Universidad de las Fuerzas Armadas ESPE

Object Oriented Programming

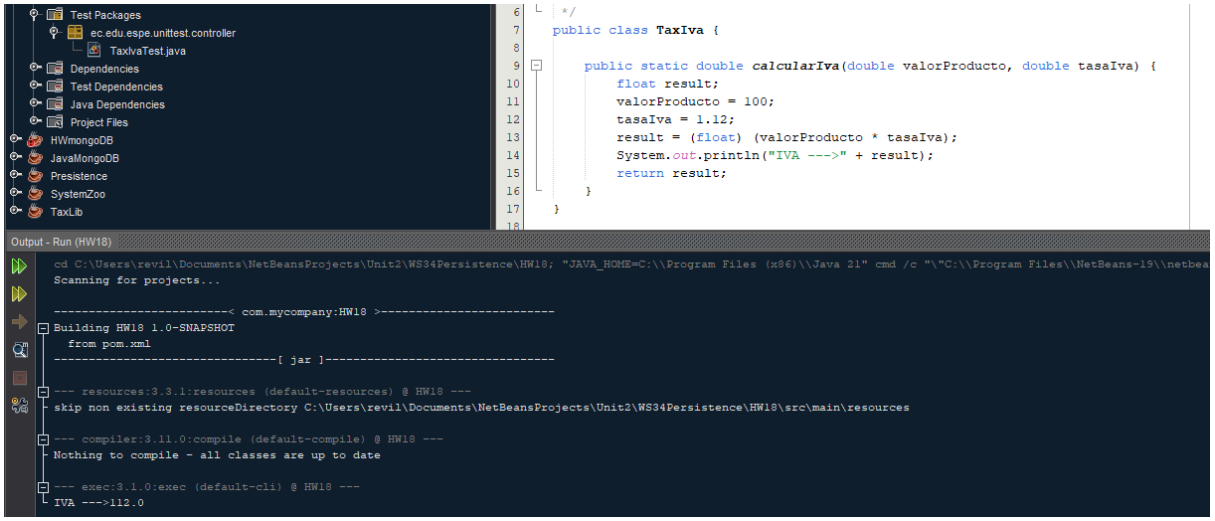
Teacher: Edison Lascano

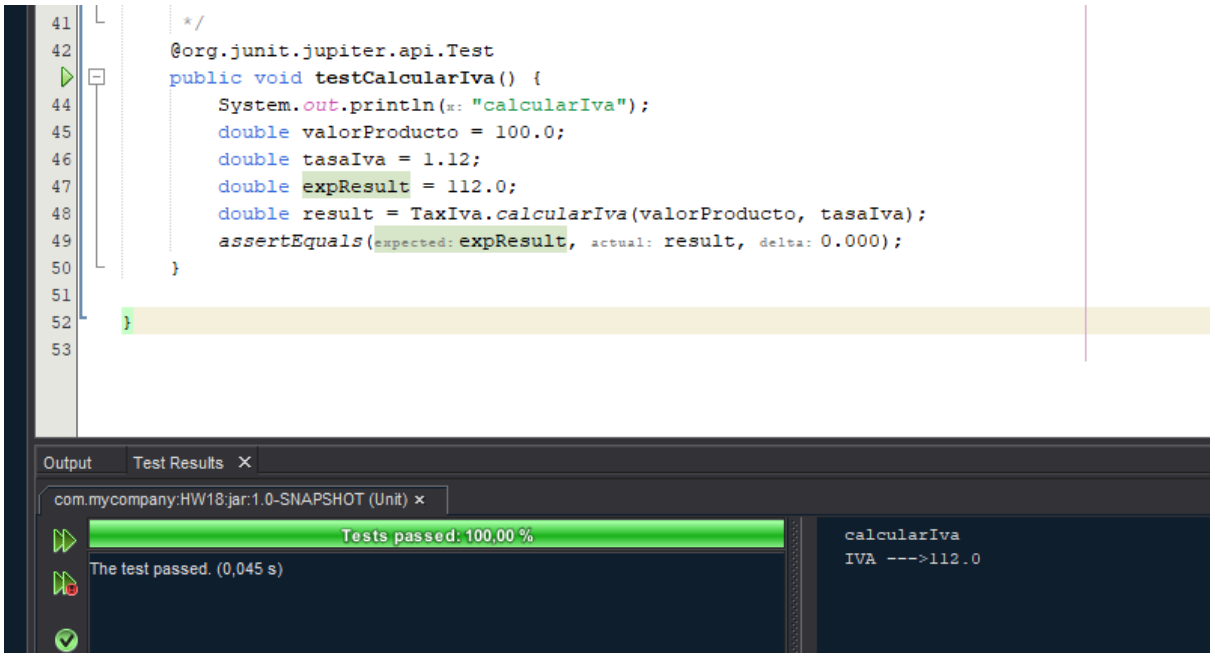
Student: Revilla Antonio

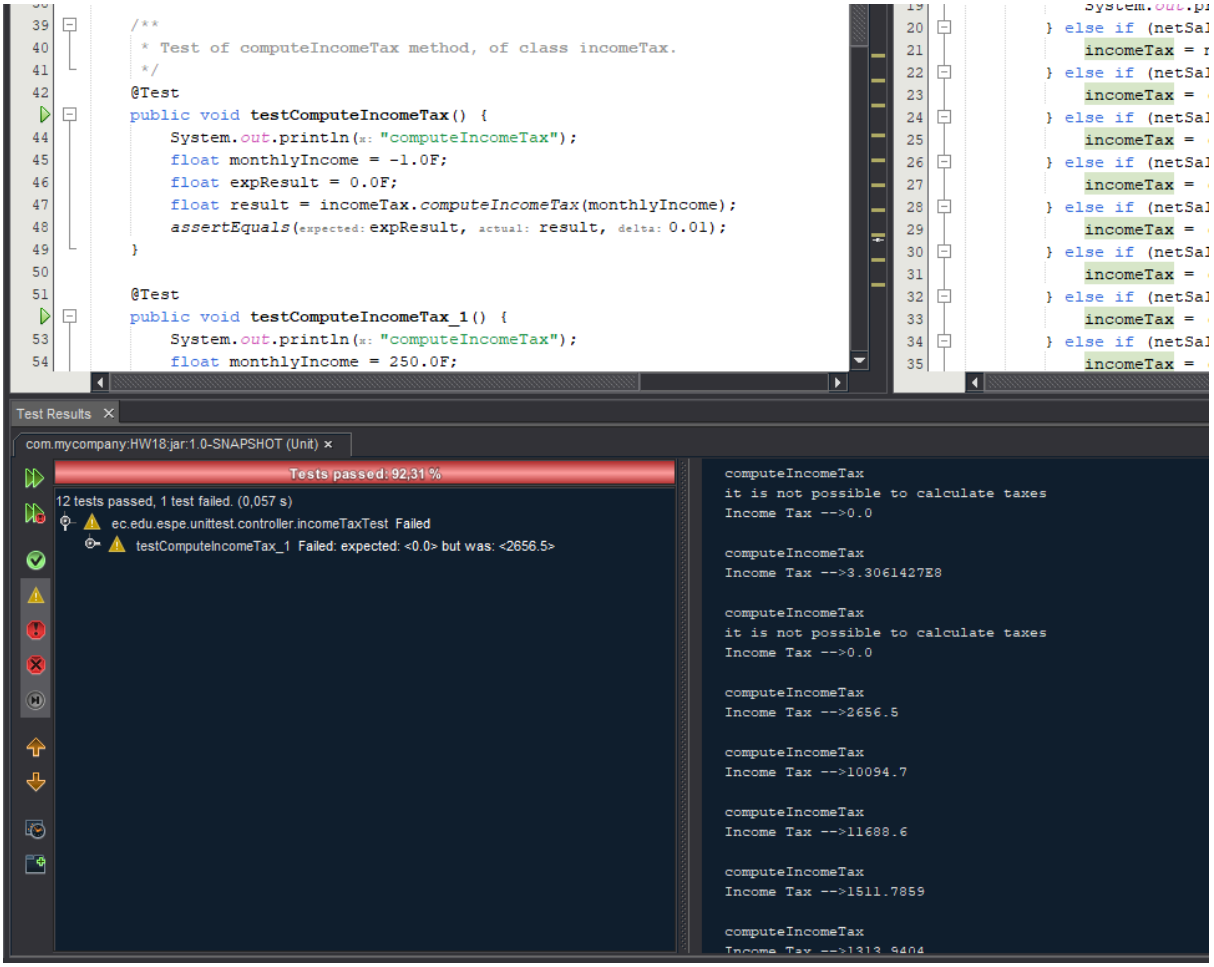
Nrc: 14575

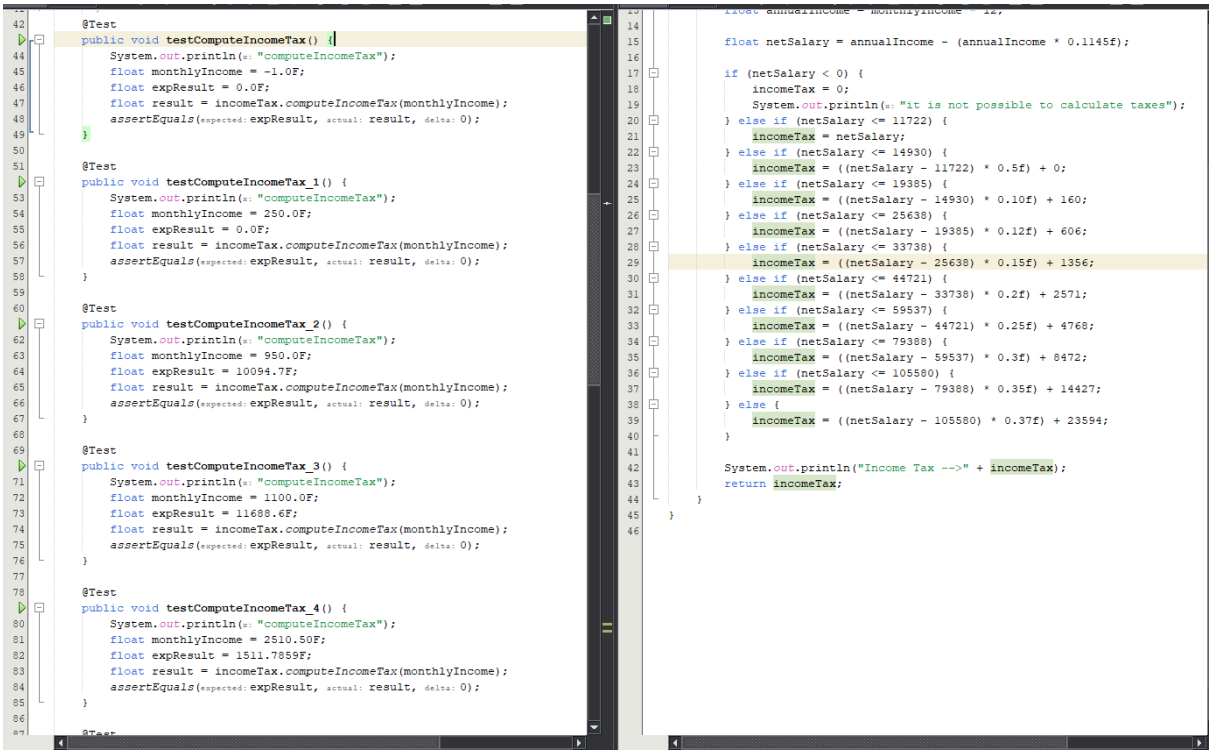
Homework #18

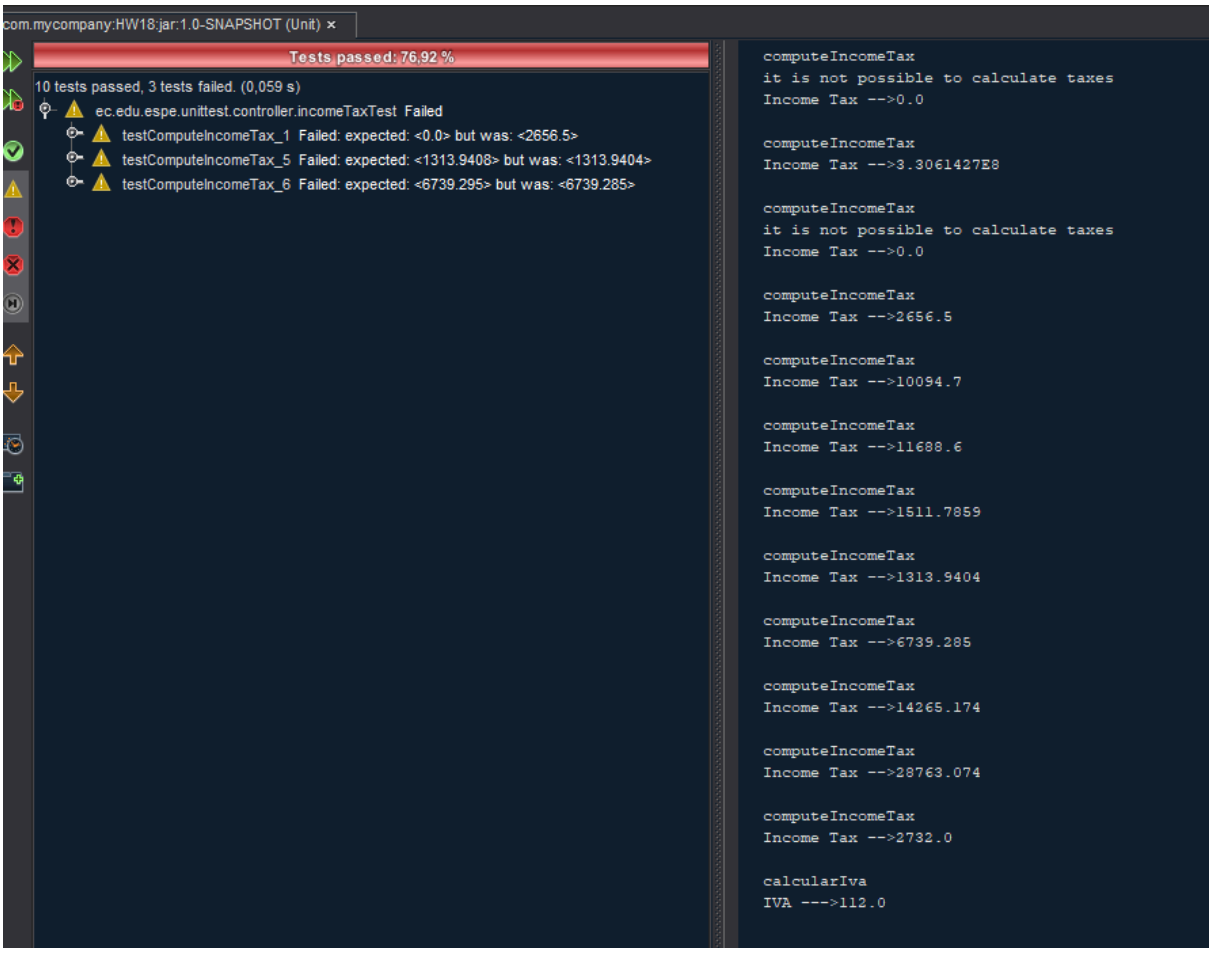
Topic: Unit Test to Taxes Lib

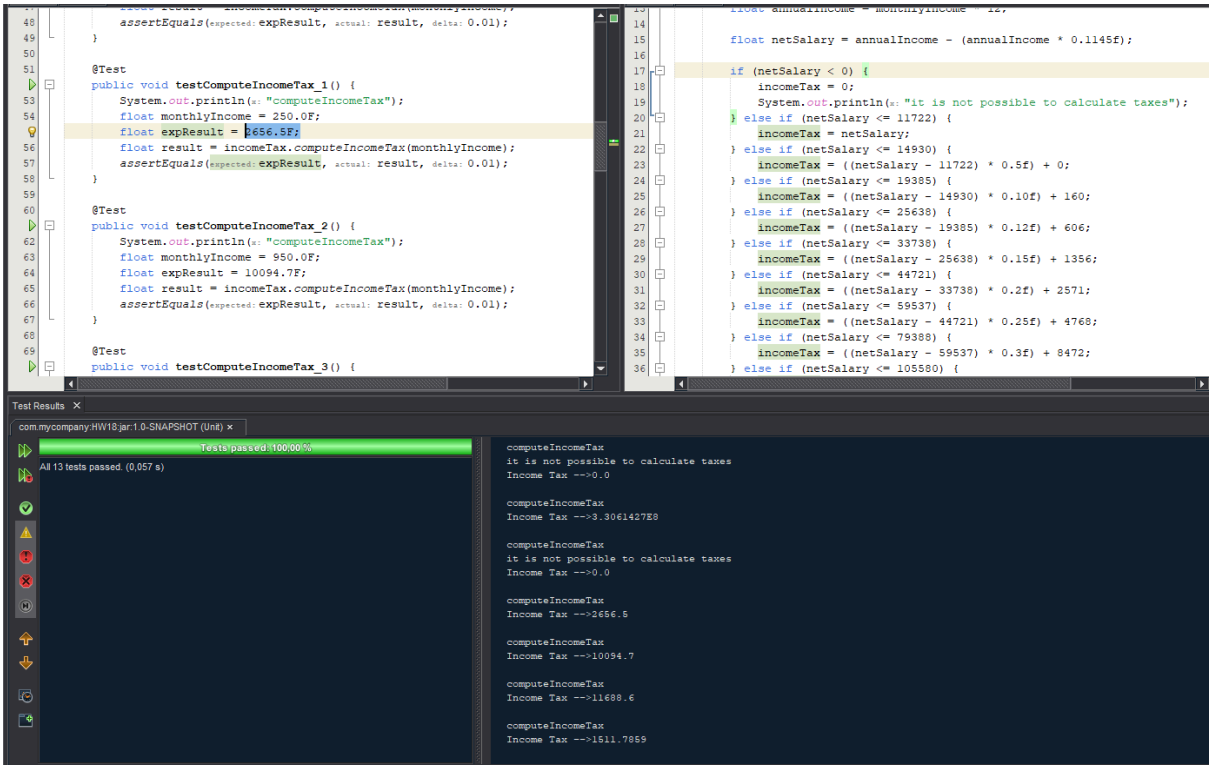

Test Packages
ec.edu.espe.unittest.controller
TaxIvaTest.java
Dependencies
Test Dependencies
Java Dependencies
Project Files
HWmongoDB
JavaMongoDB
Presistence
SystemZoo
TaxLib
*/
public class TaxIva {
public static double calcularIva(double valorProducto, double tasaIva) {
float result;
valorProducto = 100;
tasaIva = 1.12;
result = (float) (valorProducto * tasaIva);
System.out.println("IVA --->" + result);
return result;
}
}
Output - Run (HW18)
cd C:\Users\revil\Documents\NetBeansProjects\Unit2\WS34Persistence\HW18; "JAVA_HOME=C:\\Program Files (x86)\\Java 21" cmd /c "\"C:\\Program Files\\NetBeans-19\\netbea
Scanning for projects...
----------------------< com.mycompany:HW18 >----------------------
Building HW18 1.0-SNAPSHOT
from pom.xml
--------------------------------[jar]---------------------------------
--- resources:3.3.1:resources (default-resources) @ HW18 ---
skip non existing resourceDirectory C:\Users\revil\Documents\NetBeansProjects\Unit2\WS34Persistence\HW18\src\main\resources
--- compiler:3.11.0:compile (default-compile) @ HW18 ---
Nothing to compile - all classes are up to date
--- exec:3.1.0:exec (default-cli) @ HW18 ---
IVA --->112.0


*/
@org.junit.jupiter.api.Test
public void testCalcularIva() {
System.out.println(x: "calcularIva");
double valorProducto = 100.0;
double tasaIva = 1.12;
double expResult = 112.0;
double result = TaxIva.calcularIva(valorProducto, tasaIva);
assertEquals(expected: expResult, actual: result, delta: 0.000);
}
}
Output
Test Results
com.mycompany:HW18:jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
The test passed. (0,045 s)
calcularIva
IVA --->112.0


Test Results
com.mycompany:HW18:jar:1.0-SNAPSHOT (Unit)
Tests passed: 92,31 %
12 tests passed, 1 test failed. (0,057 s)
ec.edu.espe.unittest.controller.incomeTaxTest Failed
testComputeIncomeTax_1 Failed: expected: <0.0> but was: <2656.5>
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->3.3061427E8
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->2656.5
computeIncomeTax
Income Tax -->10094.7
computeIncomeTax
Income Tax -->11688.6
computeIncomeTax
Income Tax -->1511.7859




com.mycompany:HW18.jar:1.0-SNAPSHOT (Unit)
Tests passed: 76,92 %
10 tests passed, 3 tests failed. (0,059 s)
ec.edu.espe.unittest.controller.incomeTaxTest Failed
testComputeIncomeTax_1 Failed: expected: <0.0> but was: <2656.5>
testComputeIncomeTax_5 Failed: expected: <1313.9408> but was: <1313.9404>
testComputeIncomeTax_6 Failed: expected: <6739.295> but was: <6739.285>
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->3.3061427E8
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->2656.5
computeIncomeTax
Income Tax -->10094.7
computeIncomeTax
Income Tax -->11688.6
computeIncomeTax
Income Tax -->1511.7859
computeIncomeTax
Income Tax -->1313.9404
computeIncomeTax
Income Tax -->6739.285
computeIncomeTax
Income Tax -->14265.174
computeIncomeTax
Income Tax -->28763.074
computeIncomeTax
Income Tax -->2732.0
calcularIva
IVA --->112.0


assertEquals(expected: expResult, actual: result, delta: 0.01);
@Test
public void testComputeIncomeTax_1() {
System.out.println("computeIncomeTax");
float monthlyIncome = 250.0F;
float expResult = 2656.5F;
float result = incomeTax.computeIncomeTax(monthlyIncome);
assertEquals(expected: expResult, actual: result, delta: 0.01);
@Test
public void testComputeIncomeTax_2() {
System.out.println("computeIncomeTax");
float monthlyIncome = 950.0F;
float expResult = 10094.7F;
float result = incomeTax.computeIncomeTax(monthlyIncome);
assertEquals(expected: expResult, actual: result, delta: 0.01);
@Test
public void testComputeIncomeTax_3() {
float netSalary = annualIncome - (annualIncome * 0.1145f);
if (netSalary < 0) {
incomeTax = 0;
System.out.println("it is not possible to calculate taxes");
} else if (netSalary <= 11722) {
incomeTax = netSalary;
} else if (netSalary <= 14930) {
incomeTax = ((netSalary - 11722) * 0.5f) + 0;
} else if (netSalary <= 19385) {
incomeTax = ((netSalary - 14930) * 0.10f) + 160;
} else if (netSalary <= 25638) {
incomeTax = ((netSalary - 19385) * 0.12f) + 606;
} else if (netSalary <= 33738) {
incomeTax = ((netSalary - 25638) * 0.15f) + 1356;
} else if (netSalary <= 44721) {
incomeTax = ((netSalary - 33738) * 0.2f) + 2571;
} else if (netSalary <= 59537) {
incomeTax = ((netSalary - 44721) * 0.25f) + 4768;
} else if (netSalary <= 79388) {
incomeTax = ((netSalary - 59537) * 0.3f) + 8472;
} else if (netSalary <= 105580) {
Test Results
com.mycompany:HW18.jar:1.0-SNAPSHOT (Unit)
Tests passed: 100,00 %
All 13 tests passed. (0,057 s)
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->3.3061427E8
computeIncomeTax
it is not possible to calculate taxes
Income Tax -->0.0
computeIncomeTax
Income Tax -->2656.5
computeIncomeTax
Income Tax -->10094.7
computeIncomeTax
Income Tax -->11688.6
computeIncomeTax
Income Tax -->1511.7859