

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__陈姝宇__

学 号__3017218119__

年 级__2017 级__

班 级__3 班__

1. Experimental Requirements

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.

Requirements for the experiment:

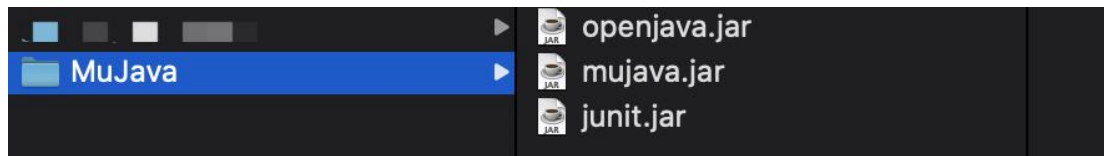
1. Finish the tasks above individually.
2. Check in your java code to github or gitee.
3. Post your experiment report to “智慧树”, the following information should be included in your report:
 - a) The brief description that you install MuJava
 - b) Steps for generating Mutants
 - c) Steps for making test sets and running mutants.
 - d) Your mutants result (The number of live mutants, killed mutants, etc.)

2. Environmental configuration

The brief description that I install MuJava:

- 1) Put the downloaded MuJava related files in a folder.

The file directory structure is:



2) Vim ~/.bash_profile and add 3 paths to the classpath:

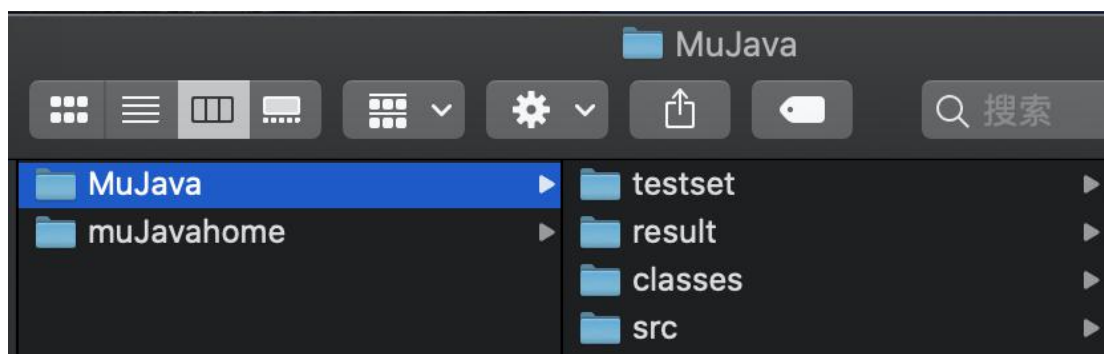
```
chenshuyu@chenshuyudeMacBook-Pro-2 ~$ vim ~/.bash_profile

JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Contents/Home/
PATH=$PATH:$JAVA_HOME/bin
CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar:/Users/chenshuyu/
Documents/MuJava/mujava.jar:/Users/chenshuyu/Documents/MuJava/junit.jar:/Use
rs/chenshuyu/Documents/MuJava/openjava.jar
export JAVA_HOME PATH CLASSPATH
```

3) Create mujava.config file in this folder, and insert mujava_home = /Users/chenshuyu/Documents/java/mujavahome into the file;

```
3. vim mujava.config (vim)
MuJava_HOME=/Users/chenshuyu/Documents/MuJava\ /muJavahome
```

4) Use java mujava.makeMuJavaStructure to create the following directory structure under mujavahome folder:

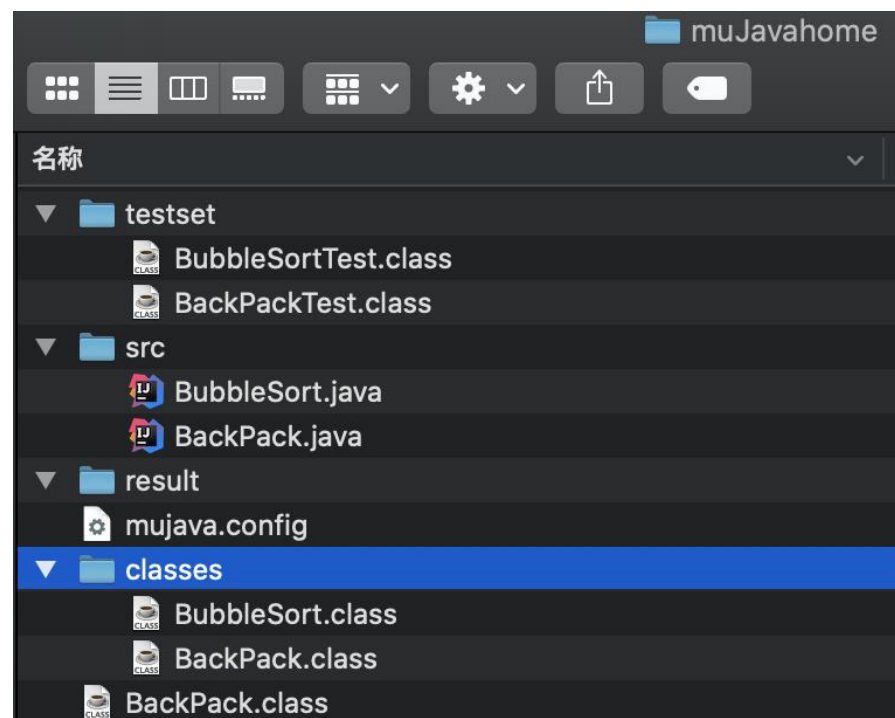
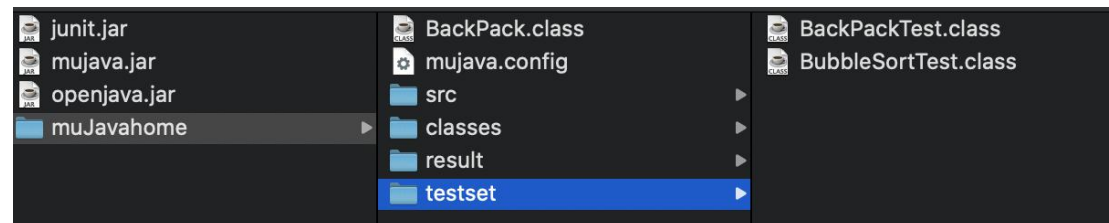


Folder means:

Directory	Explain
src	Java program source code to be tested
classes	The .class file generated by the correct compilation of the source code in SRC
testset	.class file of testset JUnit test case
result	Result generate the result file of the variant

Before using mujava, compile all .java file and put .class into file which they should be.

```
~/Documents/MuJava ➤ javac Backpack.java BackpackTest.java BubbleSort.java BubbleSortTest.java  
~/Documents/MuJava
```



3. BubbleSort.java

3.1 Generating Mutants

1) Use command to run GenMutants. Choose file and method. Click generate.

```
~/Documents/java/MuJava ➤ java mujava.gui.GenMutantsMain
```

Mutants Generator

Traditional Mutants Viewer

Class Mutants Viewer

Select a class : BubbleSort

Select a method : int_BubbleSort(int)

* Summary *

Op	#
AO...	24
AORS	2
AOIU	3
AOIS	30
AO...	0
AO...	0
ROR	18
COR	0
COD	0
COI	3
SOR	0
LOR	0
LOI	12
LOD	0
ASRS	0
SDL	9
VDL	6
CDL	5
ODL	12

Total : 124

ROR_8

AOIS_21

AOIS_19

AOIS_26

AOIS_10

AOIS_28

SDL_10

AOIS_17

ROR_1

ODL_7

ODL_9

AOIS_29

SDL_11

AOIS_11

AOIS_18

AOIS_27

ROR_9

AOIS_20

ODL_8

ODL_1

ODL_6

AOIU_3

LOI_2

ROR_10

AORB_5

ROR_17

LOI_5

AORB_2

LOI_12

AOIS_1

ROR_21

ROR_19

ODL_13

ODL_14

(line 14) int_BubbleSort(int): j < arr.length - i - 1 => j > arr.length - i - 1

Original

```

14   for (int j = 0; j < arr.length - i - 1; j++) {
15       if (arr[j + 1] < arr[j]) {
16           temp = arr[j];
17           arr[j] = arr[j + 1];
18           arr[j + 1] = temp;
19       }
20   }
21   }
22   return arr;
23   }
24   }
25   public static void main( java.lang.String[] args )
26   {
27       int[] arr = new int[]{ 1, 6, 2, 2, 5 };
28       BubbleSort.BubbleSort( arr );
29   }

```

Mutant

```

1 // This is a mutant program.
2 // Author : ysm
3
4 import java.util.Arrays;
5
6
7 public class BubbleSort
8 {
9
10    public static int[] BubbleSort( int[] arr )
11    {
12        int temp;
13        for (int i = 0; i < arr.length - 1; i++) {
14            for (int j = 0; j > arr.length - i - 1; j++) {
15                if (arr[j + 1] < arr[j]) {
16                    temp = arr[j];
17                    arr[j] = arr[j + 1];
18                    arr[j + 1] = temp;
19                }
20            }
21        }
22        return arr;
23    }
24 }

```

Mutants Generator

Traditional Mutants Viewer

Class Mutants Viewer

Select a class : BubbleSort

* Summary *

Op	#
IHI	0
IHD	0
IOD	0
IOP	0
IOR	0
ISI	0
ISD	0
IPC	0
PNC	0
PMD	0
PPD	0
PCI	0
PCC	0
PCD	0
PRV	0
OMR	0
OMD	0
OAN	0
JTI	0
JTD	0
JSI	0
JSD	0
JID	0
JDC	0
EOA	0
EOC	0
EAM	0
EMM	0

Total : 0

Original

```

19       arr[j + 1] = temp;
20   }
21   }
22   return arr;
23   }
24   }
25   public static void main( java.lang.String[] args )
26   {
27       int[] arr = new int[]{ 1, 6, 2, 2, 5 };
28       BubbleSort.BubbleSort( arr );
29       System.out.println( Arrays.toString( arr ) );
30   }
31   }
32   }

```

Mutant

3.2 Making test sets and running mutants

[BubbleSortTest.Java]

There expected array, actual array and wrong array.

