****

**软件测试技术第四次实验报告**



**学 院 智能与计算学部**

**专 业 软件学院**

**年 级 2016**

**姓 名 岳明宇**

**2019年 04 月 21 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）
2. Install MuJava. The instruction of how to install and use Mujava can be seen in <https://cs.gmu.edu/~offutt/mujava/> .
3. 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.
4. Write testing sets for 2 programs with Junit, and run mutants on the test sets with MuJava.
5. Install MuJava

1.将老师给的压缩包打开，将其中的jar包放入一个文件夹中，并将jar包的路径加入classpath中。

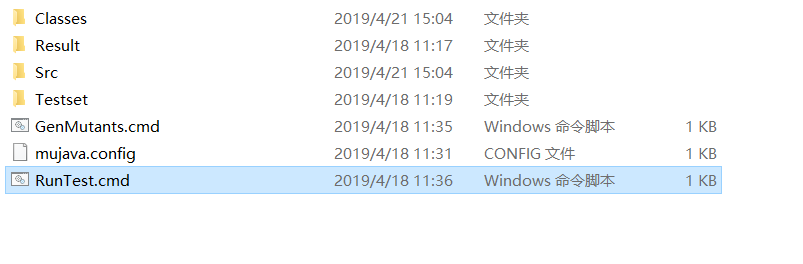


2.另外创建一个文件夹，在其中创建四个文件夹，分别为testset，src，classes，result。另外创建一个配置文件，为mujava.config，里面写上该文件夹的地址，另外编写两个cmd文件，为GenMutants.cmd和RunTest.cmd，用于运行变体产生和运行测试的命令。





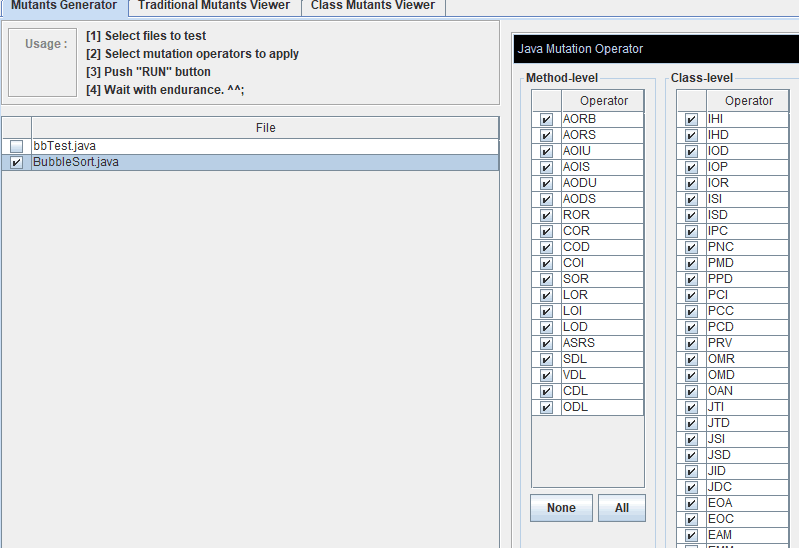
文件夹截图



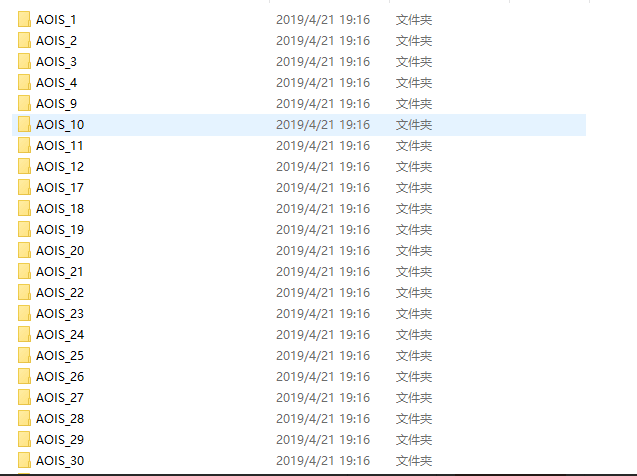
1. 详细设计

1.BubbleSort.java

将BubbleSort.java文件通过javac编译得到BubbleSort.class文件，将java文件放入src中，将class文件放入classes中，用GenMutants.cmd启动，得到程序界面



选择所有，产生变异结果



编写测试用例

import static org.junit.Assert.\*;

import org.junit.Test;

import java.util.Arrays;

public class BubbleSortTest {

@Test

public void SortTest() {

assertEquals("[1, 2, 3, 4, 5]",Arrays.toString(BubbleSort.BubbleSort(new int[]{1,3,2,4,5})));

assertEquals("[1, 2, 3, 4, 4, 5, 6, 9]",Arrays.toString(BubbleSort.BubbleSort(new int[]{1,3,2,4,5,6,4,9})));

