软件测试上机报告

第四次上机作业

学　　院__智能与计算学部__
专　　业__软件工程_____
姓　　名__马天宇_____
学　　号__3017218064_____
年　　级__2017 级_____
班　　级__软工一班_____

一、实验要求

Tasks:

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.f an error occurs, simply describe the error and analyze the cause of the error

二、源代码

TestBubbleSort.java:

```
import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.After;
```

```
import org.junit.Before;

public class TestBubbleSort {

    @Test
    public void test() {
        assertArrayEquals(BubbleSort.BubbleSort(new int[]
{2,3,5,2,1,3,0}), new int[] {0,1,2,2,3,3,5});
    }

}
```

TestBackPack.java:

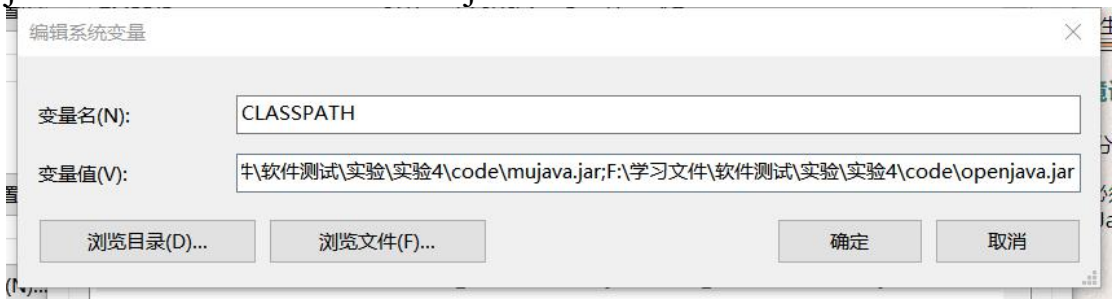
```
import static org.junit.Assert.*;

import org.junit.Test;
import org.junit.After;
import org.junit.Before;
public class TestBackPack {
```

```
    @Test
    public void test() {
        assertArrayEquals(new int[][] {{0, 0}, {0, 10}},
BackPack.BackPack_Solution(1,1,new int[] {1}, new int[] {10}));
    }

}
```

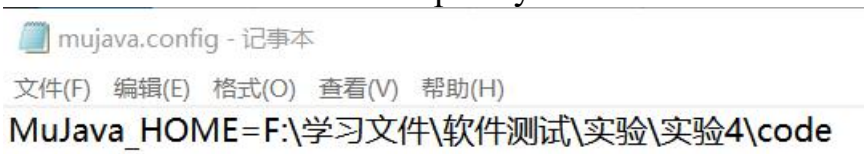
三、运行结果

1. Set environment for the muJava system:

1.1 Set CLASSPATH. The Java CLASSPATH must include two μJava jar files and one standard Java jar file.



1.2 Modify the mujava.config file to point to a directory that contains the source Java files and muJava temporary files .

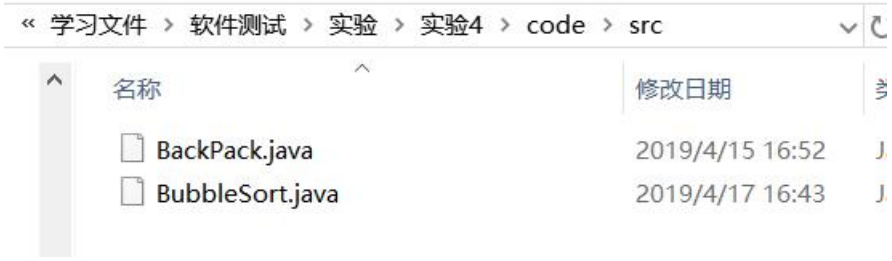


1.3 Create a directory structure for the muJava system in the $MuJava_HOME directory by using the muJava class "mujava.makeMuJavaStructure".

```
F:\学习文件\软件测试\实验\实验4\code>java mujava.makeMuJavaStructure

Make F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code directory...
F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code directory exists already.

Make F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\src directory...
F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\src directory exists already.

Make F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\classes directory...
F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\classes directory exists already.

Make F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\result directory...
F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\result directory exists already.

Make F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\testset directory...
F:\学习文件\软件测试\实验\实验4\code\= F:\学习文件\软件测试\实验\实验4\code\testset directory exists already.
```

