天津大学

软件测试技术第三次实验报告

学　　院 智能与计算学部

专　　业 软件工程

年　　级 2016

姓　　名 张环禹

2019 年 4 月 18 日

软件测试技术第三次实验报告

一、 需求分析

1. 安装 MuJava。
2. 生成 BubbleSort.java 和 BackPack.java 的变异测试文件。
3. 编写测试类并使用 MuJava 进行测试。

二、 概要设计

1. 根据官网介绍安装 Mujava。
2. 修改 mujava.config，产生出 Mujava Structure。
3. 使用命令行产生变异测试文件。
4. 进行 Junit 测试并生成结果，并进行结果分析。

三、 详细设计

在 CLASSPATH 里添加 Mujava 等工具的 jar 包，使得 SHELL 里能使用 Mujava 命令

```
zhanghuanyudeMacBook-Pro:lab4 zhanghuanyu$  cat ~/.bashrc
export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh"  # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion"  # This loads
 nvm bash_completion
alias work="cd ~/Desktop/Web/Server/work/"
alias redis-server="nohup /usr/local/redis-5.0.4/src/redis-server /usr/local/red
is-5.0.4/redis.conf  &"
alias redis-cli="/usr/local/redis-5.0.4/src/redis-cli"
alias mysql_start="/usr/local/Cellar/mysql/8.0.13/bin/mysql.server start"
alias mysql_stop="/usr/local/Cellar/mysql/8.0.13/bin/mysql.server stop"
export PATH=$PATH:/usr/local/Cellar/mysql/8.0.13/bin;
export PATH=$PATH:/sbin/;
CLASSPATH=$CLASSPATH:/Users/zhanghuanyu/Downloads/MuJava/junit.jar:/Users/zhangh
uanyu/Downloads/MuJava/mujava.jar:/Users/zhanghuanyu/Downloads/MuJava/openjava.j
ar;
export CLASSPATH;
```

查看 JDK 所在位置

```
zhanghuanyudeMacBook-Pro:lab4 zhanghuanyu$ echo $JAVA_HOME
/Library/Java/JavaVirtualMachines/jdk1.8.0_131.jdk/Contents/Home
```

修改产生 mujava.config

```
zhanghuanyudeMacBook-Pro:Lab4_Code zhanghuanyu$ cat mujava.config
MuJava_HOME=/Users/zhanghuanyu/Desktop/software_testing/Lab4/Lab4_Code
```

命令生成 Mujava 的 Structure

```
zhanghuanyudeMacBook-Pro:lab_structure zhanghuanyu$ java mujava.makeMuJavaStruct
ure

Make /Users/zhanghuanyu/Desktop/software_testing/lab_structure directory...
/Users/zhanghuanyu/Desktop/software_testing/lab_structure directory exists alrea
dy.

Make /Users/zhanghuanyu/Desktop/software_testing/lab_structure/src directory...
Making /Users/zhanghuanyu/Desktop/software_testing/lab_structure/src directory
...done.

Make /Users/zhanghuanyu/Desktop/software_testing/lab_structure/classes directory
...
Making /Users/zhanghuanyu/Desktop/software_testing/lab_structure/classes directo
ry  ...done.

Make /Users/zhanghuanyu/Desktop/software_testing/lab_structure/result directory.
..
Making /Users/zhanghuanyu/Desktop/software_testing/lab_structure/result director
y  ...done.

Make /Users/zhanghuanyu/Desktop/software_testing/lab_structure/testset directory
...
Making /Users/zhanghuanyu/Desktop/software_testing/lab_structure/testset directo
ry  ...done.
```

```
zhanghuanyudeMacBook-Pro:lab_structure zhanghuanyu$ ls -l
total 8
drwxr-xr-x  2 zhanghuanyu  staff  64  4 21 16:35 classes
-rw-r--r--  1 zhanghuanyu  staff  70  4 21 16:35 mujava.config
drwxr-xr-x  2 zhanghuanyu  staff  64  4 21 16:35 result
drwxr-xr-x  2 zhanghuanyu  staff  64  4 21 16:35 src
drwxr-xr-x  2 zhanghuanyu  staff  64  4 21 16:35 testset
```

编写测试类文件

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class BubbleSortTest {

    @Test
    public void bubbleSort() {
        int arr[] = new int[]{1,6,2,2,5};
        BubbleSort.BubbleSort(arr);
    }
}
```

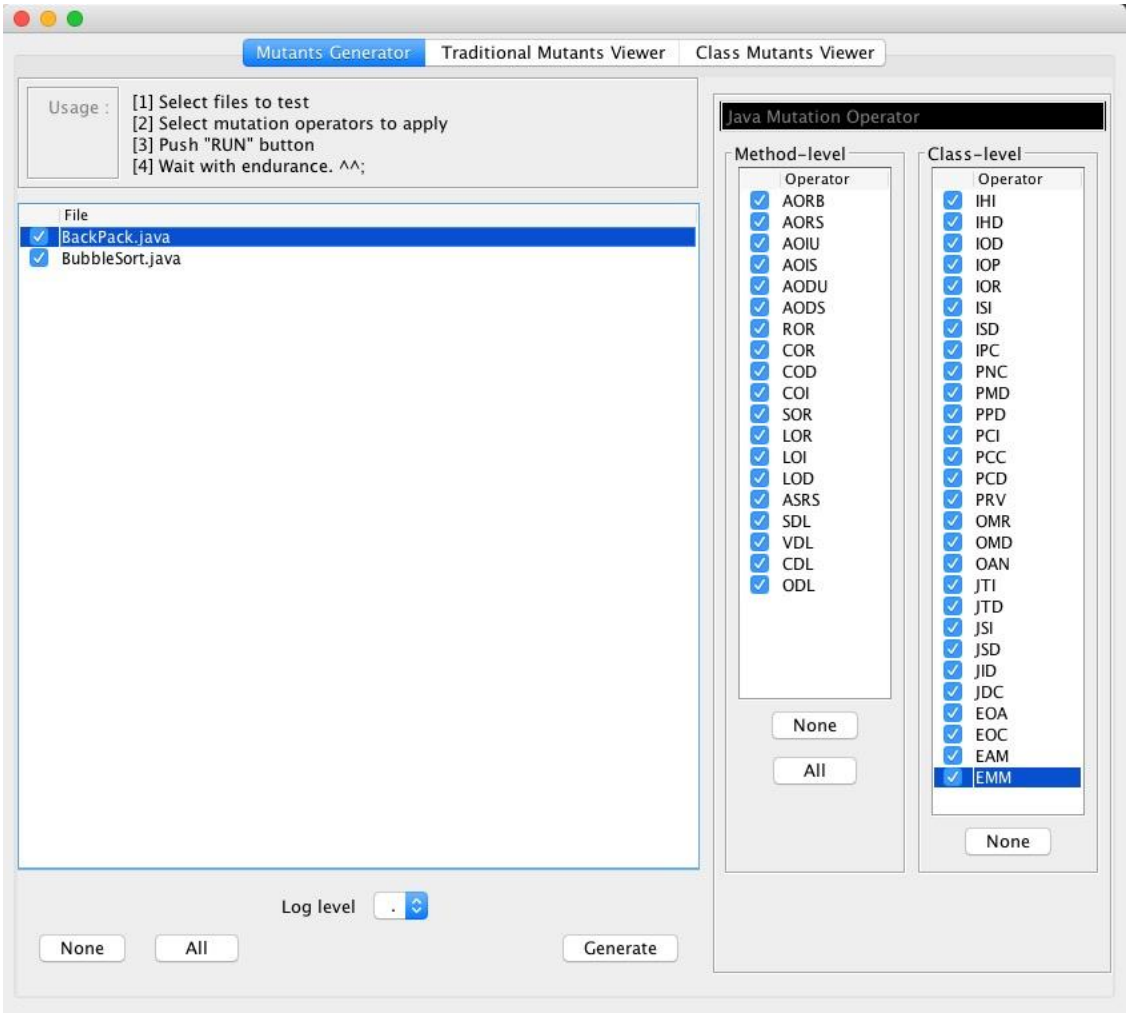
```java
import static org.junit.Assert.*;

