Program Cover Sheet

|  |
| --- |
| Name: Austin Rippee |
| Assignment: Assignment 1 – Employee Wage Calculator |
| List any parts of the assignment that do not work/were not completed:   * When switching between employees through the left and right buttons, they do not display that employee’s information, only displays the lastly entered employee’s information. |

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

Imports System.IO

Public Class frmAs1

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

'- File Name : frmAs1.frm -

'- Part of Project: EmployeeWageCalculator -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- File Purpose: -

'- This file contains the main form for the entire

'- application along with all calculations, file inputs

'- and file outputs. Finally any button action is performed

'- by the application in which resides here. -

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

'- Program Purpose: -

'- -

'- This program allows for the ability to add an employee

'- with their name, ID, hours worked, and their wage and

'- calculates the total amount earned. The program also

'- add the ability to save the info to an external file that

'- can be accessed. -

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

'- Global Variable Dictionary (alphabetically): -

'- CurrentEmp - Displays the current number of the employee

'- in the order it was added in.

'- TotalEmp - Displays the number of total employees that

'- have been added.

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

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

' Created an unused Structure for the purpose of a solution attempt

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

'Structure Employee

'Dim Name As String

'Dim ID As Integer

'Dim Hours As Double

'Dim Wage As Double

'End Structure

Dim CurrentEmp As Integer = 0

Dim TotalEmp As Integer = 0

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

'- Subprogram Name: formPayrollSystem\_Load -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

'- This subroutine is called whenever the program has been

'- called to start / on load. It simply sets which buttons

'- and textboxes are enabled/visible.–

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

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

lblCalculatedWage.Text = "" 'Sets the text of the label to nothing

txtName.Enabled = False 'Disables the name textbox

txtID.Enabled = False 'Disables the ID textbox

txtHours.Enabled = False 'Disables the hours textbox

txtWage.Enabled = False 'Disables the wage textbox

btnLeft.Enabled = False 'Disables the left button

btnRight.Enabled = False 'Disables the right button

btnSave.Enabled = False 'Disables the save button

btnCancel.Enabled = False 'Disables the cancel button

btnCalculate.Enabled = False 'Disables the calculate button

End Sub

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

'- Subprogram Name: btnAddNew\_Click -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

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

'- Add New Employee button. It's main purpose is to change

'- if the user is able to click on specific buttons and

'- change the visibility of other buttons.–

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

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

Me.Text = "Payroll System - Add New Employee" 'Changes the program title

'Sets the employee information textboxes to contain nothing

txtName.Text = ""

txtID.Text = ""

txtHours.Text = ""

txtWage.Text = ""

' Enables the textboxes that are for employee information

txtName.Enabled = True

txtID.Enabled = True

txtHours.Enabled = True

txtWage.Enabled = True

'Enables the buttons to either save info, cancel or calculate the wage

btnSave.Enabled = True

btnCancel.Enabled = True

btnCalculate.Enabled = True

'Disables the visibility of the buttons to prohibit the viewing of other records

btnAddNew.Visible = False

btnLeft.Visible = False

btnRight.Visible = False

'Enables the visibility of the save and cancel buttons

btnSave.Visible = True

btnCancel.Visible = True

End Sub

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

'- Subprogram Name: btnCalculate\_Click -

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

'- Written By: Your Name -

'- Written On: The date you wrote it -

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

'- Subprogram Purpose: -

'- -

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

'- calculate wage button. The routine will calculate the

'- employee's wage based upon if textboxes are empty, if the

'- salary checkbox has been checked and if the hours worked

'- for an employee is above 40.–

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- CalculatedWage - Does the calculation to calculate the wage

'- CalculateWageString - Converts the calculation from CalculatedWage

'- from a literal double to a string

'- CalculateWageText - Creates the text to display in front of

'- the calculated wage

'- CalcWage - Rounds the calculated wage to 2 decimal places

'- Hours - Gets the number of hours from the user inputed textbox

'- and places it in a variable

'- OvertimeHours - Calculates the overtime hours not included in

'- the normally worked hours

'- Wage - Gets the hourly wage from the user inputed textbox

'- and places it in a variable

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

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

Dim Hours As Double 'Hours is the amount of hours the employee worked

Dim Wage As Double 'Wage is the rate in which the employee makes per hour

If txtHours.Text = "" Or txtWage.Text = "" Then 'Checks if the textbox of wage or hours is empty and prompts the user to enter info

MsgBox("You need to input hours worked and a wage") 'Prompts the user

Else

Hours = txtHours.Text 'Sets whats in the hours textbox to the hours variable for later use

Wage = txtWage.Text 'Sets whats in the wage textbox to the wage variable for later use

End If

If chkbxSalary.Checked = True Then 'Checks if the salary checkbox has clicked

Dim CalculatedWage As Double = 40 \* Wage 'Sets the constant rate if the employee is salaried (40 hours multipled by the wage set by the user)

Dim CalcWage As Double = Math.Round(CalculatedWage, 2) 'Rounds the wage to 2 decimal places

Dim CalculatedWageString = CStr(CalcWage) 'Converts the decimal literal to a string

Dim CalculatedWageText As String = "Employee is Due $" 'Creates the text displayed prior to the calculated wage

lblCalculatedWage.Text = CalculatedWageText + CalculatedWageString 'Sets the label from set prior with the calculated wage from above

ElseIf Hours > 40 Then 'Checks if the employee is above 40 hours worked

Dim OvertimeHours As Double = (Wage \* 1.5) \* (Hours - 40) 'Sets the overtime hours as the wage multiplied by 1.5 and multiply that with the hours above the standard 40

Dim CalculatedWage As Double = (Wage \* 40) + OvertimeHours 'Calculates the 40 hours multiplied with the set wage and adds the overtime hours into the calculation

Dim CalcWage As Double = Math.Round(CalculatedWage, 2) 'Rounds the wage to 2 decimal places

Dim CalculatedWageString = CStr(CalcWage) 'Converts the decimal literal to a string

Dim CalculatedWageText As String = "Employee is Due $" 'Creates the text displayed prior to the calculated wage

lblCalculatedWage.Text = CalculatedWageText + CalculatedWageString 'Sets the label from set prior with the calculated wage from above

Else

Dim CalculatedWage As Double = Hours \* Wage 'Simply multiples the hours worked by the employee with the wage set in the info

Dim CalcWage As Double = Math.Round(CalculatedWage, 2) 'Rounds the wage to 2 decimal places

Dim CalculatedWageString = CStr(CalcWage) 'Converts the decimal literal to a string

Dim CalculatedWageText As String = "Employee is Due $" 'Creates the text displayed prior to the calculated wage

lblCalculatedWage.Text = CalculatedWageText + CalculatedWageString 'Sets the label from set prior with the calculated wage from above

End If

End Sub

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

'- Subprogram Name: btnCancel\_Click -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

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

'- cancel button. If the user clicks the button, it will

'- "cancel" out what the user has input into the textboxes

'- and returning to the last successfully entered employee.–

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- (None) -

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

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

Me.Text = "Payroll System" 'Reverts back to original title

'Sets each textbox to contain no text and disables them so user can't enter information

txtName.Text = ""

txtName.Enabled = False

txtID.Text = ""

txtID.Enabled = False

txtHours.Text = ""

txtHours.Enabled = False

txtWage.Text = ""

txtWage.Enabled = False

'Changes the visibility and enables/disables the corresponding button as to the original screen

btnLeft.Visible = True

btnLeft.Enabled = False

btnRight.Visible = True

btnRight.Enabled = False

btnAddNew.Visible = True

btnSave.Visible = True

btnSave.Enabled = False

btnCancel.Visible = True

btnCancel.Enabled = False

End Sub

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

'- Subprogram Name: btnSave\_Click -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

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

'- save button. If the user presses the save button, the

'- information that is in the textboxes gets stored into the

'- textboxes in which correlate to that number employee it

'- has been added. It also saves the information on a line

'- in an external file in which the location is written in

'- the routine.

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- CurrentEmployee - Converts the CurrentEmp to a string literal

'- empInfo - Creates the full string of the entire employee info

'- employee() - Creates the array of strings for each section

'- of the employee info

'- TotalEmployee - Converts the TotalEmp to a string literal

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

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

TotalEmp = TotalEmp + 1 'Adds 1 to the total amount of employees

CurrentEmp = TotalEmp 'After pressing save, go to the last saved employee

Dim CurrentEmployee = CStr(CurrentEmp) 'Converts the integer literal to a string

Dim TotalEmployee = CStr(TotalEmp) 'Converts the integer literal to a string

Dim employee(3) As String 'Creates a String array of 4 elements

'Sets what each element is part of the overall string to add to the file

employee(0) = txtName.Text

employee(1) = txtID.Text

employee(2) = txtHours.Text

employee(3) = txtWage.Text

'Creates a string to concatenate each of the elements created above

Dim empInfo As String = employee(0) + vbTab + employee(1) + vbTab + employee(2) + vbTab + employee(3) + vbCrLf

'Appends the text when the user saves the information / Creates the text file if it hasn't already been created

File.AppendAllText("C:\VBtest\Employees.txt", empInfo)

Me.Text = "Payroll System - Employee " + CurrentEmployee + "/" + TotalEmployee 'Sets the title to whichever employee the user is looking at

'Enables and disables corresponding buttons to match what the startup program looked like

txtName.Enabled = False

txtID.Enabled = False

txtHours.Enabled = False

txtWage.Enabled = False

btnSave.Visible = False

btnCancel.Visible = False

btnLeft.Visible = True

btnRight.Visible = True

btnAddNew.Visible = True

btnLeft.Enabled = True

btnRight.Enabled = True

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

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

' ALL CODE BELOW FOR REST OF SUB ROUTINE THAT WAS BEING TESTED IN FIGURING OUT A SOLUTION BUT KEPT IN CASE IT WAS NEEDED TO BE USED

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

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

'Dim empWriteFile As System.IO.StreamWriter '= System.IO.File.CreateText("C:\VBtest\Employees.txt")

'Dim empWriteFile As System.IO.StreamWriter = System.IO.File.CreateText("C:\VBtest\Employee" + CurrentEmployee + "\_Info.txt")

'For i = 0 To 3

'For i As Integer = 0 To TotalEmployee

' Dim emp = New Employee

' Dim allEmployees(TotalEmployee) As Employee

' ReDim emp.ID(CurrentEmployee)

' 'Dim empID As Integer = Val(txtID.Text)

' 'Dim empHours As Double = txtHours.Text

' 'Dim empWage As Double = txtWage.Text

' allEmployees(CurrentEmployee).ID(i) = emp.Name

' 'empWriteFile.WriteLine(allEmployees(CurrentEmployee).id(i))

' empWriteFile.WriteLine(emp.Name)

'Next

'Next

'Dim emp() As String

'For Each value As String In employee

'empWriteFile.WriteLine("Employee # " + CurrentEmployee)

'empWriteFile.WriteLine("Name: " + vbTab + txtName.Text)

'empWriteFile.WriteLine(employee(0) + vbTab + employee(1) + vbTab + employee(2) + vbTab + employee(3))

'employee.Append(vbCrLf)

'Next

'empWriteFile.Close()

'Dim emp(3) As Employee

'Dim EmpName As String

'Dim EmpID As String

'Dim EmpHours As String

'Dim EmpWage As String

'For i = 1 To CurrentEmp

' emp(i).Name = CStr(txtName.Text)

' emp(i).ID = CStr(txtID.Text)

' emp(i).Hours = CStr(txtHours.Text)

' emp(i).Wage = CStr(txtWage.Text)

' EmpName = emp(i).Name

' EmpID = emp(i).ID

' EmpHours = emp(i).Hours

' EmpWage = emp(i).Wage

' lstInfo.Items.Add(EmpName)

' lstInfo.Items.Add(EmpID)

' lstInfo.Items.Add(EmpHours)

' lstInfo.Items.Add(EmpWage)

'Next

End Sub

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

'- Subprogram Name: btnLeft\_Click -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

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

'- left button. Creates the Left button sub which moves the

'- employee back to the previously added employee –

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- CurrentEmployee - Converts the CurrentEmp to a string literal

'- eHours - Hours variable for readline file

'- eID - ID variable for readline file

'- empReadFile - Creates the StreamReader to the address of

'- the external file

'- eName - Name variable for readline file

'- eWage - Wage variable for readline file

'- TotalEmployee - Converts the TotalEmp to a string literal

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

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

If CurrentEmp <= 1 Then 'Checks if the user is attempting to move past the first record

MsgBox("You cannot move past the first record")

Else

CurrentEmp = CurrentEmp - 1 'Subtracts 1 from the current employee to denote "moving left"

Dim empReadFile As New System.IO.StreamReader("C:\VBtest\Employees.txt") 'Creates the streamreader to read in from the file

Dim eName, eID, eHours, eWage As String 'Creates the variables as Strings

eName = empReadFile.ReadLine() 'Creates the Name variable to read

eID = empReadFile.ReadLine() 'Creates the ID variable to read

eHours = empReadFile.ReadLine() 'Creates the Hours variable to read

eWage = empReadFile.ReadLine() 'Creates the Wage variable to read

empReadFile.Close() 'Closes the Streamreader

txtName.Text = eName 'Sets what was read from the name readline into the textbox

txtID.Text = eID 'Sets what was read from the ID readline into the textbox

txtHours.Text = eHours 'Sets what was read from the hours readline into the textbox

txtWage.Text = eWage 'Sets what was read from the wage readline into the textbox

Dim CurrentEmployee = CStr(CurrentEmp) 'Converts the integer literal to a string

Dim TotalEmployee = CStr(TotalEmp) 'Converts the integer literal to a string

Me.Text = "Payroll System - Employee " + CurrentEmployee + "/" + TotalEmployee 'Changes the title to the currently viewed employee

End If

End Sub

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

'- Subprogram Name: btnRight\_Click -

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

'- Written By: Austin Rippee -

'- Written On: January 18, 2022 -

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

'- Subprogram Purpose: -

'- -

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

'- right button. Creates the right button sub which moves the

'- employee forward to the newly added employee –

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

'- Parameter Dictionary (in parameter order): -

'- sender – Identifies which particular control raised the –

'- click event -

'- e – Holds the EventArgs object sent to the routine -

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

'- Local Variable Dictionary (alphabetically): -

'- CurrentEmployee - Converts the CurrentEmp to a string literal

'- eHours - Hours variable for readline file

'- eID - ID variable for readline file

'- empReadFile - Creates the StreamReader to the address of

'- the external file

'- eName - Name variable for readline file

'- eWage - Wage variable for readline file

'- TotalEmployee - Converts the TotalEmp to a string literal

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

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

If CurrentEmp >= TotalEmp Then 'Checks if the user is attempting to move past the last record added

MsgBox("You cannot move past the last record")

Else

CurrentEmp = CurrentEmp + 1 'Adds 1 to the current employee to denote "moving right"

Dim empReadFile As New System.IO.StreamReader("C:\VBtest\Employees.txt") 'Creates the streamreader to read in from the file

Dim eName, eID, eHours, eWage As String 'Creates the variables as Strings

eName = empReadFile.ReadLine() 'Creates the Name variable to read

eID = empReadFile.ReadLine() 'Creates the ID variable to read

eHours = empReadFile.ReadLine() 'Creates the Hours variable to read

eWage = empReadFile.ReadLine() 'Creates the Wage variable to read

empReadFile.Close() 'Closes the Streamreader

txtName.Text = eName 'Sets what was read from the name readline into the textbox

txtID.Text = eID 'Sets what was read from the ID readline into the textbox

txtHours.Text = eHours 'Sets what was read from the hours readline into the textbox

txtWage.Text = eWage 'Sets what was read from the wage readline into the textbox

Dim CurrentEmployee = CStr(CurrentEmp) 'Converts the integer literal to a string

Dim TotalEmployee = CStr(TotalEmp) 'Converts the integer literal to a string

Me.Text = "Payroll System - Employee " + CurrentEmployee + "/" + TotalEmployee 'Changes the title to the currently viewed employee

End If

End Sub

End Class

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated