

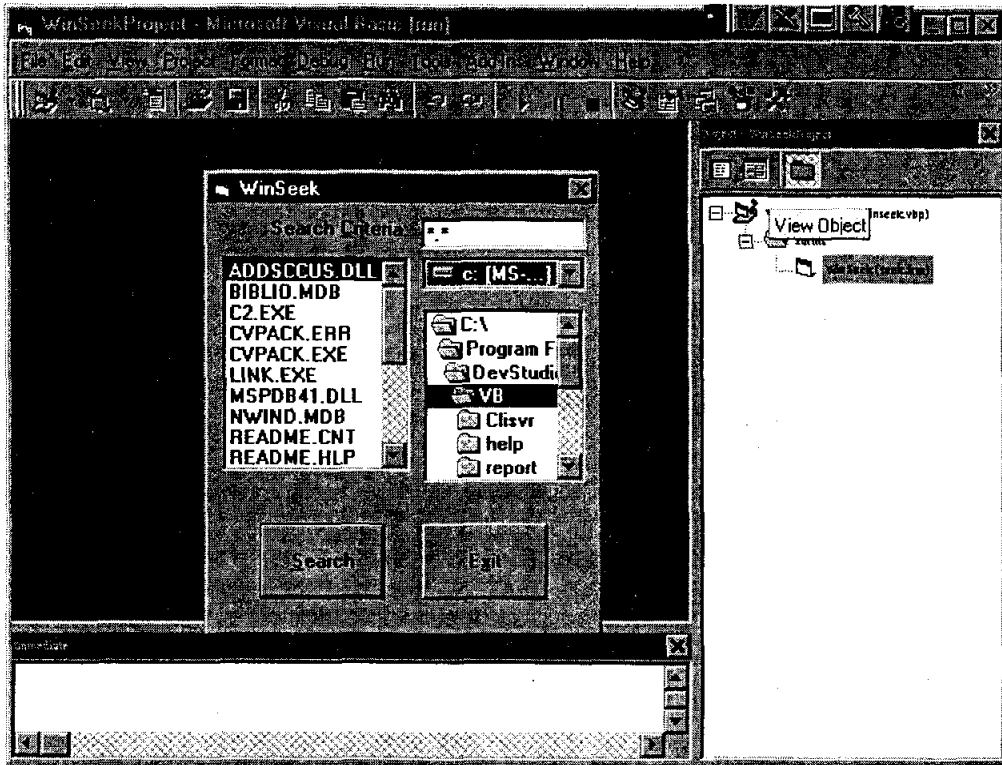
## บทที่ 14

### การเขียนโปรแกรมตามรูปแบบของ Visual

**บทนำ** การสร้างโปรแกรมในยุคปัจจุบันนี้จะเห็นว่ามี Software Technology ที่สร้างขึ้นเพื่อช่วยอำนวยความสะดวกมากมายหลายอย่างอาทิเช่น เครื่องช่วยในการออกแบบส่วนของระบบประสาน (User Interface) เครื่องมือช่วยสร้างโปรแกรม (Code Generator) รวมทั้งสิ่งอำนวยความสะดวกอื่นๆที่ช่วยในการ Debug and Testing Program ผนวกกับการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming) ซึ่งทำให้ความสามารถในเรื่องของการ “นำกลับมาใช้ใหม่ (Reuse)” โดดเด่นขึ้นมา สิ่งอำนวยความสะดวกทั้งปวงที่เกิดขึ้นจากวิวัฒนาการทาง Software Technology ล้วนแล้วแต่ทำให้อาชีพการสร้างโปรแกรมสะดวกสบายขึ้น จนเกิดสมญานามของอาชีพใหม่ขึ้นมาคือ อาชีพ “ประกอบโปรแกรม(Assembly Program)” ซึ่งจะแตกต่างจากอาชีพการเขียนโปรแกรม ความแตกต่างอยู่ในส่วนที่ว่าคนที่ประกอบโปรแกรม นั้นจะมีหน้าที่เสาะหาโปรแกรมย่อยๆต่างที่ทำสำเร็จแล้ว และตรงตามวัตถุประสงค์ แล้วนำมาเชื่อมต่อกันให้สำเร็จกลายเป็นระบบโปรแกรมขึ้นมา อาจจะเปรียบได้กับการประกอบอาหารที่ไม่ต้องเตรียมวัสดุในการปรุงอาหารขึ้นมาเอง แต่ไปซื้อของสำเร็จรูปที่ทำสำเร็จแล้วมาประกอบเป็นอาหาร อาทิเช่นเครื่องแกงสำเร็จรูป กะทิสำเร็จรูป เนื้อไก่ที่หั่นไว้เรียบร้อยแล้ว ตลอดจนเครื่องปรุงอย่างอื่นๆ แล้วนำมาประกอบกันเข้าเป็น “แกงไก่ 1 ถ้วย” ด้วยวิธีการเช่นนี้จึงเกิดมิติใหม่ของการสร้างโปรแกรม ที่เรียกว่า “Visual Style” ข้อดีของวิธีการเช่นนี้ก็คือ ง่าย เร็ว สบาย และค่อนข้างจะปลอดภัยจาก Syntax Error แต่ก็มีข้อเสียในส่วนที่ผู้สร้างจะไม่สามารถรู้สึกซึ่งถึงความเป็นมา และขั้นตอนในการเหมือนเช่นการสร้างโปรแกรมแบบ Procedural

**การสร้างโปรแกรมเชิงวิซวล** โปรแกรมเชิงวิซวลนั้นจะเป็นวิธีการสร้างโปรแกรมโดยอาศัยความช่วยเหลือในการสร้าง Interface System และ Code Generator ซอฟต์แวร์ประเภทนี้ที่ปรากฏในการใช้ในปัจจุบันนี้จะมีมากมายมา ส่วนใหญ่มักจะอิงจากภาษาคอมพิวเตอร์ที่เป็น Third Generation Language แล้วนำมาพัฒนาเพิ่มเครื่องมือในการอำนวยความสะดวกขึ้น ตัวอย่างของภาษาพวกนี้ เช่น Visual Basic , Delphi (Visual Pascal) , Asna RPG , Visual C เป็นต้น ภาษาตระกูลนี้จะมีพื้นฐานคล้ายๆกันคือสร้างเครื่องมือในการออกแบบเชิงวัตถุให้เราใช้ และมี Code Generator ให้ออกจากนั้นยังเครื่องมือช่วยเหลืออย่างอื่นในการช่วยตรวจสอบและแก้ไขโปรแกรม แต่ถึงกระนั้นก็ตามผู้สร้างโปรแกรมก็จำเป็นจะต้องมีความรู้ในส่วนของการสร้างกรรมวิธีการทำงานบางอย่างเอง เช่น การเขียนขั้นตอนในการเรียงลำดับ การเขียนคำสั่งในการรับข้อมูล ต่างๆเป็นต้น ตัวอย่างกรณีต่างๆในการสร้างโปรแกรมเชิงวิซวลนี้ จะใช้ภาษา VB (Visual Basic) เป็นเครื่องมือ

ภาพที่ 14.1 เป็นภาพของระบบประสาน (Interface) ที่ออกแบบโดยใช้ VB



โปรแกรมที่สร้างจากส่วนช่วยเหลือของ Code Generator

VERSION 5.00

Begin VB.Form WinSeek

BackColor = &H00C0C0C0&

BorderStyle = 3 'Fixed Dialog

Caption = "WinSeek"

ClientHeight = 5190

ClientLeft = 1920

ClientTop = 1890

ClientWidth = 5175

BeginProperty Font

Name = "MS Sans Serif

Size = 8.25

Charset = 222

```

Weight      = 700
Underline   = 0 'False
Italic      = 0 'False
Strikethrough = 0 'False
EndProperty
ForeColor   = &H00000080&
MaxButton   = 0 'False
MinButton   = 0 'False
ScaleHeight = 5190
ScaleWidth  = 5175
Begin VB.PictureBox Picture2
    BorderStyle = 0 'None
    Height      = 2895
    Left        = 720
    ScaleHeight = 2895
    ScaleWidth  = 3855
    TabIndex    = 8
    Top         = 1080
    Visible     = 0 'False
    Width       = 3855
Begin VB.ListBox lstFoundFiles
    Height      = 2235
    Left        = 120
    TabIndex    = 11
    Top         = 720
    Width       = 3375
End
Begin VB.Label lblCount
    Caption     = "0"
    Height      = 255
    Left        = 1200
    TabIndex    = 10
    Top         = 120
    Width       = 1095
End

