**软件测试上机报告**

****

Lab 4 Mujava

**学 院 软件学院**

**专 业 软件工程**

**姓 名 苏灿臣**

**学 号 3017218137**

**年 级 2017级**

**班 级 3**

**Requirements analysis**

1. Install 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.

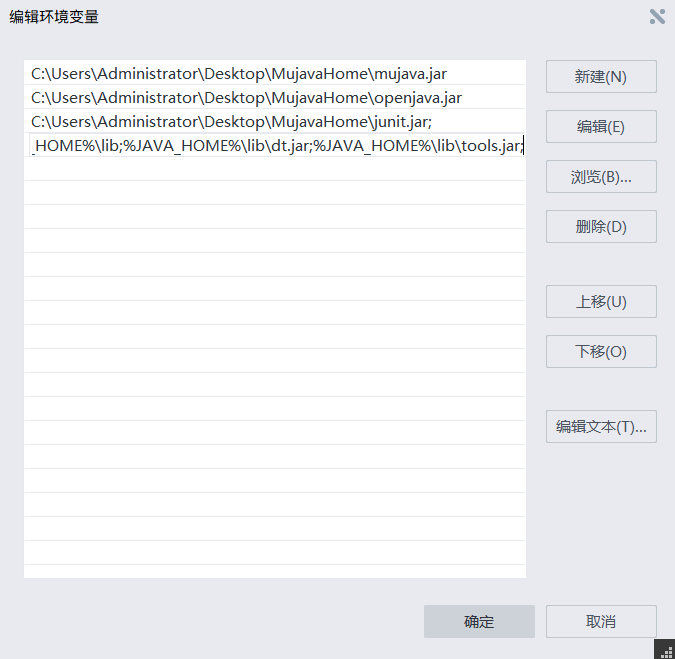
**Architectural Design**

First install the required packages, configure the environment variables, and generate the required variants. Then write the corresponding test set, and then test these variants.

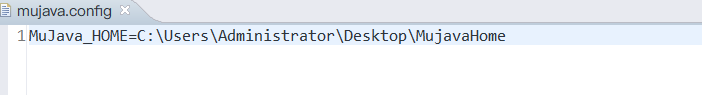
**Detailed Design**

1. Install MuJava.

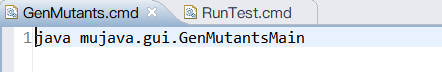
Add the downloaded jar package to the environment variables, then add the path of these jar packages in the CLASSPATH. The result is shown in the following figure



Create a new mujava.config file and write the address of MuJava\_HOME in it, which is the address of the newly created folder.



Create two new command script files, GenMutants.cmd and RunTest.cmd, and write the contents as shown in the figure below.





1. Generate variants.

First create four new folders in the MujavaHome folder, namely src, classes, result, testset.

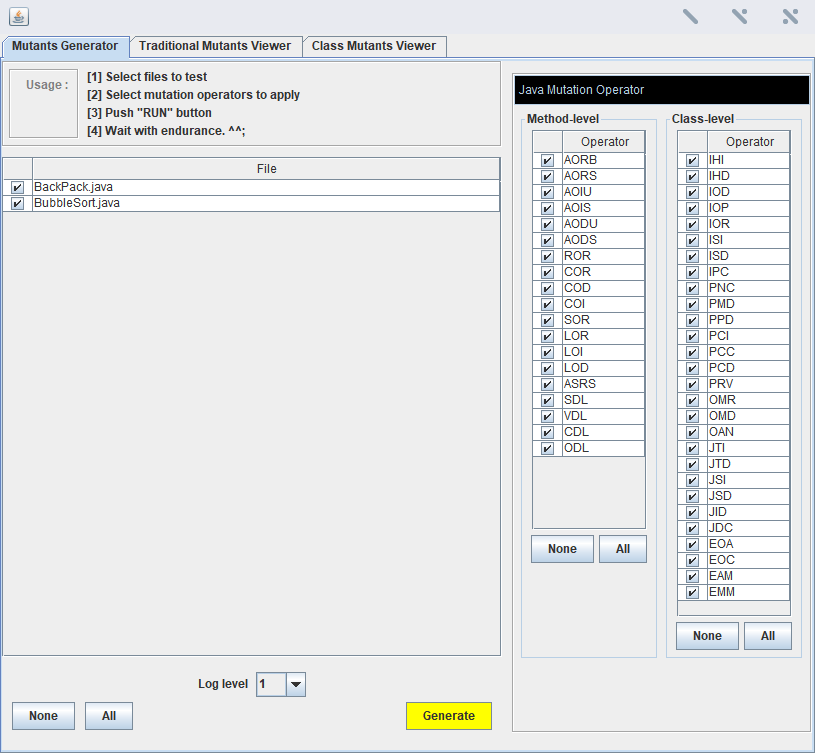


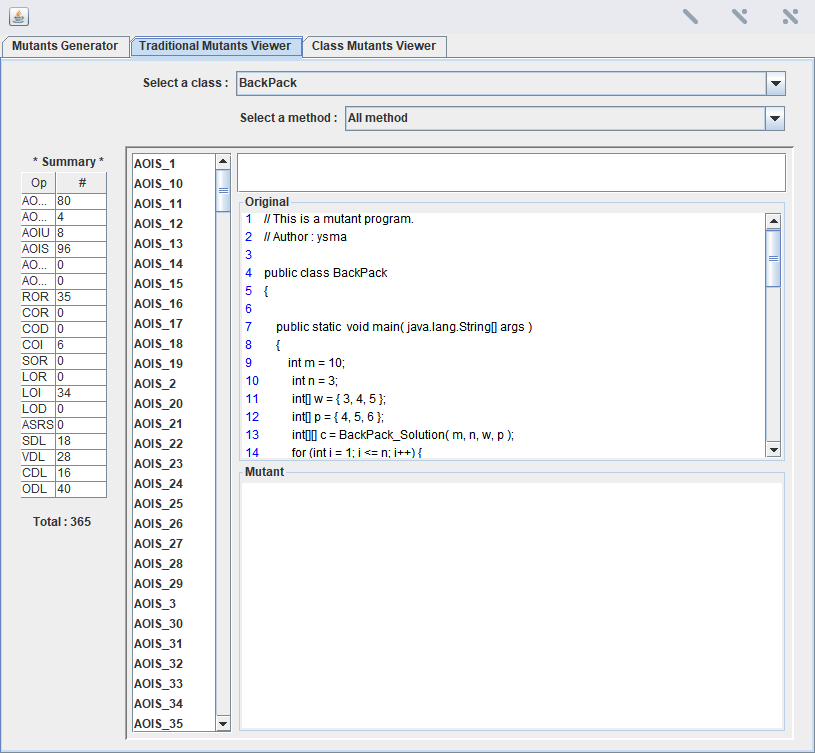
Put the two given source programs in the “src” folder, and then put the two resulting class files which is compiled by the eclipse in the “classes” folder.

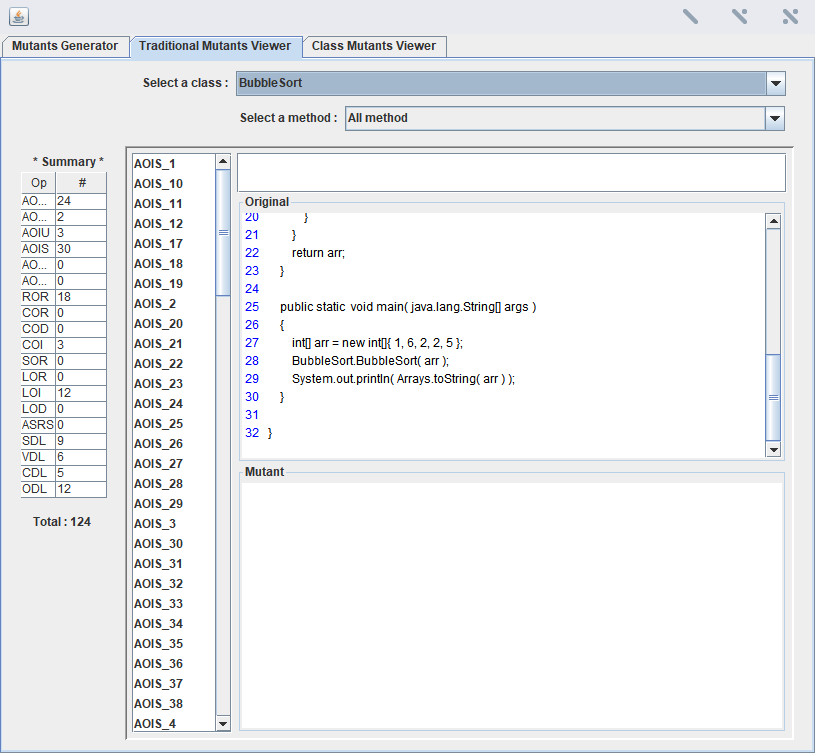




Click to run the GenMutants.cmd file, the following graphical interface pops up, and check all the mutation operators.