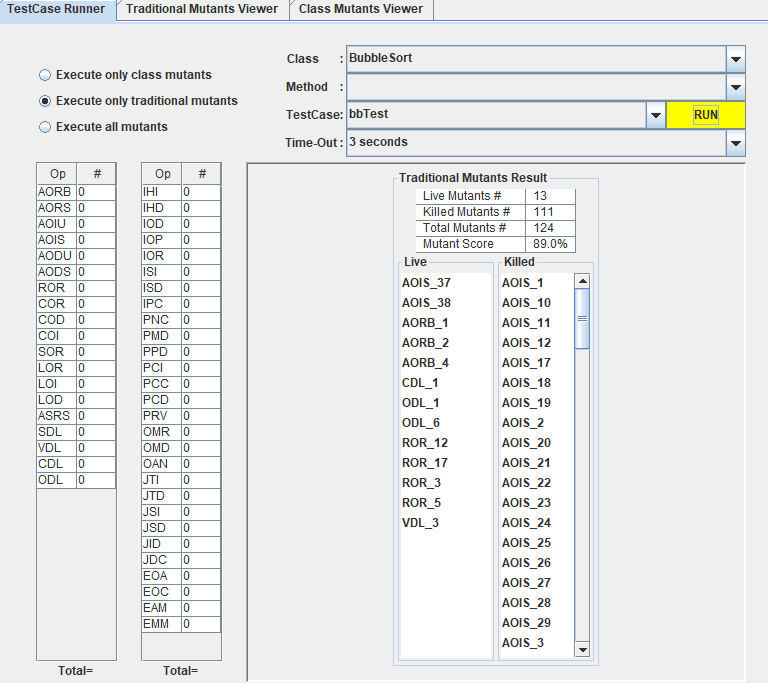
assertEquals("[0, 1, 2, 3, 4, 5]",Arrays.toString(BubbleSort.BubbleSort(new int[]{1,3,2,4,5,0})));

assertEquals("[-1, 1, 2, 3, 4, 5]",Arrays.toString(BubbleSort.BubbleSort(new int[]{1,3,2,4,5,-1})));