```

```

Begin VB.Label lblfound
  Caption      = "&Files Found:"
  Height       = 255
  Left         = 120
  TabIndex     = 9
  Top         = 120
  Width        = 1095
End
End
Begin VB.PictureBox Picture1
  BorderStyle  = 0 'None
  Height       = 2895
  Left         = 0
  ScaleHeight  = 2895
  ScaleWidth   = 3735
  TabIndex     = 2
  Top         = 120
  Width        = 3735
End
Begin VB.DriveListBox drvList
  Height       = 1530
  Left         = 2040
  TabIndex     = 7
  Top         = 480
  Width        = 1575
End
End
Begin VB.DirListBox dirList
  Height       = 1695
  Left         = 2040
  TabIndex     = 6
  Top         = 960
  Width        = 1575
End
End
Begin VB.FileListBox filList
  Height       = 2040
  Left         = 120

```

```

    TabIndex = 5
    Top = 480
    Width = 1815
End
Begin VB.TextBox txtSearchSpec
    Height = 285
    Left = 2040
    TabIndex = 4
    .Text
    Top = 120
    Width = 1575
End
Begin VB.Label lblCriteria
    Caption = "Search &Criteria:"
    Height = 255
    Left = 600
    TabIndex = 3
    Top = 120
    Width = 1335
End
End
Begin VB.CommandButton cmdSearch
    BackColor = &H00C0C0C0&
    Caption = "&Search"
    Default = -1 'True
    Height = 720
    Left = 480
    TabIndex = 0
    Top = 3000
    Width = 1200
End
Begin VB.CommandButton cmdExit
    BackColor = &H00C0C0C0&
    Caption = "E&xit"
    Height = 720

```

```

    Left      = 2040
    TabIndex  = 1
    Top       = 3000
    Width     = 1200
End
End
Attribute VB_Name = "WinSeek"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Dim SearchFlag As Integer ' Used as flag for cancel and other operations

Private Sub cmdExit_Click()
    If cmdExit.Caption = "E&xit" Then
        E n d
    Else
        If user chose Cancel, just end Search.
        SearchFlag = False
    End If
End Sub

Private Sub cmdSearch_Click()
' Initialize for search, then perform recursive search.
Dim FirstPath As String, DirCount As Integer, NumFiles As Integer
Dim result As Integer
Check what the user did last,
If cmdSearch.Caption = "&Reset" Then ' If just a reset, initialize and exit.
    ResetSearch
    txtSearchSpec.SetFocus
    Exit Sub
End If

' Update dirList.Path if it is different from the currently
selected directory: otherwise perform the search.

```

```

If dirList.Path <> dirList.List(dirList.ListIndex) Then
    dirList.Path = dirList.List(dirList.ListIndex)
    Exit Sub      Exit so user can take a look before searching
End If

' Continue with the search.
Picture2.Move 0,0
Picture1.Visible = False
Picture2.Visible = True

cmdExit.Caption = "Cancel"

filList.Pattern = txtSearchSpec.Text
FirstPath = dirList.Path
DirCount = dirList.ListCount

' Start recursive direcorey search.
NumFiles = 0          Reset found files indicator.
result = DirDiver(FirstPath, DirCount. "")
filList.Path = dirList.Path
cmdSearch.Caption = "&Reset"
cmdSearch.SetFocus
cmdExit.Caption = "E&xit"
End Sub

Private Function DirDiver(NewPath As String, DirCount As Integer, BackUp As String) As Integer
' Recursively search directories from NewPath down...
' NewPath is searched on this recursion.
BackUp is origin of this recursion.
DirCount is number of subdirectories in this directory.
Static FirstErr As Integer
Dim DirsToPeek As Integer. AbandonSearch As Integer, ind As Integer
Dim OldPath As String, ThePath As String, entry As String
Dim retval As Integer
    SearchFlag = True      ' Set flag so the user can interrupt.
    DirDiver = False      Set to True if there is an error.

