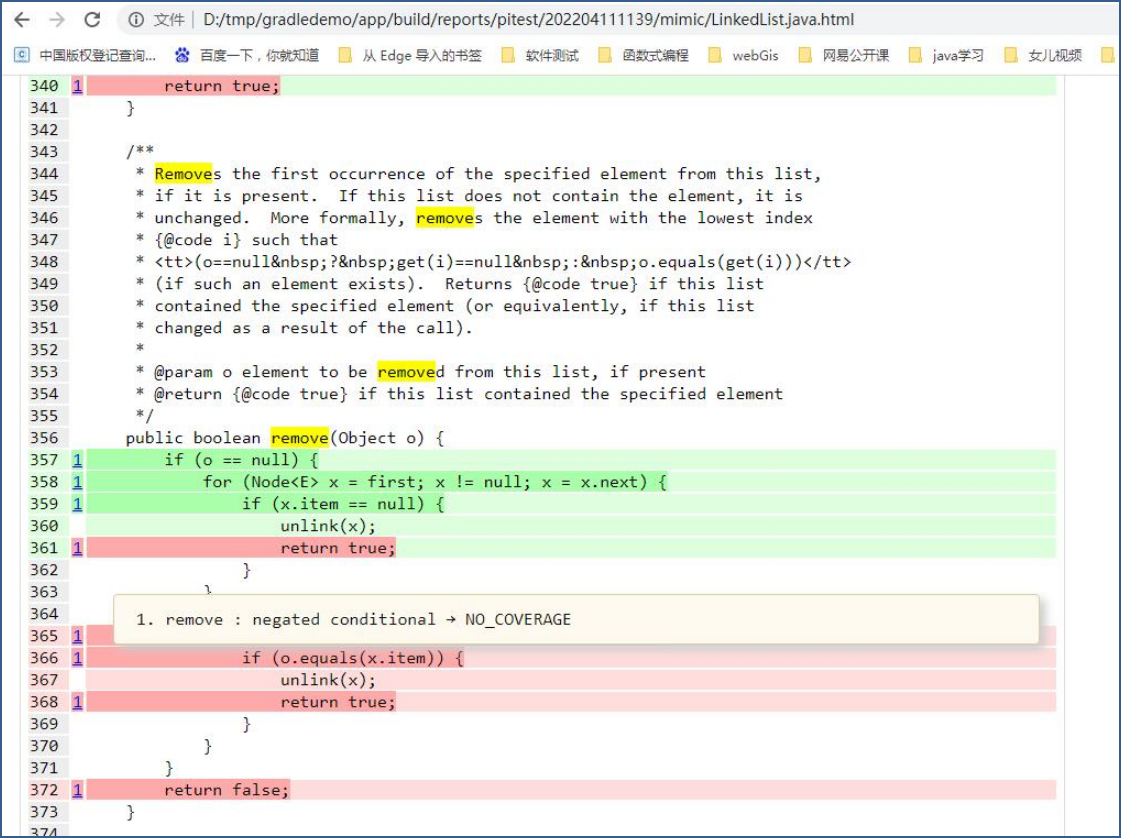
**以下代码是针对JDK8 LinkedList的remove方法实现，完成以下题目要求。具体实现可以参考OpenJDK**

1. /\*\*
2. \* Removes the first occurrence of the specified element from this list,
3. \* if it is present.  If this list does not contain the element, it is
4. \* unchanged.  More formally, removes the element with the lowest index
5. \* {@code i} such that
6. \* <tt>(o==null ? get(i)==null : o.equals(get(i)))</tt>
7. \* (if such an element exists).  Returns {@code true} if this list
8. \* contained the specified element (or equivalently, if this list
9. \* changed as a result of the call).
10. \*
11. \* @param o element to be removed from this list, if present
12. \* @return {@code true} if this list contained the specified element
13. \*/
14. **public** boolean remove(Object o) {
15. 01.  **if** (o == null) {
16. 02.    **for** (Node<E> x = first; x != null; x = x.next) {
17. 03.      **if** (x.item == null) {
18. 04.        unlink(x);
19. 05.        **return** **true**;
20. }
21. }
22. 06.  } **else** {
23. 07.    **for** (Node<E> x = first; x != null; x = x.next) {
24. 08.      **if** (o.equals(x.item)) {
25. 09.        unlink(x);
26. 10.        **return** **true**;
27. }
28. }
29. }
30. 11.  **return** **false**;
31. }
32. **配置变异测试插件Pitest**

**在build.gradle 中添加pitest插件，并配置pitest，如针对包”mimic”执行变异测试**

1. plugins {
2. id "info.solidsoft.pitest" version "1.7.4"
3. }
4. pitest {
5. targetClasses = ['mimic.\*']
6. threads = 4
7. outputFormats = ['HTML']
8. timestampedReports = **true**
9. junit5PluginVersion = '0.15'
10. }

**查看变异测试报告，并根据变异测试报告修改或补充测试用例，如下图所示**



**测试数据集：**

|  |  |  |  |
| --- | --- | --- | --- |
| 路径测试标号 | LinkedList中的数据 | 要移除的对象 | 预期输入 |
| P1 | Null,null | Null | 1 |
| P2 | A,null | Null | 1 |
| P3 | A,B | Null | 2 |
| P4 | A，A | A | 1 |
| P5 | A，B | B | 1 |
| P6 | B，B | A | 2 |

1. **代码实现：**

|  |
| --- |
| **测试代码：**  package org.example;  import org.junit.jupiter.params.ParameterizedTest;  import org.junit.jupiter.params.provider.CsvSource;  import static org.junit.jupiter.api.Assertions.assertEquals;  public class LinkedListTest {  @ParameterizedTest  @CsvSource({  "null, null, null, 1",  "A, null, null, 1",  "A, B, null, 2",  "A, A, A, 1",  "A, B, B, 1",  "B, B, A, 2"  })  void testLinkedListRemoveMethod(String a, String b, String c, int size) {  LinkedList<String> list = new LinkedList<>();  list.add("null".equals(a) ? null : a);  list.add("null".equals(b) ? null : b);  list.remove("null".equals(c) ? null : c);  assertEquals(list.size(), size);  }  }  图形用户界面, 应用程序, 表格, Excel  描述已自动生成    图形用户界面, 应用程序  描述已自动生成 |

### 五、结论分析与体会

学会了基路径测试方法

### 六、仓库地址

https://github.com/Osilly/Software-Testing-Experiment