

Homework 4: Mutation Based Testing

Software Testing 2023

2023/05/25



muJava

muJava

→ Download **mujava.jar**, **openjava.jar** and **mujava.config**

◆ From: <https://cs.gmu.edu/~offutt/mujava/>

II. Downloading μ Java

Although installing μ Java is not technically difficult or complicated, it will require a little more effort than installing many commercial packages. μ Java can run on Unix, Linux, and Windows platforms. You will need to download and deploy three files; two Java "jar" files and a configuration file.

mujava.jar	μ Java system library
openjava.jar	OpenJava library that is adapted for μ Java (OpenJava WWW page)
mujava.config	A file that specifies the μ Java system home directory

The two jar files should be placed in a directory on disk that all users who need to use μ Java can access, for example, in `C:\mujava\`.

μ Java was updated in April 2013 to work with JUnit and generics. The old versions are still available if you need them.

Version 3

mujava-v3.jar	μ Java system library
openjava2005.jar	OpenJava library that is adapted for μ Java (OpenJava WWW page)
mujava-v3.config	A file that specifies the μ Java system home directory

Issues with version 3:

- μ Java could not compile classes that use Java generics. (This was corrected in version 4.)
- Fault fix, 5-December-2011: The program did not properly initialize the time-out counter, so time-out only worked if users explicitly set it. This caused μ Java to freeze during execution sometimes.
- Fault fix, 6-December-2011: muJava implemented the ROR operator incorrectly.

μ Java was updated in December 2008 to work with Java 1.5 and 1.6. The old versions are still available if you need them.

Version 2

mujava-v2.jar	μ Java system library (Updates and fault fixes, May 2006)
adaptedOJ.jar	OpenJava library that is adapted for μ Java (OpenJava WWW page)
mujava.config	A file that specifies the μ Java system home directory

The mujava.jar file was updated in May 2006. The old version, [mujava-v1.jar](#), is still available if you need it.

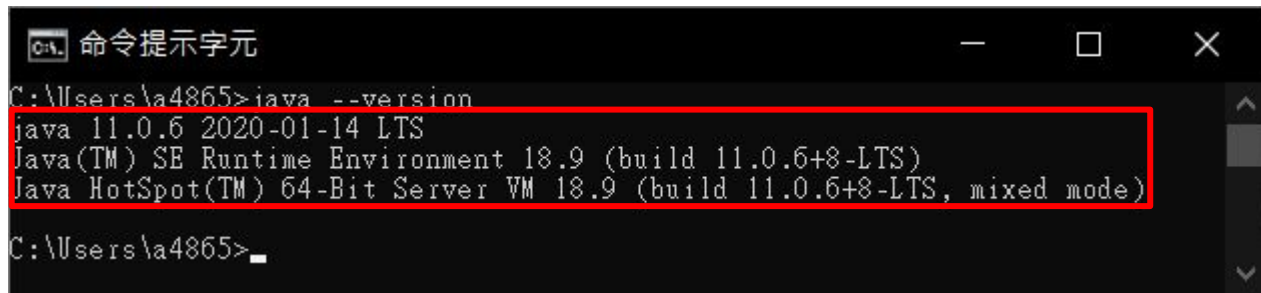
Although we have no license agreement, if you use μ Java, we would like to hear about it. Particularly, if you publish papers that rely on μ Java, please send an email to "offutt [at] gmue.edu" and we will add the reference to the μ Java web page.

Version 1

Installation - Windows 10

→ Step 1 :

- ◆ check jdk version (at least 1.8)
 - **\$ java --version**

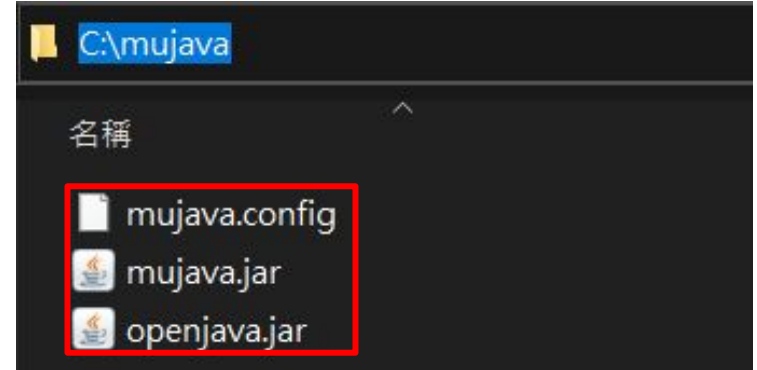


```
命令提示字元
C:\Users\A4865>java --version
java 11.0.6 2020-01-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.6+8-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.6+8-LTS, mixed mode)
C:\Users\A4865>
```

Installation - Windows 10

→ Step 2 :

