Module BookRUs

Sub Main()

'------------------------------------------------------------

'- Subprogram Name: Main -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine make sure that user is can interact with the console

'- get input from the user then print eveything that have in the console's report

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- strPath -input file path from use

'------------------------------------------------------------

'Set up the Console Screen

Console.WindowWidth = 100 'width console is 100

Console.Title = "Book 'R Us"

Console.WriteLine("Books 'R Us".PadLeft(50) + vbCrLf +

"---------------------------------".PadLeft(60) + vbCrLf + vbCrLf +

"Please enter the path and name of the file to process:")

'Request the path to txt file

Dim strPath As String = Console.ReadLine() 'request input from user

If pathValidation(strPath) = False Then

GoTo ExitConsole

End If

Console.WriteLine("Processing Completed..." + vbCrLf + vbCrLf)

'call printReport sub program by passing the path

PrintReport(strPath)

ExitConsole:

If pathValidation(strPath) = False Then

Console.WriteLine("Inapproriate Input")

Console.ReadLine()

End If

End Sub

Sub PrintReport(strPath As String)

'------------------------------------------------------------

'- Subprogram Name: PrintReport -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine print out the report using the inpiut file path of user

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- strPath -input file path from use

'- objReader - open and read the file using the path

'-llBooks - linked list of the obj books

'------------------------------------------------------------

'initiate object Reader

Dim objReader As System.IO.StreamReader

objReader = System.IO.File.OpenText(strPath) 'take in the pass in value of the file path

Dim llBooks As New LinkedList(Of clsBooks)

'while there are line in the file, read the line then,

'spit it to array of string, padd it in constructor then

'add it to linked list

While objReader.Peek() > -1

Dim bookInfo = objReader.ReadLine()

Dim info() As String = bookInfo.Split()

Dim objBook As clsBooks = New clsBooks(info)

llBooks.AddLast(objBook)

End While

'calling different report subprogram

InventoryReport(llBooks)

TotalInventoryValue(llBooks)

CategoryStat(llBooks)

OverallBookStats(llBooks)

'after eveything done, wait for user input

Console.WriteLine(vbCrLf + vbCrLf + "Thank you for using Book 'R Us Report")

Console.WriteLine("Click any key to exit...")

Console.ReadLine()

End Sub

Sub InventoryReport(llBooks As LinkedList(Of clsBooks))

'------------------------------------------------------------

'- Subprogram Name: InventoryReport -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine print out the invenory report using the lineked list

'- printing out all the books that are available in the library

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- report - LINQ result

'-strLineFormat - String format when print out output

'-llBooks - linked list of the obj books

'------------------------------------------------------------

'linq result of all the book ordered

Dim report = From book In llBooks

Order By book.strTitle

'print out the desired output

Console.WriteLine("\*\*\* Inventory Report \*\*\*".PadLeft(55))

Console.WriteLine("---------------------------------".PadLeft(60) + vbCrLf)

Console.WriteLine(" Title Category Quantity Unit Cost Extended Cost")

Console.WriteLine("------------------------ -------- -------- --------- -------------")

'using string format and for each loop

Dim strLineFormat As String = " {0,-30} {1,-10} {2,-12} {3,-12:N2} {4,-10:N2}"

For Each book As clsBooks In report

Console.WriteLine(String.Format(strLineFormat, book.strTitle, book.strCategory, book.intQuantity, book.sngPrice, book.sngInventoryTotal))

Next

Console.WriteLine()

End Sub

Sub TotalInventoryValue(llBooks As LinkedList(Of clsBooks))

'------------------------------------------------------------

'- Subprogram Name: TotalInventoryValue -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine print out the total invenory report using the lineked list

'- printing out all the books in range of prices.

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- under50 - LINQ result for book range under 50$

'- under100- LINQ result for book range under 100$

'- under150 - LINQ result for book range under 150$

'- above150 - LINQ result for book range above 150$

'-strFormat - String format when print out output

'-llBooks - linked list of the obj books

'------------------------------------------------------------

Console.WriteLine("-------------------------------------------------------------------------")

Console.WriteLine("Total Inventory Value (Quantity \* Unit Price) Statistics".PadLeft(10))

Console.WriteLine("-------------------------------------------------------------------------")

Console.WriteLine("Those books in the range of 0.00 - 50.00 are:")

Dim strFormat = " {0,-30} Price: {1,-20:C2}"

'find and print book under $50

Dim under50 = From book In llBooks

Where book.sngInventoryTotal < 50

Order By book.sngInventoryTotal

Select book.strTitle, book.sngInventoryTotal

If under50.Count > 0 Then

For Each book In under50

Console.WriteLine(String.Format(strFormat, book.strTitle, book.sngInventoryTotal))

Next

Else

Console.WriteLine("(There are no books in this range)")

End If

'find and print book under $100

Console.WriteLine(vbCrLf + "Those books in the range of 50.00 - 100.00 are:")

Dim under100 = From book In llBooks

Where book.sngInventoryTotal < 100 And book.sngInventoryTotal >= 50

Order By book.sngInventoryTotal

Select book.strTitle, book.sngInventoryTotal

If under50.Count > 0 Then

For Each book In under100

Console.WriteLine(String.Format(strFormat, book.strTitle, book.sngInventoryTotal))

Next

Else

Console.WriteLine("(There are no books in this range)")

End If

'find and print book under $150

Console.WriteLine(vbCrLf + "Those books in the range of 100.00 - 150.00 are:")

Dim under150 = From book In llBooks

Where book.sngInventoryTotal < 150 And book.sngInventoryTotal >= 100

Order By book.sngInventoryTotal

Select book.strTitle, book.sngInventoryTotal

If under150.Count > 0 Then

For Each book In under150

Console.WriteLine(String.Format(strFormat, book.strTitle, book.sngInventoryTotal))

Next

Else

Console.WriteLine("(There are no books in this range)")

End If

'find and print book above 150

Console.WriteLine(vbCrLf + "Those books in the range of 150.00 and above are:")

Dim above150 = From book In llBooks

Where book.sngInventoryTotal >= 150

Order By book.sngInventoryTotal

Select book.strTitle, book.sngInventoryTotal

If above150.Count > 0 Then

For Each book In above150

Console.WriteLine(String.Format(strFormat, book.strTitle, book.sngInventoryTotal))

Next

Else

Console.WriteLine("(There are no books in this range)")

End If

End Sub

Sub CategoryStat(llBooks As LinkedList(Of clsBooks))

'------------------------------------------------------------

'- Subprogram Name: CategoryStat -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine print out the invenory report using the lineked list

'- printing out all the count, maximum price, cheapest price book,

'-average price of the book that have the same category

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- stats - LINQ result return count, minimum, maximum of the according group

'-strFormt - String format when print out output

'-llBooks - linked list of the obj books

'------------------------------------------------------------

'print out the statistic according by category

Console.WriteLine(vbCrLf + vbCrLf + "-------------------------------------------------------------------------")

Console.WriteLine("Unit Price Range by Category Statistics".PadLeft(50))

Console.WriteLine("-------------------------------------------------------------------------")

Console.WriteLine("Category # of Titles Low Average High")

'LINQ return the result of the minmum price, average price, and the highest price along with count

'of the according to the category

Dim stats = From book In llBooks

Group By cat = book.strCategory

Into statistic = Group, Count(), Min(book.sngPrice), Average(book.sngPrice), Max(book.sngPrice)

Order By cat

'format string

Dim strFormt = " {0}............ {1,-2}............ {2,-2:N2}............ {3,4:N2}............ {4,-2:N2}"

'print to console

For Each category In stats

Console.WriteLine(String.Format(strFormt, category.cat, category.Count, category.Min, category.Average, category.Max))

Next

End Sub

Sub OverallBookStats(llBooks As LinkedList(Of clsBooks))

'------------------------------------------------------------

'- Subprogram Name: OverallBookStats -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- This subroutine print out the invenory report using the linked list

'- printing out all the count, maximum price, cheapest price book,

'-average price of the book

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically):

'- cheapest - LINQ result return the cheapest price

'- cheapBook - LINQ result return a enum set that have the cheapest price

'- expensive - LINQ result return the most expensiveprice

'- priciestBook - LINQ result return a set of books that have the priciest book

'- leastQuantity - LINQ result return the least quantity number

'- leastBook - LINQ result return return the book that have the same least quantity

'- maxQuantity - LINQ result return the maximum quantity of book

'- maxBook - LINQ result return the book the have the maximum amount

'-llBooks - linked list of the obj books

'------------------------------------------------------------

Console.WriteLine(vbCrLf + vbCrLf + "-------------------------------------------------------------------------")

Console.WriteLine("Overall Book Statistics".PadLeft(50))

Console.WriteLine("-------------------------------------------------------------------------")

'find the cheapest price and the set of book that have that price using LINQ

Dim cheapest = Aggregate book In llBooks

Into Min(book.sngPrice)

Dim cheapBook = From book In llBooks

Where book.sngPrice = cheapest

Select book.strTitle

'print the book title and prices

Console.WriteLine("The cheapest book title(s) at a unit price of " & String.Format("{0:C2}", cheapest) & " are:")

For Each book In cheapBook

Console.WriteLine(book)

Next

Console.WriteLine()

'find the priciest price and the set of book that have that price using LINQ

Dim expensive = Aggregate book In llBooks

Into Max(book.sngPrice)

Dim priciestBook = From book In llBooks

Where book.sngPrice = expensive

Select book.strTitle

'print the book title and prices

Console.WriteLine("The priciest book title(s) at a unit price of " & String.Format("{0:C2}", expensive) & " are:")

For Each book In priciestBook

Console.WriteLine(book)

Next

Console.WriteLine()

'find the least anoubt and the set of book that have that amount using LINQ

Dim leastQuantity = Aggregate book In llBooks

Into Min(book.intQuantity)

Dim leastBook = From book In llBooks

Where book.intQuantity = leastQuantity

Select book.strTitle

'print the book title and quantity

Console.WriteLine("The title(s) with the least quantity on hand at " & String.Format("{0:D2}", leastQuantity) & " units are:")

For Each book In leastBook

Console.WriteLine(book)

Next

Console.WriteLine()

'find the max amount and the set of book that have that amount using LINQ

Dim maxQuantity = Aggregate book In llBooks

Into Max(book.intQuantity)

Dim maxBook = From book In llBooks

Where book.intQuantity = maxQuantity

Select book.strTitle

'print the book title and quantity

Console.WriteLine("The title(s) with the most quantity on hand at " & String.Format("{0:D2}", maxQuantity) & " units are:")

For Each book In maxBook

Console.WriteLine(book)

Next

End Sub

Function pathValidation(ByVal strPath As String) As Boolean

'------------------------------------------------------------

'- Subprogram Name: pathValidation -

'------------------------------------------------------------

'- Written By: Binh Dang -

'- Written On: Feb 25, 2022 -

'------------------------------------------------------------

'- Subprogram Purpose: -

'- -

'- This subroutine make sure that user is input the right value

'- else it will check and allow user to enter another input

'------------------------------------------------------------

'------------------------------------------------------------

'- Local Variable Dictionary (alphabetically): None

If String.IsNullOrWhiteSpace(strPath) Then 'if the input is white or nothing

Console.WriteLine()

Return False

ElseIf Not strPath.EndsWith(".txt") Then 'check if the file is txt or ot

Console.WriteLine()

Return False

ElseIf Dir(strPath) = "" Then ' Check if file exist

Console.WriteLine()

Return False

Else 'else continue to the loops

Console.WriteLine()

Return True

End If

Return False

End Function

End Module