' Michael Voell

' Aluminum Recycling Campaign

' 1/14/2016

' Purpose: Use subprocedures to calculate different types of equations.

Option Strict On

Public Class frmAluminumRecycling

' Class variables

Private \_strCansNum As String = "0,000"

Private \_strCansText As String = "cans need to be collected to reach your goal"

Private \_strAmountNum As String = "$0,000"

Private \_strAmountText As String = "will be earned by collecting cans for recycling"

Private \_strGoalAmount As String = "Target Goal Amount: "

Private \_strCansCollected As String = "Cans Collected: "

Private Sub frmAluminumRecycling\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

Threading.Thread.Sleep(4000)

End Sub

Private Sub btnClear\_Click(sender As System.Object, e As System.EventArgs) Handles btnClear.Click

cboRecyclingNeed.Text = "Select an Item"

txtAmount.Text = ""

lblOutput.Text = \_strCansNum

lblOutputText.Text = \_strCansText

lblGoalAndCans.Text = \_strGoalAmount

End Sub

Private Sub cboRecyclingNeed\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles cboRecyclingNeed.SelectedIndexChanged

' Variables

Dim intRecyclingNeed As Integer

' Getting case information

intRecyclingNeed = cboRecyclingNeed.SelectedIndex

Select Case intRecyclingNeed

Case 0

CansNeeded()

Case 1

AmountEarned()

End Select

End Sub

Private Sub CansNeeded()

' Setting the output labels to the number and description respectively.

lblOutput.Text = \_strCansNum

lblOutputText.Text = \_strCansText

lblGoalAndCans.Text = \_strGoalAmount

txtAmount.Focus()

End Sub

Private Sub AmountEarned()

' Setting the output labels to the number and description respectively.

lblOutput.Text = \_strAmountNum

lblOutputText.Text = \_strAmountText

lblGoalAndCans.Text = \_strCansCollected

txtAmount.Focus()

End Sub

Private Sub btnCalculate\_Click(sender As System.Object, e As System.EventArgs) Handles btnCalculate.Click

' Variables

Dim intAmountWanted As Integer

Dim intRecyclingNeed As Integer

Dim intTargetTotal As Integer

Dim intFindAmountTotal As Integer

' Getting case information

intRecyclingNeed = cboRecyclingNeed.SelectedIndex

Select Case intRecyclingNeed

Case 0

CansNeeded()

intTargetTotal = CInt(FindTargetAmount(intAmountWanted))

lblOutput.Text = intTargetTotal.ToString()

Case 1

AmountEarned()

intFindAmountTotal = CInt(FindAmountEarned(intAmountWanted))

lblOutput.Text = intFindAmountTotal.ToString("C")

End Select

End Sub

Private Function FindTargetAmount(ByVal intCanAmount As Integer) As Decimal

' Equation will be: 24 \* (Amount Wanted ($) / 0.75)

' Variables

Dim intAmountWanted As Integer

Dim strAmountWanted As String

Dim intTargetTotal As Integer

'Try / Catch exceptions

Try

' Getting information from text box

strAmountWanted = txtAmount.Text.ToString()

' Conversions

intAmountWanted = Convert.ToInt32(strAmountWanted)

' Calculations

intTargetTotal = CInt(24 \* (intAmountWanted / 0.75))

Catch ex As DivideByZeroException

MsgBox("Attempting to divide by 0. Not allowed.")

CansNeeded()

Catch ex As OverflowException

MsgBox("The number was too large.")

CansNeeded()

Catch ex As FormatException

MsgBox("The input was not a number. Please use a number.")

CansNeeded()

Catch ex As Exception

MsgBox("General exception has been handled. Please try again.")

CansNeeded()

End Try

' Return intTargetTotal to main sub

Return intTargetTotal

End Function

Private Function FindAmountEarned(ByVal intCansCollectedTotal As Integer) As Decimal

' Equation will be: 0.75 \* (Cans Collected / 24)

' Variables

Dim strCansCollected As String

Dim intCansCollected As Integer

Dim intCansTotal As Integer

'Try / Catch exceptions

Try

' Getting information for the string

strCansCollected = txtAmount.Text.ToString()

' Conversions

intCansCollected = Convert.ToInt32(strCansCollected)

' Calculations

intCansTotal = CInt(0.75 \* (intCansCollected / 24))

Catch ex As DivideByZeroException

MsgBox("Attempting to divide by 0. Not allowed.")

AmountEarned()

Catch ex As OverflowException

MsgBox("The number was too large.")

AmountEarned()

Catch ex As FormatException

MsgBox("The input was not a number. Please use a number.")

AmountEarned()

Catch ex As Exception

MsgBox("General exception has been handled. Please try again.")

AmountEarned()

End Try

' Return intCansTotal to main sub

Return intCansTotal

End Function

End Class