- ◆ Put the file in here :
 - C:\mujava

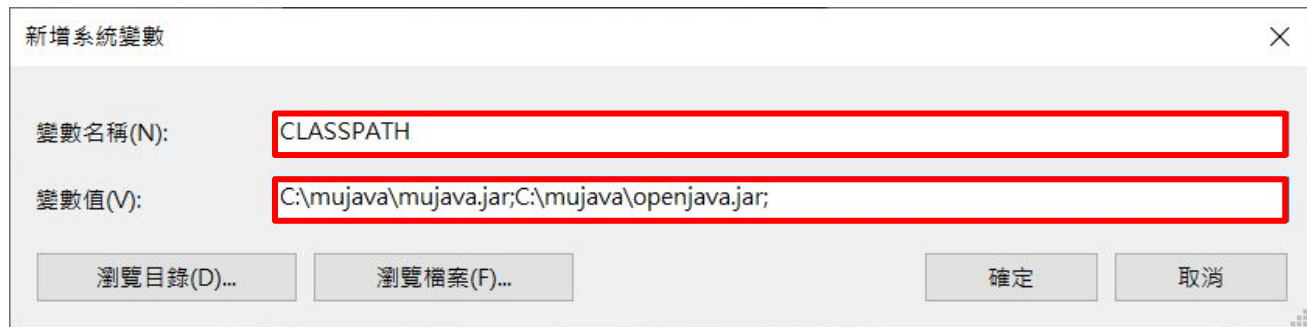


Installation - Windows 10

→ Step 3 :

◆ Set CLASSPATH

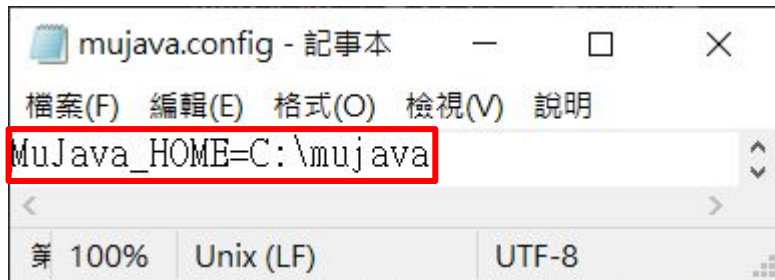
- 本機 → 內容 → 進階系統設定 → 進階 → 環境變數 → 系統變數 → 新增
 - 變數名稱: CLASSPATH
 - 變數值: C:\mujava\mujava.jar;C:\mujava\openjava.jar;



Installation - Windows 10

→ Step 4 :

- ◆ Modify the content of ***mujava.config***
 - MuJava_HOME=C:\mujava



Installation - macOS 11.4

→ Steps

- ◆ `$ export CLASSPATH=$CLASSPATH:/Users/chtsai/muJava/mujava.jar:/Users/chtsai/muJava/openjava.jar`
 - e.g. `/Users/chtsai/muJava`
- ◆ `$ java --version`
 - `openjdk 11.0.9 2020-10-20`
 - `OpenJDK Runtime Environment (build 11.0.9+11)`
 - `OpenJDK 64-Bit Server VM (build 11.0.9+11, mixed mode)`



Installation - Ubuntu 20.04

→ Steps

- ◆ install openjdk-11
 - `sudo apt-get install openjdk-11-jdk`
- ◆ export CLASSPATH
 - `export CLASSPATH=/DIR/mujava.jar:/DIR/openjava.jar`
- ◆ set mujava.config
 - `MuJava_HOME=/DIR`

```
tl455047@tl455047-X556UR ❏ ~/st_hw4 ❏ java --version
openjdk 11.0.11 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
tl455047@tl455047-X556UR ❏ ~/st_hw4 ❏ echo $CLASSPATH
/home/tl455047/st_hw4/mujava.jar:/home/tl455047/st_hw4/openjava.jar
tl455047@tl455047-X556UR ❏ ~/st_hw4 ❏ cat mujava.config
MuJava_HOME=/home/tl455047/st_hw4
```

