16:

Public Class frm16

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

Dim RHR As Integer = txtRHR.Text

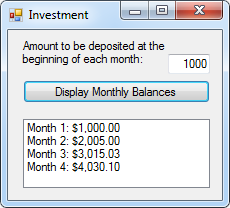
Dim MHR As Integer = 220 - CInt(txtAge.Text)

Dim THR As Integer = ((MHR - RHR) \* 0.6) + RHR

txtTHR.Text = THR

End Sub

End Class



24:

Public Class frm16

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

Dim RHR As Integer = txtRHR.Text

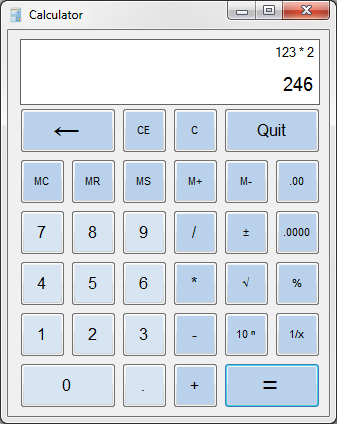
Dim MHR As Integer = 220 - CInt(txtAge.Text)

Dim THR As Integer = ((MHR - RHR) \* 0.6) + RHR

txtTHR.Text = THR

End Sub

End Class

Calculator:

Public Class frmCalc

Dim Working, Storage, Memory As Decimal

Dim IsAnswer As Boolean = False

Dim IsMemory As Boolean = False

Dim IsClear As Boolean = False

Dim Symbol As String

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

lblWorking.Text = 0

IsClear = True

End Sub

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

End

End Sub

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

If lblWorking.Text.Length > 0 Then lblWorking.Text = lblWorking.Text.Remove(lblWorking.Text.Length - 1)

End Sub

Private Sub Clear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click, btnClearEntry.Click

If sender Is btnClear Then lblWorking.Text = 0 : lblStorage.Text = ""

If sender Is btnClearEntry Then lblWorking.Text = 0

IsClear = True

End Sub

Private Sub Memory\_Click(sender As Object, e As EventArgs) Handles btnMemoryClear.Click, btnMemoryReturn.Click, btnMemorySave.Click, btnMemoryAdd.Click, btnMemorySubtract.Click

IsMemory = True

If lblWorking.Text.Length > 0 Then Working = CDec(lblWorking.Text)

If sender Is btnMemoryClear Then Memory = 0

If sender Is btnMemoryReturn Then lblWorking.Text = Memory

If sender Is btnMemorySave Then Memory = Working

If sender Is btnMemoryAdd Then Memory = Memory + Working

If sender Is btnMemorySubtract Then Memory = Memory - Working

End Sub

Private Sub Decimals\_Click(sender As Object, e As EventArgs) Handles btnTwoDecimals.Click, btnFourDecimals.Click

If sender Is btnTwoDecimals And lblWorking.Text.Length > 0 Then lblWorking.Text = FormatNumber(lblWorking.Text, 2)

If sender Is btnFourDecimals And lblWorking.Text.Length > 0 Then lblWorking.Text = FormatNumber(lblWorking.Text, 4)

End Sub

Private Sub Numbers\_Click(sender As Object, e As EventArgs) Handles btnZero.Click, btnOne.Click, btnTwo.Click, btnThree.Click, btnFour.Click, btnFive.Click, btnSix.Click, btnSeven.Click, btnEight.Click, btnNine.Click, btnDecimal.Click

If IsAnswer Then lblWorking.Text = "" : IsAnswer = False

If IsClear Then lblWorking.Text = "" : IsClear = False

If IsMemory Then lblWorking.Text = "" : IsMemory = False

If sender Is btnZero Then lblWorking.Text += "0"

If sender Is btnOne Then lblWorking.Text += "1"

If sender Is btnTwo Then lblWorking.Text += "2"

If sender Is btnThree Then lblWorking.Text += "3"

If sender Is btnFour Then lblWorking.Text += "4"

If sender Is btnFive Then lblWorking.Text += "5"

If sender Is btnSix Then lblWorking.Text += "6"

If sender Is btnSeven Then lblWorking.Text += "7"

If sender Is btnEight Then lblWorking.Text += "8"

If sender Is btnNine Then lblWorking.Text += "9"

If sender Is btnDecimal And lblWorking.Text.IndexOf(".") = -1 Then

If sender Is btnDecimal And lblWorking.Text.Length > 0 Then lblWorking.Text += "."

If sender Is btnDecimal And lblWorking.Text.Length = 0 Then lblWorking.Text += "0."

End If

End Sub

Private Sub Operations\_Click(sender As Object, e As EventArgs) Handles btnAdd.Click, btnSubtract.Click, btnMultiply.Click, btnDivide.Click

If sender Is btnAdd And lblWorking.Text.Length > 0 Then Storage = CDec(lblWorking.Text) : Symbol = "+" : lblStorage.Text = lblWorking.Text & " " & Symbol & " "

If sender Is btnSubtract And lblWorking.Text.Length > 0 Then Storage = CDec(lblWorking.Text) : Symbol = "-" : lblStorage.Text = lblWorking.Text & " " & Symbol & " "

If sender Is btnMultiply And lblWorking.Text.Length > 0 Then Storage = CDec(lblWorking.Text) : Symbol = "\*" : lblStorage.Text = lblWorking.Text & " " & Symbol & " "

If sender Is btnDivide And lblWorking.Text.Length > 0 Then Storage = CDec(lblWorking.Text) : Symbol = "/" : lblStorage.Text = lblWorking.Text & " " & Symbol & " "

lblWorking.Text = ""

End Sub

Private Sub Functions\_Click(sender As Object, e As EventArgs) Handles btnSign.Click, btnSqrt.Click, btnPercent.Click, btnExponent.Click, btnInverse.Click

If sender Is btnSign Then

lblStorage.Text = "Sign(" & lblWorking.Text & ")"

lblWorking.Text = -1 \* lblWorking.Text

End If

If sender Is btnSqrt Then

lblStorage.Text = "Sqrt(" & lblWorking.Text & ")"

lblWorking.Text = Math.Sqrt(lblWorking.Text)

End If

If sender Is btnPercent And lblWorking.Text.IndexOf("%") = -1 Then

lblStorage.Text = "Percent(" & lblWorking.Text & ")"

lblWorking.Text = FormatPercent(lblWorking.Text)

End If

If sender Is btnExponent Then

lblStorage.Text = "Exponent(" & lblWorking.Text & ")"

lblWorking.Text = 10 ^ lblWorking.Text

End If

If sender Is btnInverse Then

lblStorage.Text = "Inverse(" & lblWorking.Text & ")"

lblWorking.Text = 1 / lblWorking.Text

End If

End Sub

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

lblStorage.Text = lblStorage.Text & lblWorking.Text

Working = CDec(lblWorking.Text)

If Symbol = "+" Then lblWorking.Text = Storage + Working

If Symbol = "-" Then lblWorking.Text = Storage - Working

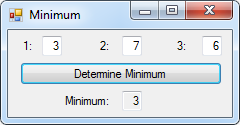
If Symbol = "\*" Then lblWorking.Text = Storage \* Working

If Symbol = "/" Then lblWorking.Text = Storage / Working

IsAnswer = True

End Sub

End Class

22:

Public Class frm22

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

If (txt1.Text < txt2.Text) And (txt1.Text < txt3.Text) Then txtResult.Text = txt1.Text

If (txt2.Text < txt1.Text) And (txt2.Text < txt3.Text) Then txtResult.Text = txt2.Text

If (txt3.Text < txt1.Text) And (txt3.Text < txt2.Text) Then txtResult.Text = txt3.Text

End Sub

End Class