Click the “Generate” button to generate variants. The following figure shows the variants of BackPack and BubbleSort: 

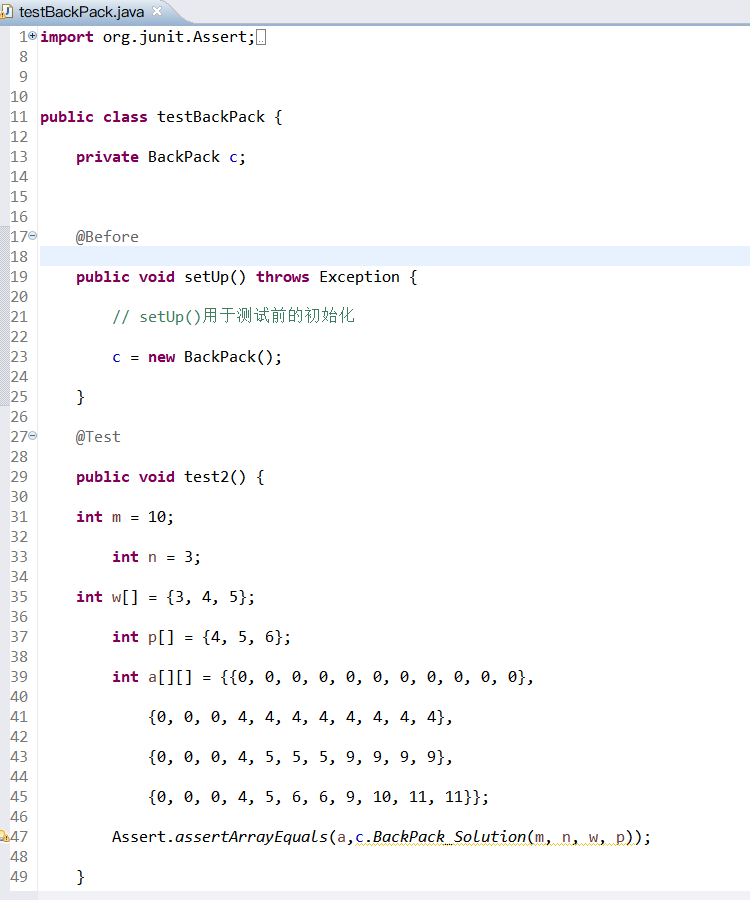


Among them, only method-level variants are generated, and no class-level variants are generated.

After generating the variant, you can find the generated variant file in the result folder.



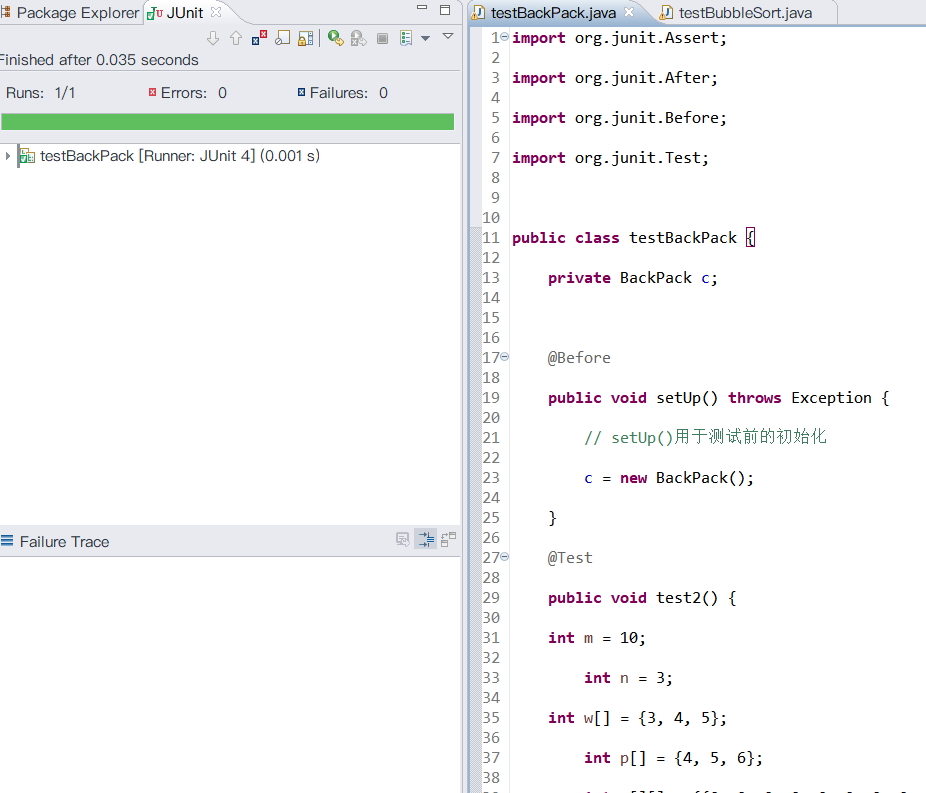
1. Write the test cases.
2. backpack的测试用例：