In test, equal and notsame assertion are used.

```
public class BubbleSortTest2 {

    private int[] expected;
    private int[] actual;
    private int[] wrong;

    @Before
    public void before() {
        int length = 10;
        int max=100,min = 0;

        // generate the actual array
        actual = new int[length];
        expected = new int[length];
        wrong = new int[length];

        for (int i = 0; i < length; ++i) {
            int num = (int) (Math.random() * (max - min + min));
            actual[i] = num;
            expected[i] = num;
            wrong[i] = num;
        }

        Arrays.sort(expected);
        Arrays.sort(wrong);

        for (int i = 1; i < length; ++i) {
```

```

        if(wrong[i-1]!=wrong[i]){
            int temp = wrong[i-1];
            wrong[i-1] = wrong[i];
            wrong[i] = temp;
            break;
        }
    }
}

@After
public void after() {

}

@Test
public void bubbleSortTest() {

Assert.assertEquals(this.expected,BubbleSort.BubbleSort(this.actual))
;

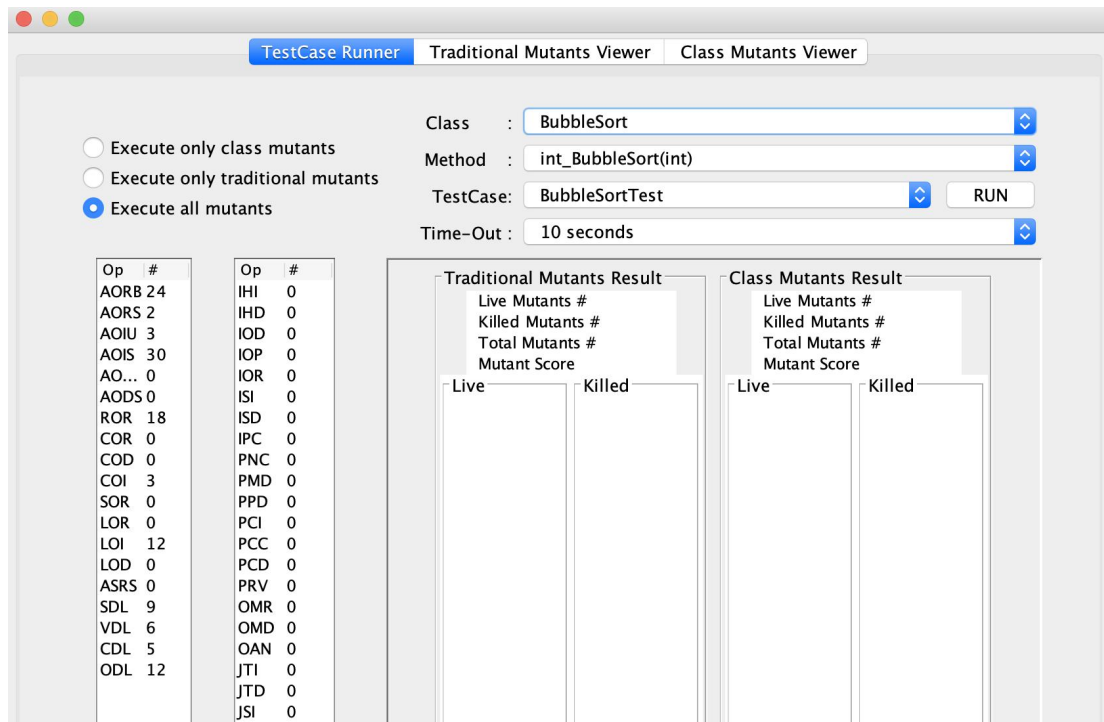
    Assert.assertNotSame(Arrays.toString(this.wrong),
Arrays.toString(BubbleSort.BubbleSort(this.actual)));

}
}

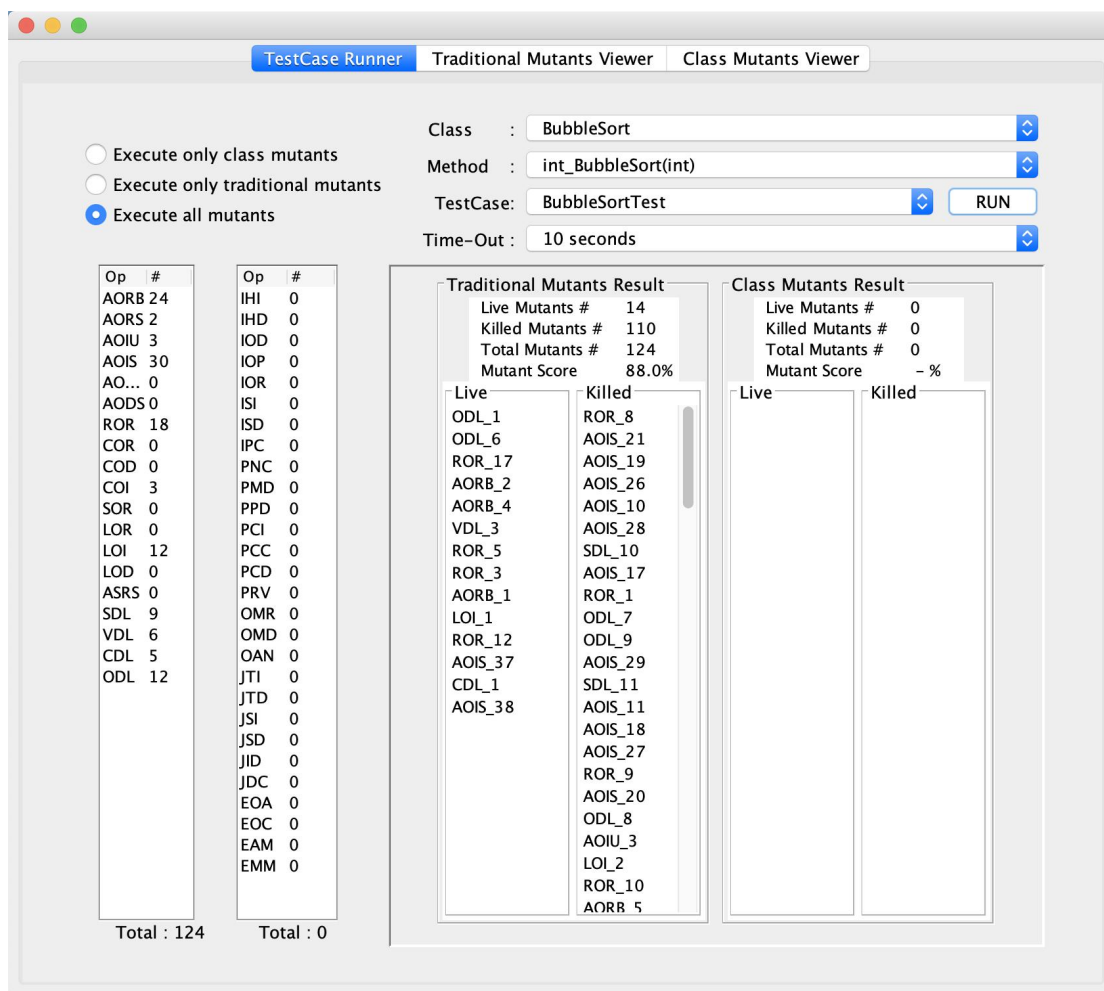
```

Run test and click RUN.

```
~/Documents/MuJava/muJavahome ➤ java mujava.gui.RunTestMain
```



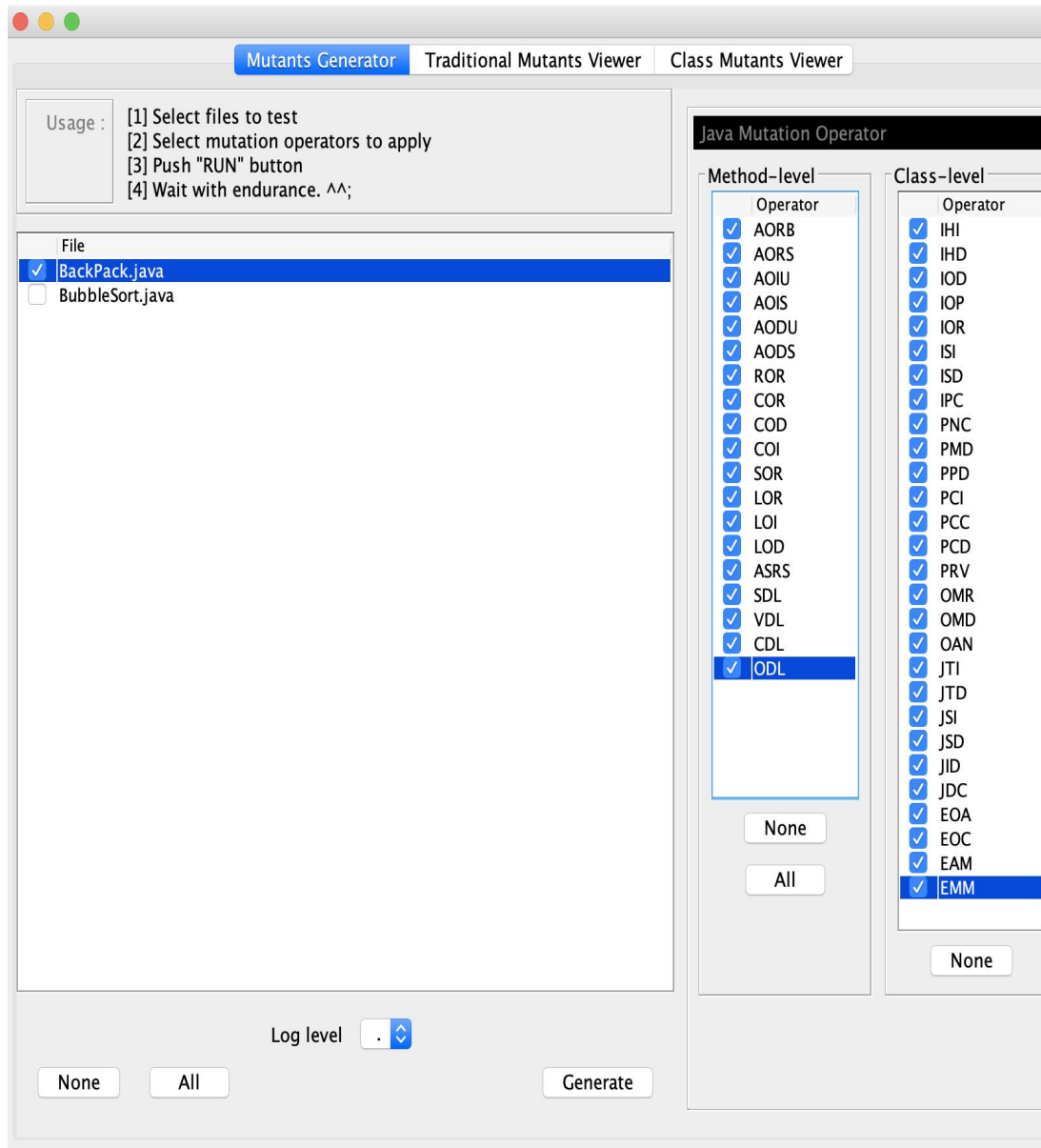
3.3 Mutants Result



4. Backpack.java

4.1 Generating Mutants

1) Choose file and method. Click generate.



2) View Traditional Viewer and Class Mutants Viewer.

Mutants Generator

Traditional Mutants Viewer

Class Mutants Viewer

Select a class : Backpack

Select a method : int_BackPack_Solution(int,int,int,int)

* Summary *

Op	#
AO...	80
AORS	4
AOIU	8
AOIS	96
AO...	0
AO...	0
ROR	35
COR	0
COD	0
COI	6
SOR	0
LOR	0
LOI	34
LOD	0
ASRS	0
SDL	18
VDL	28
CDL	16
ODL	40

Total : 365

SDL_19

ROR_8

AOIS_21

AOIS_19

SDL_21

AOIS_26

AOIS_10

SDL_17

AOIS_28

SDL_10

AOIS_17

ROR_1

AOIS_43

AOIS_88

AOIS_44

AOIS_86

AOIS_72

AOIS_75

AOIS_81

AOIS_29

AOIS_16

AOIS_11

SDL_16

AOIS_18

SDL_20

AOIS_27

ROR_9

SDL_18

AOIS_20

AOIS_80

AOIS_74

AOIS_73

AOIS_87

AOIS_45

(line 10) int_BackPack_Solution(int,int,int,int): i < n + 1 ==> i > n + 1

Original

```

10 for (int i = 0; i < n + 1; i++) {
11     c[i][0] = 0;
12 }
13 for (int j = 0; j < m + 1; j++) {
14     c[0][j] = 0;
15 }
16 for (int i = 1; i < n + 1; i++) {
17     for (int j = 1; j < m + 1; j++) {
18         if (w[i - 1] <= j) {
19             if (c[i - 1][j] < c[i - 1][j] - w[i - 1] + p[i - 1]) {
20                 c[i][j] = c[i - 1][j] - w[i - 1] + p[i - 1];
21             } else {
22                 c[i][j] = c[i - 1][j];
23             }
24         } else {
25             c[i][j] = c[i - 1][j];
26         }
27     }
28 }

```

Mutant

```

1 // This is a mutant program.
2 // Author : ysma
3
4 public class Backpack
5 {
6
7     public static int[][] Backpack_Solution( int m, int n, int[] w, int[] p )
8     {
9         int[][] c = new int[n + 1][m + 1];
10         for (int i = 0; i > n + 1; i++) {
11             c[i][0] = 0;
12         }
13         for (int j = 0; j < m + 1; j++) {
14             c[0][j] = 0;
15         }
16     }
17 }

```

Mutants Generator

Traditional Mutants Viewer

Class Mutants Viewer

Select a class : BubbleSort

* Summary *

Op	#
IHI	0
IHD	0
IOD	0
IOP	0
IOR	0
ISI	0
ISD	0
IPC	0
PNC	0
PMD	0
PPD	0
PCI	0
PCC	0
PCD	0
PRV	0
OMR	0
OMD	0
OAN	0
JTI	0
JTD	0
JSI	0
JSD	0
JID	0
JDC	0
EOA	0
EOC	0
EAM	0
EMM	0

Total : 0

Original

```

12 int temp;
13 for (int i = 0; i < arr.length - 1; i++) {
14     for (int j = 0; j < arr.length - i - 1; j++) {
15         if (arr[j + 1] < arr[j]) {
16             temp = arr[j];
17             arr[j] = arr[j + 1];
18             arr[j + 1] = temp;
19         }
20     }
21 }
22 return arr;
23 }
24
25 }

```

Mutant

4.2 Making test sets and running mutants

[BackPackTest.java]

```
public class BackPackTest {  
    int m; // capacity  
    int n; // number  
    int w[]; // weight  
    int p[]; // value  
    int expected[][];  
    int wrong[][];  
    @Before  
    public void before() {  
        m = 10;  
        n = 3;  
        w = new int[]{3, 4, 5}; // weight  
        p = new int[]{4, 5, 6}; // value  
        expected = new int[][]{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 4, 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}};  
        wrong = new int[][]{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 3, 4, 4, 4,  
4, 4, 4, 4, 4}, {0, 0, 0, 3, 5, 5, 5, 9, 9, 9, 9}, {0, 0, 0, 3, 5, 6, 6, 9, 10, 11, 11}};  
    }  
  
    @After  
    public void after() {  
    }  
  
    private String toString(int array[]){  
        StringBuilder sb = new StringBuilder();  
        for (int[] line:array){  
            sb.append(Arrays.toString(line));  
        }  
        return sb.toString();  
    }  
}
```

```

}

@Test
public void bubbleSortTest() {
    String actualS = toString(BackPack.BackPack_Solution(m,n,w,p));
    Assert.assertEquals(toString(this.expected), actualS);
    Assert.assertNotSame(toString(this.wrong), actualS);
}
}

```

Rerun test and click RUN.

```
~/Documents/MuJava/muJavahome ➤ java mujava.gui.RunTestMain
```

4.3 Mutants Result

The screenshot shows the MuJava GUI with the 'Test Case Runner' tab selected. The interface includes fields for Class, Method, Test Case, and Time-Out, along with a 'RUN' button. Below these fields are two tables showing mutant results for the 'BackPack' class.

