Marco Seman

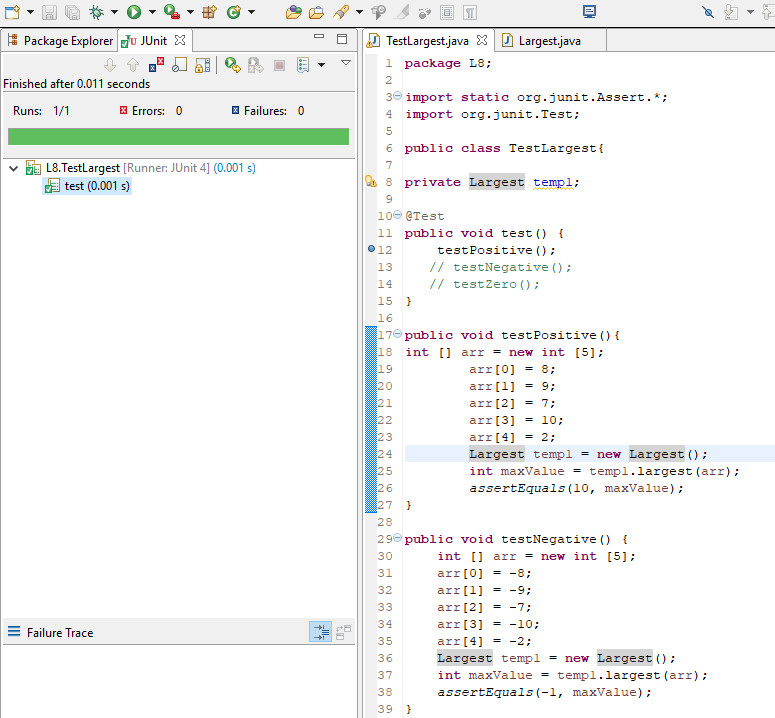
CIS 285 Software Engineering Tools

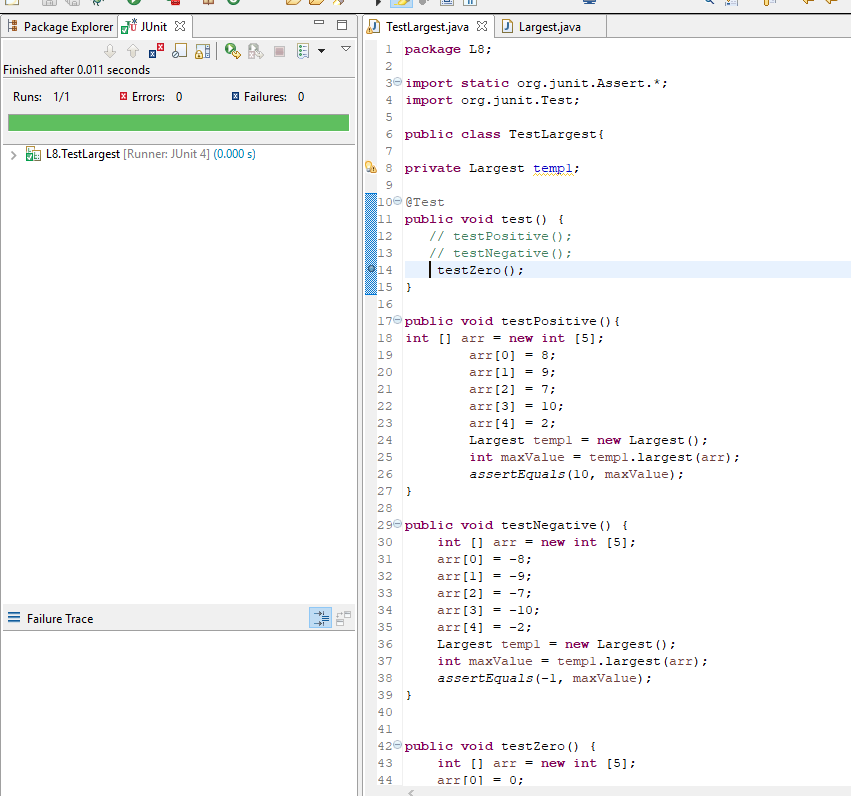
March 28, 2019

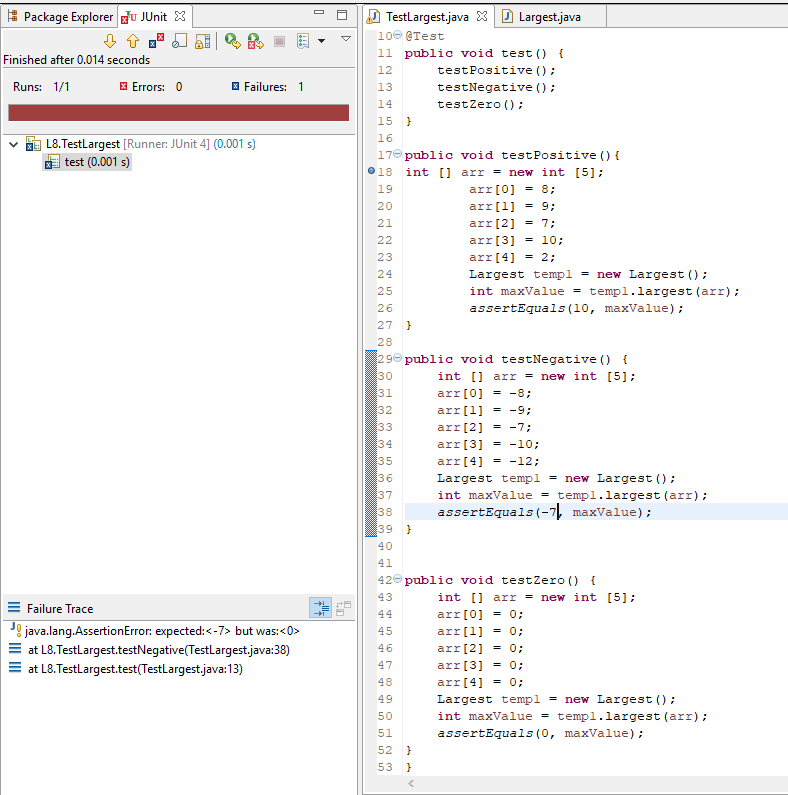
Lab 8

1. Outputs of unit tests. Include both failed and passed unit tests. In the case of failed unit tests, indicate what was wrong and what was done to fix the problem.

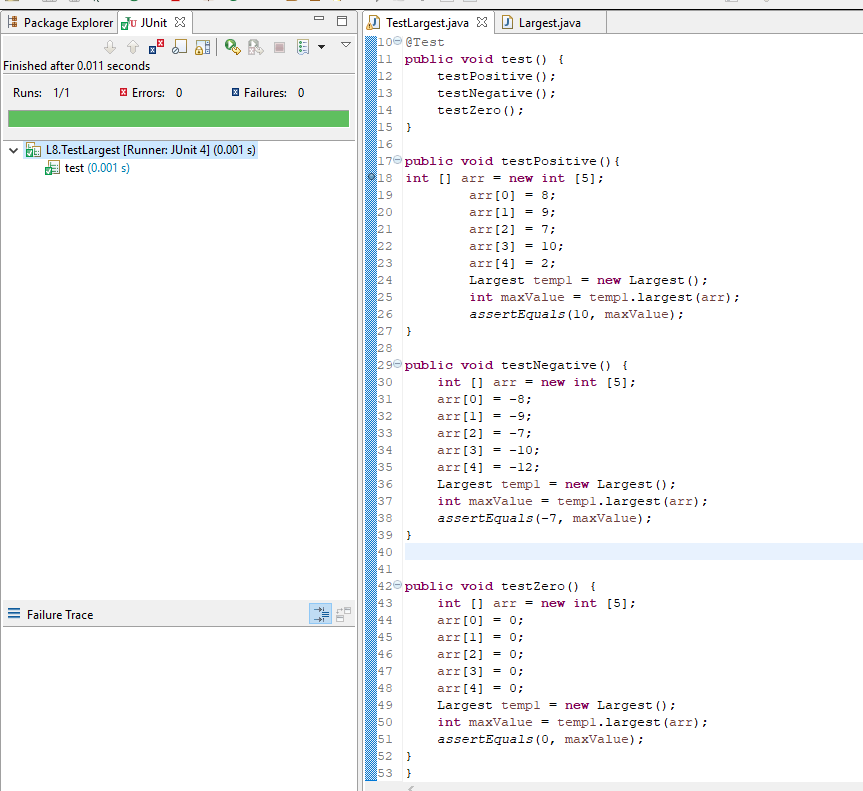
The testPositive and testZero have passed both test cases. However, the testNegative has failed. The testNegative failed because it is looking for a zero in the assertEquals(-7, maxValue); part of the code when it is supposed to be the highest negative value. It was looking for zero because in the source code for Largest, max was initially assigned to zero. To fix this problem, I had to make modifications to the Largest.java class and convert int max into an array of a list because max must be assigned a value in the array function. Max investigates each of the array list in the particular function and determines which number in the array list is the largest.







b. Final outputs of all unit tests showing successful pass for all



c. Source code of final testLargest.java class

**package** L8;

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

**import** org.junit.Test;

**public** **class** TestLargest{

**private** Largest temp1;

@Test

**public** **void** test() {

testPositive();

testNegative();

testZero();

}

**public** **void** testPositive(){

**int** [] arr = **new** **int** [5];

arr[0] = 8;

arr[1] = 9;

arr[2] = 7;

arr[3] = 10;

arr[4] = 2;

Largest temp1 = **new** Largest();

**int** maxValue = temp1.largest(arr);

*assertEquals*(10, maxValue);

}

**public** **void** testNegative() {

**int** [] arr = **new** **int** [5];

arr[0] = -8;

arr[1] = -9;

arr[2] = -7;

arr[3] = -10;

arr[4] = -12;

Largest temp1 = **new** Largest();

**int** maxValue = temp1.largest(arr);

*assertEquals*(-7, maxValue);

}

**public** **void** testZero() {

**int** [] arr = **new** **int** [5];

arr[0] = 0;

arr[1] = 0;

arr[2] = 0;

arr[3] = 0;

arr[4] = 0;

Largest temp1 = **new** Largest();

**int** maxValue = temp1.largest(arr);

*assertEquals*(0, maxValue);

}

}

d. Source code of final Largest.java class after making correction

**package** L8;

**public** **class** Largest {

**public** Largest() {

}

**public** **int** largest(**int** [] list){

**int** index;

**int** max = list[0];

**for**(index = 0; index < list.length-1; index++){

**if**(list[index] > max) {

max = list[index];

}

}

**return** max;

}

}