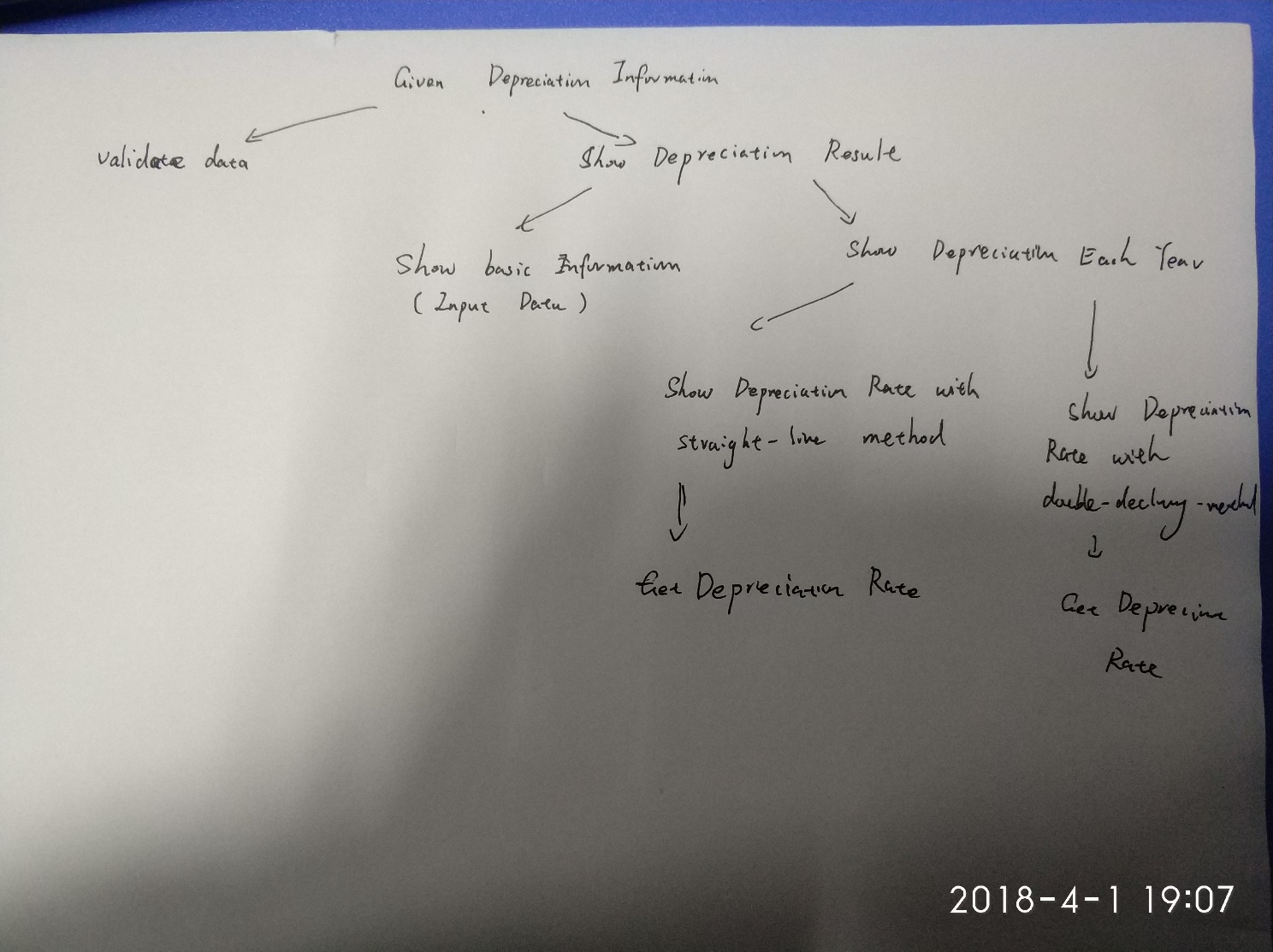
1. **A Structured Diagram**



1. **Pseudocode**

Click button Calcalate (Sub procedure btnCalculate\_Click)

VALIDATE data (Function TrueData)

SHOW depreciation result (Sub procedure ShowDepreciationResult)

SHOW basic info (Sub procedure ShowBasicInfo)

If Method = straight-line

Compute depreciation rate (Function GetDepreciation Rate)

SHOW Depreciation with straight-line Method (Sub procedure ShowSLDepreciation)

For 1 to usageLife

Calculate depreciation value, current value based on depreciation method

OUTPUT each year depreciation info

Else

Compute depreciation rate (Function GetDepreciation Rate)

SHOW Depreciation with double-declining Method (Sub procedure ShowDDBDepreciation

For 1 to usageLife

Calculate depreciation value, current value based on depreciation method

OUTPUT each year depreciation info

Click Button Clear (Sub procedure btnClear\_Click)

Clear all the text box and reset the radio box

Click Button Exit (Sub procedure btnExit\_Click)

Exit Application

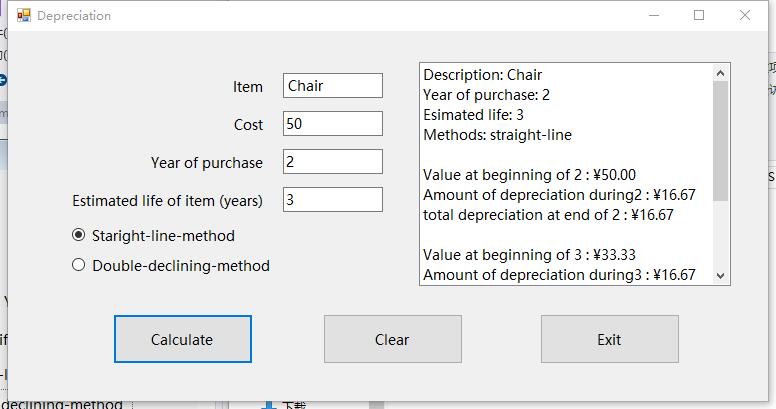
1. **Object table**

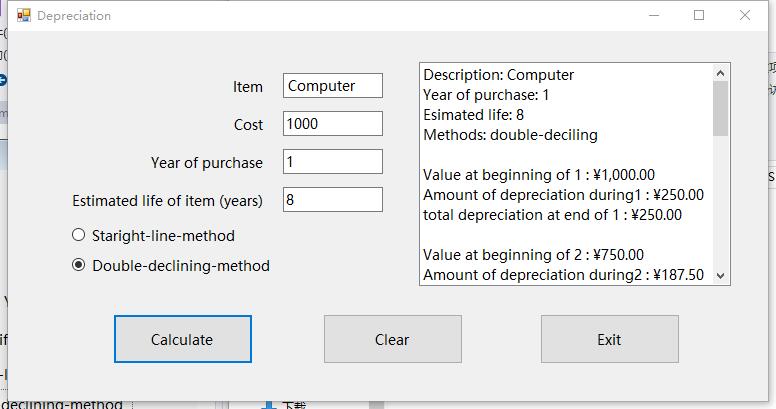
| **Object** | **Property** | **Setting** |
| --- | --- | --- |
| lblCost | Text | Item price |
| lblItem | Text | Item name |
| lblYearOfPurchase | Text | Purchase year |
| lblEstimatedLifeOfItem | Text | Item‘s estimated life |
| txtItem | Text | Item name |
| txtCost | Text | Item price |
| txtYearOfPurchase | Text | Purchase year |
| txtEstimatedLifeOfItem | Text | Item‘s estimated life |
| radStraight | Text | Straight-line method |
| radStraight | Checked | True |
| radDouble | Text | Double-declining  method |
| radDouble | Checked | False |
| lstResult | Text | Show info |
| btnCalculate | Text | Calculate value |
| btnClear | Text | Empty value |
| btnExit | Text | Exit procedure |
| grpMethods | Text | Depreciation Methods |

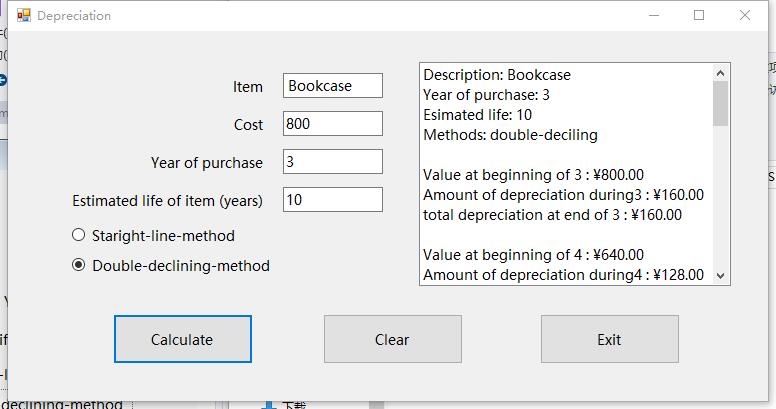
**3. Table of Products and details**

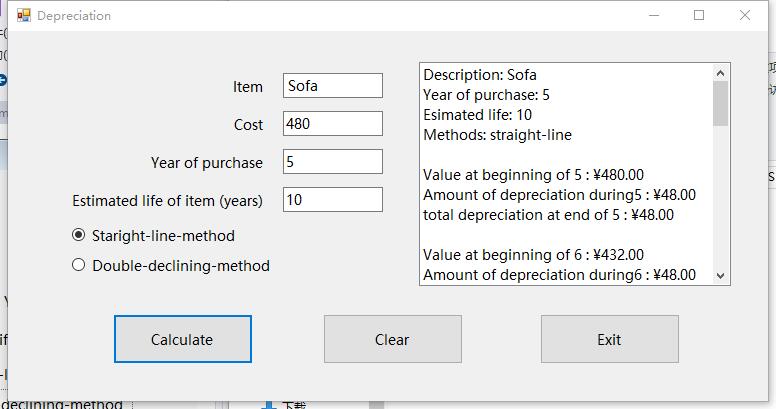
| **Product Name** | **Original Cost** | **Years of purchase** | **Estimated Life of Product** | **Depreciation Type** |
| --- | --- | --- | --- | --- |
| Chair | $50 | 2 | 3 | Straight line |
| Computer | $1000 | 1 | 8 | Double-declining |
| Bookcase | $800 | 3 | 10 | Double-declining |
| Sofa | $480 | 5 | 10 | Straight line |