Test Case Runner Configuration:

- Class: BackPack
- Method: int_BackPack_Solution(int,int,int,int)
- Test Case: BackPackTest
- Time-Out: 10 seconds
- Execute all mutants (selected)

Mutant Results:

Op	#	Op	#
AORB	80	IHI	0
AORS	4	IHD	0
AOIU	8	IOD	0
AOIS	96	IOP	0
AO...	0	IOR	0
AODS	0	ISI	0
ROR	35	ISD	0
COR	0	IPC	0
COD	0	PNC	0
COI	6	PMD	0
SOR	0	PPD	0
LOR	0	PCI	0
LOI	34	PCC	0
LOD	0	PCD	0
ASRS	0	PRV	0
SDL	18	OMR	0
VDL	28	OMD	0
CDL	16	OAN	0
ODL	40	JTI	0
		JTD	0
		JSI	0
		JSD	0
		JID	0
		JDC	0
		EOA	0
		EOC	0
		EAM	0
		EMM	0

Total : 365

Traditional Mutants Result		Class Mutants Result	
Live Mutants #	365	Live Mutants #	0
Killed Mutants #	0	Killed Mutants #	0
Total Mutants #	365	Total Mutants #	0
Mutant Score	0.0%	Mutant Score	- %

The 'Live' column in the 'Traditional Mutants Result' table contains a list of mutants: SDL_19, ROR_8, AOIS_21, AOIS_19, SDL_21, AOIS_26, AOIS_10, SDL_17, AOIS_28, SDL_10, AOIS_17, ROR_1, AOIS_43, AOIS_88, AOIS_44, AOIS_86, AOIS_72, AOIS_75, AOIS_81, AOIS_29, AOIS_16, AOIS_11, and SDI_16.

5. Exception and Solution

5.1 Error for class

Exception

```
chenshuyu@chenshuyudeMacBook-Pro-2 ~/Documents/MuJava/muJavahome
java mujava.gui.GenMutantsMain
The main method starts
[ERROR] for class BubbleSort => BubbleSort (wrong name: xyz/chenshuyu/bubbleSort/BubbleSort)
```

Reason:

Cannot use .class file which compile by IDE.

Solution:

Compile .java file by javac in terminal.

```
~/Documents/MuJava javac BubbleSort.java
```

5.2 Cannot find symbol

Exception

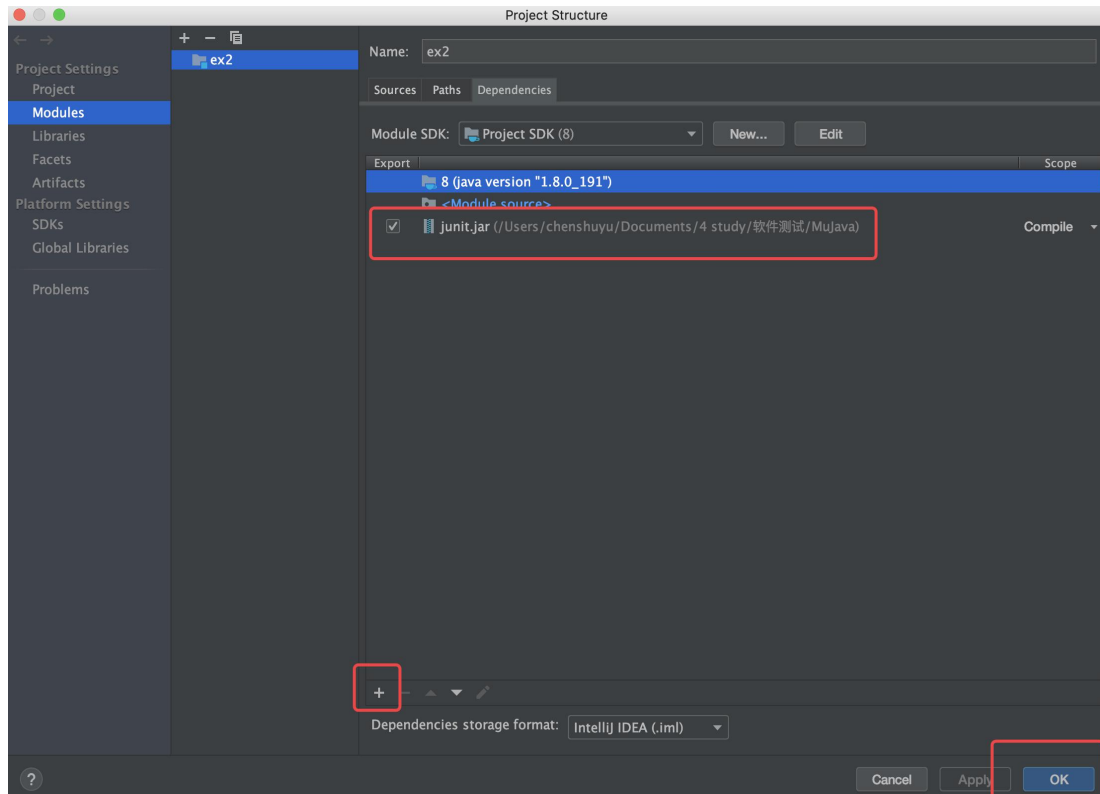
```
yu/bubbleSort master ➤ javac BubbleSortTest.java
BubbleSortTest.java:75: 错误: 找不到符号
    String actualS = Arrays.toString(BubbleSort.BubbleSort(this.actual));
                                   ^
  符号:   变量 BubbleSort
  位置: 类 BubbleSortTest
BubbleSortTest.java:77: 错误: 找不到符号
    Assert.assertEquals(Arrays.toString(this.wrong), actualS);
    ^
  符号:   方法 assertEquals(String,String)
  位置: 类 Assert
2 个错误
```

