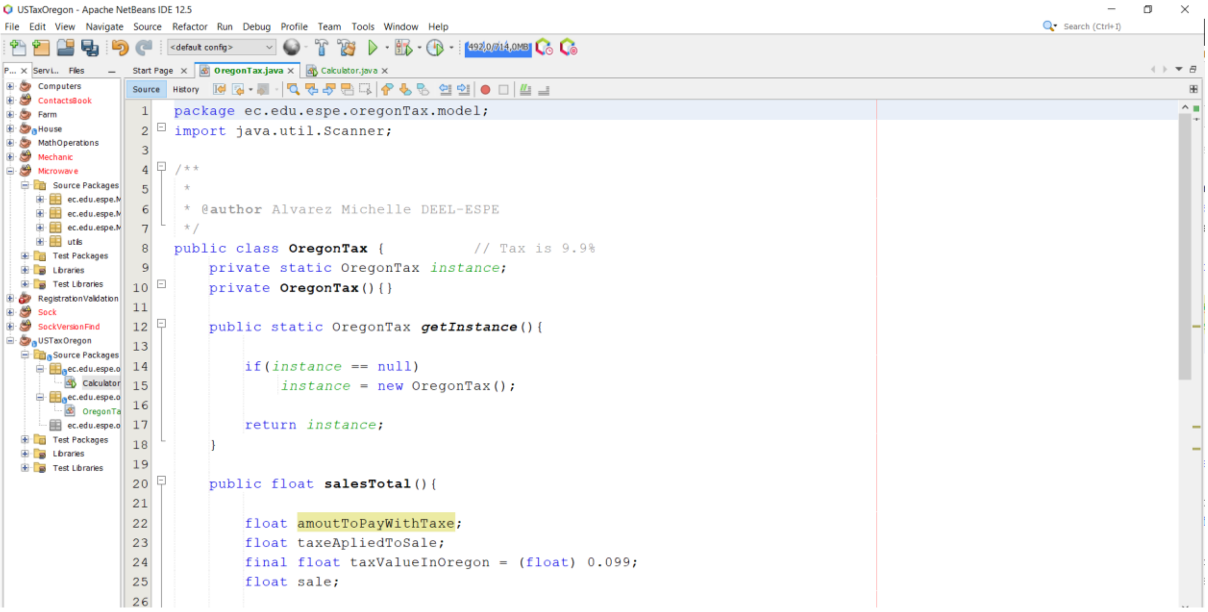
WS34 - SINGLETON PATTERN

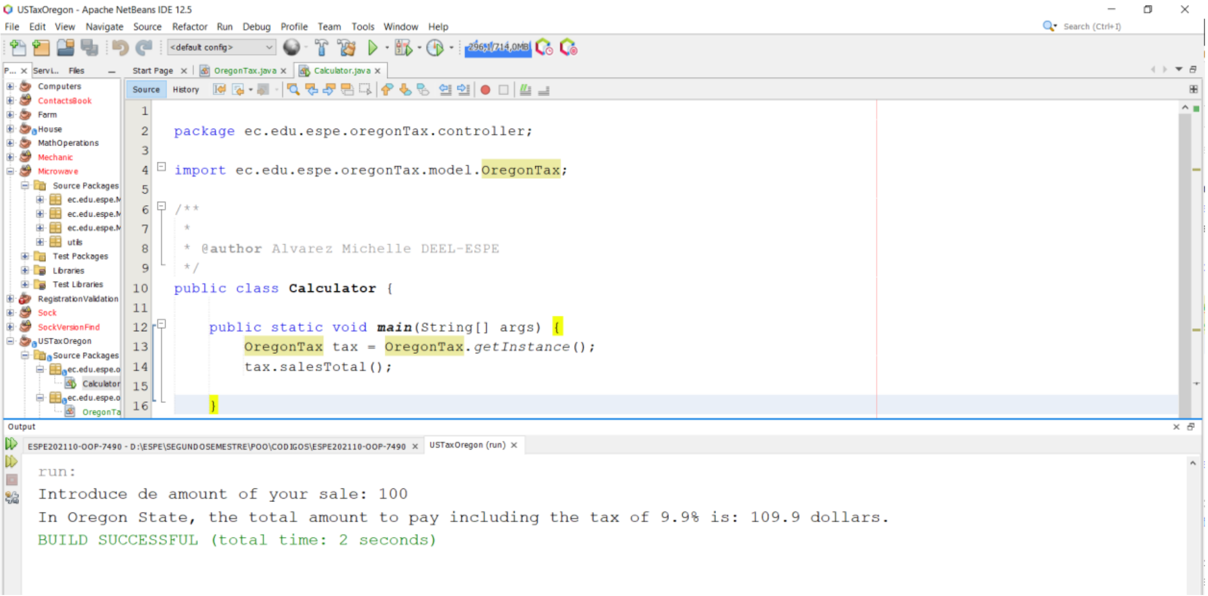
DATE: 15th february 2022

3   ALVAREZ RAMIREZ MICHELLE ESTEFANIA

A single Pattern is one of the simplest design patterns in Java. It makes sure that **a class only has one instance**. This class provides a way to access its only object which can be accessed directly without need to instantiate the object of the class.


USTaxOregon - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Start Page
OregonTax.java
Calculator.java
Source
History

```
package ec.edu.espe.oregonTax.model;
import java.util.Scanner;

/**
 *
 * @author Alvarez Michelle DEEL-ESPE
 */
public class OregonTax {          // Tax is 9.9%
    private static OregonTax instance;
    private OregonTax(){}

    public static OregonTax getInstance(){

        if(instance == null)
            instance = new OregonTax();

        return instance;
    }

    public float salesTotal(){

        float amoutToPayWithTaxe;
        float taxeApliedToSale;
        final float taxValueInOregon = (float) 0.099;
        float sale;
```


USTaxOregon - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Start Page
OregonTax.java
Calculator.java
Source
History
Output
USTaxOregon (run)

```

package ec.edu.espe.oregonTax.controller;

import ec.edu.espe.oregonTax.model.OregonTax;

/**
 *
 * @author Alvarez Michelle DEEL-ESPE
 */
public class Calculator {

    public static void main(String[] args) {
        OregonTax tax = OregonTax.getInstance();
        tax.salesTotal();

    }
```

```
run:
Introduce de amount of your sale: 100
In Oregon State, the total amount to pay including the tax of 9.9% is: 109.9 dollars.
BUILD SUCCESSFUL (total time: 2 seconds)
```