1. **Deliver 4 Printouts**









**5. Coding Design**

Public Class Depreciation

Private Sub Label1\_Click(sender As Object, e As EventArgs) Handles lblItem.Click

End Sub

' Determine whether the data format is correct

Function TrueData() As String

If txtItem.Text = "" Then

Return "Please enter right item value!"

ElseIf txtCost.Text = "" Or isNumeric(txtCost.Text) = False Then

Return "Please enter right cost value!"

ElseIf txtYearOfPurchase.Text = "" Or IsNumeric(txtYearOfPurchase.Text) = False Then

Return "Please enter right Tear of Purchase value!"

ElseIf txtEstimatedLifeOfItem.Text = "" Or IsNumeric(txtEstimatedLifeOfItem.Text) = False Then

Return "Please enter right Estimated Life of Item!"

ElseIf CDbl(txtEstimatedLifeOfItem.Text) = 0 Then

Return "Please enter a EstimatedLifeOfItem value > 0 !!"

End If

' If the data is properly formatted, return "OK"

Return "OK"

End Function

' Display the basic information section on list

Private Sub DisplayBasicInfo(startYear As String, usageLife As String, itemName As String, depreciationMethod As String)

lstResult.Items.Clear()

lstResult.Items.Add("Description: " & itemName)

lstResult.Items.Add("Year of purchase: " & startYear)

lstResult.Items.Add("Esimated life: " & usageLife)

lstResult.Items.Add("Methods: " & depreciationMethod)

lstResult.Items.Add("")

End Sub

' Calculation of depreciation rate according to the depreciation method

Function GetDepreciationRate(usageLife As Double, depreciationMethod As String) As Double

Dim depreciationRate As Double

If depreciationMethod = "straight-line" Then

depreciationRate = 1.0 / usageLife

Else

depreciationRate = 2.0 / usageLife

End If

Return depreciationRate

End Function

' Display the annual depreciation information

Private Sub ShowDepreciationResult(ByVal startYear As Integer, usageLife As Double, ByVal itemValue As Double, itemName As String, depreciationMethod As String)

Dim depreciationRate, depreciationValue, totalDepreciationValue As Double

Dim currentValue As Double = itemValue

' Show the basic information

DisplayBasicInfo(startYear, CStr(usageLife), CStr(itemName), depreciationMethod)

' Compute depreciation rate

depreciationRate = GetDepreciationRate(usageLife, depreciationMethod)

' Set the default amount of depreciation, that is, the amount of depreciation in the case of straight-line.

depreciationValue = depreciationRate \* itemValue

' The initial total depreciation is 0

totalDepreciationValue = 0

For index = 1 To usageLife

If currentValue <= 0 Then

Exit For

End If

lstResult.Items.Add("Value at beginning of " & CStr(startYear) & " : " & currentValue.ToString("C"))

' According to the depreciation method, calculate depreciation value, current value based on depreciation method

If depreciationMethod = "straight-line" Then

currentValue -= depreciationValue

totalDepreciationValue += depreciationValue

Else

depreciationValue = currentValue \* depreciationRate

totalDepreciationValue += depreciationValue

currentValue -= depreciationValue

End If

lstResult.Items.Add("Amount of depreciation during" & CStr(startYear) & " : " & depreciationValue.ToString("C"))

lstResult.Items.Add("total depreciation at end of " & CStr(startYear) & " : " & totalDepreciationValue.ToString("C"))

lstResult.Items.Add("")

startYear += 1

Next

End Sub

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

Dim method As String

If TrueData() = "OK" Then

' Check depreciation method

If radStraight.Checked = True Then

method = "straight-line"

Else

method = "double-deciling"

End If

ShowDepreciationResult(CInt(txtYearOfPurchase.Text), CDbl(txtEstimatedLifeOfItem.Text), CDbl(txtCost.Text), txtItem.Text, method)

Else

MessageBox.Show(TrueData(), "Warning")

End If

End Sub

' Click Clear to empty all values and restore the single box to the default value (straight-line)

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

txtCost.Clear()

txtItem.Clear()

txtYearOfPurchase.Clear()

txtEstimatedLifeOfItem.Clear()

radDouble.Checked = False

radStraight.Checked = True

End Sub

' Click Exit to exit applications

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

End

End Sub

End Class