Reason:

1. JUnit version in my IDE is different from JUnit version set in class home.
2. BubbleSortTest.java is supposed to be compiled with BubbleSort.java.

Solution:

1. Reset JUnit in my IDE. And change methon in BubbleSortTest.java and BackPackTest.java



Assert.assertEquals ---> Assert.assertArrayEquals

Assert.assertNotEquals ---> Assert.assertNotSame

```
@Test
public void bubbleSortTest() {
    String actualS = toString(BackPack.BackPack_Solution(m,n,w,p));
    Assert.assertEquals(toString(this.expected), actualS);
    Assert.assertNotSame(toString(this.wrong), actualS);
}
```

```
@Test
public void bubbleSortTest() {
    Assert.assertArrayEquals(this.expected, BubbleSort.BubbleSort(this.actual));
    Assert.assertNotSame(Arrays.toString(this.wrong), Arrays.toString(BubbleSort.BubbleSort(this.actual)));
}
```

2. Recompile file

```
/MuJava/muJavahome ➔ javac BubbleSort.java BubbleSortTest.java
```


5.3 NullPointerException

Exception

```
chenshuyu@chenshuyudeMacBook-Pro-2 ~/Documents/MuJava/muJavahome java mujava.gu
java.lang.InstantiationException: BubbleSortTest
java.lang.NullPointerException
    at mujava.TestExecuter.computeOriginalTestResults(TestExecuter.java:250)
    at mujava.gui.RunTestPanel.testRunB_mouseClicked(RunTestPanel.java:461)
    at mujava.gui.RunTestPanel$3.mouseClicked(RunTestPanel.java:235)
    at java.awt.AWTEventMulticaster.mouseClicked(AWTEventMulticaster.java:270)
    at java.awt.Component.processMouseEvent(Component.java:6542)
    at javax.swing.JComponent.processMouseEvent(JComponent.java:3324)
    at java.awt.Component.dispatchEvent(Component.java:6304)
    at java.awt.Container.dispatchEvent(Container.java:2239)
    at java.awt.Component.dispatchEventImpl(Component.java:4889)
    at java.awt.Container.dispatchEventImpl(Container.java:2297)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4904)
    at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4544)
    at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4476)
    at java.awt.Container.dispatchEventImpl(Container.java:2283)
    at java.awt.Window.dispatchEventImpl(Window.java:2746)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:760)
    at java.awt.EventQueue.access$500(EventQueue.java:97)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:778)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:138)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:131)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:123)
    at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)
No mutants have been generated for the class BubbleSort
[Exception 2]java.lang.InstantiationException: BubbleSortTest
```

Reason:

Use parameterized

Solution:

Modify BubbleSortTest.java without parameterized.

```
@RunWith(Parameterized.class)
public class BubbleSortTest {
```

```
public class BubbleSortTest2 {
```