```

```

retval = DoEvents()      ' Check for events (for instance, if the user chooses Cancel).
If SearchFlag = False Then
    DirDiver = True
    Exit Function
End If
On Local Error GoTo DirDriverHandler
DirsToPeek = dirList.ListCount      How many directories below this?
Do While DirsToPeek > 0 And SearchFlag = True
    OldPath = dirList.Path          Save old path for next recursion.
    dirList.Path = NewPath
    If dirList.ListCount > 0 Then
        ' Get to the node bottom.
        dirList.Path = dirList.List(DirsToPeek - 1)
        AbandonSearch = DirDiver((dirList.Path), DirCount%, OldPath)
    End If
    ' Go up one level in directories.
    DirsToPeek = DirsToPeek - 1
    If AbandonSearch = True Then Exit Function
Loop
' Call function to enumerate files.
If fileList.ListCount Then
    If Len(dirList.Path) <= 3 Then      ' Check for 2 bytes/character
        ThePath = dirList.Path        ' If at root level, leave as is...
    Else
        ThePath = dirList.Path + "\   ' Otherwise put "\" before the filename.
    End If
    For ind = 0 To fileList.ListCount - 1      Add conforming files in this directory to the list box
        entry = ThePath + fileList.List(ind)
        lstFoundFiles.AddItem entry
        lblCount.Caption = Str(Val(lblCount.Caption) + 1)
    Next ind
End If
If BackUp <> "" Then      ' If there is a superior directory, move It.
    dirList.Path = BackUp
End If

```



```

Exit Function

DirDriverHandler:
    If Err = 7 Then          ' If Out of Memory error occurs, assume the list box just got full.
        DirDiver = True    ' Create Msg and set return value AbandonSearch.
        MsgBox "You've filled the list box. Abandoning search..."
        Exit Function      ' Note that the exit procedure resets Err to 0.
    Else                    ' Otherwise display error message and quit.
        MsgBox Error
    End
End If

End Function

Private Sub DirList_Change()
    ' Update the file list box to synchronize with the directory list box.
    filList.Path = dirList.Path
End Sub

Private Sub DirList_LostFocus()
    dirList.Path = dirList.List(dirList.ListIndex)
End Sub

Private Sub DrvList_Change()
    On Error GoTo DriveHandler
    dirList.Path = drvList.Drive
Exit Sub'

DriveHandler:
    drvList.Drive = dirList.Path
Exit Sub

End Sub

Private Sub Form_Load()
    Picture2.Move 0, 0
    Picture2.Width = WinSeek.ScaleWidth
    Picture2.BackColor = WinSeek.BackColor

```

```

IblCount.BackColor = WinSeek.BackColor
IblCriteria.BackColor = WinSeek.BackColor
Iblfound.BackColor = WinSeek.BackColor
Picture1 .Move 0, 0
Picture1 .Width = WinSeek.ScaleWidth
Picture1 .BackColor = WinSeek.BackColor
End Sub

Private Sub Form-Unload(Cancel As integer)
    End
End Sub

Private Sub ResetSearch()
    ' Reinitialize before starting a new search.
    lstFoundFiles.Clear
    IblCount.Caption = 0
    SearchFlag = False          ' Flag indicating search in progress.
    Picture2.Visible = False
    cmdSearch.Caption = "&Search"
    cmdExit.Caption = "E&xit"
    Picture1 .Visible = True
    dirList.Path = CurDir: drvList.Drive = dirList.Path ' Reset the path.
End Sub

Private Sub txtSearchSpec_Change()
    ' Update file list box if user changes pattern.
    filList.Pattern = txtSearchSpec.Text
End Sub

Private Sub txtSearchSpec_GotFocus()
    txtSearchSpec.SelStart = 0    ' Highlight the current entry.
    txtSearchSpec.SelLength = Len(txtSearchSpec.Text)
End Sub

