Payroll Application [Element 4]

1. Introduction

This payroll application allows you to enter a Job Title and the corresponding Rate of Pay. Then you add an Employee Name and Start the Calculations to determine the amount to pay according to the hours working.

The program consist of two main forms, one is called "Welcome" and the other "Payroll". It also has an additional "About" form

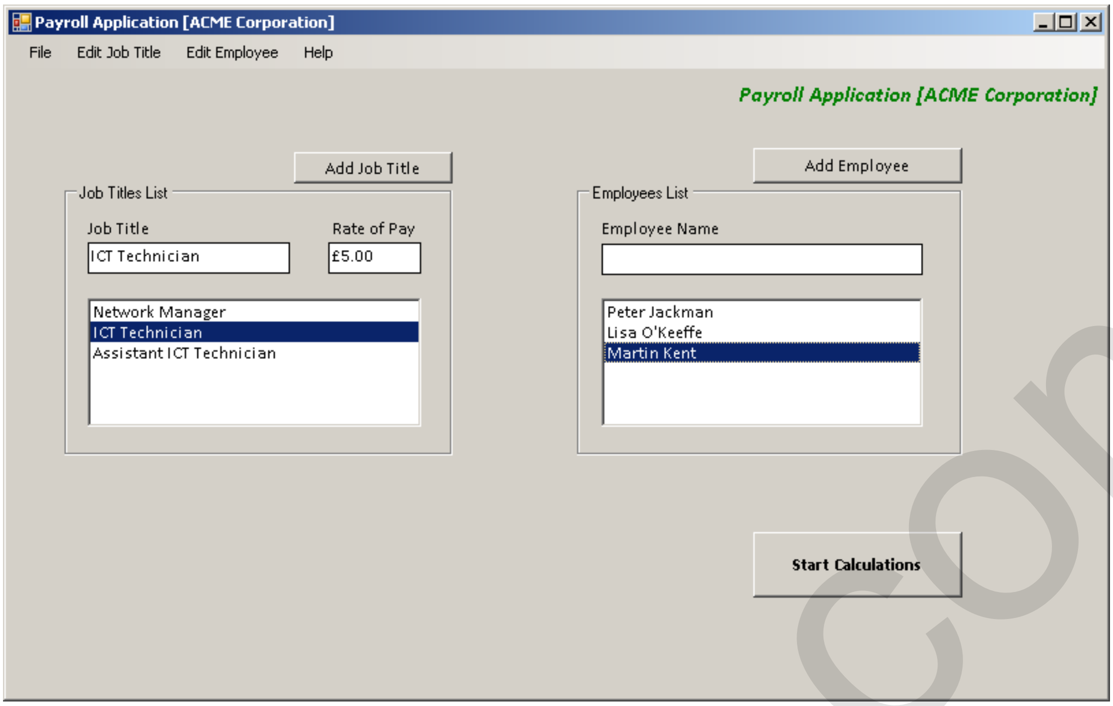
2. The Forms

The program will start with the Welcome form, requiring the user to input the data. Only when data has been selected, the user is allowed to continue to the next form, where the calculation can be made.

The program also contains a "Add" form, intended to enter data on the Job Lists, but I did not know how to fully configure it, therefore it is hidden and the button that will show it up is disabled.

The Welcome Form

In this form there are two lists, one for Job Titles and the other for the Employees. Each one of the list have got a 'add' button on the top where the user has to click in order to input the data.



Payroll Application [ACME Corporation]

Job Titles List

Job Title: ICT Technician

Rate of Pay: £5.00

- Network Manager
- ICT Technician
- Assistant ICT Technician

Employees List

Employee Name:

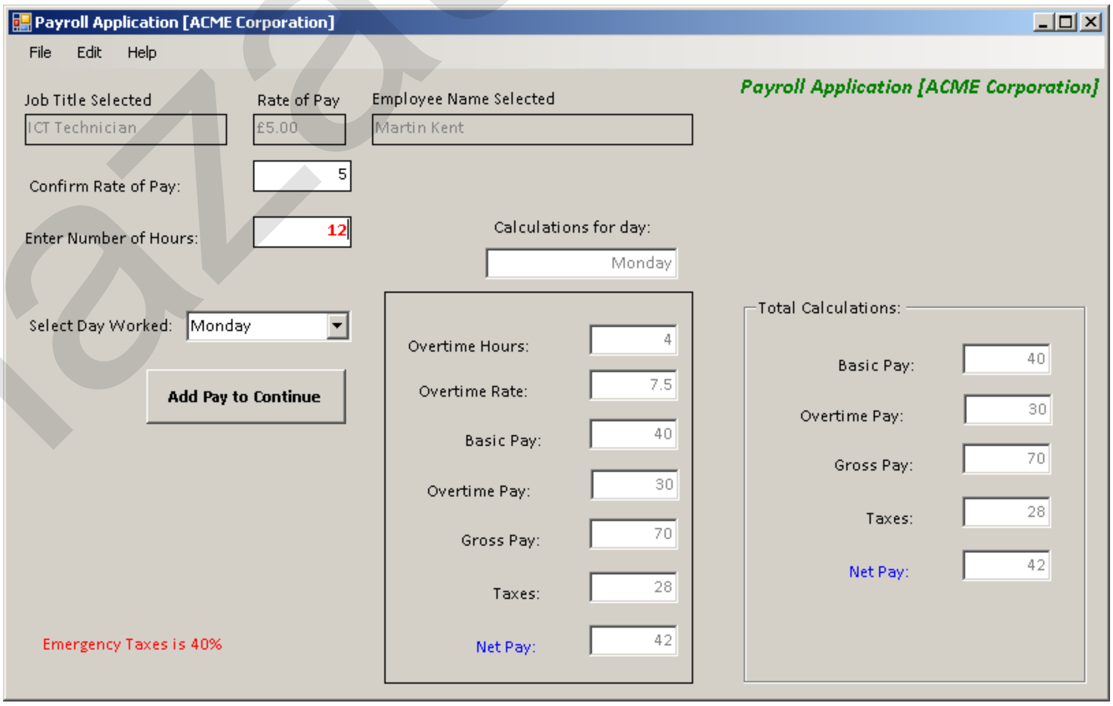
- Peter Jackman
- Lisa O'Keeffe
- Martin Kent

The Pay roll from

In this form the data that the user selected in the previous form is transferred and appears as greyed out.

The user is required to confirm the rate of pay that is grey out, enter the number of hours worked, select a date and then start the calculations for that particular date.

This form displays the calculations made for the date selected and also the sum of the total of calculations made for the other worked dates.



Payroll Application [ACME Corporation]

Job Title Selected: ICT Technician

Rate of Pay: £5.00

Employee Name Selected: Martin Kent

Confirm Rate of Pay: 5

Enter Number of Hours: 12

Select Day Worked: Monday

Calculations for day: Monday

Overtime Hours: 4

Overtime Rate: 7.5

Basic Pay: 40

Overtime Pay: 30

Gross Pay: 70

Taxes: 28

Net Pay: 42

Total Calculations:

Basic Pay: 40

Overtime Pay: 30

Gross Pay: 70

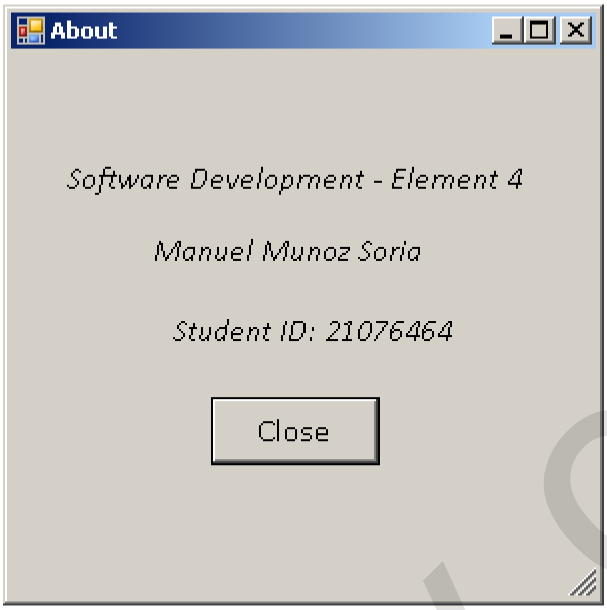
Taxes: 28

Net Pay: 42

Emergency Taxes is 40%

The About Form

This form displays the credits and version of the program



About

Software Development - Element 4

Manuel Munoz Soria

Student ID: 21076464

3. Program Structure

In order to share data among the forms, the program contains a module called "Data", where I declared the items that will be transferred from the "Welcome" form to the "Payroll" form.

The items declared involved the Job Titles list and the Employees List.

The program will open with the Welcome form, expecting the user to input the details of the data needed. Validation entries are made in the 'Add' button of the Job Title, in order to ensure that the Rate of Pay is a number.

Each one of the lists contains a top menu called "Edit Job Title" and "Edit Employee". Only the Amend for the Job Title and the Delete for both lists are working in these menus. The other ones will be configured on a later version of the program.

The "File" menu also contains option to "Open" and "Save As" files, but they are not yet configured as I did not know how to do it well. The Exit button is working, displaying a message of confirmation before leaving the application.

After the user has selected on item in both the Job Lists and the Employee list, he/she is allowed to click on the "Start Calculations" button and go to the Payroll form where the calculation are made for the data selected. There are several validations made on this form in order to prevent the user for entering wrong or invalid data, like character instead of numbers for the number of hours worked.

4. Program Operation

The main program operation, the nutshell of it, is based in the payroll form, particularly in the "Add Pay and Continue" button. After the user click on this button, the date is being validated, ensuring that there is only numbers in the Rate of Pay, that a date has been selected and the following validations are made for the hours worked:

- Ensure the input is only numbers
- Ensure the input is not negative
- Ensure the input does not contain decimals
- Ensure the input is lower than 16

Then, in sequence, the "Add Pay and Continue" button start calculating the basic pay, then the gross, taxes and net pay, then the overtime calculations and finally it display all the items in the form in the corresponding string fields.

Right at the end, the button clear the inputted field, leaving them ready for the next input, and one last crucial step is to add the items just added to the memory, where the data accumulates every time the user input more figures.

This memory location is created when opening the form and set to 'zero' when the form loads.

5. Main Procedures Explained

For the Welcome form, one of the main subroutines is the Job Titles, where the integer "indexJobTitle" index the items. The Rate of Pay is created on a array of 21 members only (20 plus 0) and is display in its corresponding field with the pound sign (£). Only after at least one item is entered in the Job List, the Amen button becomes enabled.

The other major subroutine of this form is the lstEmployees array, where the integer "intIndexEmployee" helps keeping track of the item selected that will be called upon going to the Payroll form. In the add button for the employee, there is a validation that ensures no blank name can be entered.

The final bit of the Welcome Form, is the "Start Calculations" button, that ensure one item have been selected on every list, closing the welcome form and opening the Payroll form only if it is true that items have been selected.

The Payroll form starts by declaring variables and then set the memory allocated to zero. It also disables menus that I do not know yet how to configure, and show up in greyed out fields the index transferred from the Welcome form. These indexes are the items selected by the users, and they can be moved from one form to the other thanks to the module data where these items have been declared for public use.

The Payroll form also contains a string called lstDates, where I manually entered the dates as a collection of items using the Graphical Interface of Visual Basic Studio.

The main subroutine of this form is the bntNetPay button, which I have explained in the Program Operation section of this document.

6. The Code

• Module Data

```
Module Data

    'Declare items for Job Titles List
    Public sngRatePay(20) As Single
    Public strJobTitle As String
    Public indexJobTitle As Integer
    Public nItems As Integer

    'Declare items for Employees List
    Public lstEmployees As Array
    Public intIndexEmployee As Integer
    Public strEmployee As String


End Module
```

• Welcome Form Code

```
Public Class frmWelcome

    Private Sub Welcome_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        'Focus cursor on Job Title
        txtJobTitle.Focus()
        'Disable Amen menus
        mnuAmendJobTitle.Enabled = False
        mnuAmendEmployee.Enabled = False
        'Disable Add menus untill I know how to code them
        mnuAddJobTitle.Enabled = False
        mnuAddEmployee.Enabled = False

    End Sub

    Private Sub lstJobTitles_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstJobTitles.SelectedIndexChanged

        'Index the Job Title selected
        indexJobTitle = lstJobTitles.SelectedIndex
        'Display the selection of Job Titles in text boxes
        If indexJobTitle >= 0 Then
            strJobTitle = lstJobTitles.Items.Item(indexJobTitle)
            txtJobTitle.Text = strJobTitle
            txtRatePay.Text = Format(sngRatePay(indexJobTitle), "£0.00")
        End If
        mnuAmendJobTitle.Enabled = True

    End Sub

    Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddJobTitle.Click

        'Add the Job Title
        lstJobTitles.Items.Add(txtJobTitle.Text)
        If IsNumeric(txtRatePay.Text) = False Then
            MsgBox("Please input only numbers in the Rate of Pay field")
            txtRatePay.Focus()
        Else
            'Add price to the array
            sngRatePay(nItems) = CSng(txtRatePay.Text)
            nItems = nItems + 1
        End If
        'clear the text fields
        txtJobTitle.Clear()
        txtRatePay.Clear()

    End Sub

    Private Sub mnuAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuAddJobTitle.Click
        frmAdd.ShowDialog()
    End Sub

    Private Sub mnuAmend_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuAmendJobTitle.Click

        'Determine the product selected
        indexJobTitle = lstJobTitles.SelectedIndex

        'display the items so that you can change them
        strJobTitle = CStr(txtJobTitle.Text)
        sngRatePay(nItems) = CSng(txtRatePay.Text)

        'and now puts them back in memory
        sngRatePay(indexJobTitle) = sngRatePay(nItems)
        lstJobTitles.Items(indexJobTitle) = strJobTitle

    End Sub

    Private Sub mnuDelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuDeleteJobTitle.Click

        'Remove and validate that Job Title is selected
        With lstJobTitles
            If .SelectedIndex < 0 Then
                MsgBox("Please select Job Title to remove")
            End If
            If .SelectedIndex >= 0 Then
                If MsgBox("Are you sure you want to delete this Job Title?",
MsgBoxStyle.OkCancel, "Yes") = MsgBoxResult.Ok Then
                    .Items.RemoveAt(lstJobTitles.SelectedIndex)
                End If
            End If
        End With
    End Sub

    Private Sub lstEmployees_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles lstEmployees.SelectedIndexChanged

        'Display values on the employee list
        intIndexEmployee = lstEmployees.SelectedIndex
        If intIndexEmployee >= 0 Then
            strEmployee = lstEmployees.Items.Item(intIndexEmployee)
            txtEmployee.Text = strEmployee
        End If
        txtEmployee.Clear()

    End Sub
    Private Sub btnAddEmployee_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAddEmployee.Click

        'Add and validates that blank employee cannot be entered
        If txtEmployee.Text = "" Then
            MsgBox("Please enter the name of the employee")
            txtEmployee.Focus()
        Else
            lstEmployees.Items.Add(txtEmployee.Text)
            txtEmployee.Clear()
        End If

    End Sub

    Private Sub mnuDeleteEmployee_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuDeleteEmployee.Click

        'Remove and validate that Employee is selected
        With lstEmployees
            If .SelectedIndex < 0 Then
                MsgBox("Please select Employee to remove")
            End If
            If .SelectedIndex >= 0 Then
```

```
                If MsgBox("Are you sure you want to delete this Employee?",
MsgBoxStyle.OkCancel, "Yes") = MsgBoxResult.Ok Then
                    .Items.RemoveAt(lstEmployees.SelectedIndex)
                End If
            End If
        End With

    End Sub
    Private Sub btnToPayroll_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnToPayroll.Click

        If lstJobTitles.SelectedItem = "" Then
            MsgBox("Please select a Job Title from the list to continue")
            lstEmployees.Focus()
        ElseIf lstEmployees.SelectedItem = "" Then
            MsgBox("Please select an Employee from the list to continue")
            lstEmployees.Focus()
        Else
            My.Forms.frmPayroll.Show()
            Me.Hide()
        End If

    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click

        Dim msg As String
        Dim style As MsgBoxStyle
        Dim response As MsgBoxResult
        msg = "Are you sure you want to to EXIT the Application?"
        style = MsgBoxStyle.YesNo
        response = MsgBox(msg, style)
        If response = MsgBoxResult.Yes Then
            Me.Close()
            My.Forms.frmPayroll.Close() 'Close both forms if YES
        Else
            ' Do nothing if NO
        End If

    End Sub

    Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AboutToolStripMenuItem.Click
        frmAbout.ShowDialog()
    End Sub
End Class
```

- **Payroll Form Code**

```
Public Class frmPayroll
    'Declare global variables needed for the program
    Dim intIndexDate As Integer

    'Declar global variables needed for the total calculations stored in Memory
    Dim intMemBasicPay, intMemOverTimePay, intMemGrossPay, intMemNetPay As Integer
    Dim sngMemTaxesPay As Single

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Disable menus untill I know how to configure them
        mnuOpen.Enabled = False
        mnuSave.Enabled = False
        mnuClose.Enabled = False
        mnuNew.Enabled = False
        mnuAmend.Enabled = False
        mnuDelete.Enabled = False

        'Set to zero the value of the global variables stored in Memory
        intMemBasicPay = 0
        intMemOverTimePay = 0
        intMemGrossPay = 0
        sngMemTaxesPay = 0
        intMemNetPay = 0

        'Show selected items from welcome form
        txtEmployeeSelected.Text = CStr(strEmployee)
        txtJobTitleSelected.Text = CStr(strJobTitle)
        'txtRatePaySelected.Text = CStr(sngRatePay(indexJobTitle))
        txtRatePaySelected.Text = CStr(Format(sngRatePay(indexJobTitle), "£0.00"))

    End Sub

    Private Sub lstDates_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles lstDates.SelectedIndexChanged
        Dim strDate As String
        'Index the dates
        intIndexDate = lstDates.SelectedIndex
        If intIndexDate >= 0 Then
            strDate = lstDates.Items.Item(intIndexDate)
            txtDate.Text = strDate
        End If
    End Sub

    Private Sub btnNetPay_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnNetPay.Click
        'Declare items needed for calculations of net pay
        Dim sngRatePayConfirm, sngOverTimeRate, sngTaxesPay As Single
        Dim intNoHours, intBasicHours, intOverTimeHours, intBasicPay, intOverTimePay,
intGrossPay, intNetPay As Integer
        'Declare items needed for calcualtions of overtime
        Dim intTotalBasicPay, intTotalOverTimePay, intTotalGrossPay, intTotalNetPay As
Integer
        Dim sngTotalTaxesPay As Single

        'Start validation of input data
        If sngRatePayConfirm = sngRatePay(indexJobTitle) Then

        ElseIf IsNumeric(txtRateOfPay.Text) = False Then
            MsgBox("Please input only numbers in the Rate of Pay field")
            txtRateOfPay.Focus()
        ElseIf IsNumeric(txtNoHours.Text) = False Then
            MsgBox("Please input only numbers in the Number of Hours field")
            txtNoHours.Focus()
        ElseIf Fix(CSng(txtNoHours.Text)) <> (CSng(txtNoHours.Text)) Then
            MsgBox("Please DO NOT put decimals in the Number of Hours field")
            txtNoHours.SelectAll()
            txtNoHours.Focus()
        ElseIf lstDates.SelectedItem = "" Then
            MsgBox("Please select a Date")
            txtDate.Focus()
        Else
            'Start to display input values
            sngRatePayConfirm = CSng(txtRateOfPay.Text)
            intNoHours = CInt(txtNoHours.Text)

            'Start validating overtime, hours worked and basic pay
            If intNoHours <= 0 Then
                MsgBox("Please ensure number of hours is a positive number")
            ElseIf intNoHours <= 8 Then
                intOverTimeHours = 0
                sngOverTimeRate = 0
            ElseIf intNoHours >= 8 Then
                intBasicHours = 8
            ElseIf intNoHours > 16 Then
                MsgBox("It is illegal to work more than 16 hours a day. Please check
the hours and try again")
                intNoHours = 0
                sngOverTimeRate = 0
            End If

            If sngRatePayConfirm < -1 Then
                MsgBox("Please ensure Rate of Pay is a positive number")
            Else
                intOverTimeHours = intNoHours - 8
                sngOverTimeRate = sngRatePayConfirm * 1.5
            End If

            'Start calculating Basic Pay
            intBasicPay = intBasicHours * sngRatePayConfirm
            'Start calculating Overtime
            intOverTimePay = intOverTimeHours * sngOverTimeRate

            'Start calculating Gross Pay, Taxes and Net Pay
            intGrossPay = intBasicPay + intOverTimePay
            sngTaxesPay = intGrossPay * 0.4
            intNetPay = intGrossPay - sngTaxesPay

            'Display overtime calculations
            txtOverTimeHours.Text = CStr(intOverTimeHours)
            txtOverTimeRate.Text = CStr(sngOverTimeRate)

            'Display the payments calculations
            txtBasicPay.Text = CStr(intBasicPay)
            txtOverTimePay.Text = CStr(intOverTimePay)
            txtGrossPay.Text = CStr(intGrossPay)
            txtTaxesPay.Text = CStr(sngTaxesPay)
            txtNetPay.Text = CStr(intNetPay)

            'Clear the input fields and focus on Rate of Pay
            txtRateOfPay.Clear()
            txtNoHours.Clear()
            txtRateOfPay.Focus()

            'Start calculating the Totals by adding the memory values
            intTotalBasicPay = CInt(intBasicPay)
            intMemBasicPay = intMemBasicPay + intTotalBasicPay
            intTotalOverTimePay = CInt(intOverTimePay)

            intMemOverTimePay = intMemOverTimePay + intTotalOverTimePay
            intTotalGrossPay = CInt(intGrossPay)
            intMemGrossPay = intMemGrossPay + intTotalGrossPay
            sngTotalTaxesPay = CSng(sngTaxesPay)
            sngMemTaxesPay = sngMemTaxesPay + sngTotalTaxesPay
            intTotalNetPay = CInt(intNetPay)
            intMemNetPay = intMemNetPay + intTotalNetPay

            'Display the Totals values
            txtTotalBasicPay.Text = CStr(intMemBasicPay)
            txtTotalOverTimePay.Text = CStr(intMemOverTimePay)
            txtTotalGrossPay.Text = CStr(intMemGrossPay)
            txtTotalTaxesPay.Text = CStr(sngMemTaxesPay)
            txtTotalNetPay.Text = CStr(intMemNetPay)
        End If

    End Sub

    Private Sub mmuExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuExit.Click
        Dim msg As String
        Dim style As MsgBoxStyle
        Dim response As MsgBoxResult
        msg = "Are you sure you want to to EXIT the Application?"
        style = MsgBoxStyle.YesNo
        response = MsgBox(msg, style)
        If response = MsgBoxResult.Yes Then
            Me.Close()
            My.Forms.frmWelcome.Close() 'Close both forms if YES
        Else
            ' Do nothing if NO
        End If
    End Sub

    Private Sub mnuOpen_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuOpen.Click
        FileOpen(1, "PayrollData.txt", OpenMode.Output)
    End Sub


    Private Sub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AboutToolStripMenuItem.Click
        frmAbout.ShowDialog()
    End Sub
End Class
```

- **About Form Code**

```
Public Class frmAbout

    Private Sub btnClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnClose.Click
        Me.Close()
    End Sub
End Class
```