BeckAddisonChapter3Lab

Public Class Stock

'Programmed By: Addison Beck

'Chapter 3 Lab

'Create a stock class

Implements IComparable

'With the following properties:

Private m\_stockSymbol As String 'Stock Symbol (String)

Private m\_price As Double 'Price (Double)

Private m\_earning As Double 'Earnings (Double)

Public Property StockSymbol() As String

Get

Return m\_stockSymbol

End Get

Set(value As String)

m\_stockSymbol = value

End Set

End Property

Public Property Price() As Double

Get

Return m\_price

End Get

Set(value As Double)

m\_price = value

End Set

End Property

Public Property Earnings() As Double

Get

Return m\_earning

End Get

Set(value As Double)

m\_earning = value

End Set

End Property

'Create a default constructor that initializes these values to 0 or empty strings

Public Sub New()

m\_stockSymbol = ""

m\_price = 0

m\_earning = 0

End Sub

'Create an additional constructor that initializes these values based on user input

Public Sub New(ByRef p\_stockSymbol, ByRef p\_price, ByRef p\_earnings)

m\_stockSymbol = p\_stockSymbol

m\_price = p\_price

m\_earning = p\_earnings

End Sub

'Create a ReadOnly method/property named PeRatio that returns the stock’s price divided by the earnings

Public ReadOnly Property PeRatio() As Double

Get

Return (m\_price / m\_earning)

End Get

End Property

'Sorts by ticker

Private Function IComparable\_CompareTo(obj As Object) As Integer Implements IComparable.CompareTo

Return m\_stockSymbol.CompareTo(CType(obj, Stock).m\_stockSymbol)

End Function

End Class

Public Class HomePage

'Create an application that sorts a list of Stock objects based on:

' Ticker Symbol

' Price-to-Earnings (PE) Ratio (Allow the user can select whether to sort in ascending Or descending order

' you will need comparator methods For the List.Sort)

Dim Stock As New Stock

Dim stockList As New List(Of Stock)()

'Use the following sample stocks (you can build a list of Stock objects in the Form\_Load event)

Private Sub HomePage\_Load\_1(sender As Object, e As EventArgs) Handles MyBase.Load

stockList.Add(New Stock() With {

.StockSymbol = "MMA",

.Price = 40,

.Earnings = 5})

stockList.Add(New Stock() With {

.StockSymbol = "XYZ",

.Price = 77.5,

.Earnings = 4.25})

stockList.Add(New Stock() With {

.StockSymbol = "LLT",

.Price = 43.25,

.Earnings = 5.5})

stockList.Add(New Stock() With {

.StockSymbol = "SBA",

.Price = 42,

.Earnings = 7.1})

For Each Stock In stockList

dgvStocks.Rows.Add(Stock.StockSymbol, Stock.Price, Stock.Earnings, Stock.PeRatio)

Debug.Write(Stock.StockSymbol)

Next

End Sub

'Sorts by price

Public Function ComparePrice(ByVal x As Stock, ByVal y As Stock) As Integer

Return x.Price.CompareTo(y.Price)

End Function

'Sorts by PE

Public Function ComparePeRation(ByVal x As Stock, ByVal y As Stock) As Integer

Return x.PeRatio.CompareTo(y.PeRatio)

End Function

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

dgvStocks.Rows.Clear()

cmbOrder.Text = ""

stockList.Sort()

If cmbOrder.Text = "Dec" Then 'Reverses list based on cmb input

stockList.Reverse()

End If

For Each Stock In stockList

dgvStocks.Rows.Add(Stock.StockSymbol, Stock.Price, Stock.Earnings, Stock.PeRatio)

Debug.Write(Stock.StockSymbol)

Next

End Sub

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

dgvStocks.Rows.Clear()

stockList.Sort(AddressOf ComparePrice)

If cmbOrder.Text = "Dec" Then 'Reverses list based on cmb input

stockList.Reverse()

End If

For Each Stock In stockList

dgvStocks.Rows.Add(Stock.StockSymbol, Stock.Price, Stock.Earnings, Stock.PeRatio)

Debug.Write(Stock.StockSymbol)

Next

End Sub

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

dgvStocks.Rows.Clear()

stockList.Sort(AddressOf ComparePeRation)

If cmbOrder.Text = "Dec" Then 'Reverses list based on cmb input

stockList.Reverse()

End If

For Each Stock In stockList

dgvStocks.Rows.Add(Stock.StockSymbol, Stock.Price, Stock.Earnings, Stock.PeRatio)

Debug.Write(Stock.StockSymbol)

Next

End Sub

End Class

