CSC1021 Project 1 Documentation

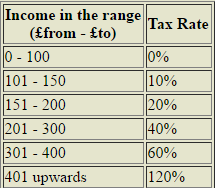
Author: Matas Zilaitis

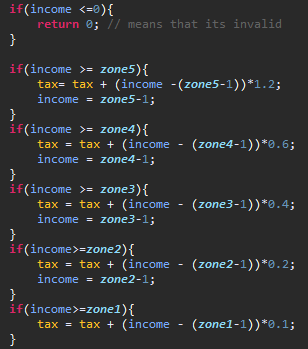
2016-11-09

How the program works:

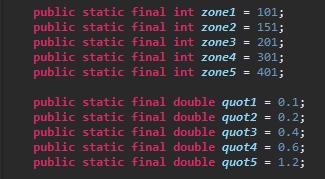
1. We input values into the array
2. In BOGOFF class, draw() method is called.
3. For statement is initialized for every value in the array
4. Draw() method then retrieves values from both methods (taxPayable,incomeLeft) in TaxCalculator class. Meanwhile methods retrieve values from BOGOFF class, array called incomeArray ( see below). Taxes are calculated by the constants given in Blackboard.

How the tax is calculated:





Ranges and tax rates can be changed anytime in TaxCalculator class:



1. Lets say income is 175, that means it’s in range 151-200. We subtract: 200-175 and multiply the difference by the quotient. So 25 \* 0.2 = 5. Calculating in this range is done so we set the income to 150.
2. Now income (150) is in range 101-150. We subtract: 150-100 and multiply the difference by the quotient. So 50 \* 0.1 = 5. Calculating in this range is done so we set the income to 100.
3. Now income is 100, and first 100 pounds are not taxable, therefore tax calculating is done. 5+ 5 =10, that means that if a child earns 175, 10 must be paid in taxes.
4. Values from methods are used to create two bars ( incomeLeftBar,taxes).
5. Taxes bar is put on top of the incomeLeftBar.
6. Repeat steps 4-6 until we reach the end of the array.

Examples

1. First example will include the outputs from array given in Blackboard



Output chart :



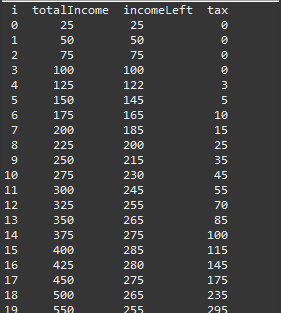
Right here red colour represents tax size

yellow colour - income left after taxes

To represent integer values, we output

Index, income left ( yellow colour in the chart), and taxes (red colour in the chart).

The program automatically rounds up the taxes .



Calculations:

Income: 25, tax = 0. Correct

Income: 50, tax = 0. Correct

Income: 75, tax = 0. Correct

Income: 100, tax = 0. Correct

Income: 125, tax = 3. 25\*0.1 = 2.5 = 3 (rounded up). Correct

Income: 150, tax =5. 50\*0.1 = 5. Correct

Income: 175, tax = 10. 50\*0.1+25\*0.2 = 10. Correct

Income: 200, tax = 15. 50\*0.1+50\*0.2 = 15. Correct

Income: 225, tax = 25. 50\*0.1+50\*0.2+25\*0.4 = 25. Correct

Income: 250, tax = 35. 50 \*0.1+50\*0.2+50\*0.4 = 35. Correct

Income: 275, tax = 45. 50\*0.1+50\*0.2+75\*0.4 = 45. Correct

Income: 300, tax = 55. 50\*0.1+50\*0.2+100\*0.4 = 55. Correct

Income: 325, tax = 70. 50\*0.1+50\*0.2+100\*0.4+25\*0.6 = 70. Correct

Income: 350, tax = 85. 50\*0.1+50\*0.2+100\*0.4+50\*0.6 = 85. Correct

Income: 375, tax = 100. 0\*0.1+50\*0.2+100\*0.4+75\*0.6 = 100. Correct

Income: 400, tax = 115. 50\*0.1+50\*0.2+100\*0.4+100\*0.6 = 115. Correct

Income: 425, tax = 145. 50\*0.1+50\*0.2+100\*0.4+100\*0.6+25\*1.2 = 145. Correct

Income: 450, tax = 175. 50\*0.1+50\*0.2+100\*0.4+100\*0.6+50\*1.2 = 175. Correct

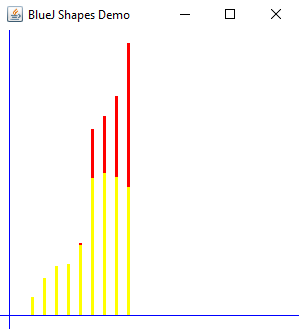
Income: 500, tax = 235. 50\*0.1+50\*0.2+100\*0.4+100\*0.6+100\*1.2 = 235. Correct

Income: 550, tax = 295. 50\*0.1+50\*0.2+100\*0.4+100\*0.6+150\*1.2 = 295. Correct

1. For second example we will delete and add some values, to see if the program adapts to the change of the array size.

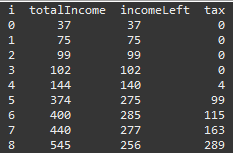


Output chart:



The size is alright, we have 9 elements in array and 9 bars.

Console:



Lets see if it calculates values correctly

Income = 37 , tax = 0. Correct.

Income = 75, tax = 0. Correct

Income = 99, tax = 0. Correct;

Income = 102, tax= 0. Correct, because 2\*0.1 = 0.2 and rounds up to 0.

Income = 144, tax = 4. Correct, because 44\*0.1 = 4.4 and rounds up to 4

Income = 374, tax = 99. Correct, because 50\*0.1+50\*0.2+100\*0.4+74\*0.6 = 99.4 and rounds up to 99

Income = 400, tax = 115. Correct ,because 50\*0.1+50\*0.2+100\*0.4+100\*0.6=115

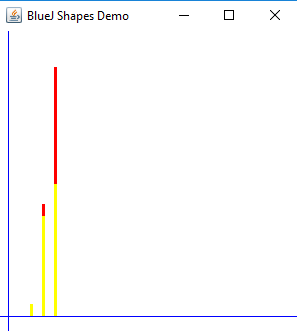
Income = 440, tax = 163. Correct, because 50\*0.1+50\*0.2+100\*0.4+100\*0.6+40\*1.2=163

Income = 545, tax =289. Correct, because 50\*0.1+50\*0.2+100\*0.4+100\*0.6+145\*1.2 = 289

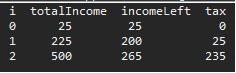
1. For third example, lets add just three values to see how scaling works.



Output chart:



Console:



Everything works fine, array size is 3 and we see 3 bars.

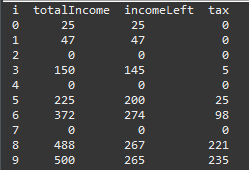
1. For fourth example lets add a few negative numbers and see how the program responds.



Output chart:



Console:



As income can’t be negative, program automatically assigns it to 0 and therefore bar with height 0 is not visible.

1. For fifth and final example lets input a string and see how program responds.



The graph is not outputted because of error(string inside of int array):