public class BackPackTest {
    @org.junit.Test
    public void backPack_Solution() {
        int m = 10;
        int n = 3;
        int w[] = {3, 4, 5};
        int p[] = {4, 5, 6};
        BackPack.BackPack_Solution(m,n,w,p);
    }
}
```

编译所给的 java 类文件和测试类文件，并将 java 类文件放入 src 内，类文件的 class 文件放入 classes 内，测试类文件和测试类文件的 class 文件放入 testset 内。

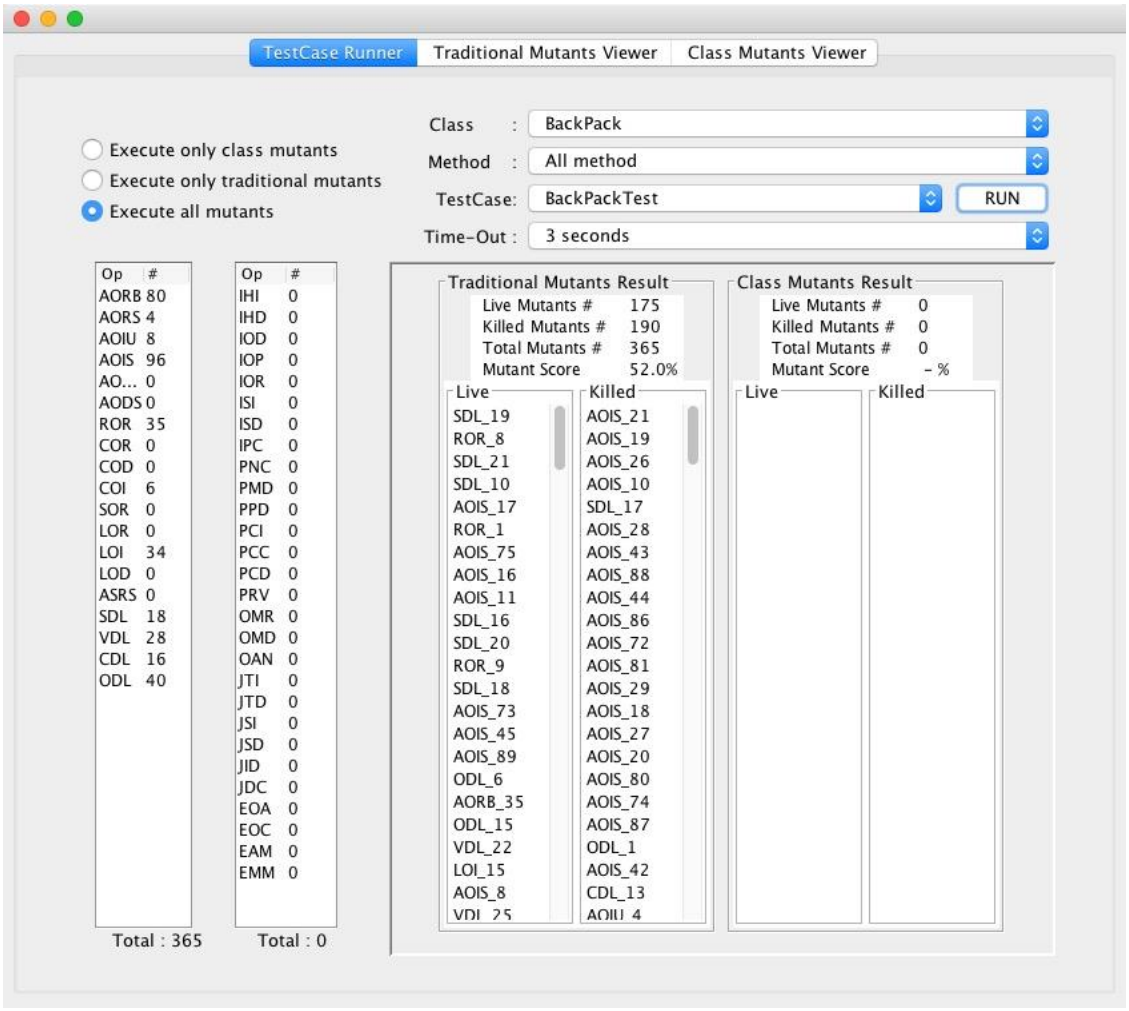
```
zhanghuanyudeMacBook-Pro:Lab4_Code zhanghuanyu$ ls -R classes
BackPack.class          BubbleSort.class
zhanghuanyudeMacBook-Pro:Lab4_Code zhanghuanyu$ ls -R testset
BackPackTest.class      BubbleSortTest.class
BackPackTest.java       BubbleSortTest.java
zhanghuanyudeMacBook-Pro:Lab4_Code zhanghuanyu$ ls -R src
BackPack.java   BubbleSort.java
```

接下来使用 mujava 进行产生变异测试文件