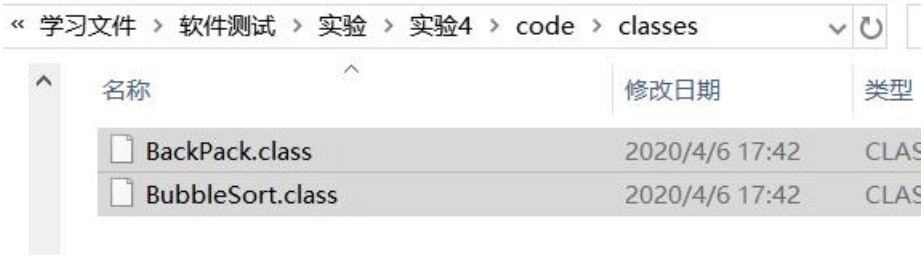
2.Generating Mutants with muJava

2.1 Put the source files to test to MuJava_HOME\src directory.



2.2 Compile all the Java files in MuJava_HOME\src and copy the .class files into the MuJava_HOME\classes\ directory.

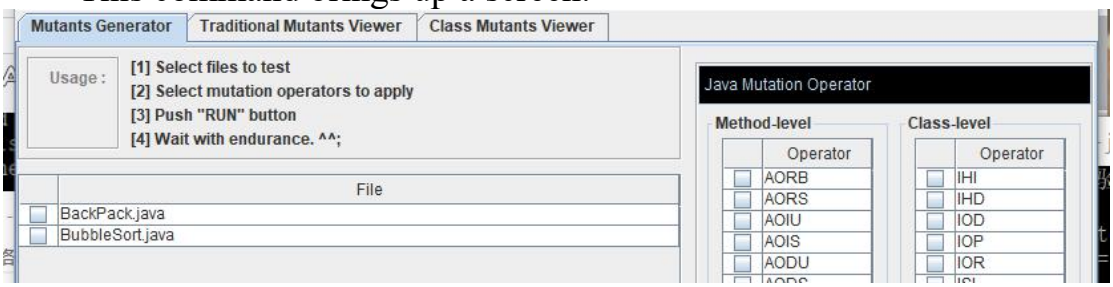
```
实验4\code\src>javac -encoding UTF-8 BackPack.java BubbleSort.java

实验4\code\src>
```



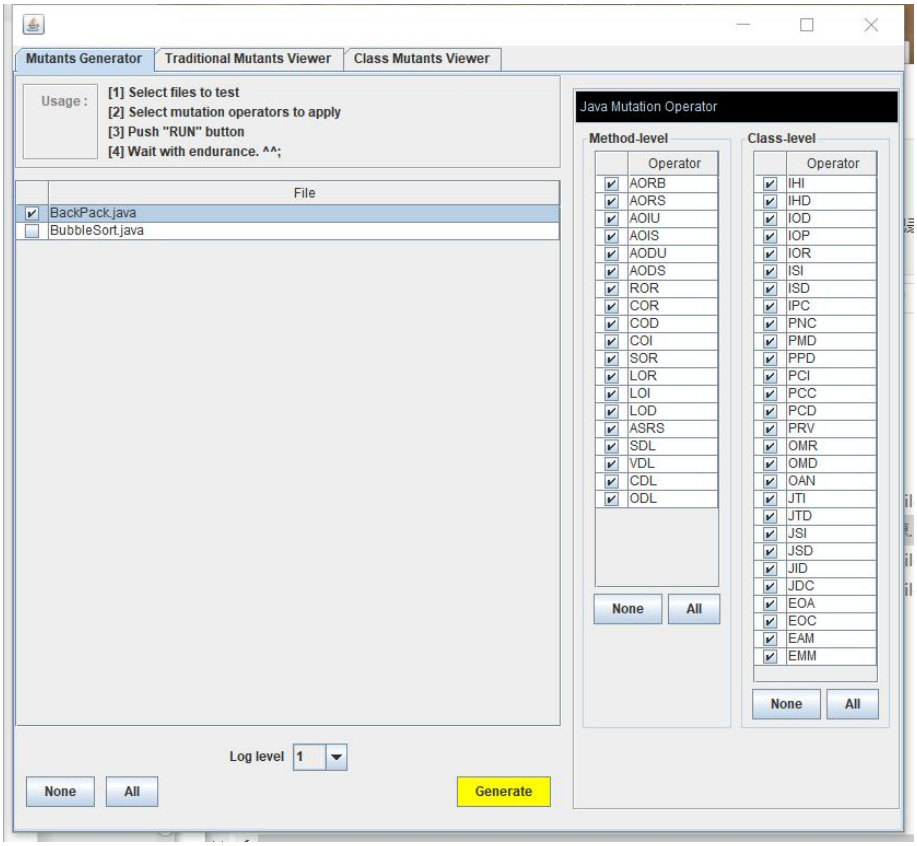
2.3 Start the GUI from the command line. Use it to generate mutants:

```
F:\学习文件\软件测试\实验\实验4\code>java mujava.gui.GenMutantsMain
The main method starts
```

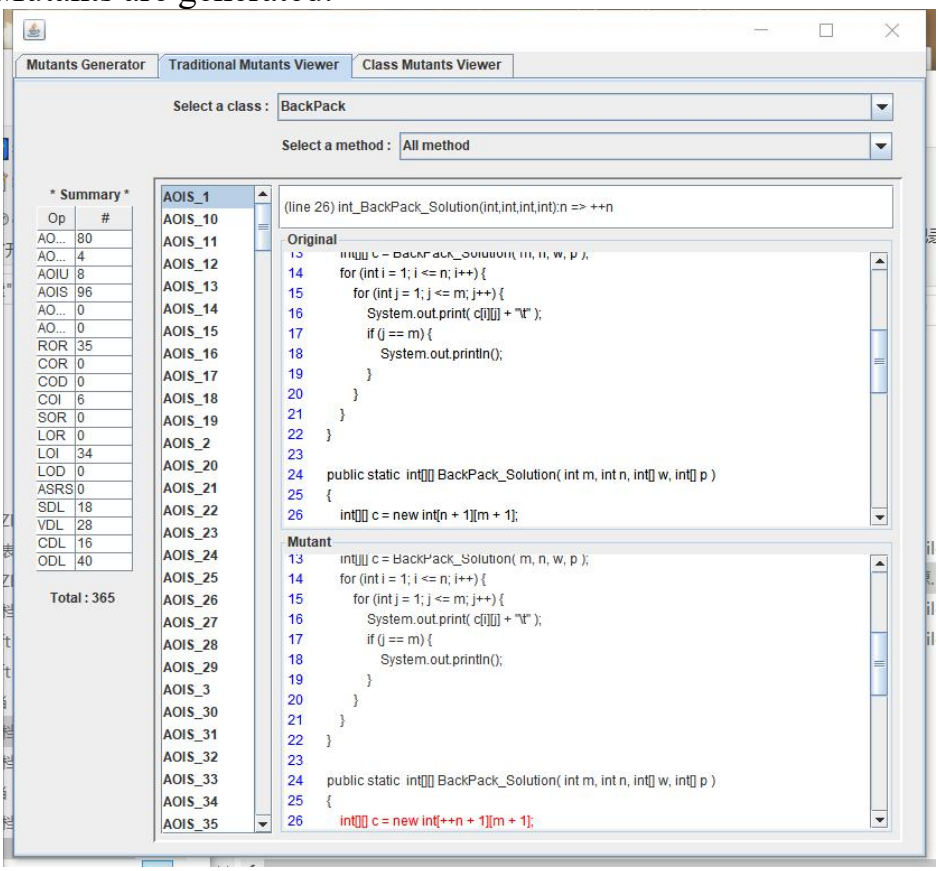
This command brings up a screen:



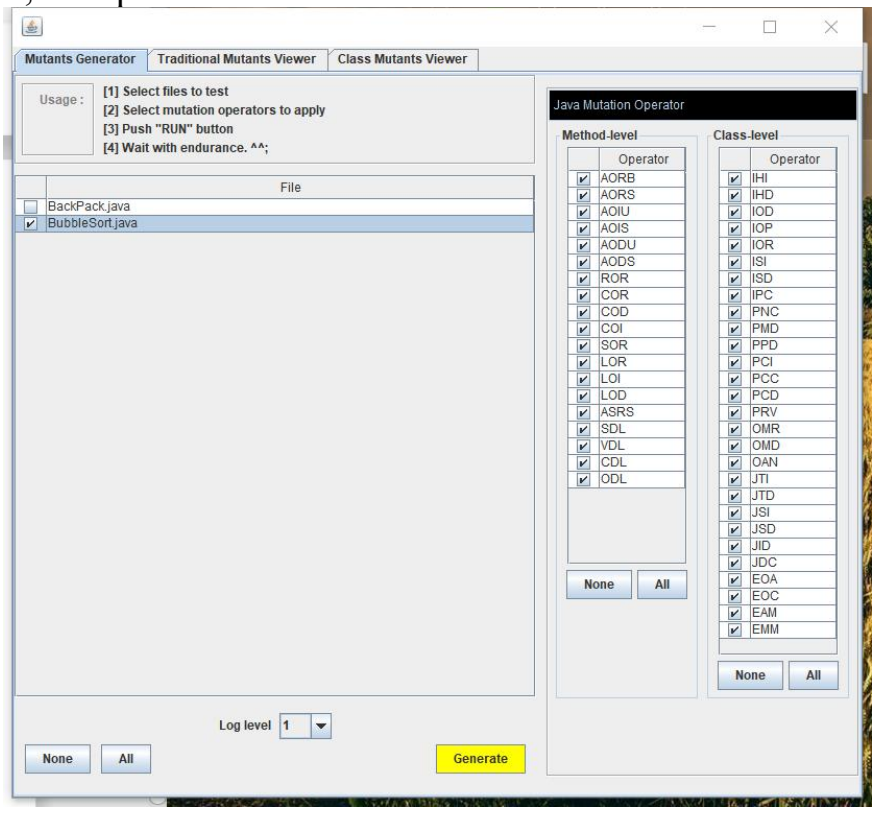
2.4 Select BackPack.java to mutate and select all mutation operators, then push RUN:



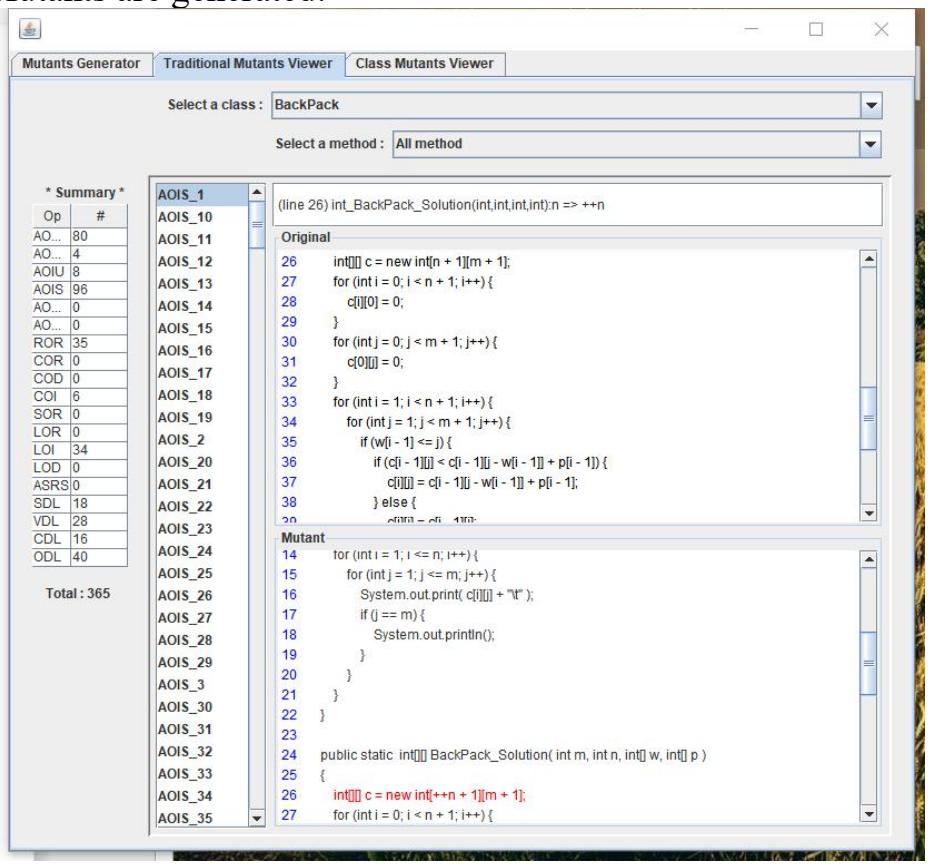
Mutants are generated:



2.5 Select BubbleSort.java to mutate and select all mutation operators, then push RUN:

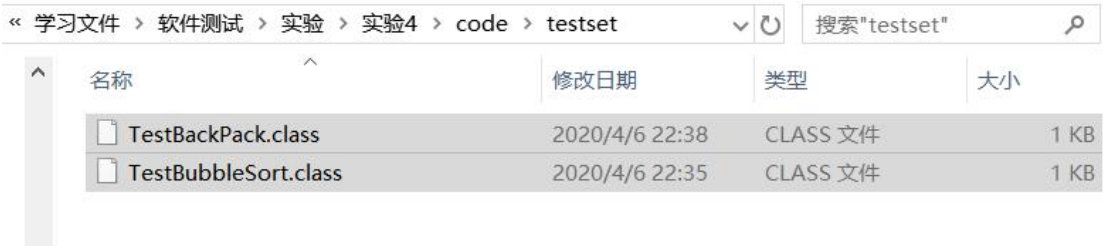


Mutants are generated:



3. Making a test set

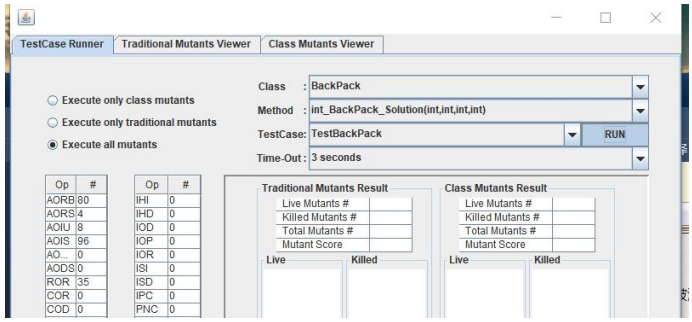
3.1 Make JUnit test case for these two file, compile, and move them to the directory MuJava_HOME\testset\.