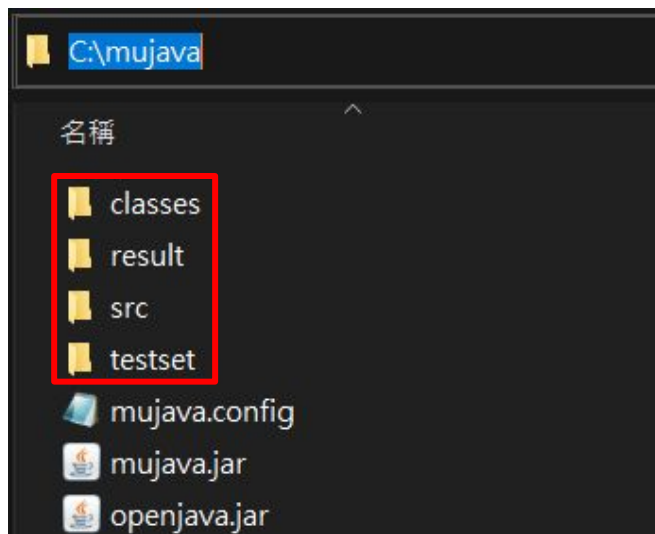
muJava

→ Steps

- ◆ Create src, classes, result, testset directories
 - **\$ java mujava.makeMuJavaStructure**



```
命令提示字元
C:\mujava> java mujava.makeMuJavaStructure
Make C:\mujava directory...
C:\mujava directory exists already.
Make C:\mujava\src directory...
Making C:\mujava\src directory ...done.
Make C:\mujava\classes directory...
Making C:\mujava\classes directory ...done.
Make C:\mujava\result directory...
Making C:\mujava\result directory ...done.
Make C:\mujava\testset directory...
Making C:\mujava\testset directory ...done.
C:\mujava>
```



muJava

→ Steps

- ◆ Add JUnit4 , hamcrest to CLASSPATH
 - [junit-4.12.jar](#)
 - [hamcrest-core-1.3.jar](#)

muJava

→ Steps

- ◆ Put your target java file in /src directory
 - [Cal.java](#)
- ◆ Write test cases (CalTest.java) and compile
 - `$ javac ./testset/CalTest.java ./src/Cal.java`
 - output:
 - `./testset/CallTest.class`
 - `./src/Cal.class`

muJava

testset/CalTest.java

```
1 import org.junit.Assert;
2 import org.junit.Test;
3
4
5 public class CalTest {
6     // Normal year
7     @Test
8     public void test1() {
9         Assert.assertEquals(59, Cal.cal(1, 1, 3, 1, 2022));
10    }
11
12 }
```

muJava

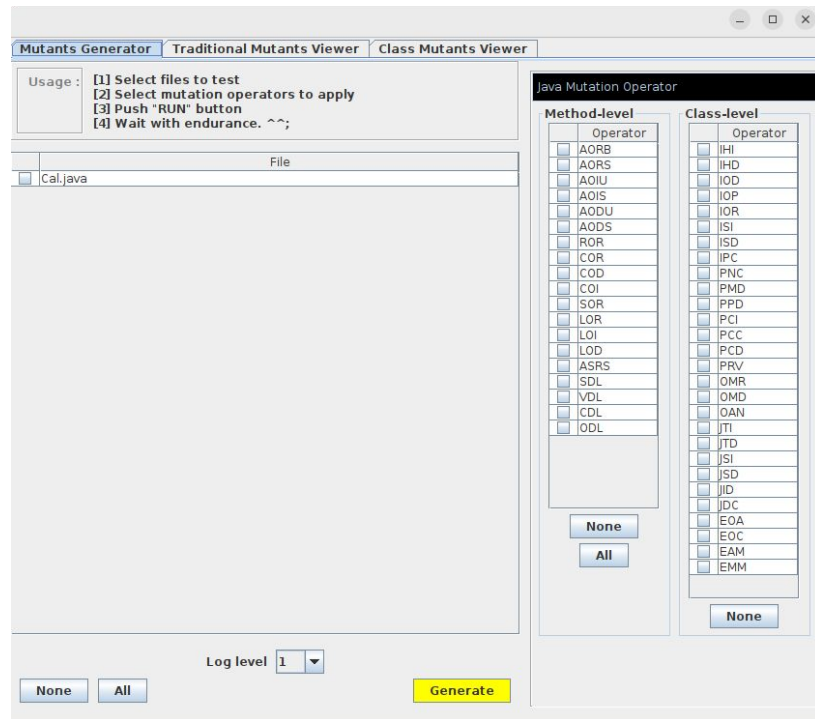
compile

```
oceans@lab547 ~/st_hw4> pwd
/home/oceans/st_hw4
oceans@lab547 ~/st_hw4> ls src
Cal.java
oceans@lab547 ~/st_hw4> ls testset/
CalTest.java
oceans@lab547 ~/st_hw4> javac ./testset/CalTest.java ./src/Cal.java
oceans@lab547 ~/st_hw4> ls src
Cal.class  Cal.java
oceans@lab547 ~/st_hw4> ls testset/
CalTest.class  CalTest.java
oceans@lab547 ~/st_hw4>
```