```
 -- BackPack.java class contains 'static void main()' method.
    Pleas note that mutants are not generated for the 'static void main()' metho
d
File /Users/zhanghuanyu/IdeaProjects/Lab4_2.0/src/BackPack.java
File /Users/zhanghuanyu/IdeaProjects/Lab4_2.0/src/BackPack.java
 -- BubbleSort.java class contains 'static void main()' method.
    Pleas note that mutants are not generated for the 'static void main()' metho
d
File /Users/zhanghuanyu/IdeaProjects/Lab4_2.0/src/BubbleSort.java
File /Users/zhanghuanyu/IdeaProjects/Lab4_2.0/src/BubbleSort.java
----------------------------------------------------------------------
All files are handled
```

接下来运行 mujava 进行不同变异测试文件并进行统计



四、 调试分析

使用 jdk11 运行 classes 文件下存在 class 文件的 mujava 结构报错

```
[zhanghuanyudeMacBook-Pro:Lab4_Code zhanghuanyu$ ./GenMutants.cmd
The main method starts
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by mujava.MutationSystem (file:/Users/zhanghu
anyu/Downloads/MuJava/mujava.jar) to method java.net.URLClassLoader.addURL(java.
net.URL)
WARNING: Please consider reporting this to the maintainers of mujava.MutationSys
tem
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflect
ive access operations
WARNING: All illegal access operations will be denied in a future release
Exception in thread "main" java.lang.IllegalArgumentException: object is not an
instance of declaring class
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Nativ
e Method)
        at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(Native
MethodAccessorImpl.java:62)
        at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(De
legatingMethodAccessorImpl.java:43)
        at java.base/java.lang.reflect.Method.invoke(Method.java:566)
        at mujava.MutationSystem.addURL(MutationSystem.java:548)
        at mujava.MutationSystem.recordInheritanceRelation(MutationSystem.java:4
80)
        at mujava.gui.GenMutantsMain.main(GenMutantsMain.java:57)
```

经查询，mujava 并不适配于 jdk11，重新安装 jdk8 即可解决此问题。

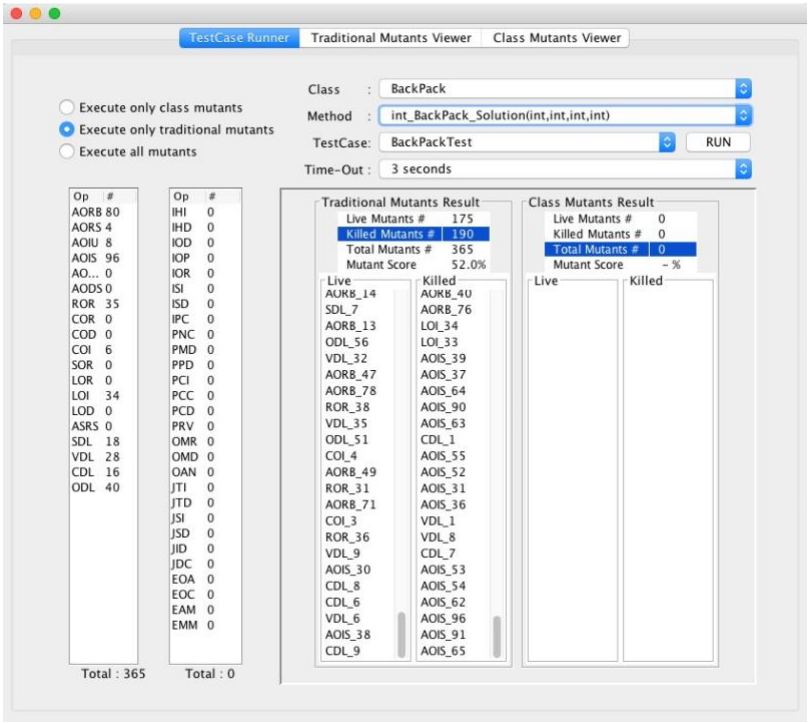
五、 测试结果

本次实验测试用例为随机选取数据，即

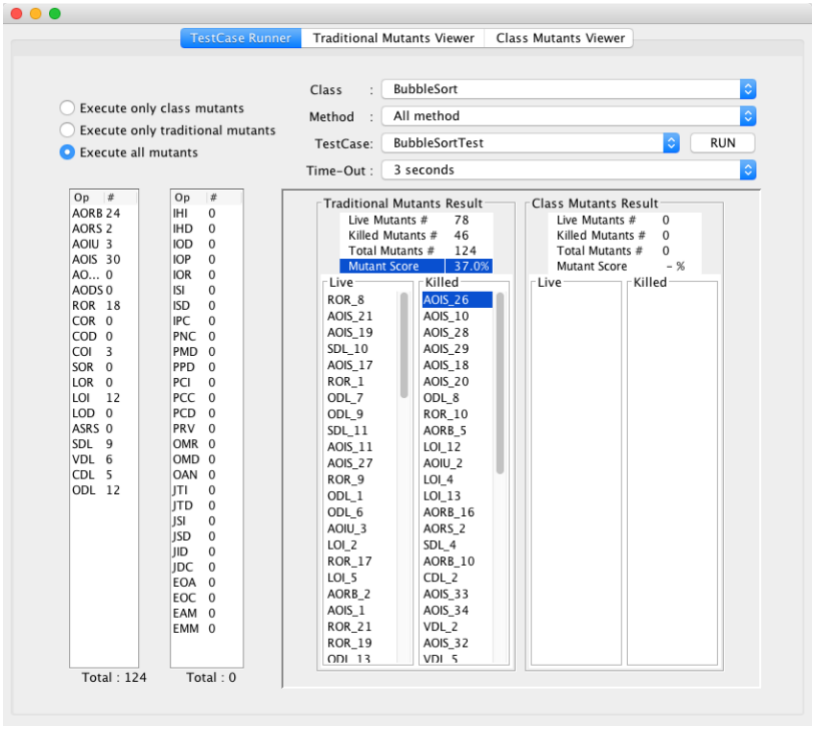
```
int m = 10;
int n = 3;
int w[] = {3, 4, 5};
int p[] = {4, 5, 6};
BackPack.BackPack_Solution(m,n,w,p);
```