```

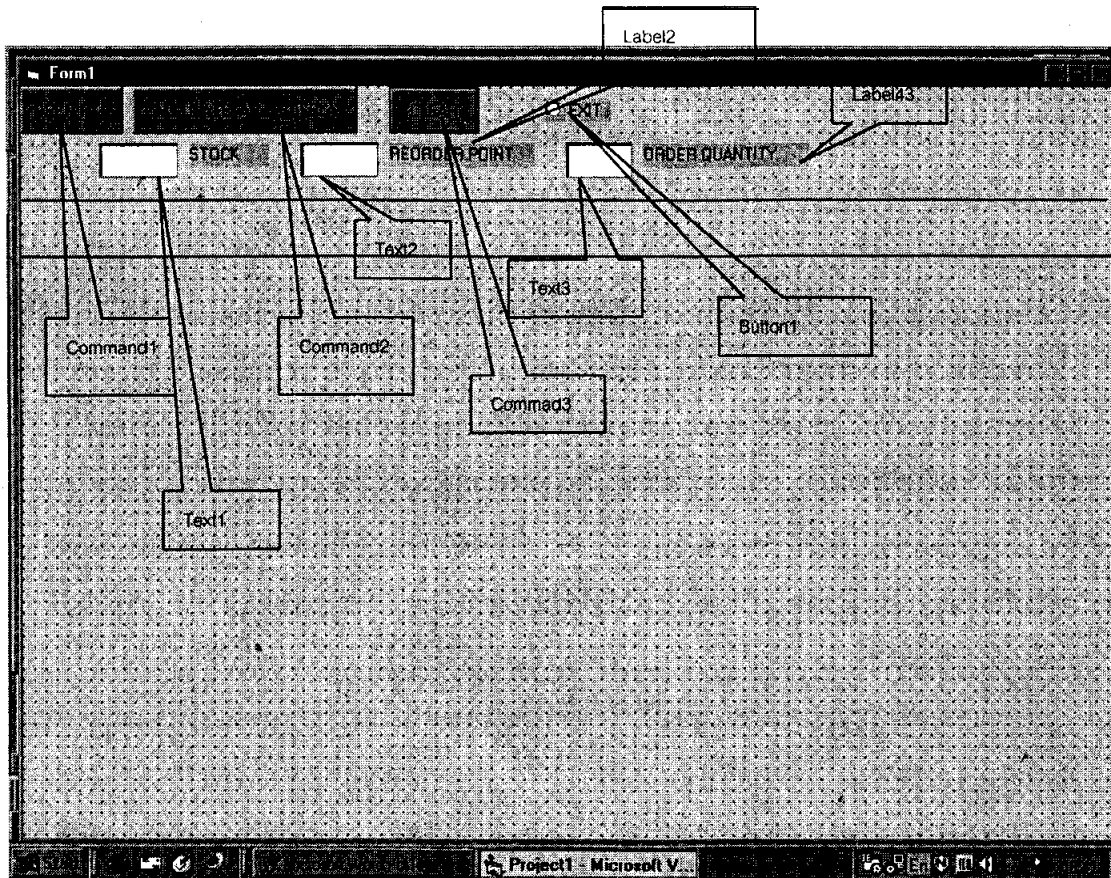
## การสร้างโปรแกรมแบบวิซวล (Visual Programming)

การสร้างโปรแกรมแบบวิซวลนั้นจะมีขั้นตอนเพิ่มเติมนอกเหนือจากการเขียนโปรแกรมในรูปแบบของ Procedural ในส่วนที่จะต้องมีการนำวัตถุที่อยู่ในคลังมาออกแบบระบบประสาน ซึ่งจะเรียกว่าการออกแบบระบบประสาน (User Interface) โดยที่ภายหลังการออกแบบแล้วในส่วนนี้ซอฟต์แวร์ที่ใช้จะทำหน้าที่ สร้าง Code ของโปรแกรมให้เอง นอกเหนือจากนี้ผู้สร้างจะต้องออกแบบเหตุการณ์ (Event) จะมีหน้าที่ผลักดัน ให้วัตถุนั้นสามารถดำเนินการได้ ตัวอย่างของเหตุการณ์ที่กำหนดนั้นถ้าใช้ Mouse เป็นเครื่องมือก็อาจจะใช้ On Mouse Click , On Mouse Double Click , On Mouse Move หรือถ้าใช้เป็นพิมพ์ก็อาจจะใช้เหตุการณ์ที่เป็นเรื่องการบังคับเป็นพิมพ์ เช่น On KeyPress , On Key Change เป็นต้น การเลือกเหตุการณ์ที่จะใช้นั้น ผู้ออกแบบต้องนึกถึงการใช้งานของผู้ใช้ประกอบด้วยว่าลักษณะใดที่เป็นการอำนวยความสะดวกให้กับผู้ใช้งานที่สุด ลำดับถัดไปหลังจากที่ออกแบบระบบประสาน และเลือก Event ที่จะใช้งานแล้ว ผู้สร้างโปรแกรมจะต้องกำหนด กรรมวิธีที่จะดำเนินงานภายใน Object นั้นๆ ร่วมกับการออกแบบโครงสร้างของตัวแปรด้วย ซึ่งนี่ไม่พ้นในการที่จะนำความรู้จากบทที่แล้วๆ มาใช้งาน ดังนั้นถึงแม้ว่าจะใช้การสร้างโปรแกรมเชิงวิซวลแล้วก็ตาม แต่พื้นฐานส่วนหนึ่งก็ยังคงต้องใช้ความรู้การเขียนโปรแกรมแบบดั้งเดิมเข้ามาร่วมด้วย

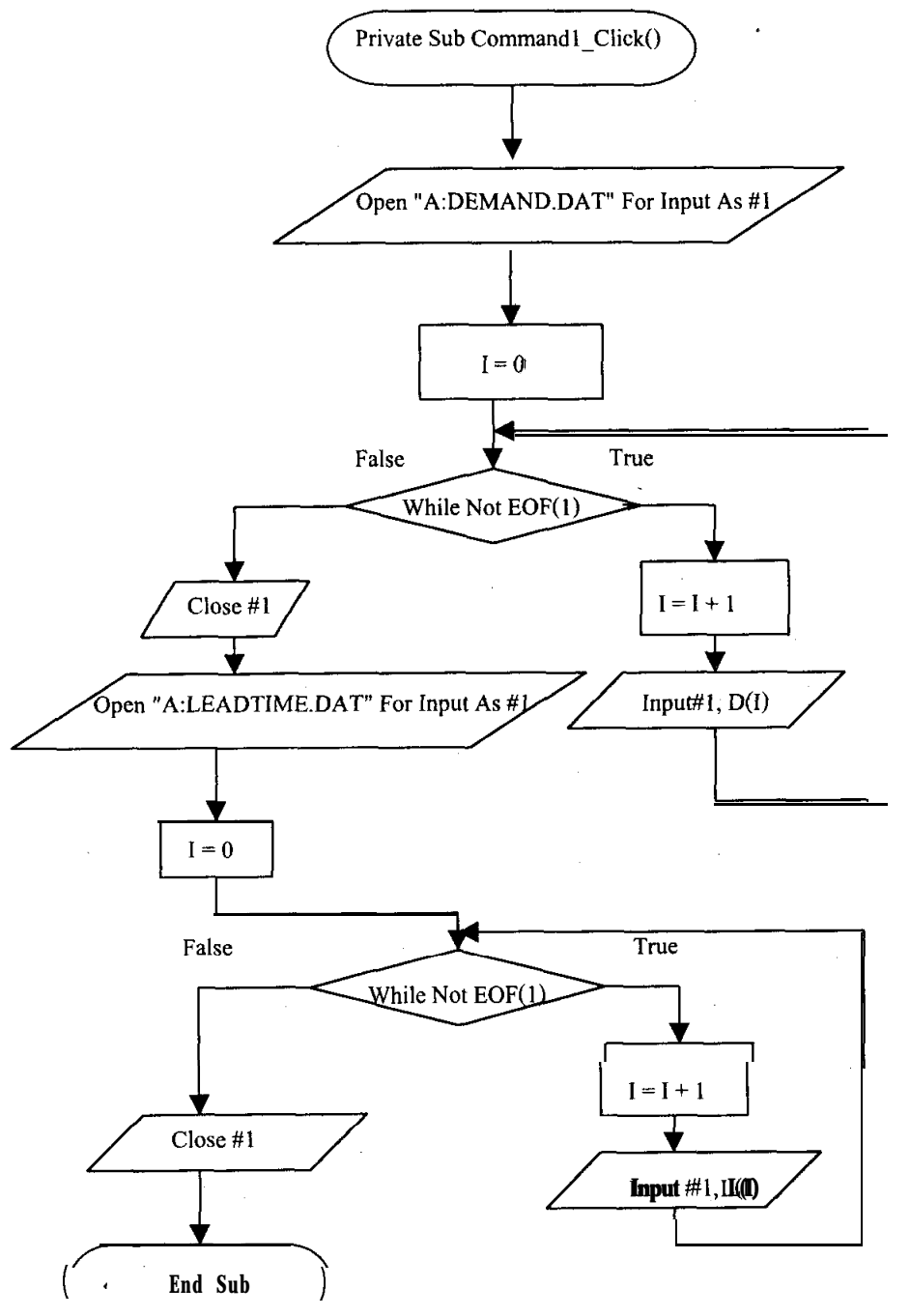
การออกแบบระบบประสาน เป็นขั้นตอนที่ผู้ออกแบบโปรแกรมจะต้องแบ่งแยกกิจกรรมของระบบงานให้ออกเป็นส่วนๆ ซึ่งเป็นลักษณะเดียวกับการออกแบบเชิง Modular Approach เพียงแต่จะแตกต่างกันในตรงที่ว่าแทนที่จะมองเป็น โมดูล เรากลับมองเป็นวัตถุ (Object ) แทน และแทนที่จะสั่งให้ Procedure นั้นทำงานโดยการสั่งด้วยข้อมูลในรูปแบบที่เป็น Data Driven Program กลับกำหนดให้เหตุการณ์เป็นผู้สั่งให้วัตถุนั้นดำเนินงาน (Event Driven Program)

ตัวอย่างกรณีศึกษาต่อไปนี้จะเป็นการสร้างโปรแกรมเชิงวิซวล

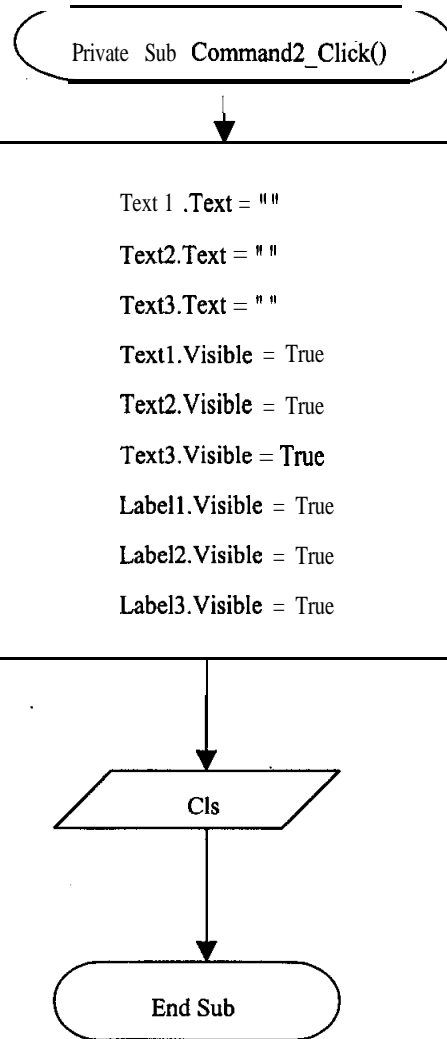
กรณีศึกษาที่ 1 เป็นกรณีของการทำงานเพื่อตั้งค่าจุดสั่งซื้อที่เหมาะสมสำหรับสินค้าชนิดหนึ่ง