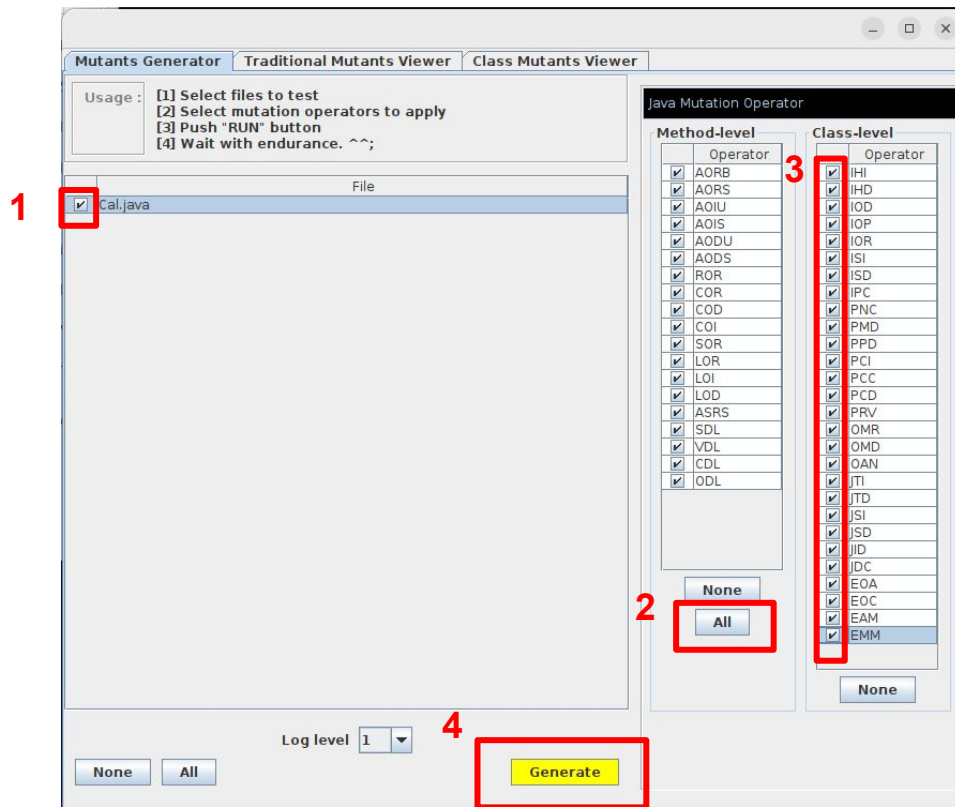
muJava

→ Run muJava

- `$ java mujava.gui.GenMutantsMain`



muJava



muJava

```
oceane@lab547 ~/st_hw4> java mujava.gui.GenMutantsMain
The main method starts
File /home/oceane/st_hw4/src/Cal.java
Exception at generating IOR mutant. file : AllMutantsGenerator.java
File /home/oceane/st_hw4/src/Cal.java
-----
All files are handled
MutantsViewerPanel.MList_mouseClicked - mutant_log != null
MutantsViewerPanel.getMutatedContent --- start_index =7 : end_index =10
```

muJava

The screenshot shows the 'Traditional Mutants Viewer' window in muJava. The 'Select a class' dropdown is set to 'Cal' and 'Select a method' is set to 'All method'. On the left, a summary table lists mutants with their IDs and counts. The main area displays the original code for the 'int_cal' method and a mutant version with several changes highlighted in red.

*** Summary ***

Op	#
AO...	36
AO...	1
AOIU	13
AOIS	76
AO...	0
AO...	0
ROR	34
COR	4
COD	0
COI	7
SOR	0
LOR	0
LOI	20
LOD	0
AS...	0
SDL	12
VDL	13
CDL	5
ODL	22

Total : 243

Select a class : Cal
Select a method : All method

(line 18) int_cal(int,int,int,int,int):year % 100 ==> year - 100

Original

```
18 int m100 = year % 100;
19 int m400 = year % 400;
20 if (m4 != 0 || m100 == 0 && m400 != 0) {
21     daysIn[2] = 28;
22 } else {
23     daysIn[2] = 29;
24 }
25 numDays = day2 + (daysIn[month1] - day1);
26 for (int i = month1 + 1; i <= month2 - 1; i++) {
27     numDays = daysIn[i] + numDays;
28 }
29 }
30 return numDays;
31 }
32
33 public static void main( java.lang.String[] argv )
34 {
```

Mutant

```
12 int numDays;
13 if (month2 == month1) {
14     numDays = day2 - day1;
15 } else {
16     int[] daysIn = { 0, 31, 0, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
17     int m4 = year % 4;
18     int m100 = year - 100;
19     int m400 = year % 400;
20     if (m4 != 0 || m100 == 0 && m400 != 0) {
21         daysIn[2] = 28;
22     } else {
23         daysIn[2] = 29;
24     }
25     numDays = day2 + (daysIn[month1] - day1);
26     for (int i = month1 + 1; i <= month2 - 1; i++) {
27         numDays = daysIn[i] + numDays;
28     }
29 }
```

muJava

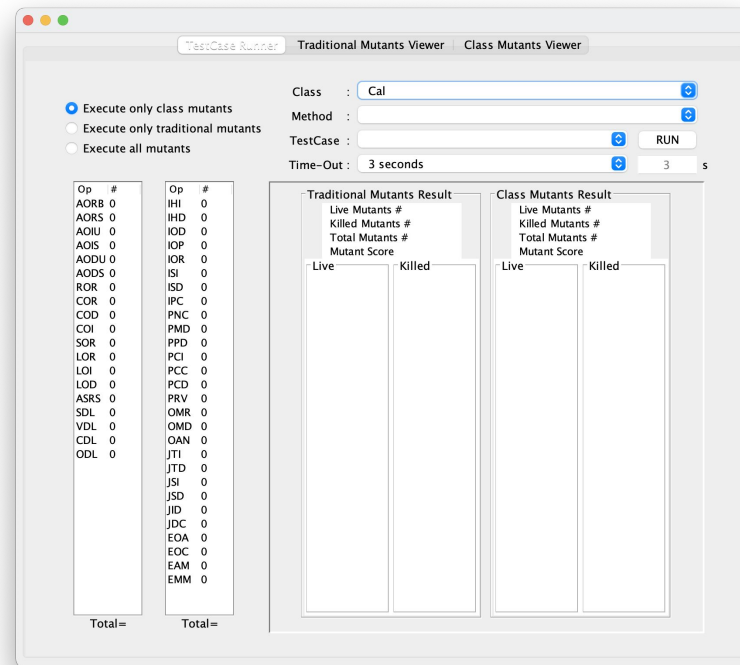
→ **After** GenMutantsMain

- a. Avoid score is always 100%
- b. Prevent the warning
- c. **cp ./src/Cal.class ./classes**

muJava

→ Run Test

◆ `$ java mujava.gui.RunTestMain`



muJava

muJava Test Case Runner Interface

Test Case Runner | Traditional Mutants Viewer | Class Mutants Viewer

☐ Execute only class mutants
☐ Execute only traditional mutants
☒ Execute all mutants

Class : Cal
Method : Cal

TestCase : CalTest RUN

Time-Out : 3 seconds 3 s

Op	#	Op	#
AORB	0	IHI	0
AORS	0	IHD	0
AOIU	0	IOD	0
AOIS	0	IOP	0
AODU	0	IOR	0
AODS	0	ISI	0
ROR	0	ISD	0
COR	0	IPC	0
COD	0	PNC	0
COI	0	PMD	0
SOR	0	PPD	0
LOR	0	PCI	0
LOI	0	PCC	0
LOD	0	PCD	0
ASRS	0	PRV	0
SDL	0	OMR	0
VDL	0	OMD	0
CDL	0	OAN	0
ODL	0	JTI	0
		JTD	0
		JSI	0
		JSD	0
		JID	0
		JDC	0
		EOA	0
		EOC	0
		EAM	0
		EMM	0
Total=		Total=	

Traditional Mutants Result

Live Mutants #	
Killed Mutants...	
Total Mutants...	
Mutant Score	

Live	Killed

Class Mutants Result

Live Mutants #	
Killed Mutants...	
Total Mutants...	
Mutant Score	

Live	Killed

muJava

The screenshot shows the muJava TestCase Runner interface. The 'Traditional Mutants Viewer' tab is active. On the left, there are three radio buttons for execution options: 'Execute only class mutants', 'Execute only traditional mutants', and 'Execute all mutants' (which is selected). Below these are two tables of mutants, each with columns 'Op' and '#'. The first table has 24 mutants, and the second has 0. At the bottom, it shows 'Total : 243' and 'Total : 0'. On the right, there are dropdown menus for 'Class' (set to 'Cal'), 'Method' (set to 'All method'), and 'Test Case' (set to 'int_cal(int,int,int,int,int)'). The 'Time-Out' is set to 'void main(java.lang.String[])'. Below these are two result panels: 'Traditional Mutants Result' and 'Class Mutants Result'. Each panel has a table with columns 'Live' and 'Killed' and rows for 'Live Mutants #', 'Killed Mutants...', 'Total Mutants...', and 'Mutant Score'. The 'Test Case' dropdown is highlighted with a red box.

TestCase Runner Traditional Mutants Viewer Class Mutants Viewer

☐ Execute only class mutants
☐ Execute only traditional mutants
☒ Execute all mutants

Class : Cal
Method : All method
Test Case : int_cal(int,int,int,int,int)
Time-Out : void main(java.lang.String[])
int_getN0

Op	#
AO...	36
AO...	1
AOIU	13
AOIS	76
AO...	0
AO...	0
ROR	34
COR	4
COD	0
COI	7
SOR	0
LOR	0
LOI	20
LOD	0
ASRS	0
SDL	12
VDL	13
CDL	5
ODL	22

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 : 243 Total : 0

Traditional Mutants Result

Live	Killed
Live Mutants #	
Killed Mutants...	
Total Mutants...	
Mutant Score	

Class Mutants Result

Live	Killed
Live Mutants #	
Killed Mutants...	
Total Mutants...	
Mutant Score	

muJava

TestCase Runner Traditional Mutants Viewer Class Mutants Viewer

☐ Execute only class mutants
☐ Execute only traditional mutants
☒ Execute all mutants

Class : Cal
Method : int_cal(int,int,int,int,int)
TestCase : CalTest
Time-Out : 3 seconds

RUN

Op	#
AO...	36
AO...	1
AOIU	12
AOIS	74
AO...	0
AO...	0
ROR	34
COR	4
COD	0
COI	7
SOR	0
LOR	0
LOI	19
LOD	0
ASRS	0
SDL	10
VDL	13
CDL	5
ODL	22

Total : 237

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

Traditional Mutants Result

Live	Killed

Class Mutants Result

Live	Killed

muJava

```
AOIU_2{test2=pass, test1=pass}
AORB_22{test2=test2: 15; expected:<121> but was:<90>, test1=test1: 9; expected:<59> but was:<28>}
AOIS_43{test2=pass, test1=pass}
ODL_4{test2=pass, test1=pass}
AOIS_24{test2=pass, test1=pass}
AOIS_52{test2=pass, test1=pass}
AOIS_58 time out: more than 3000 milliseconds
VDL_15{test2=test2: 15; expected:<121> but was:<31>, test1=test1: 9; expected:<59> but was:<31>}
AOIS_64{test2=test2: 15; expected:<121> but was:<91>, test1=pass}
VDL_14{test2=test2: 15; expected:<121> but was:<152>, test1=test1: 9; expected:<59> but was:<90>}
AOIS_73{test2=pass, test1=pass}
AORB_26{test2=test2: 15; expected:<121> but was:<152>, test1=test1: 9; expected:<59> but was:<90>}
AOIS_30{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
LOI_7{test2=pass, test1=pass}
AOIS_63{test2=test2: 15; Index 13 out of bounds for length 13, test1=test1: 9; Index 13 out of bounds for length 13}
SDL_9{test2=test2: 15; expected:<121> but was:<92>, test1=pass}
AORB_5{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
COR_2{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
AOIS_69{test2=test2: 15; expected:<121> but was:<124>, test1=test1: 9; expected:<59> but was:<60>}
AOIS_40{test2=pass, test1=pass}
AORB_24{test2=test2: 15; expected:<121> but was:<61>, test1=test1: 9; expected:<59> but was:<-1>}
AOIS_53{test2=test2: 15; expected:<121> but was:<92>, test1=test1: 9; expected:<59> but was:<31>}
ODL_32{test2=test2: 15; expected:<121> but was:<30>, test1=test1: 9; expected:<59> but was:<28>}
AOIS_57{test2=test2: 15; expected:<121> but was:<62>, test1=test1: 9; expected:<59> but was:<31>}
ODL_21{test2=test2: 15; expected:<121> but was:<122>, test1=test1: 9; expected:<59> but was:<60>}
AORB_34{test2=test2: 15; / by zero, test1=test1: 9; expected:<59> but was:<0>}
AOIU_9{test2=test2: 15; Index 13 out of bounds for length 13, test1=test1: 9; Index 13 out of bounds for length 13}
ROR_19{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
CDL_8{test2=test2: 15; expected:<121> but was:<152>, test1=test1: 9; expected:<59> but was:<90>}
AOIS_48{test2=test2: 15; expected:<121> but was:<152>, test1=test1: 9; expected:<59> but was:<90>}
ROR_6{test2=test2: 15; expected:<121> but was:<0>, test1=test1: 9; expected:<59> but was:<0>}
LOI_1{test2=pass, test1=pass}
AOIS_49{test2=test2: 15; expected:<121> but was:<120>, test1=test1: 9; expected:<59> but was:<58>}
ODL_8{test2=pass, test1=pass}
AOIS_44{test2=pass, test1=pass}
ROR_13{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
AOIU_7{test2=test2: 15; expected:<121> but was:<119>, test1=test1: 9; expected:<59> but was:<57>}
SDL_3{test2=test2: 15; expected:<121> but was:<0>, test1=test1: 9; expected:<59> but was:<0>}
COI_7{test2=test2: 15; expected:<121> but was:<31>, test1=test1: 9; expected:<59> but was:<31>}
ROR_5{test2=test2: 15; expected:<121> but was:<0>, test1=test1: 9; expected:<59> but was:<0>}
AOIU_10{test2=test2: 15; expected:<121> but was:<31>, test1=test1: 9; expected:<59> but was:<31>}
ROR_9{test2=test2: 15; expected:<121> but was:<120>, test1=pass}
AOIS_60
```


muJava

muJava interface showing the Traditional Mutants Viewer.

Test Case Configuration:

- Class : Cal
- Method : int_cal(int,int,int,int,int)
- TestCase : CalTest
- Time-Out : 3 seconds

Execution Options:

- ☐ Execute only class mutants
- ☐ Execute only traditional mutants
- ☒ Execute all mutants

Test Results:

Traditional Mutants Result (highlighted in red):

Live Mutants #	97
Killed Mutants...	140
Total Mutants...	237
Mutant Score	59.0%

Class Mutants Result:

Live Mutants #	0
Killed Mutants...	0
Total Mutants...	0
Mutant Score	- %

Op # Table:

Op	#
AO...	36
AO...	1
AOIU	12
AOIS	74
AO...	0
AO...	0
ROR	34
COR	4
COD	0
COI	7
SOR	0
LOR	0
LOI	19
LOD	0
ASRS	0
SDL	10
VDL	13
CDL	5
ODL	22

Total : 237

Op # Table:

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

muJava

TestCase Runner Traditional Mutants Viewer Class Mutants Viewer

☐ Execute only class mutants
☐ Execute only traditional mutants
☒ Execute all mutants

Class : Cal
Method : int_cal(int,int,int,int,int)
TestCase : CalTest RUN
Time-Out : 3 seconds 3 s

Op	#	Op	#
AO...	36	IHI	0
AO...	1	IHD	0
AOIU	12	IOD	0
AOIS	74	IOP	0
AO...	0	IOR	0
AO...	0	ISI	0
ROR	34	ISD	0
COR	4	IPC	0
COD	0	PNC	0
COI	7	PMD	0
SOR	0	PPD	0
LOR	0	PCI	0
LOI	19	PCC	0
LOD	0	PCD	0
ASRS	0	PRV	0
SDL	10	OMR	0
VDL	13	OMD	0
CDL	5	OAN	0
ODL	22	JTI	0
		JTD	0
		JSI	0
		JSD	0
		JID	0
		JDC	0
		EOA	0
		EOC	0
		EAM	0
		EMM	0

Total : 237 Total : 0

Traditional Mutants Result

Live Mutants #	97
Killed Mutants...	140
Total Mutants...	237
Mutant Score	59.0%

Live

- AORB_12
- AORB_13
- AOIS_46**
- AORB_11
- AOIU_2
- AOIS_43
- ODL_4
- AOIS_24
- AOIS_52
- AOIS_73
- LOI_7
- AOIS_40
- LOI_1
- ODL_8
- AOIS_44
- ROR_18
- AOIS_33
- ROR_28
- AORB_13
- LOI_2
- LOI_3
- ROR_25
- AOIU_3

Killed

- ODL_33
- AORB_31
- LOI_13
- LOI_14
- AOIS_62
- ODL_25
- LOI_15
- AORB_22
- AOIS_58
- VDL_15
- AOIS_64
- VDL_14
- AORB_26
- AOIS_30
- AOIS_63
- SDL_9
- AORB_5
- COR_2
- AOIS_69
- AORB_24
- AOIS_53
- ODL_32
- AOIS_57

Class Mutants Result

Live Mutants #	0
Killed Mutants...	0
Total Mutants...	0
Mutant Score	-%

Live

Killed

muJava

The screenshot shows the muJava Traditional Mutants Viewer interface. The top tabs are "TestCase Runner", "Traditional Mutants Viewer" (selected), and "Class Mutants Viewer". Below the tabs, there are two dropdown menus: "Select a class : Cal" and "Select a method : All method".

On the left side, there is a "Summary" table with columns "Op" and "#". It lists various mutants, with "AOIS_46" highlighted in a red box. Below the table, it says "Total : 243".

The main area displays the "Original" code for the method `int cal(int month1, int day1, int month2, int day2, int year)`. The code is as follows:

```
10 public static int cal(int month1, int day1, int month2, int day2, int year )
11 {
12     int numDays;
13     if (month2 == month1) {
14         numDays = day2 - day1;
15     } else {
16         int[] daysIn = { 0, 31, 0, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
17         int m4 = year % 4;
18         int m100 = year % 100;
19         int m400 = year % 400;
20         if (m4 != 0 || m100 == 0 && m400 != 0) {
21             daysIn[2] = 28;
22         } else {
23             daysIn[2] = 29;
24         }
25         numDays = day2 + (daysIn[month1] - day1);
26     }
27 }
```

Below the original code, the "Mutant" code is shown. It is identical to the original code, but with a change on line 25: `numDays = day2 + (daysIn[--month1] - day1);`.

muJava

modify testset/CalTest.java

```
1 import org.junit.Assert;
2 import org.junit.Test;
3
4
5 public class CalTest {
6     // Normal year
7     @Test
8     public void test1() {
9         Assert.assertEquals(59, Cal.cal(1, 1, 3, 1, 2022));
10    }
11
12    // leap year
13    @Test
14    public void test2() {
15        Assert.assertEquals(121, Cal.cal(1, 1, 5, 1, 2020));
16    }
17
18    // (m100 == 0) && (m400 != 0)
19    @Test
20    public void test3() {
21        Assert.assertEquals(181, Cal.cal(1, 1, 7, 1, 2100));
22    }
23
24 }
```

muJava

```
oceane@lab547 ~/st_hw4> vim testset/CalTest.java
oceane@lab547 ~/st_hw4> javac ./testset/CalTest.java ./src/Cal.java
oceane@lab547 ~/st_hw4> java mujava.gui.RunTestMain
```

muJava

muJava interface showing the Traditional Mutants Viewer.