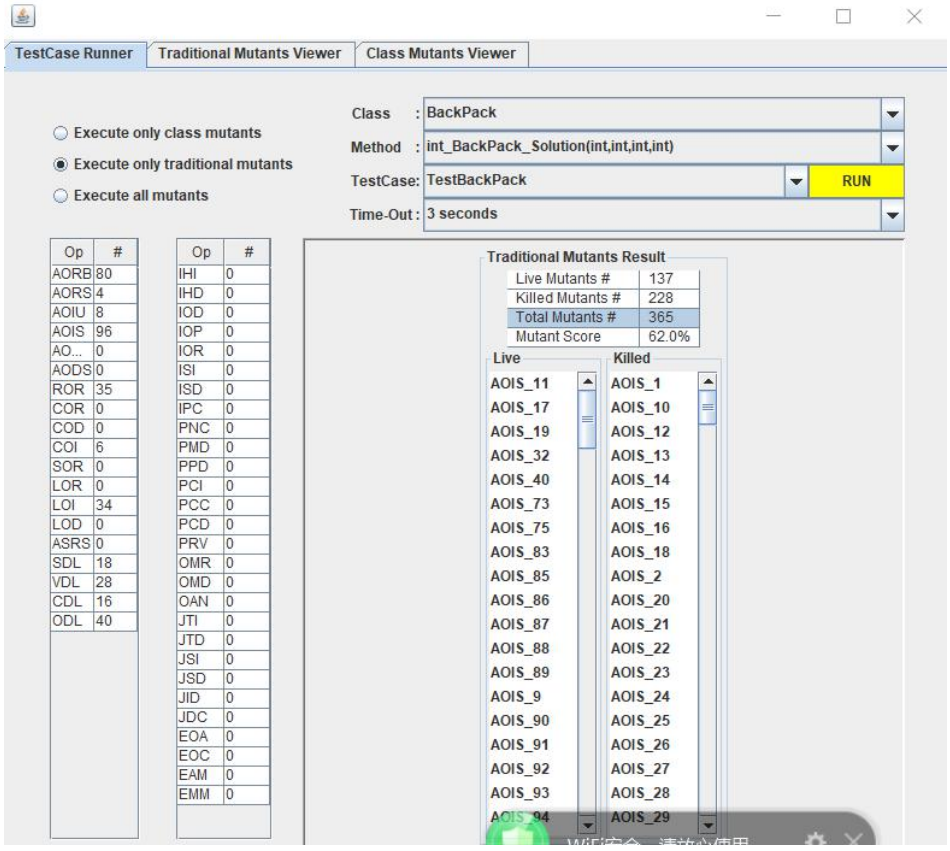
4. Run mutants.

```
F:\>cd F:\学习文件\软件测试\实验\实验4\code

F:\学习文件\软件测试\实验\实验4\code>java mujava.gui.RunTestMain
```

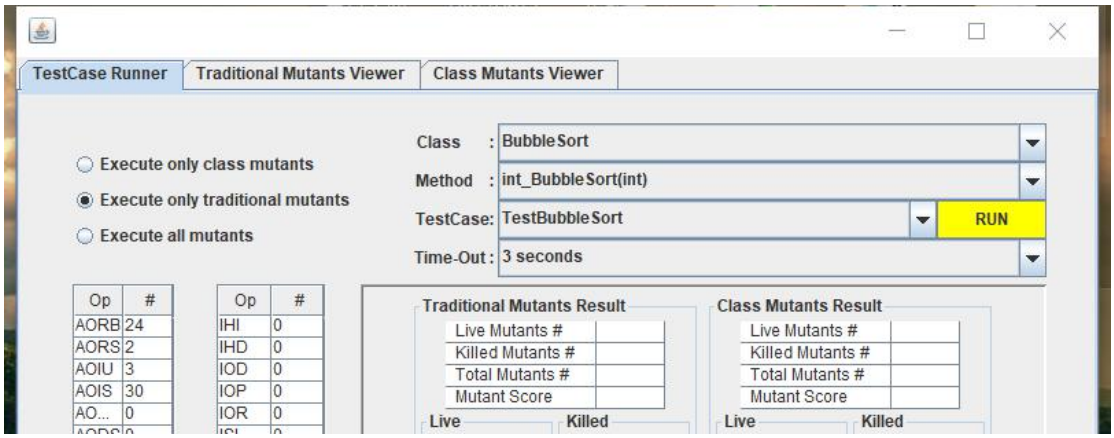
4.1 Run test case for BackPack:



Results:



4.2 Run test case for BubbleSort:



Results: