****

**软件测试技术第四次实验报告**



**学 院 智能与计算学部**

**专 业 软件工程**

**年 级 2016**

**姓 名 傅淏**

**2019年 4 月 18 日**

# 软件测试技术第四次实验报告

1. 需求分析（描述具体需求）

1. Install MuJava. The instruction of how to install and use MuJava can be seen in https://cs.gmu.edu/~offutt/mujava/.

2. Two small programs are given for your task. BubbleSort.java is an implementation of bubble sort algorithm and BackPack.java is a solution of 01 backpack problem. Try to generate Mutants of 2 given programs with MuJava.

3. Write testing sets for 2 programs with Junit, and run mutants on the test sets with MuJava.

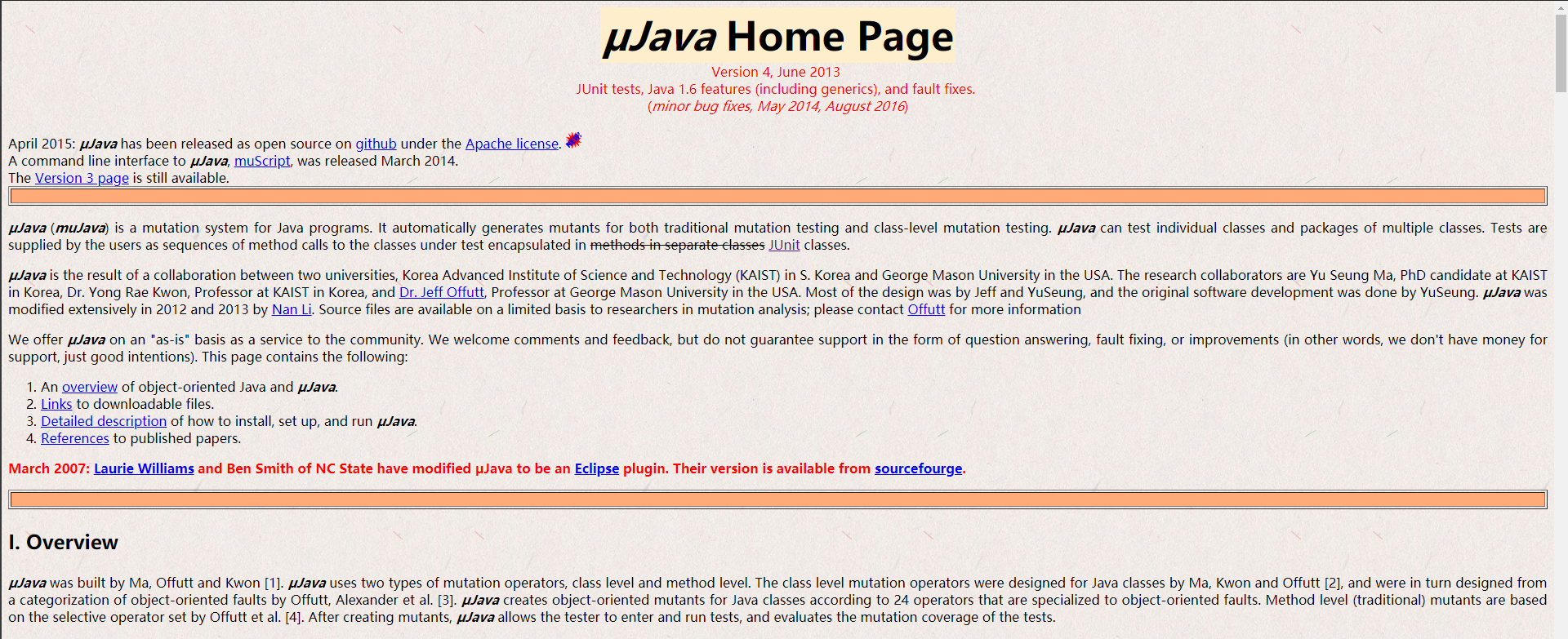
1. 概要设计

**设计思路：**

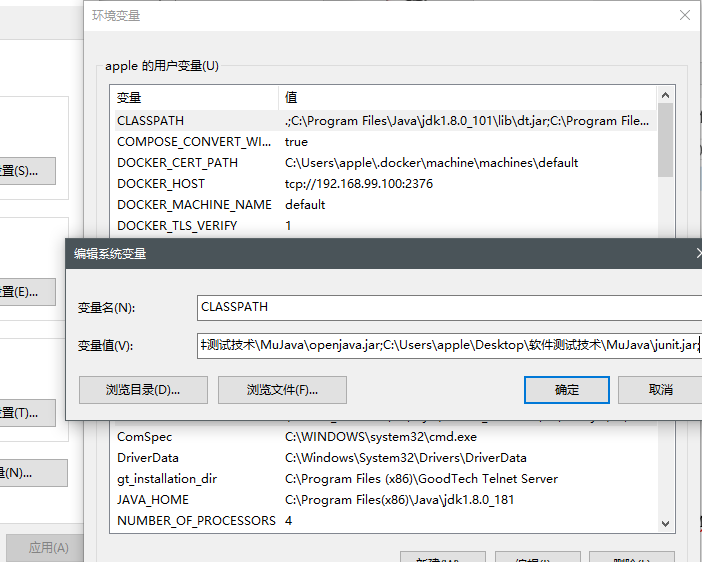
根据所给程序编写Junit测试程序，覆盖程序的所有分支与路径。

1. 详细设计（详细描述具体如何实现，附代码及说明）

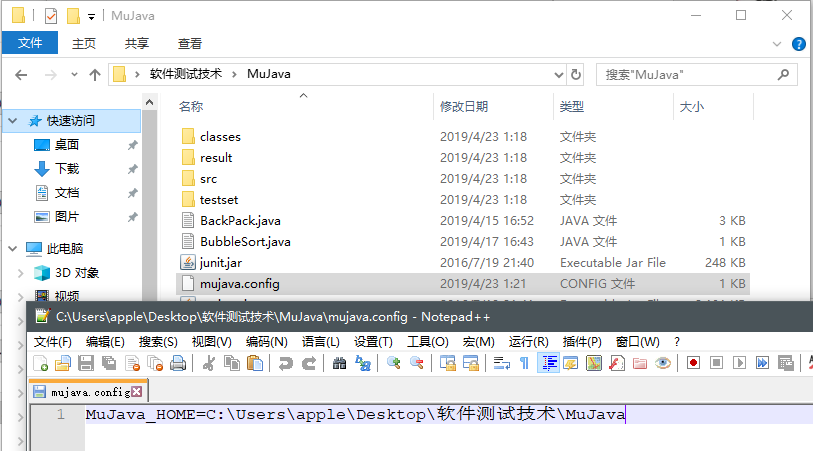
1、根据MuJava官网教程进行下载与安装；



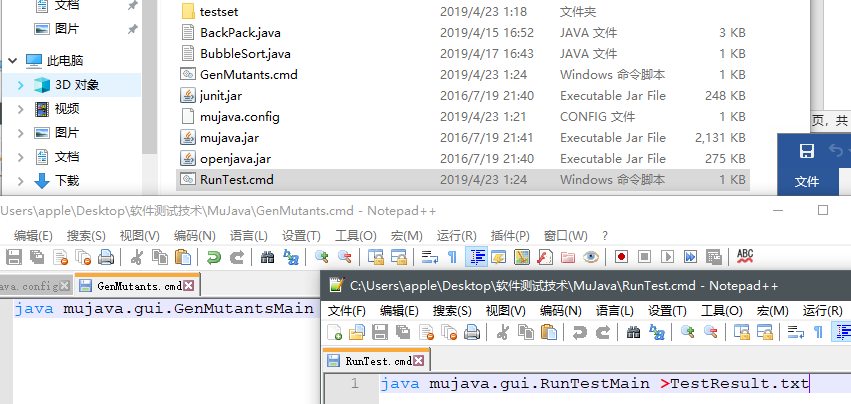
2、配置MuJava环境变量；



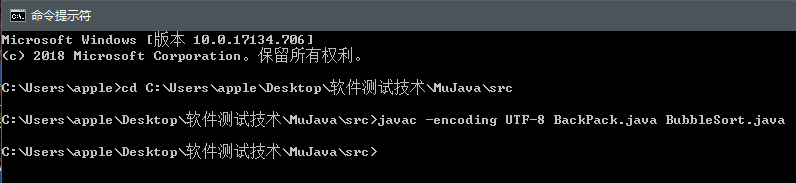
3、创建文件夹并配置config文件

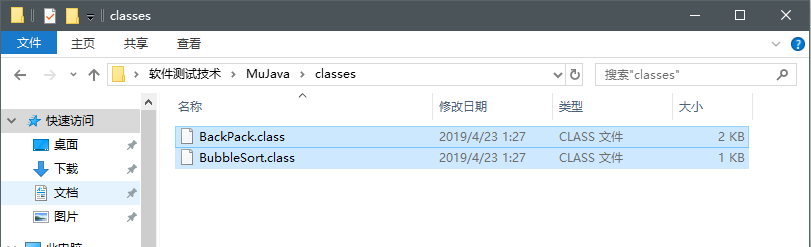


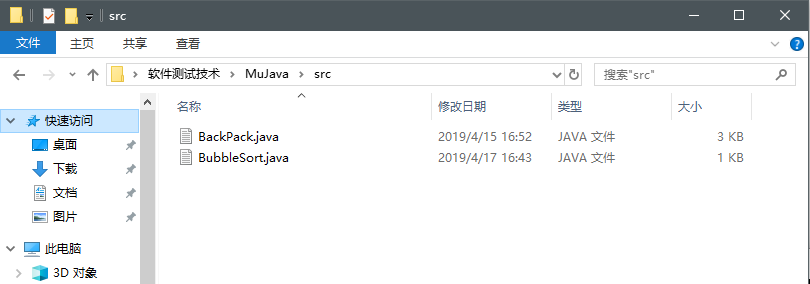
4、创建两个cmd程序用于直接运行；



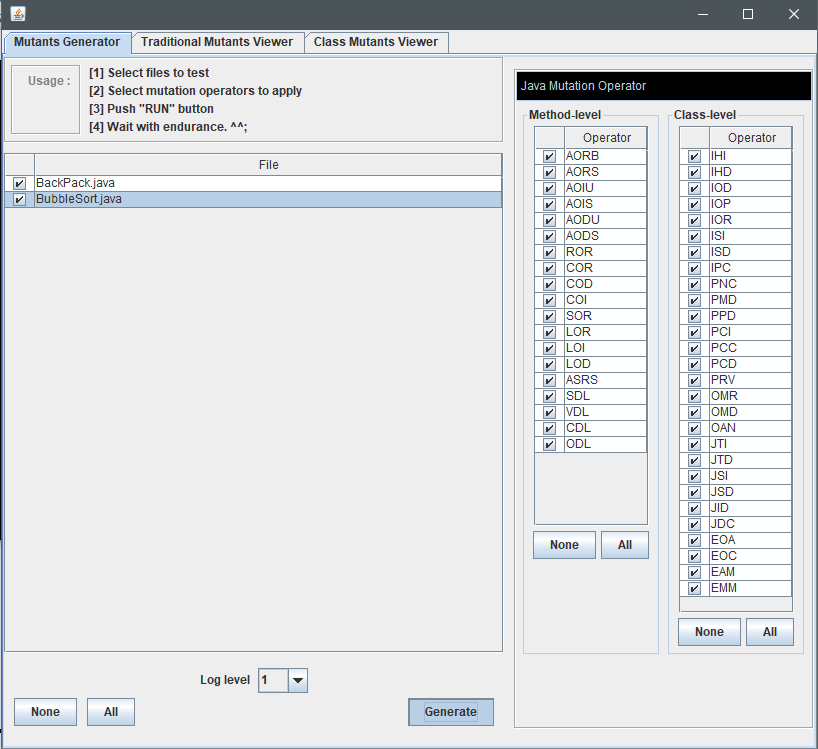
5、生成两个文件的变体，分别放入不同的文件夹中；

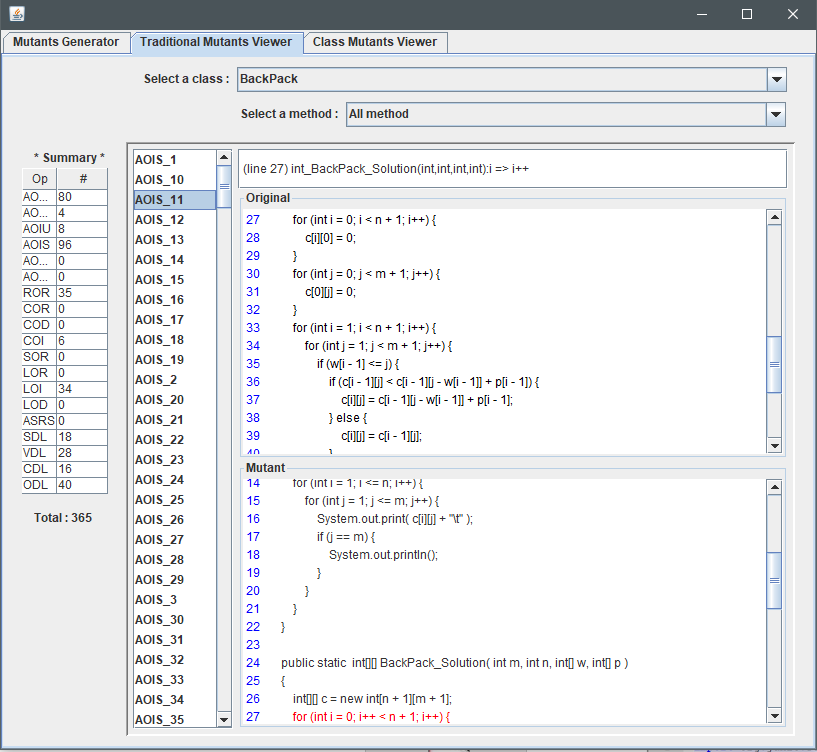


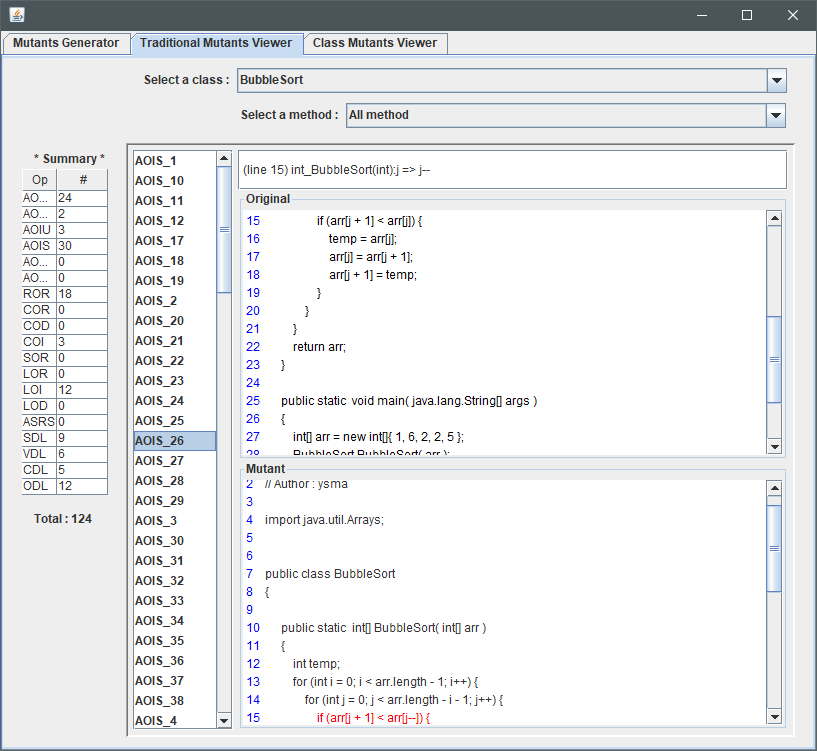




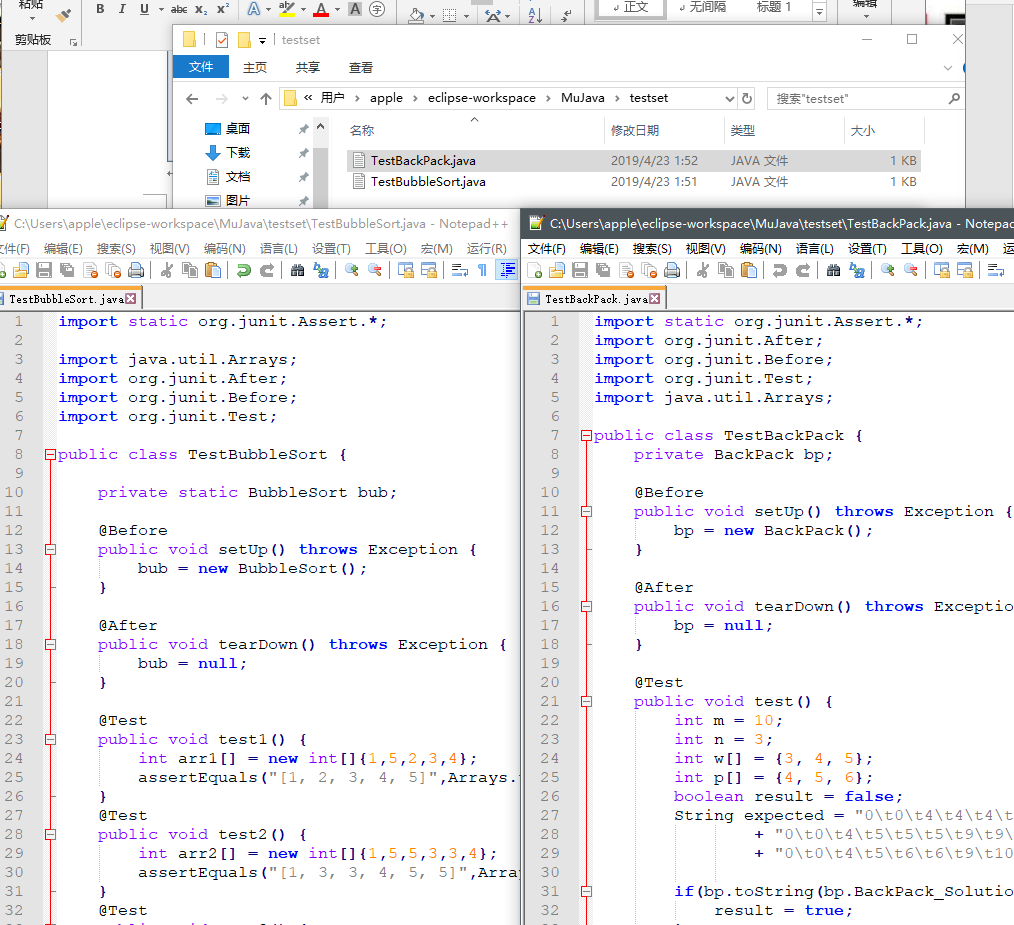
6、运行GenMutants.cmd，进入MuJava的GUI界面，生成mutant；





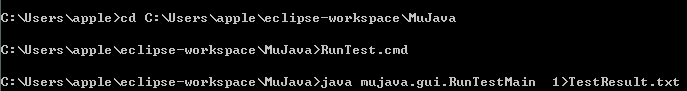


7、编写测试程序，进行变异测试；





8、设置参数进行测试；



1. 调试分析（在实验过程中遇到的问题以及如何解决）

1、缺少MuJava.config文件导致MuJava无法运行，添加路径时句子末尾无任何符号；

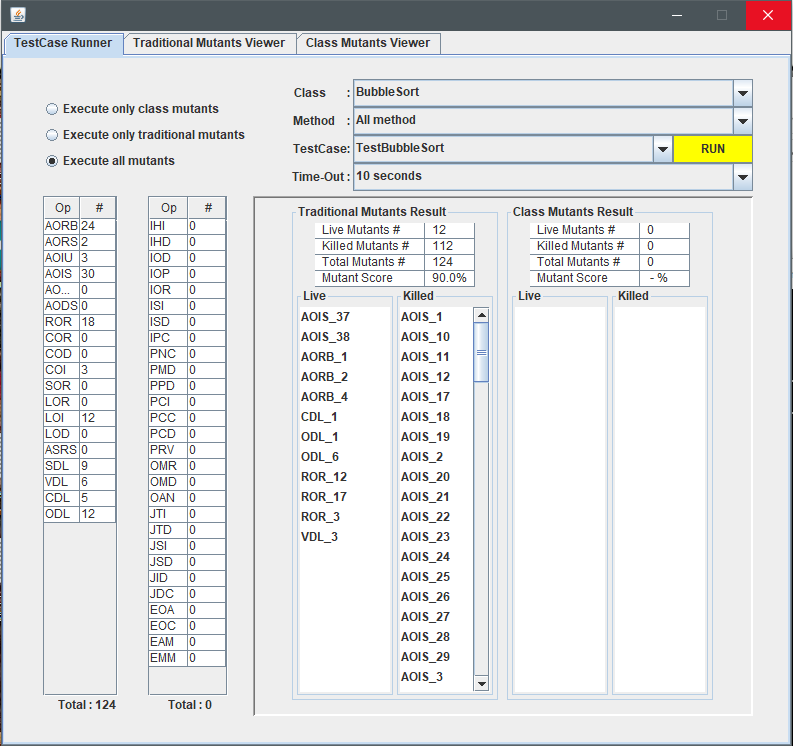
2、无法编译Test文件，将源文件与编写的Test文件放在同一目录下再编译；

3、每次测试时电脑较卡，所以有几次的TestResult文件没有截取下来；

1. 测试结果（描述输入和输出）

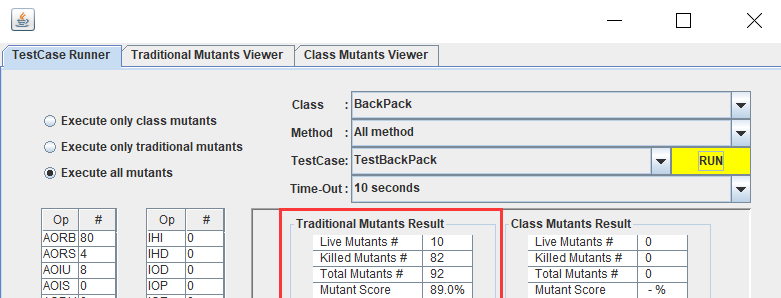
BubbleSort：

Kill112个，Live12个，MutantScore为90%

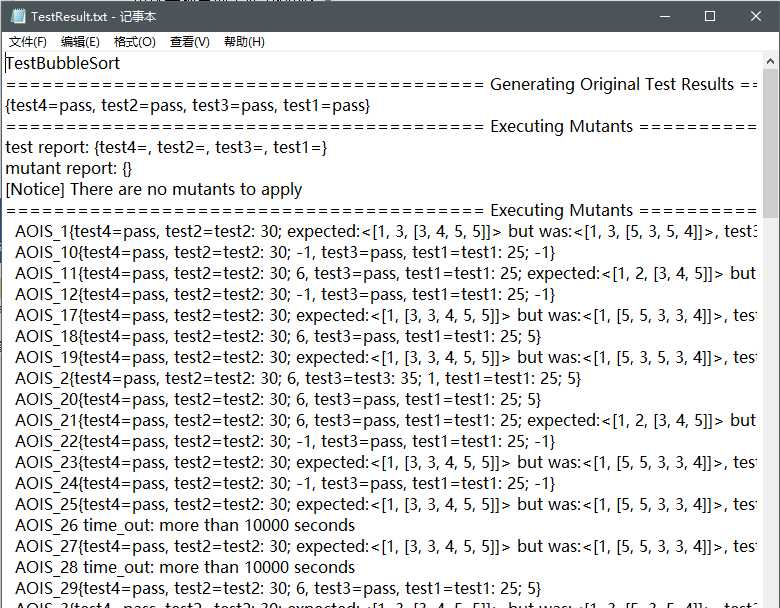


BackPack：

Kill82个，Live10个，MutantScore为89%



部分TestResult：



1. 总结

通过这次实验，我学会了使用MuJava进行变异测试，使用Junit进行单元测试，并且也温习了冒泡排序与背包问题。

七、GitHub的URL

https://github.com/HaoFuTju