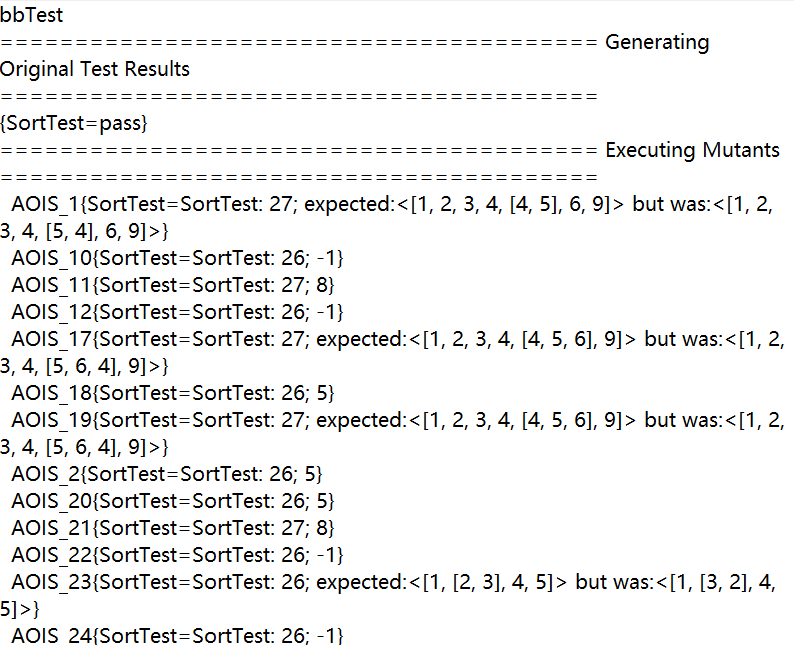
}

}

使用javac命令编译，将这两个文件放入testset中，运行RunTest.cmd，点击run命令，得到结果：

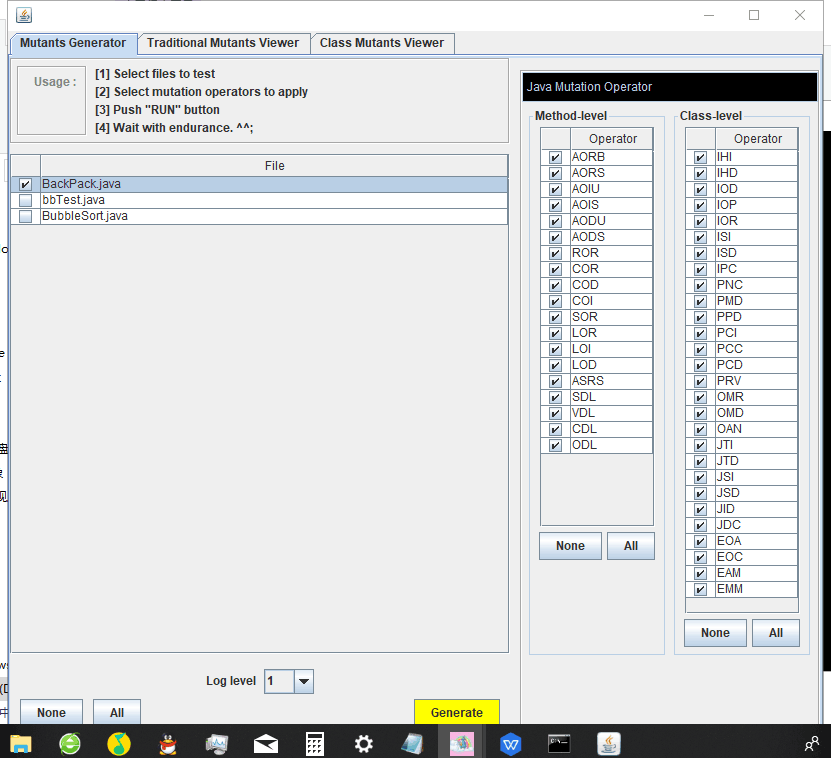


结果

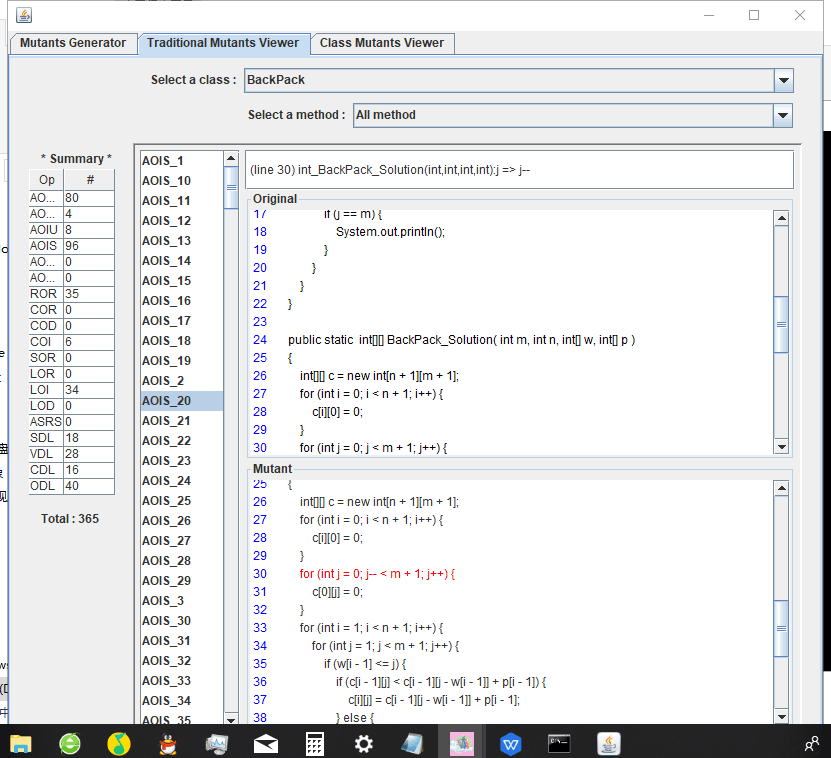


2.BackPack.java

与前一个实验类似，将BackPack.java通过javac命令编译成.class文件，分别将.java文件跟.class文件放入src文件夹及classes文件夹中，打开GenMutants.cmd，运行



得到mutants



编写测试用例：

import static org.junit.Assert.assertEquals;

import java.util.Arrays;

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

public class bpTest {

private BackPack b;

int m = 10;

int n =3;

int w[] = {3, 4, 5};

int p[] = {4, 5, 6};

int c[][] = {{0},{0,0,0,4,4,4,4,4,4,4,4},{0,0,0,4,5,5,5,9,9,9,9},{0,0,0,4,5,6,6,9,10,11,11}};

@Before

public void setup() throws Exception{

b = new BackPack();

}

@After

public void down() throws Exception{

b = null;

}

@Test

public void SortTest() {

for(int i =1;i<=n;i++) {

for(int j =1;j<=m;j++) {

assertEquals(c[i][j],BackPack.BackPack\_Solution(m,n,w,p)[i][j]);

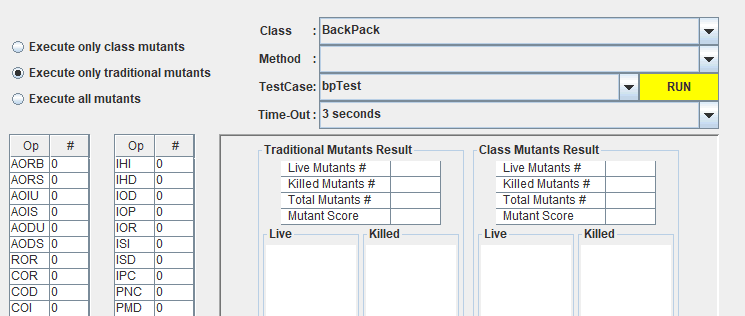
}

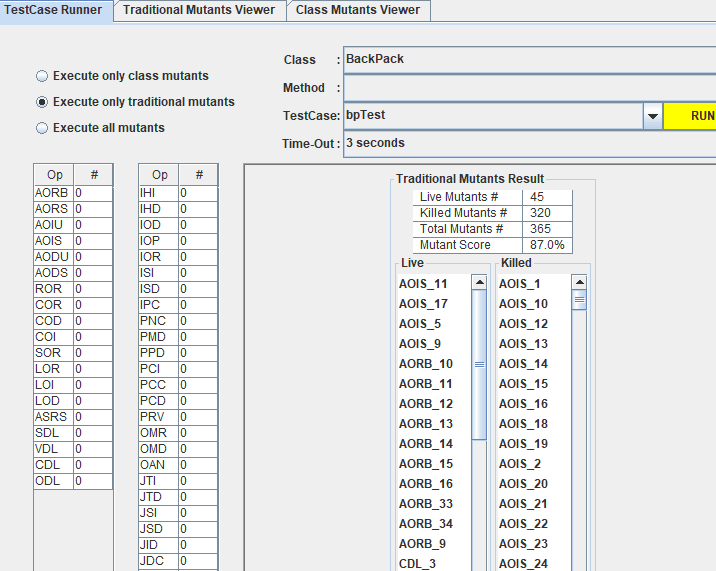
}

}

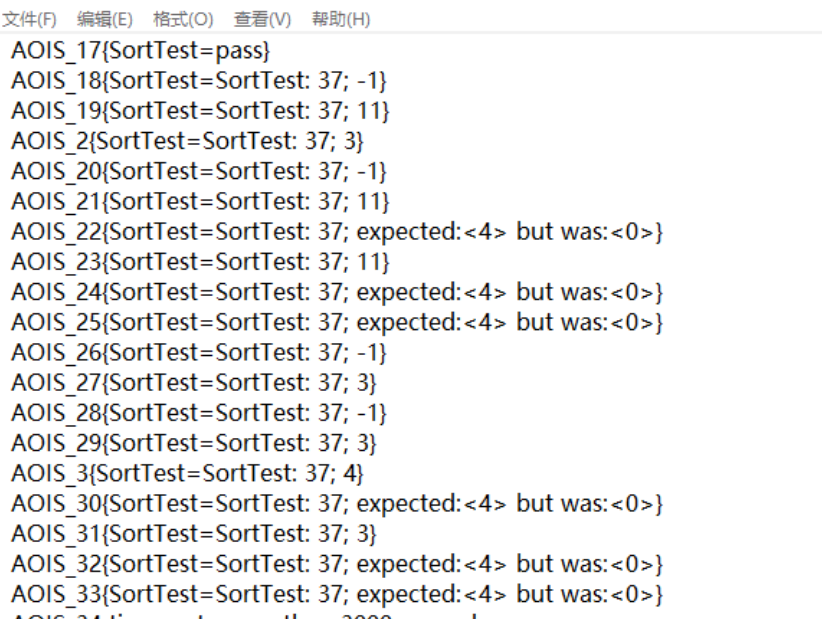
}

运行RunTest.cmd





得到结果



1. 总结

实验通过mujava程序对某一程序产生mutants并能够用测试人员编写好的测试用例对mutants判定，得到live及killed mutants，最后生成测试报告。生成并测试mutants的过程能帮助我们编写足够的测试用例开发软件，能够帮助我们找到程中的错误并改正。