目录

[1. the equals() method from the java.util.AbstractList<E> class 2](#_Toc62916745)

[1.1 (a)Draw a control flow graph for this method. (b) Label edges and nodes in the graph with the corresponding code fragments 2](#_Toc62916746)

[1.2 (c) Node coverage requires (at least) four tests on this graph. Explain why. 3](#_Toc62916747)

[1.3 (d) Provide four tests (as calls to equals()) that satisfy node coverage on this graph 4](#_Toc62916748)

[1.4 (e) Give test requirements and test values for prime path coverage 4](#_Toc62916749)

[2 Control Flow Coverage Tools 5](#_Toc62916750)

[1工具安装 5](#_Toc62916751)

[简介 5](#_Toc62916752)

[安装 6](#_Toc62916753)

[2 选择的类 6](#_Toc62916754)

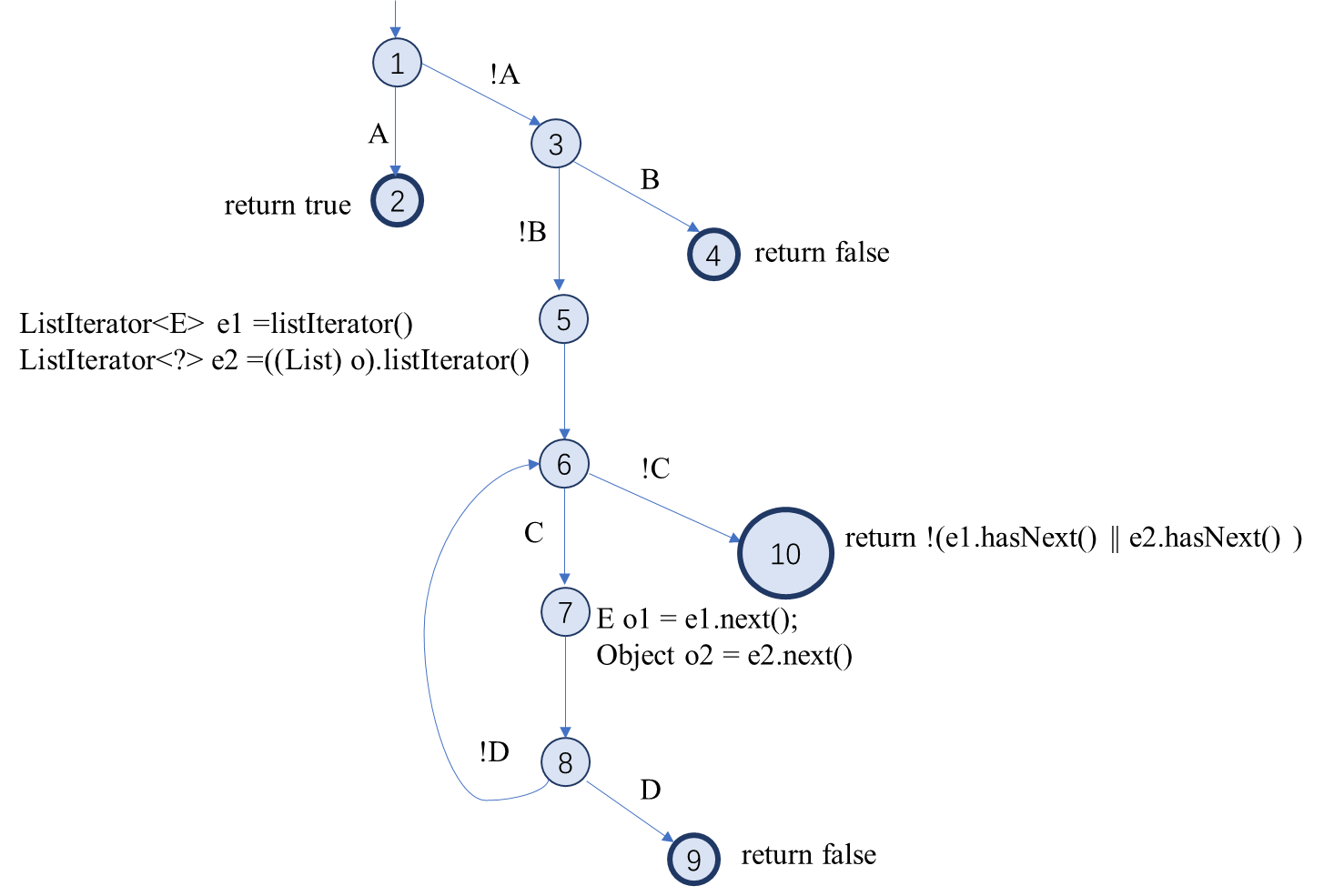
[3 进行测试 6](#_Toc62916755)

[4 测试结果 7](#_Toc62916756)

[5 附件 9](#_Toc62916757)

# 1. the equals() method from the java.util.AbstractList<E> class

## (a)Draw a control flow graph for this method. (b) Label edges and nodes in the graph with the corresponding code fragments



## (c) Node coverage requires (at least) four tests on this graph. Explain why.

equals()方法有3个没有else的if结构和1个while结构，有4个final nodes, 要满足节点覆盖标准，至少需要如下4条路径来访问到每个节点，因此也就需要至少4个测试 如t1, t2, t3, t4

path(t1) = [1,2]

path(t2) = [1,3,5,6,7,8,9]

path(t3) = [1,3,4]

path(t4) = [1,3,5,6,7,8,6,10]

## (d) Provide four tests (as calls to equals()) that satisfy node coverage on this graph



## (e) Give test requirements and test values for prime path coverage

prime path ={ [1,2]

[1,3,4]

[1,3,5,6,10]

[1,3,5,6,7,8,9]

[6,7,8,6]

[7,8,6,7]

[8,6,7,8]

[7,8,6,10] }

TR = { tour[1,2]

tour[1,3,4]

tour[1,3,5,6,10]

tour[1,3,5,6,7,8,9]

tour[6,7,8,6]

tour[7,8,6,7]

tour[8,6,7,8]

tour[7,8,6,10] }

path(t1) = [1,2]

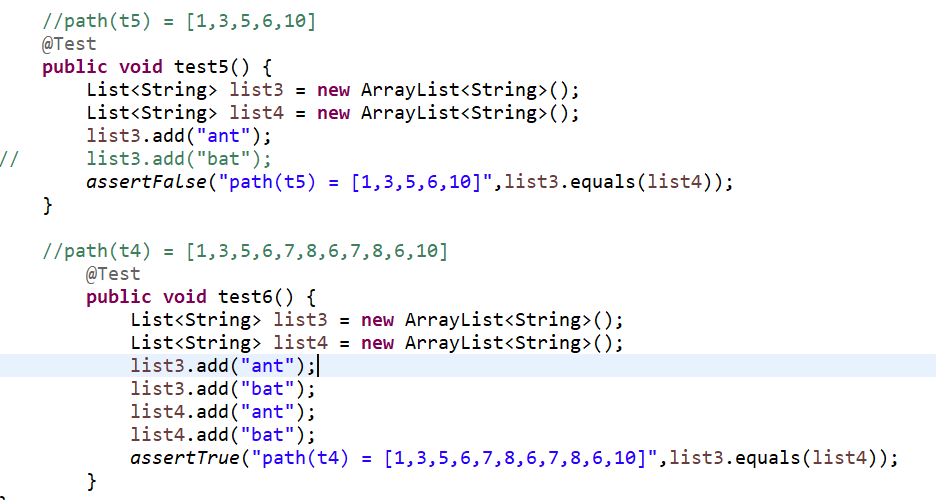
path(t2) = [1,3,5,6,7,8,9]

path(t3) = [1,3,4]

path(t4) = [1,3,5,6,7,8,6,7,8,6,10]

path(t5) = [1,3,5,6,10]

t1, t2, t3同(d)的t1,t2,t3 因此测试同(d)的test1 , test2 , test3。如下是t4 , t5



# 2 Control Flow Coverage Tools

## 1工具安装

我选择的工具是eclemma

### 简介

Eclemma是一个优秀的开源软件测试工具eclipse的一个插件能够对由 Java 语言编写的程序进行覆盖测试。

### 安装

集成到：Eclipse Photon Release (4.8.0)

Eclemma有很多安装方法，我使用的是本地安装：

将eclemma压缩包下载下来后解压到Eclipse本地目录的dropins下，重启Eclipse

eclemma压缩包下载地址：<https://sourceforge.net/projects/eclemma/>

eclemma插件安装成功标志：在Eclipse上会出现新的图标



使用：直接点击上述图标或右键任意一个java文件，选中Coverage As选项

## 选择的类

我选择的是java.util. ArrayList<E>

由于要测试覆盖率，我将ArrayList<E>类的代码复制为本地文件，ArrayListDemo.java，由于remove(), add()等多个方法涉及访问parent.modCount，而AbstractList<E>类的成员变量**protected** **transient** **int** modCount = 0;在包外无法访问，我新建了AbstractListDemo.java文件, 将AbstractList<E>类的代码复制到此文件中。在ArrayListDemo.java中引入了需要使用一些方法所属的类如下。

**import** java.util.\*;

**import** java.util.function.Consumer;

**import** java.util.function.Predicate;

**import** java.util.function.UnaryOperator;

**import** sun.misc.SharedSecrets;

## 进行测试

我采用源码中方法的顺序逐一进行测试，因为需要而会调用到其他方法，就一并测试，不调用其他方法的方法会优先测试。

下面记录测试中一些特别的地方

我先进行了3个构造方法的测试，采用ArrayListDemo()，ArrayListDemo(**int** initialCapacity)，ArrayListDemo(Collection<? **extends** E> c)的顺序，

ArrayListDemo()没有分支结构，ArrayListDemo(**int** initialCapacity)有判断参数是否>0, =0, <0的3个分支，使用不同的参数可以覆盖到不同的分支，<0会抛出异常，用try-catch块捕捉异常，在这里用的org.evosuite.runtime.EvoAssertions.*verifyException*方法验证希望的异常抛出；后面涉及到分支和异常的测试都使用类似的方法，测试异常还使用到了[org](eclipse-javadoc:%E2%98%82=HelloDemo/D:%5C/Program%20Files%20%5C(x86)%5C/eclipse-java-photon-R-win32-x86_64%5C/eclipse%5C/plugins%5C/org.junit_4.12.0.v201504281640%5C/junit.jar%3Corg).[junit](eclipse-javadoc:%E2%98%82=HelloDemo/D:%5C/Program%20Files%20%5C(x86)%5C/eclipse-java-photon-R-win32-x86_64%5C/eclipse%5C/plugins%5C/org.junit_4.12.0.v201504281640%5C/junit.jar%3Corg.junit).[Test](eclipse-javadoc:%E2%98%82=HelloDemo/D:%5C/Program%20Files%20%5C(x86)%5C/eclipse-java-photon-R-win32-x86_64%5C/eclipse%5C/plugins%5C/org.junit_4.12.0.v201504281640%5C/junit.jar%3Corg.junit(Test.class%E2%98%83Test).expected()，org.junit.Assert.fail(String message)。

测试add()时，涉及调用ensureCapacityInternal(size + 1)，ensureCapacityInternal方法又调用ensureExplicitCapacity(calculateCapacity(elementData, minCapacity))，我使用不同的ArrayListDemo对象调用add方法将这3个方法都测试了。

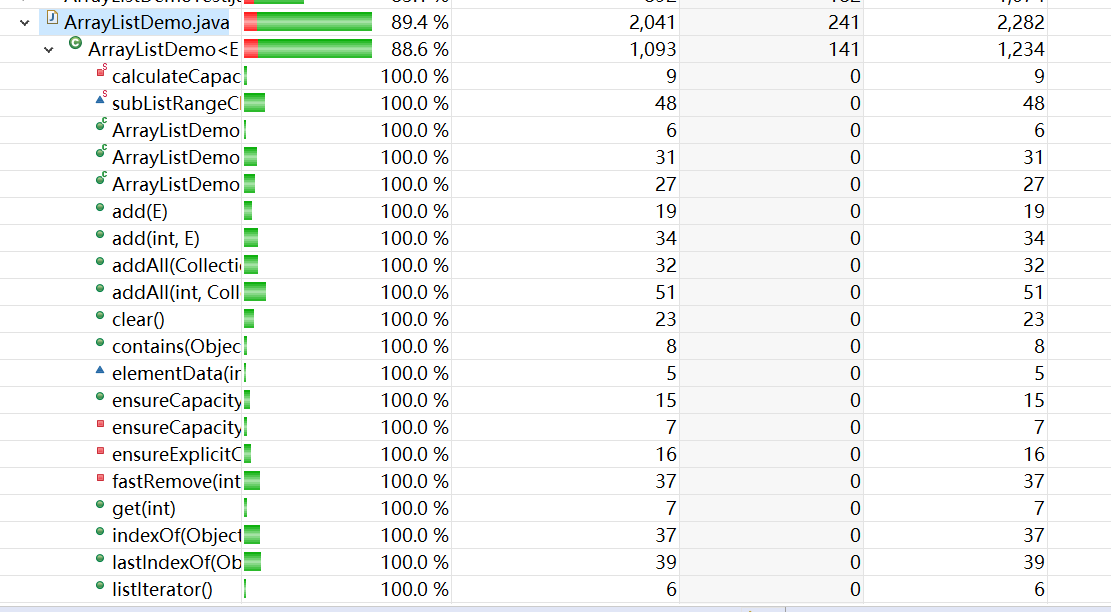
后面的类中的Itr, ListItr,SubList等类都是私有的，它们的有些方法不是public无法被调用，我将有的修饰符改成了public，有的放在其他方法里调用，其中SubList的removeRange放在remove里调用了，由于sublist方法返回的是List<E>, 而这个类没有removeRange方法。

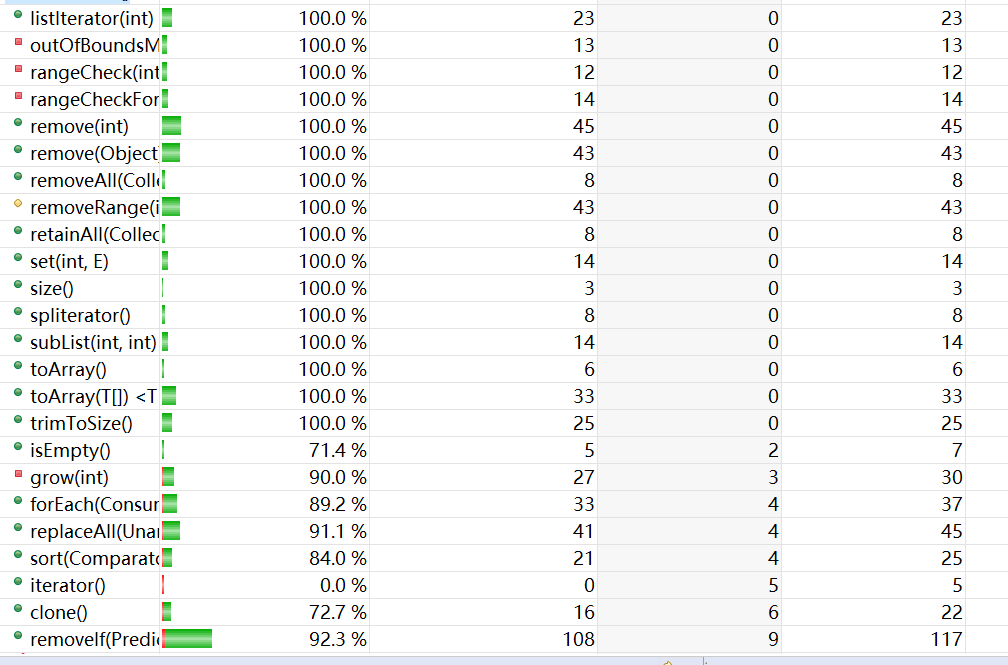
writeObject，readObject方法没有被测试到，它们是private的，也没有被其他方法调用。SubList类的大部分方法都和ArrayListDemo类似，有些测试我使用了ArrayListDemo中的代码修改而成。checkForComodification()的if分支不知道如何测到，它要求对列表结构性修改的次数与被记录的不一致，而这个条件我在看完整个类代码后并未发现如何达到。最后3个方法removeIf, replaceAll, sort的参数有些特殊和相似，都使用到了lambda表达式，当然，使用到的还有forEach等。

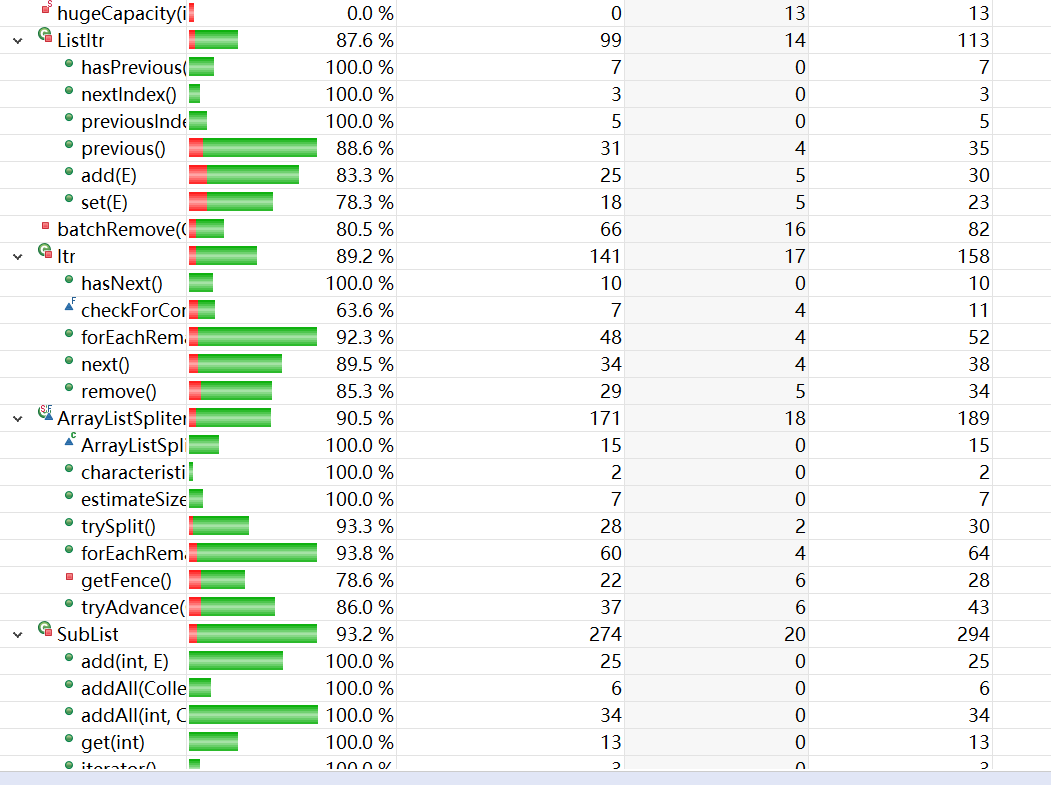
## 测试结果

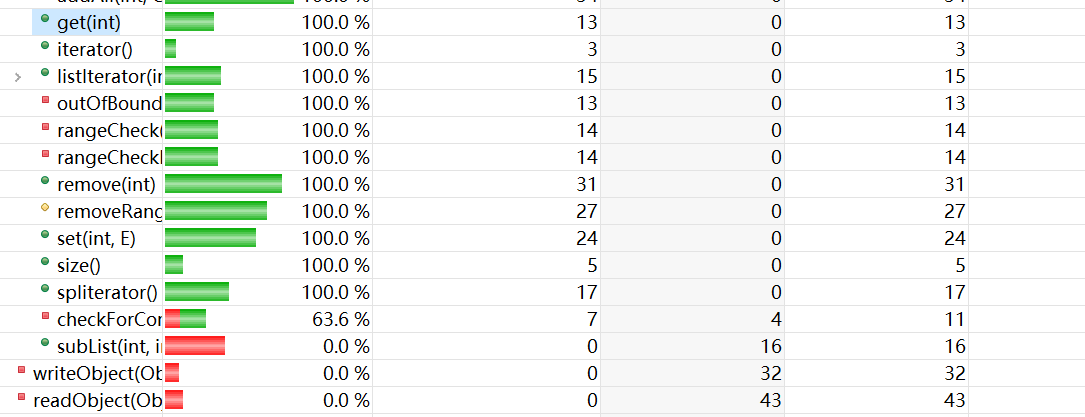
对绝大多数方法实现了语句覆盖和分支覆盖，一部分方法实现了部分分支覆盖，少数方法没有被测试到，功能性的代码几乎完全测试到。

使用EclEmma工具测试到的测试覆盖率：总覆盖率89.4%









## 附件

源码：<ArrayListDemo.java>

Junit测试：[ArrayListDemo .java](ArrayListDemoTest.java)