Program Cover Sheet

|  |
| --- |
| Name: Madison Kell |
| Assignment: 5 |
| List any parts of the assignment that do not work/were not completed: |

|  |
| --- |
| Instructor’s Comments: |
| Grade: |

Program Submission Requirements: (1) all files, zipped and uploaded to Canvas and (2) a completed cover sheet, program execution screenshots and source code printed, **stapled** and turned in during class. Failure to follow the submission requirements will result in points lost on that particular assignment.

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

'- File Name : Assignment5 -

'- Part of Project: Assign5 -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- File Purpose: -

'- This file contains the main application form where all

'- functions and subprograms are held. The file allows for

' the compiler to sort through a file containing books.

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

'- Program Purpose: -

'- -

'- This program allows the user to

' input a file path and the application will sort

'- through the list of books and display data relating to

' the books title, price, and quanity.

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

'- Global Variable Dictionary (alphabetically): -

'- none

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

Imports System.IO

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

'- Subprogram Name: Class books -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- class that holds the information we want,

'' along with a simple constructor and an information

' overridden specific version Of the ToString method

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

'- Parameter Dictionary (in parameter order):

' (none)

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

'- Local Variable Dictionary (alphabetically): -

'- intQuantity - public property to hold value of the book quantity later

'- sngInventoryTotal - public property to hold value of the book inventory totals later

'- sngPrice - - public property to hold value of the book prices later

'- strCategory - public property to hold value of the book categories later

'- strTitle - public property to hold value of the book titles later

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

Class books

'creating a public propery of the category of the books

Public Property strCategory As String

'creating a public propery of the quantity of the books

Public Property intQuantity As Integer

'creating a public propery of the price of the books

Public Property sngPrice As Single

'creating a public propery of the title of the books

Public Property strTitle As String

'creating a public propery of the inventory total of the books

Public Property sngInventoryTotal As Single

'constructor to set all values of the properties to params of the sub, so I can manipulate the public properties and assign them to the

'existing value in class books

Public Sub New(ByVal Title As String, ByVal Category As String, ByVal Quantity As Integer, ByVal Price As Single, ByVal Total As Single)

Me.strTitle = Title

Me.strCategory = Category

Me.intQuantity = Quantity

Me.sngPrice = Price

Me.sngInventoryTotal = Total

End Sub

'override tostring method to help with some fomatting of the first section of the data

Public Overrides Function ToString() As String

Return String.Format(" {0, -30} {1, 15} {2, 17} {3, 17:.00} {4, 20:.00}", strTitle, strCategory, intQuantity, sngPrice, sngInventoryTotal)

End Function

End Class

Module Module1

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

'- Subprogram Name: sub main -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the user clicks the -

'- starts the program. All of the functions are called here

'– and the data is sorted, filed, and compiled here. Headings

'- of the sections are here as well.

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

'- Parameter Dictionary (in parameter order): -

'- none

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

'- Local Variable Dictionary (alphabetically): -

'- list- created to hold a list of all of the data in the

'- file

'- amount- a query that is looped through to order the titles

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

Sub Main()

'created to hold a list of all of the data in the file

Dim list As New List(Of books)

'try to validate the file

Try

'call the function that gets the file

getFile(list)

'catching the error

Catch ex As Exception

'message box that closes the program if the user inputs a bad path name

MsgBox("An invalid file path was entered. Goodbye.", Title:="Error!")

'exit the program softly

Exit Sub

End Try

'header section for the first section of data

Console.WriteLine(vbNewLine & StrDup(53, " ") & "Books 'R' Us")

Console.WriteLine(StrDup(48, " ") & StrDup(3, "\*") & " Inventory Report " & StrDup(3, "\*") & vbNewLine & StrDup(45, " ") & StrDup(30, "-"))

Console.WriteLine(vbNewLine & String.Format(" {0, -38} {1, -15} {2, -17} {3, -17} {4}", "Title: ", "Category: ", "Quantity: ", "Unit Cost: ", "Extended Cost: "))

Console.WriteLine(String.Format(" {0, -38} {1, -15} {2, -17} {3, -15} {4}", StrDup(20, "-"), StrDup(10, "-"), StrDup(10, "-"), StrDup(15, "-"), StrDup(20, "-")))

'setting all of the data to the list in a string

list.ToString()

'sorting the data by titl

Dim amount = From allBooks In list

Order By allBooks.strTitle

'looping through each title and printing the data out

For Each book In amount

Console.WriteLine(book.ToString)

Next

'header section for the second section of data

Console.WriteLine(vbNewLine & StrDup(25, " ") & StrDup(60, "-"))

Console.WriteLine(StrDup(28, " ") & "Total Inventory Value (Quantity \* Unit Price) Statistics ")

Console.WriteLine(StrDup(25, " ") & StrDup(60, "-") & vbNewLine)

'calling the function that is gettingt hte data for the total inventory section

invValue(list)

'header section for the third section of data

Console.WriteLine(vbNewLine & StrDup(25, " ") & StrDup(60, "-"))

Console.WriteLine(StrDup(35, " ") & "Unit Price Range by Category Statistics")

Console.WriteLine(StrDup(25, " ") & StrDup(60, "-"))

Console.WriteLine(vbNewLine & String.Format(" {0, -15} {1, -15} {2, -15} {3, -15} {4}", "Category", "# of Titles", "Low", "Ave", "High"))

'adding a line before the call of the next function

Console.WriteLine()

'calling the function that prints the category (directions said that the categories will always be the same) data

unitPrice("F", list)

unitPrice("N", list)

unitPrice("S", list)

'calling the function that prints the stats of unit price

stats(list)

'that pause to let the reader read the information

Console.ReadLine()

End Sub

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

'- Subprogram Name: Function getFile -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- This function is called when the main calls for the file

'– information.

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

'- Parameter Dictionary (in parameter order): -

'- List – passing the information of the file in -

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

'- Local Variable Dictionary (alphabetically): -

'- arrLines- the only array. to separate the file and manipulate

'- the data

'- dblExtendedCost - holding the extended cost

'- fileInfo - the acutal data from the file path that the user

'- gives

'- strExisitingFilepath - string holding the input file the user

'- types in

'- strFileReader - stream reader to read the file

'- tmp-a temp string in the for

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

Function getFile(list As List(Of books))

' string holding the input file the user

Dim strExisitingFilepath As String

'asking the user to enter the file path

Console.WriteLine("Please enter the file path: ")

'reading the file path that the user gave

strExisitingFilepath = Console.ReadLine()

' the acutal data from the file path that the user

Dim fileInfo = File.ReadAllLines(strExisitingFilepath).Length

'a stream reader so the program can read in the file

Dim strFileReader As StreamReader = New StreamReader(strExisitingFilepath)

'the only array. to separate the file and manipulate the data

Dim arrLines() As String

'looping through all of the data

For i As Integer = 0 To fileInfo - 1

'splitting the lines by the spaces

arrLines = strFileReader.ReadLine.Split(" ")

'settign a temp to replace the book titles with spaces in it

Dim tmp As String = ""

'if the index is greater than 3 spaces loop through

For j As Integer = 3 To arrLines.Length - 1

'set the temp to the array and add the space afterwards

tmp = tmp + arrLines(j) + " "

Next

'creating a variable to hold the extended cost

Dim dblExtendedCost As Double

'setting the extended cost to the category \* the quantity

dblExtendedCost = arrLines(1) \* arrLines(2)

'adding the split pieces of the array to the list

list.Add(New books(tmp, arrLines(0), arrLines(1), arrLines(2), dblExtendedCost))

Next

'closing the stream

strFileReader.Close()

End Function

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

'- Subprogram Name: invValue -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- This function is called when the main calls for the books

'– that are between certain prices

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

'- Parameter Dictionary (in parameter order): -

'- List – passing the information of the file in -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

'- Returns: -

'- fiftyToHundred - query to hold all book titles that are 50-100

'- moreThanHundred - query to hold all book titles that are >150

'- oneHundredToOneFifty -query to hold all book titles that are 100-50

'- zeroToFifty -query to hold all book titles that are 0-50

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

Function invValue(list As List(Of books))

'query to go through the list and find the values that are between 0 adn 50 and order them

' and then select the title and the price from that smaller list

Dim zeroToFifty = From books In list

Where books.sngInventoryTotal > 0 And books.sngInventoryTotal < 50

Order By books.sngInventoryTotal Ascending

Select books.strTitle, books.sngInventoryTotal

'writing the sub header

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

'looping through the data from the query above and print out the title of the book and the inventory total

For Each book In zeroToFifty

Console.WriteLine(String.Format(" {0, -50} {1} {2, 5:C2}", book.strTitle, "Price: ", book.sngInventoryTotal))

Next

Console.WriteLine()

'query to go through the list and find the values that are between 50 adn 100 and order them

' and then select the title and the price from that smaller list

Dim fiftyToHundred = From books In list

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

Order By books.sngInventoryTotal Ascending

Select books.strTitle, books.sngInventoryTotal

'writing the sub header

Console.WriteLine(" The books in the range of 50.00 - 100.00 are: " & vbNewLine)

'looping through the data from the query above and print out the title of the book and the inventory total

For Each book In fiftyToHundred

Console.WriteLine(String.Format(" {0, -50} {1} {2, 5:C2}", book.strTitle, "Price: ", book.sngInventoryTotal))

Next

Console.WriteLine()

'query to go through the list and find the values that are between 100 adn 150 and order them

' and then select the title and the price from that smaller list

Dim oneHundredToOneFifty = From books In list

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

Order By books.sngInventoryTotal Ascending

Select books.strTitle, books.sngInventoryTotal

'writing the sub header

Console.WriteLine(" The books in the range of 100.00 - 150.00 are: " & vbNewLine)

'looping through the data from the query above and print out the title of the book and the inventory total

For Each book In oneHundredToOneFifty

Console.WriteLine(String.Format(" {0, -50} {1} {2, 5:C2}", book.strTitle, "Price: ", book.sngInventoryTotal))

Next

Console.WriteLine()

'query to go through the list and find the values that are 150+ and order them

' and then select the title and the price from that smaller list

Dim moreThanHundred = From books In list

Where books.sngInventoryTotal >= 150 And books.sngInventoryTotal < 150

Order By books.sngInventoryTotal Ascending

Select books.strTitle, books.sngInventoryTotal

'writing the sub header

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

'looping through the data from the query above and print out the title of the book and the inventory total

For Each book In moreThanHundred

Console.WriteLine(String.Format(" {0, -50} {1} {2, 5:C2}", book.strTitle, "Price: ", book.sngInventoryTotal))

Next

End Function

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

'- Subprogram Name: unit price -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- This function is called when the main calls for the books

'– that sorted by the category and the data that corresponds

''- with that

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

'- Parameter Dictionary (in parameter order): -

'- individualCat – passes the string value of the category

'- list – passing the information of the file in

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

'- Local Variable Dictionary (alphabetically):

'-theAvg -aggregate that holds the value of the average cost from the specifc category

'-theCount - -aggregate that holds the value of the count of the specifc category

'-theLow -aggregate that holds the value of the low cost from the specifc category

'-theMax -aggregate that holds the value of the max cost from the specifc category

'- query -aggregate that holds all values from the lsit created above

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

Function unitPrice(individualCat As String, list As List(Of books))

'aggregate that holds all values from the lsit created above

'then the query finds the information pertaining to each category

'and selecting the name and the price from each

Dim query = From book In list

Where book.strCategory = individualCat

Select book.strCategory, book.sngPrice

'aggregate that holds the value of the count of the specifc category

'and then groups it into the build in count()

Dim theCount = Aggregate book In query

Into Count(book.sngPrice)

'aggregate that holds the value of the low amount of the specifc category

'and then groups it into the build in count()

Dim theLow = Aggregate book In query

Into Min(book.sngPrice)

'aggregate that holds the value of the max amount of the specifc category

'and then groups it into the build in count()

Dim theMax = Aggregate book In query

Into Max(book.sngPrice)

'aggregate that holds the value of the average of the specifc category

'and then groups it into the build in count()

Dim theAvg = Aggregate book In query

Into Average(book.sngPrice)

'printing them all out to the console formatted

Console.WriteLine(String.Format(" {0, -15} {1, -15} {2, -15:.00} {3, -15:.00} {4, -15:.00}", individualCat, theCount, theLow, theAvg, theMax))

End Function

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

'- Subprogram Name: stats -

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

'- Written By: Madison Kell -

'- Written On: 02/26/2022 -

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

'- Subprogram Purpose: -

'- -

'- This function is called when the main calls for the overall book

'- statistics

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

'- Parameter Dictionary (in parameter order): -

'- list – passing the information of the file in

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

'- Local Variable Dictionary (alphabetically): -

'- index- creating an index variable to increase in loops

' so the header for each sub sub section only prints once -

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

Function stats(list As List(Of books))

'printing the heading

Console.WriteLine(vbNewLine & StrDup(25, " ") & StrDup(60, "-"))

Console.WriteLine(StrDup(44, " ") & "Overall Book Statistics")

Console.WriteLine(StrDup(25, " ") & StrDup(60, "-") & vbNewLine)

'query getting the category, price, and quantity from the list

Dim query = From book In list

Select book.strCategory, book.sngPrice, book.intQuantity

'taking the aggregate from the query and adding into min

Dim theLowPrice = Aggregate book In query Into Min(book.sngPrice)

'creating a linq to select the books in the list that were also in

'the function above and selecting the title and the price

Dim cheapest = From book In list

Order By book.sngPrice

Where book.sngPrice = theLowPrice

Select book.strTitle, book.sngPrice

'taking the aggregate from the query and adding into max

Dim theMaxPrice = Aggregate book In query Into Max(book.sngPrice)

'creating a linq to select the books in the list that were also in

'the function above and selecting the title and the price

Dim priciest = From book In list

Order By book.sngPrice Descending

Where book.sngPrice = theMaxPrice

Select book.strTitle, book.sngPrice

'taking the aggregate from the query and adding into min quantity

Dim theLow = Aggregate book In query Into Min(book.intQuantity)

'creating a linq to select the books in the list that were also in

'the function above and selecting the title and the quantity

Dim leastQuantity = From book In list

Order By book.intQuantity

Where book.intQuantity = theLow

Select book.strTitle, book.intQuantity

'taking the aggregate from the query and adding into max quantity

Dim theMax = Aggregate book In query Into Max(book.intQuantity)

'creating a linq to select the books in the list that were also in

'the function above and selecting the title and the quantity

Dim mostQuantity = From book In list

Order By book.intQuantity Descending

Where book.intQuantity = theMax

Select book.strTitle, book.intQuantity

'creating an index variable to increase in loops

'so the header for each sub sub section only prints once

Dim index As Integer = 0

'for each book in the cheapest books query

For Each book In cheapest

'if the index is 0 (only once) print the heading

If (index = 0) Then

'write the title

Console.WriteLine(String.Format("{0} {1:C2} {2}", "The cheapest book title(s) at a unit price of", book.sngPrice, "are:"))

End If

'write each corresponding title

Console.WriteLine(StrDup(10, " ") & book.strTitle & vbNewLine)

'increasing index so it does not run the loop with the sub sub head twice

index = index + 1

Next

'resetting the index

index = 0

'for each book in the priciest books query

For Each book In priciest

'if the index is 0 (only once) print the heading

If (index = 0) Then

'write the title

Console.WriteLine(String.Format("{0} {1:C2} {2}", "The priciest book title(s) at a unit price of", book.sngPrice, "are:"))

End If

'write each corresponding title

Console.WriteLine(StrDup(10, " ") & book.strTitle & vbNewLine)

'increasing index so it does not run the loop with the sub sub head twice

index = index + 1

Next

'resetting the index

index = 0

'for each book in the leastQuantity books query

For Each book In leastQuantity

'if the index is 0 (only once) print the heading

If (index = 0) Then

'write the title

Console.WriteLine(String.Format("{0} {1} {2}", "The title(s) with the least quantity on hand at", book.intQuantity, "units are:"))

End If

'write each corresponding title

Console.WriteLine(StrDup(10, " ") & book.strTitle & vbNewLine)

'increasing index so it does not run the loop with the sub sub head twice

index = index + 1

Next

'resetting the index

index = 0

'for each book in the mostQuantity books query

For Each book In mostQuantity

'if the index is 0 (only once) print the heading

If (index = 0) Then

'write the title

Console.WriteLine(String.Format("{0} {1} {2}", "The title(s) with the most quantity on hand at", book.intQuantity, "units are:"))

End If

'write each corresponding title

Console.WriteLine(StrDup(10, " ") & book.strTitle)

'increasing index so it does not run the loop with the sub sub head twice

index = index + 1

Next

End Function

End Module