1. bubblesort的测试用例：



1. Test variants



Then put testBackPack.java, testBubbleSort.java, testBackPack.class, testBubbleSort.class into the “testset” folder.



Then click to run the RunTest.cmd file, open the graphical interface, set the parameters, and test the variants of the two programs.

**Debug analysis (problems encountered during the experiment and how to solve them)**

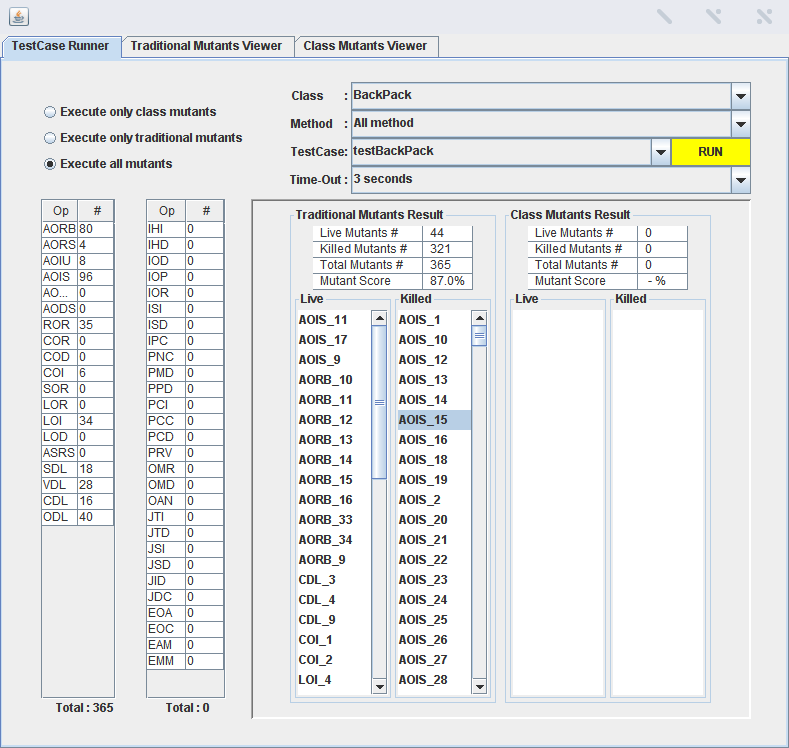
Problems caused by incorrect configuration of environment variables.

After clicking the ‘Generate’ button, the variant was not successfully generated and an error was reported at the command desk: Error in update() in TraditionalMutantsViewerPanel.

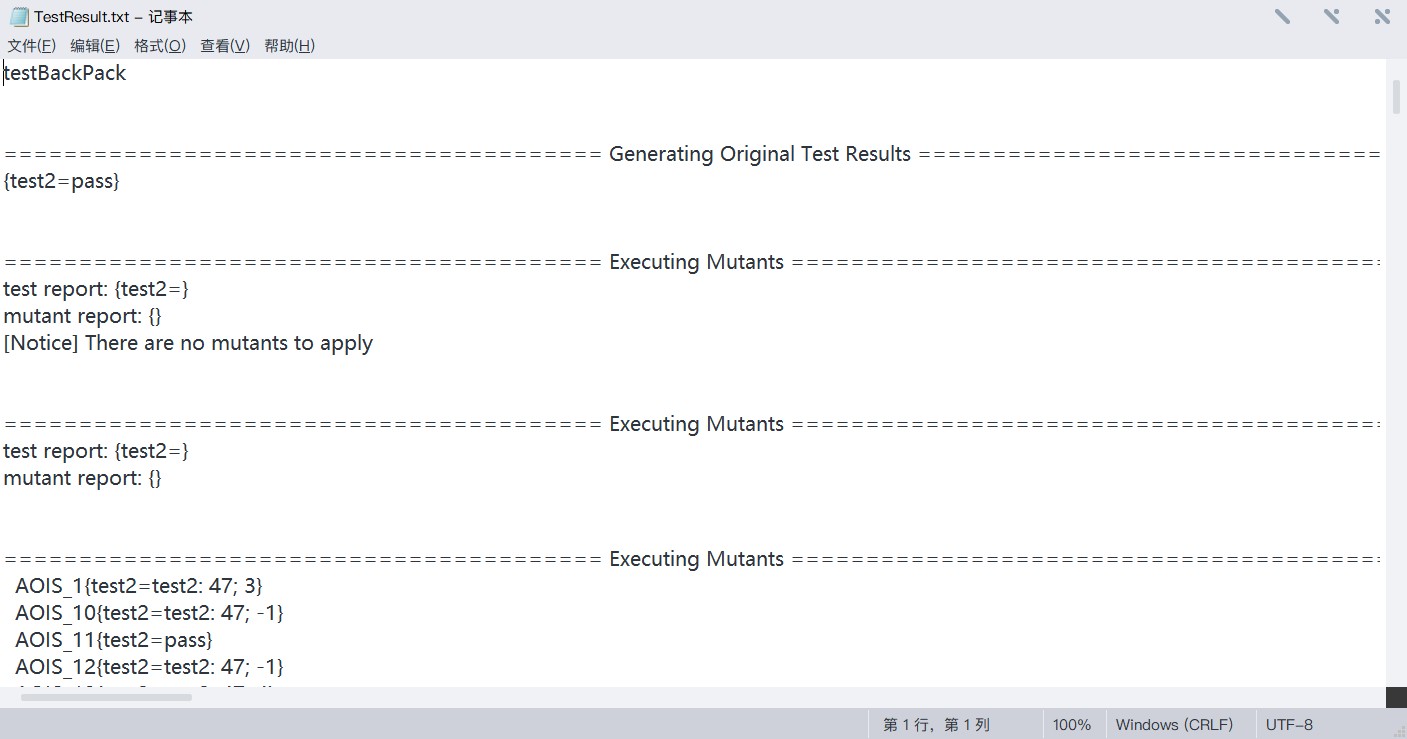
Solved by re-creating the cmd script file.

**Test results (describe input and output)**

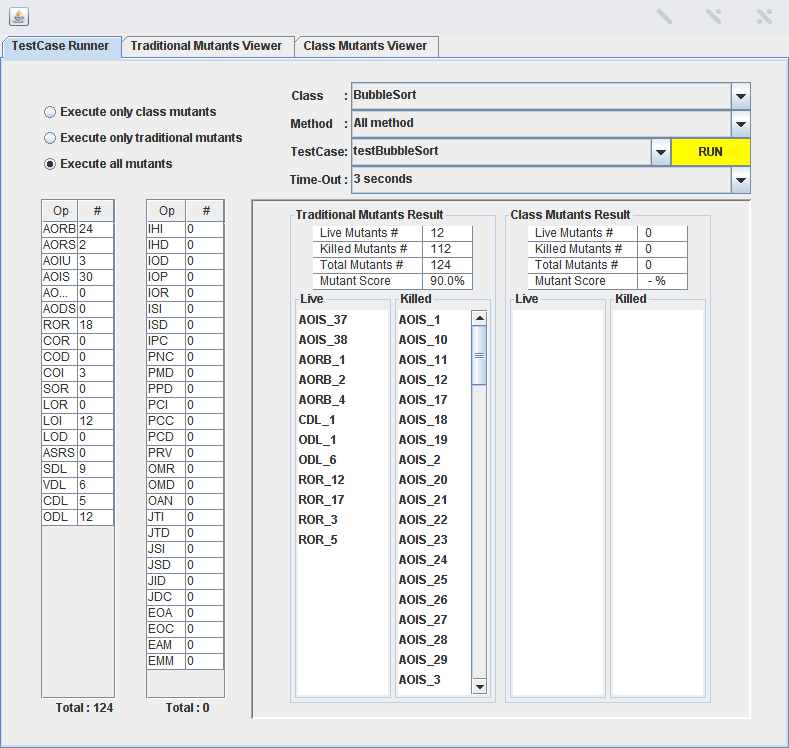
1. The following figure shows the test result of the BackPack program and the output result.



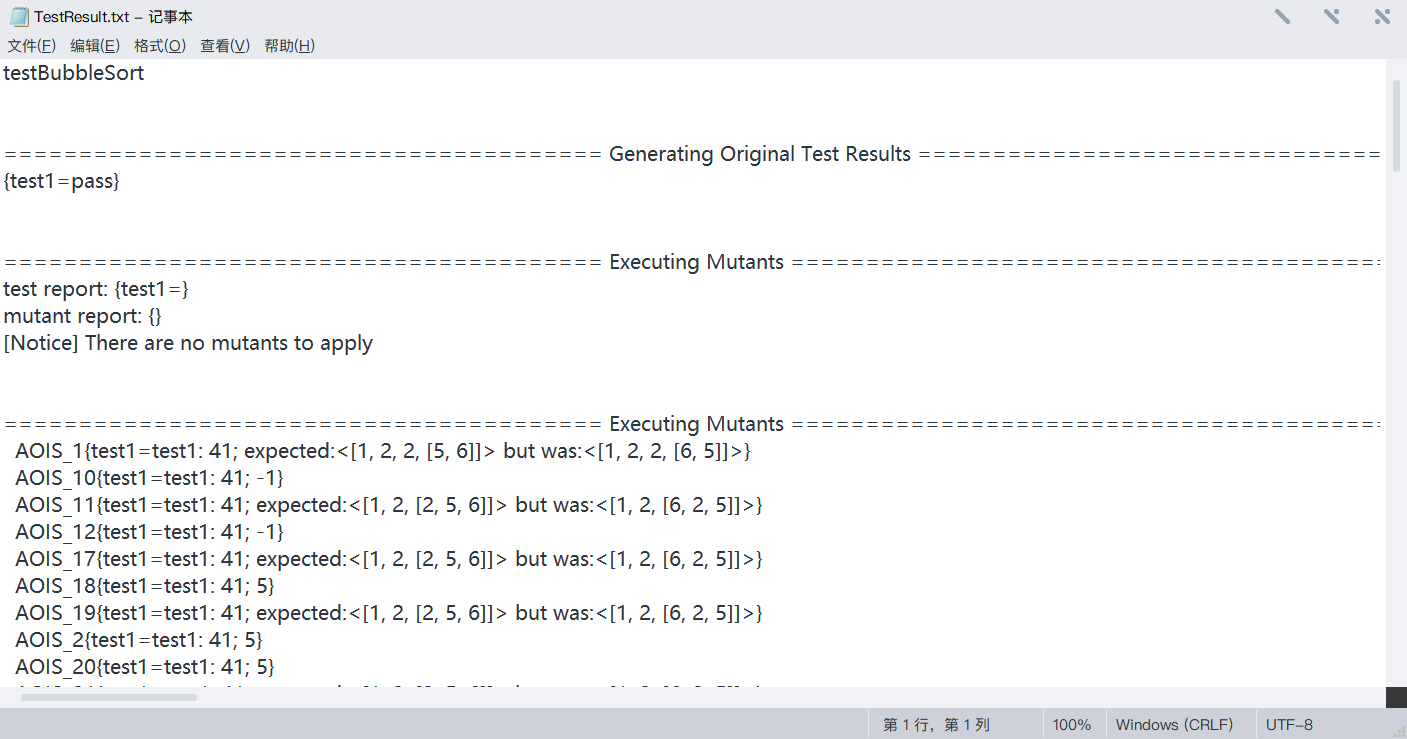
It can be seen that there are a total of 365 variants and 321 variants that have been killed.



1. The following figure shows the test result of the BubbleSort program and the output result.



It can be seen that there are a total of 124 variants and 112 variants that have been killed.



**Summary**

Through this experiment, I practiced the configuration method of environment variables, and also learned the technique of using mujava to test the mutation of the program. Including how to classify files, generate variants and run mutation tests. When writing test cases for a program, the function assertArrayEquals is used to test arrays. I also learned how to write the command to the cmd file to implement the command.