/\*\*

\*

\* @group of ZhenhuaYang, YingyaLi, AmrithaRaveendran

\*/

import java.text.DecimalFormat;

public class SalesTax {

public static void main( String [] args ){

DecimalFormat twoDecimals = new DecimalFormat("0.00");

// i. Create a 2D double array variable named US-States

// and assign it the value returned by create2DArray() static method.

double [][] usStates = create2DArray();

// ii. Print the string returned by getString() static method

System.out.println( "\nii.Print the string: \n" + getString(usStates));

// iii. Print the string returned by taxRatesForSpecificState()

System.out.println( "\niii.taxRatesForSpecificState: " + taxRatesForSpecificState( usStates, 6));

// iv. Print the string returned by taxRatesForSpecificYear()

System.out.println( "\niv.taxRatesForSpecificYear: \n" + taxRatesForSpecificYear( usStates, 7));

System.out.println();

// v.Print the value returned by stateWithHighestTaxRate()

System.out.println( "\nv.State with Highest Tax Rate: \n" + stateWithHighestTaxRate(usStates));

System.out.println();

// vi.  Create an int array variable statesWithLowSalesTaxRate

// and assign it the value returned by statesWithLowSalexTaxRate()

int [] statesWithLowSalesTaxRate = statesWithLowSalexTaxRate(usStates);

// Print the array

System.out.println("\nvi.States with Low Sales Tax Rate: ");

for( int i = 0; i < statesWithLowSalesTaxRate.length; i++ ){

System.out.println(statesWithLowSalesTaxRate[i]);

}

// vii. Create a double array variable statesHighestSalesTaxRate

// and assign it the value returned by statesHighestSalesTaxRate ().

double [] statesHighestSalesTaxRate = statesHighestSalesTaxRate(usStates);

// Print the array

System.out.println("\nvii.Highest Sales Tax Rate of each state over the 10 years: ");

for( int i = 0; i < statesHighestSalesTaxRate .length; i++ ){

System.out.println(twoDecimals.format(statesHighestSalesTaxRate [i]));

}

}

/\*\*

\* the static method that create a 2D Array.

\* @return a 2D Array with double elements.

\*/

public static double[][] create2DArray(){

double [][] taxRates = new double[50][10];

for( int i = 0; i < taxRates.length; i++){

for( int j = 0; j < taxRates[i].length; j++ ){

taxRates[i][j] = Math.random()\*0.06 ;

}

}

return taxRates;

}

/\*\*

\* the method that return all the elements in the 2D array in a String.

\* @param array the array that is going to be returned as a String.

\* @return all the elements of the array in a string.

\*/

public static String getString( double [][] array){

// create a DecimalFormat object

DecimalFormat twoDecimal = new DecimalFormat("0.00");

String str = "";

for( int i = 0; i < array.length; i++){

for( int j = 0; j < array[i].length; j++ ){

str += twoDecimal.format(array[i][j]) + "\t" ; // format the elements in the 2D-array

}

str += "\n";

}

return str;

}

/\*\*

\* The method that return the tax rates in each year of selected state.

\* @param array the 2-d array that will passed into the method

\* @param stateIndex the index of the selected state.

\* @return a string of the tax rates in each year.

\*/

public static String taxRatesForSpecificState( double [][] array, int stateIndex ){

String str = "";

// create a DecimalFormat object

DecimalFormat twoDecimal = new DecimalFormat("0.00");

for( int i = 0; i < array[stateIndex].length; i++ ){

str += twoDecimal.format( array[stateIndex][i] )+ "\t";

}

return str;

}

/\*\*

\* The method that returns a string of tax rate of each state in a certain year

\* @param array the 2-d array that will passed into the method

\* @param yearIndex the index of the selected year.

\* @return string of tax rate of each state in a certain year

\*/

public static String taxRatesForSpecificYear( double [][] array, int yearIndex ){

String str = "";

// create a DecimalFormat object

DecimalFormat twoDecimal = new DecimalFormat("0.00");

for( int i = 0; i < array.length; i++ ){

if( i % 10 == 0 && i !=0 )

str += twoDecimal.format(array[i][yearIndex]) + "\n";

else

str += twoDecimal.format(array[i][yearIndex]) + "\t";

}

return str;

}

/\*\*

\* The method that return the index of the state that has the highest tax rate in the array.

\* @param array the 2-d array that includes the tax rates of states over 10 years.

\* @return the index of the state that has the highest tax rate.

\*/

public static int stateWithHighestTaxRate( double [][] array ) {

double maxAvg = 0;

double sumYear = 0;

int maxIndex = 0;

for( int i = 0; i < array.length; i++ ){

for( int j = 0; j < array[i].length; j++ ){

sumYear += array[i][j];

}

if( maxAvg < sumYear / array[i].length ){

maxIndex = i;

maxAvg = sumYear / array[i].length;

}

}

return maxIndex;

}

/\*\*

\* method that find the states with Low Sales Tax Rate

\* @param array the 2-d array that includes the tax rates of states over 10 years.

\* @return the array of index of states with Low Sales Tax Rate

\*/

public static int[] statesWithLowSalexTaxRate( double [][] array ){

int[] arr = new int [array.length];

int count = 0;

for( int i = 0; i < array.length; i++ ){

for( int j = 0; j < array[i].length; j++ ){

if( array[i][j] < 0.001 ){

arr[count] = i;

count++;

break;

}

}

}

int [] temp = new int[count];

for( int i = 0; i < count; i++ ){

temp[i] = arr[i];

}

return temp;

}

/\*\*

\* the method that finds the highest Sales Tax Rate of each state over the 10 years

\* @param array the 2-d array that includes the tax rates of states over 10 years.

\* @return the array of tax rate of each state over the 10 years.

\*/

public static double[] statesHighestSalesTaxRate( double [][] array ){

double[] arr = new double[array.length];

double max = 0;

for(int i = 0; i < array.length; i++){

for( int j = 0; j < array[i].length; j++ ){

if( max < array[i][j]){

max = array[i][j];

}

arr[i]= max;

max = 0;

}

}

return arr;

}

}

Run 1:

ii.Print the string:

0.00 0.03 0.02 0.04 0.04 0.01 0.05 0.04 0.02 0.01

0.05 0.02 0.00 0.05 0.04 0.05 0.01 0.05 0.01 0.05

0.05 0.03 0.05 0.04 0.05 0.05 0.04 0.04 0.02 0.06

0.03 0.00 0.04 0.03 0.02 0.04 0.02 0.00 0.02 0.01

0.06 0.01 0.03 0.02 0.06 0.02 0.04 0.02 0.05 0.04

0.05 0.02 0.03 0.01 0.01 0.05 0.03 0.02 0.03 0.03

0.04 0.05 0.05 0.05 0.02 0.04 0.04 0.04 0.06 0.03

0.03 0.03 0.04 0.01 0.03 0.05 0.03 0.03 0.01 0.05

0.02 0.01 0.04 0.05 0.01 0.03 0.03 0.04 0.05 0.03

0.03 0.03 0.01 0.04 0.05 0.03 0.04 0.01 0.05 0.01

0.02 0.06 0.02 0.01 0.02 0.01 0.02 0.02 0.01 0.02

0.05 0.02 0.04 0.02 0.02 0.03 0.04 0.06 0.03 0.04

0.04 0.05 0.05 0.02 0.04 0.03 0.03 0.06 0.03 0.02

0.02 0.05 0.06 0.05 0.04 0.02 0.03 0.06 0.05 0.06

0.06 0.05 0.00 0.04 0.03 0.02 0.05 0.05 0.04 0.02

0.04 0.06 0.03 0.04 0.03 0.06 0.02 0.04 0.03 0.02

0.00 0.05 0.00 0.04 0.03 0.05 0.04 0.03 0.05 0.03

0.04 0.03 0.00 0.00 0.01 0.00 0.06 0.05 0.04 0.03

0.01 0.02 0.01 0.03 0.01 0.06 0.04 0.05 0.06 0.03

0.03 0.05 0.02 0.03 0.06 0.02 0.06 0.04 0.05 0.04

0.05 0.03 0.04 0.02 0.05 0.03 0.03 0.06 0.03 0.05

0.05 0.04 0.01 0.01 0.03 0.06 0.02 0.04 0.00 0.04

0.05 0.05 0.06 0.04 0.03 0.02 0.04 0.02 0.01 0.03

0.02 0.01 0.06 0.02 0.01 0.01 0.04 0.06 0.03 0.03

0.01 0.05 0.01 0.06 0.05 0.05 0.01 0.01 0.03 0.01

0.05 0.00 0.05 0.01 0.02 0.01 0.03 0.01 0.00 0.04

0.01 0.06 0.02 0.01 0.04 0.02 0.05 0.06 0.06 0.01

0.04 0.04 0.02 0.06 0.05 0.05 0.02 0.04 0.04 0.00

0.05 0.02 0.00 0.00 0.02 0.05 0.05 0.03 0.05 0.03

0.06 0.03 0.05 0.06 0.02 0.05 0.03 0.00 0.03 0.05

0.05 0.02 0.01 0.04 0.04 0.00 0.04 0.02 0.01 0.03

0.00 0.03 0.02 0.02 0.05 0.05 0.04 0.03 0.05 0.03

0.06 0.01 0.02 0.05 0.04 0.06 0.06 0.06 0.06 0.04

0.05 0.05 0.04 0.05 0.05 0.04 0.01 0.02 0.06 0.02

0.06 0.02 0.06 0.03 0.03 0.00 0.01 0.00 0.05 0.03

0.01 0.03 0.06 0.04 0.01 0.03 0.06 0.04 0.06 0.05

0.03 0.06 0.04 0.04 0.01 0.05 0.04 0.06 0.04 0.04

0.02 0.02 0.03 0.02 0.01 0.04 0.03 0.02 0.06 0.06

0.02 0.03 0.05 0.05 0.05 0.02 0.05 0.05 0.03 0.02

0.05 0.03 0.03 0.01 0.06 0.01 0.03 0.02 0.00 0.05

0.04 0.04 0.02 0.03 0.03 0.03 0.00 0.06 0.01 0.03

0.03 0.02 0.03 0.05 0.02 0.00 0.05 0.01 0.02 0.01

0.03 0.02 0.03 0.03 0.00 0.03 0.01 0.03 0.06 0.04

0.02 0.03 0.02 0.00 0.02 0.03 0.06 0.01 0.01 0.04

0.05 0.01 0.06 0.01 0.05 0.03 0.04 0.04 0.04 0.03

0.02 0.04 0.03 0.00 0.00 0.01 0.01 0.02 0.04 0.06

0.00 0.05 0.00 0.02 0.02 0.01 0.04 0.02 0.02 0.05

0.01 0.01 0.03 0.05 0.02 0.03 0.02 0.01 0.01 0.06

0.04 0.04 0.01 0.05 0.03 0.03 0.02 0.05 0.03 0.02

0.02 0.04 0.01 0.04 0.05 0.03 0.03 0.02 0.05 0.02

iii.taxRatesForSpecificState: 0.04 0.05 0.05 0.05 0.02 0.04 0.04 0.04 0.06 0.03

iv.taxRatesForSpecificYear:

0.04 0.05 0.04 0.00 0.02 0.02 0.04 0.03 0.04 0.01 0.02

0.06 0.06 0.06 0.05 0.04 0.03 0.05 0.05 0.04 0.06

0.04 0.02 0.06 0.01 0.01 0.06 0.04 0.03 0.00 0.02

0.03 0.06 0.02 0.00 0.04 0.06 0.02 0.05 0.02 0.06

0.01 0.03 0.01 0.04 0.02 0.02 0.01 0.05 0.02

v.State with Highest Tax Rate:

49

vi.States with Low Sales Tax Rate:

3

17

31

34

41

45

vii.Highest Sales Tax Rate of each state over the 10 years:

0.01

0.05

0.06

0.01

0.04

0.03

0.03

0.05

0.03

0.01

0.02

0.04

0.02

0.06

0.02

0.02

0.03

0.03

0.03

0.04

0.05

0.04

0.03

0.03

0.01

0.04

0.01

0.00

0.03

0.05

0.03

0.03

0.04

0.02

0.03

0.05

0.04

0.06

0.02

0.05

0.03

0.01

0.04

0.04

0.03

0.06

0.05

0.06

0.02

0.02

BUILD SUCCESSFUL (total time: 1 second)

Run 2:

ii.Print the string:

0.01 0.03 0.01 0.02 0.05 0.04 0.04 0.03 0.00 0.00

0.04 0.04 0.06 0.02 0.04 0.04 0.02 0.02 0.01 0.02

0.04 0.03 0.04 0.04 0.05 0.02 0.05 0.04 0.03 0.03

0.00 0.05 0.02 0.06 0.04 0.06 0.00 0.02 0.06 0.05

0.01 0.05 0.03 0.06 0.05 0.05 0.01 0.05 0.02 0.03

0.00 0.04 0.04 0.05 0.02 0.01 0.02 0.01 0.01 0.00

0.05 0.03 0.04 0.00 0.00 0.01 0.05 0.05 0.05 0.02

0.05 0.06 0.04 0.04 0.03 0.06 0.01 0.06 0.01 0.04

0.00 0.01 0.05 0.03 0.06 0.02 0.02 0.04 0.00 0.04

0.05 0.00 0.01 0.05 0.01 0.04 0.03 0.05 0.06 0.06

0.00 0.04 0.05 0.02 0.04 0.05 0.05 0.02 0.00 0.02

0.01 0.06 0.01 0.03 0.06 0.04 0.02 0.05 0.03 0.04

0.05 0.01 0.00 0.06 0.05 0.02 0.05 0.03 0.01 0.01

0.04 0.01 0.04 0.05 0.03 0.01 0.04 0.01 0.03 0.02

0.03 0.01 0.01 0.05 0.02 0.02 0.02 0.04 0.06 0.00

0.03 0.01 0.00 0.00 0.00 0.02 0.00 0.02 0.01 0.06

0.04 0.00 0.00 0.03 0.04 0.04 0.02 0.03 0.04 0.03

0.02 0.04 0.03 0.01 0.03 0.02 0.03 0.01 0.01 0.00

0.01 0.04 0.04 0.04 0.02 0.01 0.05 0.03 0.06 0.05

0.03 0.03 0.05 0.06 0.01 0.03 0.05 0.00 0.06 0.01

0.03 0.00 0.04 0.04 0.02 0.02 0.01 0.04 0.02 0.03

0.00 0.04 0.02 0.01 0.04 0.03 0.06 0.05 0.00 0.03

0.02 0.00 0.03 0.00 0.01 0.05 0.01 0.00 0.02 0.05

0.01 0.00 0.05 0.05 0.05 0.01 0.01 0.02 0.03 0.02

0.03 0.06 0.03 0.01 0.01 0.05 0.06 0.02 0.01 0.01

0.00 0.01 0.02 0.01 0.02 0.05 0.03 0.04 0.02 0.00

0.04 0.03 0.05 0.06 0.03 0.03 0.03 0.03 0.01 0.03

0.06 0.01 0.02 0.04 0.05 0.04 0.04 0.06 0.04 0.00

0.01 0.02 0.06 0.00 0.04 0.02 0.05 0.05 0.01 0.05

0.02 0.05 0.02 0.06 0.03 0.02 0.00 0.01 0.00 0.06

0.04 0.03 0.03 0.03 0.04 0.06 0.05 0.03 0.05 0.04

0.04 0.04 0.05 0.00 0.02 0.04 0.05 0.01 0.01 0.00

0.03 0.00 0.04 0.02 0.01 0.05 0.05 0.04 0.05 0.03

0.03 0.01 0.02 0.01 0.04 0.01 0.00 0.04 0.03 0.06

0.05 0.05 0.03 0.01 0.05 0.06 0.05 0.05 0.00 0.02

0.03 0.04 0.06 0.03 0.02 0.02 0.04 0.02 0.02 0.00

0.02 0.00 0.04 0.00 0.03 0.05 0.04 0.00 0.05 0.01

0.06 0.05 0.04 0.03 0.05 0.05 0.03 0.01 0.05 0.04

0.04 0.03 0.05 0.02 0.02 0.04 0.02 0.04 0.04 0.02

0.05 0.00 0.00 0.04 0.02 0.02 0.01 0.05 0.05 0.00

0.05 0.04 0.03 0.05 0.06 0.01 0.01 0.02 0.01 0.02

0.01 0.05 0.02 0.02 0.03 0.01 0.05 0.04 0.02 0.04

0.02 0.01 0.05 0.04 0.01 0.02 0.02 0.04 0.05 0.04

0.03 0.00 0.06 0.00 0.04 0.01 0.00 0.00 0.01 0.00

0.05 0.00 0.02 0.03 0.03 0.03 0.02 0.03 0.00 0.02

0.01 0.03 0.05 0.02 0.06 0.03 0.04 0.02 0.02 0.03

0.05 0.06 0.02 0.06 0.00 0.06 0.02 0.02 0.02 0.02

0.00 0.01 0.03 0.03 0.00 0.05 0.05 0.03 0.06 0.02

0.05 0.00 0.03 0.02 0.02 0.00 0.00 0.00 0.00 0.02

0.01 0.00 0.04 0.05 0.00 0.04 0.01 0.03 0.06 0.00

iii.taxRatesForSpecificState: 0.05 0.03 0.04 0.00 0.00 0.01 0.05 0.05 0.05 0.02

iv.taxRatesForSpecificYear:

0.03 0.02 0.04 0.02 0.05 0.01 0.05 0.06 0.04 0.05 0.02

0.05 0.03 0.01 0.04 0.02 0.03 0.01 0.03 0.00 0.04

0.05 0.00 0.02 0.02 0.04 0.03 0.06 0.05 0.01 0.03

0.01 0.04 0.04 0.05 0.02 0.00 0.01 0.04 0.05 0.02

0.04 0.04 0.00 0.03 0.02 0.02 0.03 0.00 0.03

v.State with Highest Tax Rate:

49

vi.States with Low Sales Tax Rate:

6

9

20

22

25

36

39

48

49

vii.Highest Sales Tax Rate of each state over the 10 years:

0.00

0.02

0.03

0.05

0.03

0.00

0.02

0.04

0.04

0.06

0.02

0.04

0.01

0.02

0.00

0.06

0.03

0.00

0.05

0.01

0.03

0.03

0.05

0.02

0.01

0.00

0.03

0.00

0.05

0.06

0.04

0.00

0.03

0.06

0.02

0.00

0.01

0.04

0.02

0.00

0.02

0.04

0.04

0.00

0.02

0.03

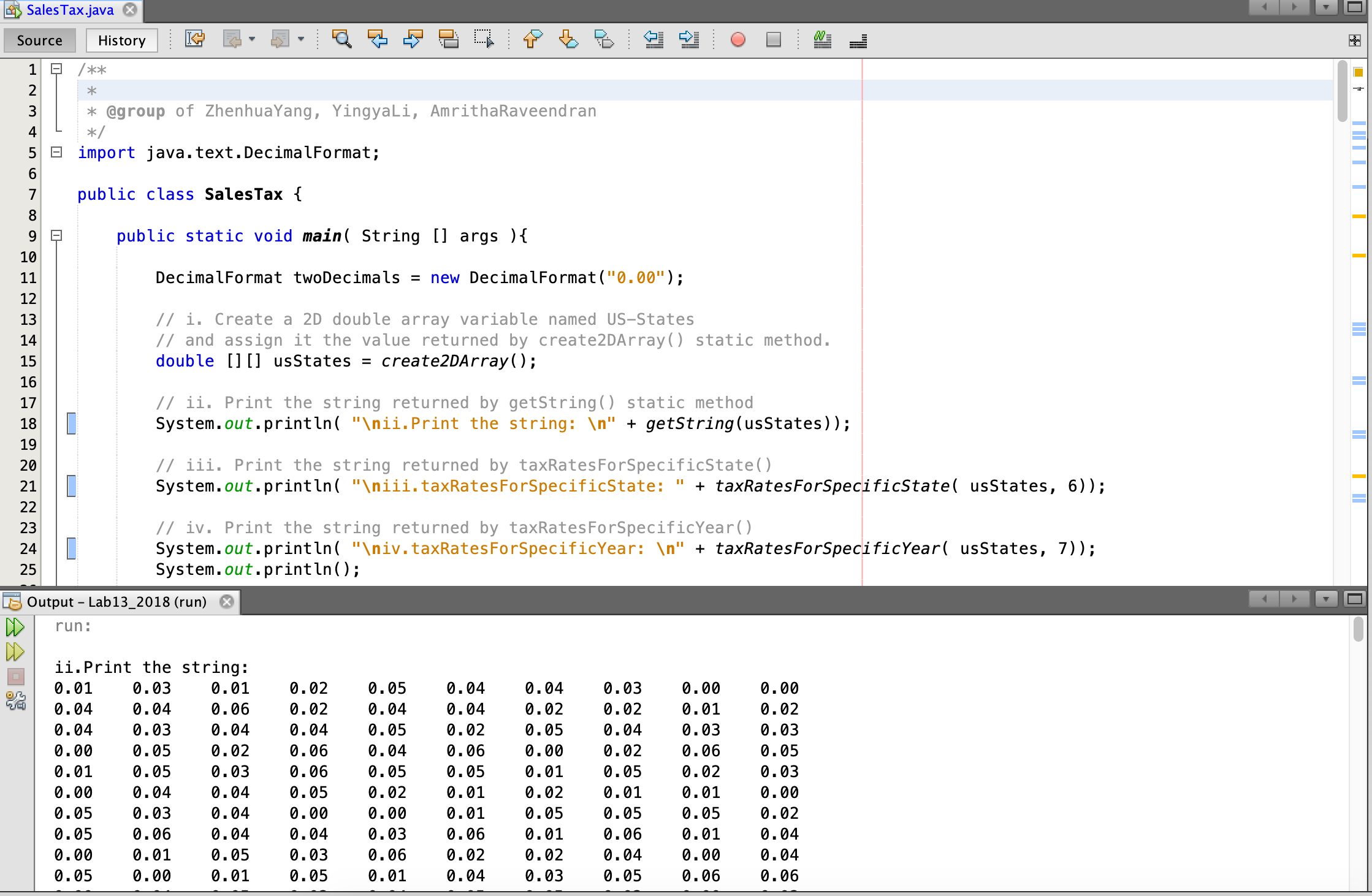
0.02

0.02

0.02

0.00

BUILD SUCCESSFUL (total time: 0 seconds)

****

**Lab Assignment – 13**

**Total Points: 100**

**Due Date: 11/21 (Wednesday), 11.59 pm**

You can work in groups of up to 3 students each on this assignment. Each group will make only one submission. Any student from the group can make the submission along with the names of the group members. All the students in the group will receive the same grade.

**Multi-Dimensional Arrays**

1. Create a Class called SalesTax with the following ***static methods*** in it:

*Please note there are no instance variables and no non-static methods in this class. All the methods in the class are static methods just like the Math class.*

* + ***create2DArray( )*** - public static method that does not take any parameter and returns a 2-Dimensional array of values representing sales tax rates of 50 states for 10 years. Dimension 1 represents the state (there are 50 states) and dimension 2 represents the year. The sales tax values you would populate this structure will be a random value between 0.0 and 0.06 using Math.random( ) method.
  + ***getString( )*** - public static method that takes a 2D double array as a parameter and returns a string containing sales tax rates for all the 10 states for all the 50 states.
  + ***taxRatesForSpecificState( )*** - public static method that takes a 2D double array as a parameter and a int value representing the index of a specific state and returns a string containing the sales tax rates for all the 10 years for a specific state.
  + ***taxRatesForSpecificYear( )*** - public static method that that takes a 2D double array as a parameter and a int value representing the index representing a specific year and returns a string containing the sales tax rates for a specific year for all the 50 states.
  + ***stateWithHighestTaxRate( )*** - public static method that takes a 2D double array as a parameter and returns the index of the state that has the highest average tax rate over the years.
  + ***statesWithLowSalexTaxRate( )*** - public static method that that takes a 2D double array as a parameter and returns an array containing the indexes of the states that have had at least one year with tax rate less than 0.001. Hint: You would need to use a nested for loop.
  + ***statesHighestSalesTaxRate ( )*** - public static method that that takes a 2D double array as a parameter and returns an array containing the highest sales tax rate for each state over the 10 years.
  + Implement a ***main method*** in the same program and do the following inside the main method.
    1. Create a 2D double array variable named US-States and assign it the value returned by ***create2DArray( )*** static method.
    2. Print the string returned by ***getString( )*** static method on the output window.
    3. Print the string returned by ***taxRatesForSpecificState( )*** on the output window.
    4. Print the string returned by ***taxRatesForSpecificYear( )*** on the output window.
    5. Print the value returned by ***stateWithHighestTaxRate( )*** on the output window.
    6. Create an int array variable statesWithLowSalesTaxRate and assign it the value returned by ***statesWithLowSalexTaxRate( )***. Run a for loop on this array to print it’s contents on the output window.
    7. Create a double array variable ***statesHighestSalesTaxRate*** and assign it the value returned by ***statesHighestSalesTaxRate ( )***. Run a for loop on this array to print it’s contents on the output window.

Note: Please note to make your output verbose clearly describing each of the values that you are printing to the output window.

**Things to submit:**

* + In the word document, put the names of the group members at the top.
  + Copy and paste the source code.
  + Copy and paste the output of **two** runs of the program.
  + Copy and paste the screen shot of the program.
* Zip your NetBeans project folder and submit the same with word document.