Test Case Configuration:

- Class : Cal
- Method : int_cal(int,int,int,int,int)
- TestCase : CalTest
- Time-Out : 3 seconds

Execution Options:

- ☐ Execute only class mutants
- ☐ Execute only traditional mutants
- ☒ Execute all mutants

Test Results Summary:

Op	#	Op	#
AO...	36	IHI	0
AO...	1	IHD	0
AOIU	12	IOD	0
AOIS	74	IOP	0
AO...	0	IOR	0
AO...	0	ISI	0
ROR	34	ISD	0
COR	4	IPC	0
COD	0	PNC	0
COI	7	PMD	0
SOR	0	PPD	0
LOR	0	PCI	0
LOI	19	PCC	0
LOD	0	PCD	0
ASRS	0	PRV	0
SDL	10	OMR	0
VDL	13	OMD	0
CDL	5	OAN	0
ODL	22	JTI	0
		JTD	0
		JSI	0
		JSD	0
		JID	0
		JDC	0
		EOA	0
		EOC	0
		EAM	0
		EMM	0

Total : 237

Traditional Mutants Result (Highlighted):

Live Mutants #	73
Killed Mutants...	164
Total Mutants...	237
Mutant Score	69.0%

Class Mutants Result:

Live Mutants #	0
Killed Mutants...	0
Total Mutants...	0
Mutant Score	- %

Mutant Lists:

Live:

- AOIS_10
- AOIS_46
- AOIU_2
- AOIS_43
- ODL_4
- AOIS_24
- AOIS_52
- AOIS_73
- LOI_7
- AOIS_40
- LOI_1
- AOIS_44
- ROR_18
- AORB_13
- LOI_2
- LOI_3
- AOIU_3
- AOIS_36
- AOIS_35
- AOIS_27
- AOIS_32
- AOIS_72
- ROR_8

Killed:

- ODL_33
- AORB_12
- AORB_31
- AORB_11
- LOI_13
- LOI_14
- AOIS_62
- ODL_25
- LOI_15
- AORB_22
- AOIS_58
- VDL_15
- VDL_14
- AORB_26
- AOIS_30
- AOIS_63
- SDL_9
- AORB_5
- COR_2
- AOIS_69
- AORB_24
- AOIS_53

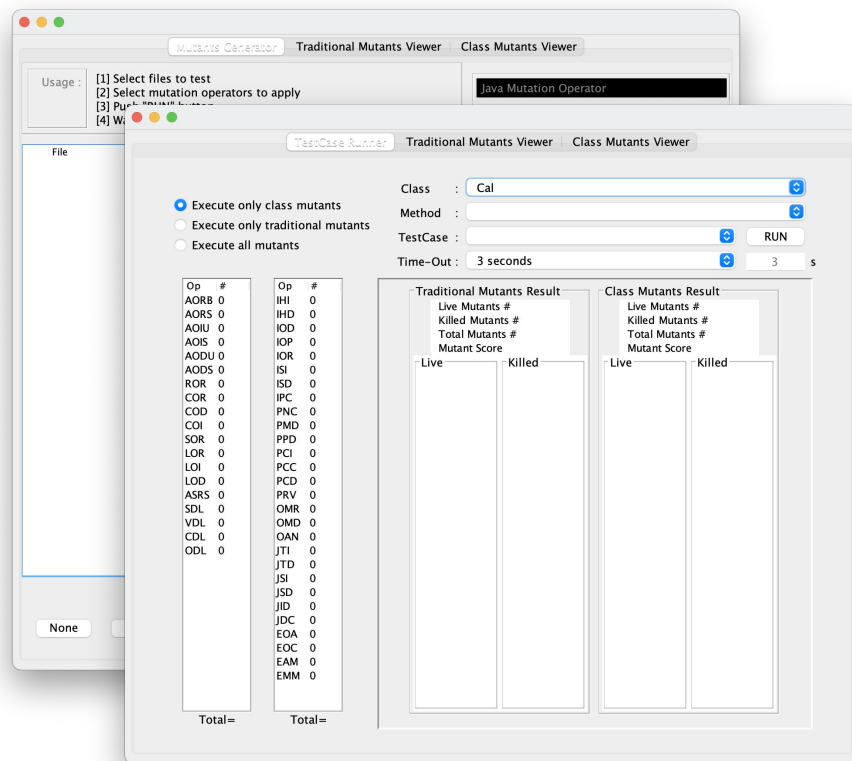
Homework

→ Installation

- Get Java version
- Export CLASSPATH
 - jar files
- Set muJava configuration
 - specify muJava home directory

→ Run muJava

- `$ java mujava.gui.GenMutantsMain`
- `$ java mujava.gui.RunTestMain`



Hint

1. Junit 4 only
2. Method is blank
 - a. choose “**class**” list first

