Lance Brown

10/24/19

Exam 2 pseudocode/logic

This has been heavily edited after completing the program compared to when I began. Things got out of hand as I tried to make the program "better" for no real practical reason. I just wanted to mess around with VB and see how to handle things I know how to do in JS. I'd like to clarify some of the decisions I made. I put all of the ranges of employment into a two dimensional rectangular array and the discount values in two other arrays so that the values could be easily changed (as the company discount policy changed or what have you), and it is scalable to accommodate more or fewer tenure brackets. I used the ReadOnly designation because that's the closest I could come up with to a constant for Arrays, but through a little bit of testing it looks like that only protects the array itself, not the values it contains.

My Select Cases for determining the discount bracket are clunky at this scale, so I left the hardcoded Select Case in as a comment block just to show that I can do it within the purview of this class.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Class

Const maxDiscount

ReadOnly employmentRanges()()

ReadOnly hourlyDiscout()

ReadOnly managerDiscount()

Dim dblDailyTotal

Dim dblDailyDiscountedTotal

Sub exit

Close()

Sub NextEmployee

Clear output fields

Clear Input fields

Focus Name

Sub Calculate

Reset error colors

Dim strName As String

Dim intYears As Integer

Dim dblYtdPurchase As Double

Dim dblPurchase As Double

Dim blnHourly As Boolean

Dim dblDiscount As Double

Dim dblYtdDiscount As Double

Dim dblDiscountDollars As Double

Dim dblDiscountedTotal As Double

Dim dblOverage As Double

Validate inputs, if fail, message box, error color, focus and end sub

strName = txtName

intYears = txtYears

dblYtdPurchase = txtYtdPurchase

dblPurchase = txtPurchase

blnHourly = radHourly.Checked

Increment Daily Running Total by dblPurchase

If Hourly Then

For each tenure range

Select Case intYears

Case rangeMin To rangeMax

Discount = hourlyDiscount(range.index)

Else

Do the same for management discount array

Calculate YTD discount to account for max discount before applying to purchase

If dblYtdDiscount > maxDiscount then dblYtdDiscount = maxDiscount (you couldn’t have gotten more than the max even if your previous purchases exceed the max)

Calculate dollar amount of discount

If dblDiscount + dblYtdDiscount >= maxDiscount Then

Find by how much it exceeds the max, and subtract the difference from dblDiscount

dblDiscountedTotal = dblPurchase – dblDiscoutDollars

Increment daily discounted total by discounted total

Display everything

Sub Summary

Display running totals

Testing:

Input: nn, 0, 0, hourly, 100

Expected Output: nn, 0, 0, 100, 0, 100

Output: nn, 0, 0, 100, 0, 100

Input: nn, 1, 0, hourly, 100

Expected Output: nn, 10, 0, 100, 10, 90

Output: nn, 10, 0, 100, 10, 90

Input: nn, 5, 0, hourly, 100

Expected Output: nn, 14, 0, 100, 14, 86

Output: nn, 14, 0, 100, 14, 86

Input: nn, 25, 0, hourly, 100

Expected Output: nn, 30, 0, 100, 30, 70

Output: nn, 30, 0, 100, 30, 70

Input: nn, 1, 0, management, 100

Expected Output: nn, 20, 0, 100, 20, 80

Output: nn, 20, 0, 100, 20, 80

Input: nn, 1, 100, management, 100

Expected Output: nn, 20, 20, 100, 20, 80

Output: nn, 20, 20, 100, 20, 80

Input: nn, 1, 100, hourly, 100

Expected Output: nn, 10, 10, 100, 10, 90

Output: nn, 10, 10, 100, 10, 90

Input: nn, 20, 400, management, 200

Expected Output: nn, 40, 160, 200, 40, 160

Output: nn, 40, 160, 200, 40, 160

Input: nn, 20, 10000, management, 100

Expected Output: nn, 40, 200, 100, 0, 100

Output: nn, 40, 200, 100, 0, 100