```
int arr[] = new int[]{1,6,2,2,5};
BubbleSort.BubbleSort(arr);
```

结果如下

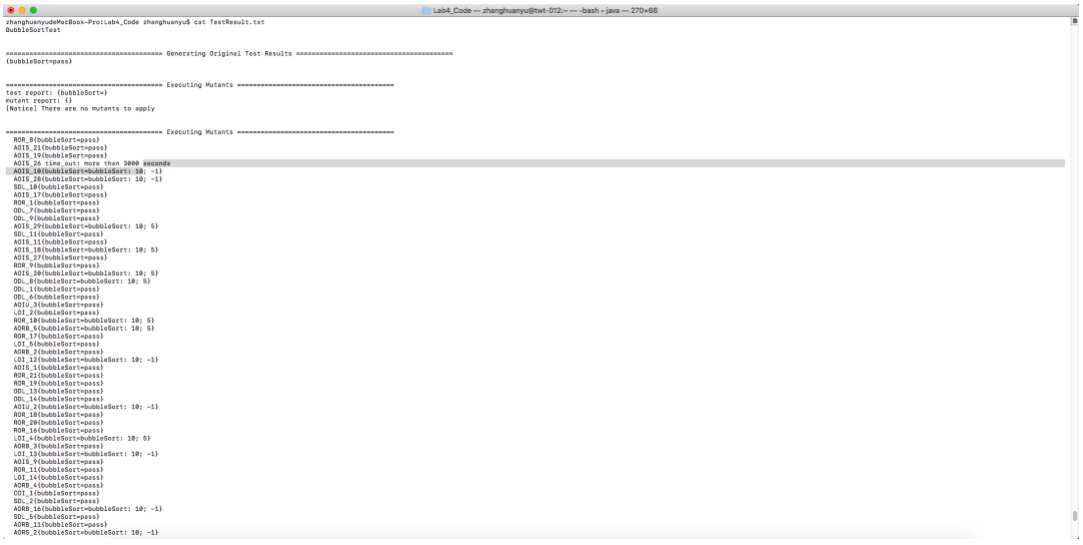


BackPackTest 内 Live 175，Killed 190



BubbleSortTest 内 Live 78，Killed 46. 可见测试集效果不佳，故应重新设计多个测试集

对之前数据得到如下测试结果



六、 总结

本次试验遇到了许多问题 ,最主要的是 JDK 版本过高 ,导致不匹配 Mujava，查询了较久的解决方法，大大影响了实验的完成效率。

同时，我也学会了 mujava 的环境配置方法和使用方法。对于此次试验中的被测试 Java 类，Java 代码无错误，但由于测试数据不充分，导致 kill 率较低，经过多次试验未找到合适测试类。