การสั่ง Command1 โดยใช้ Mouse Click



CT 211 283

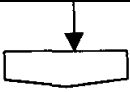


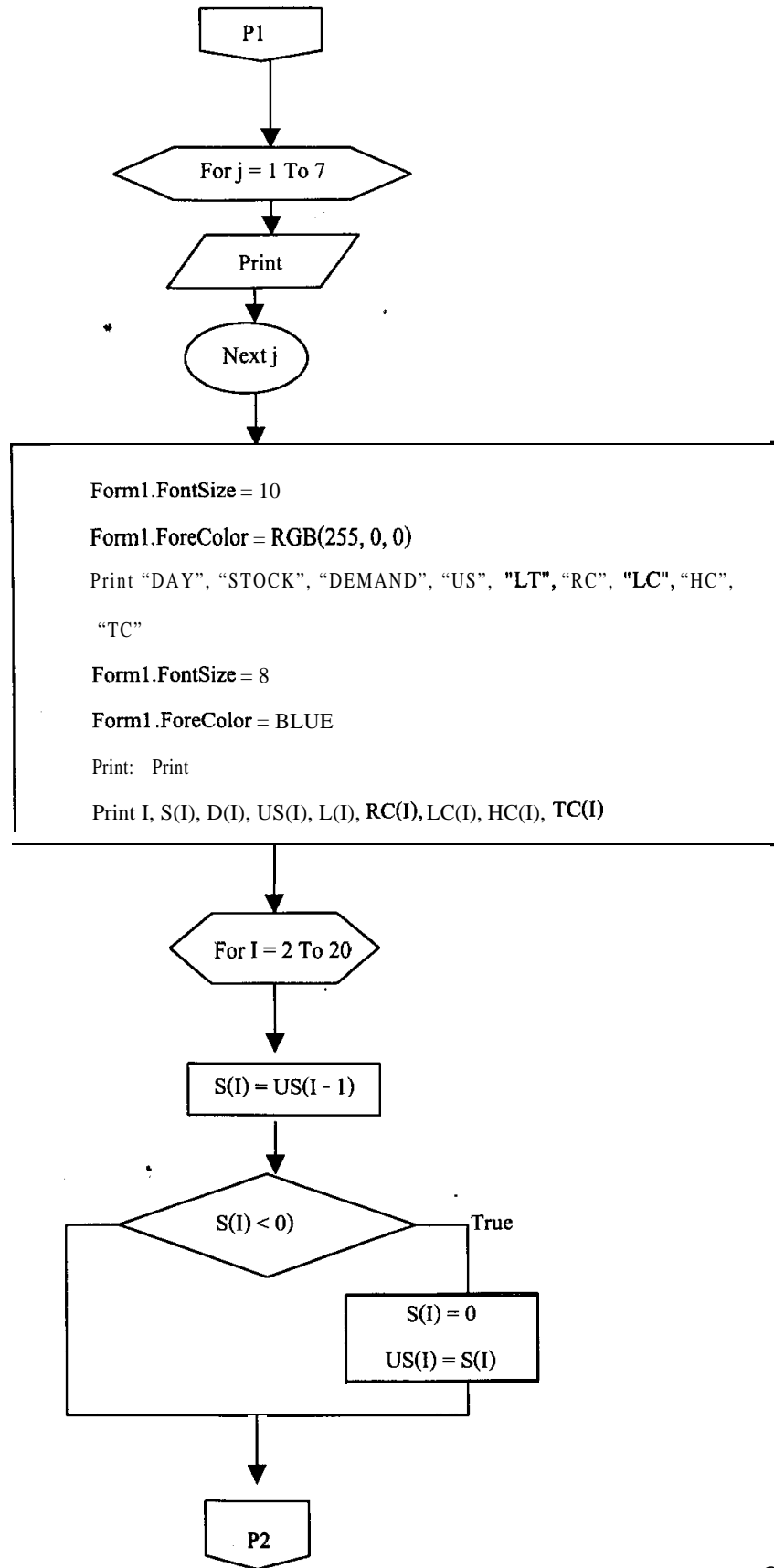
Private Sub Command3\_Click()

```
Dim S(20), US(20), RC(20) As String
Dim HC(20), LC(20) As Integer
Dim TC(20) As Long
Dim TEST As Boolean
```

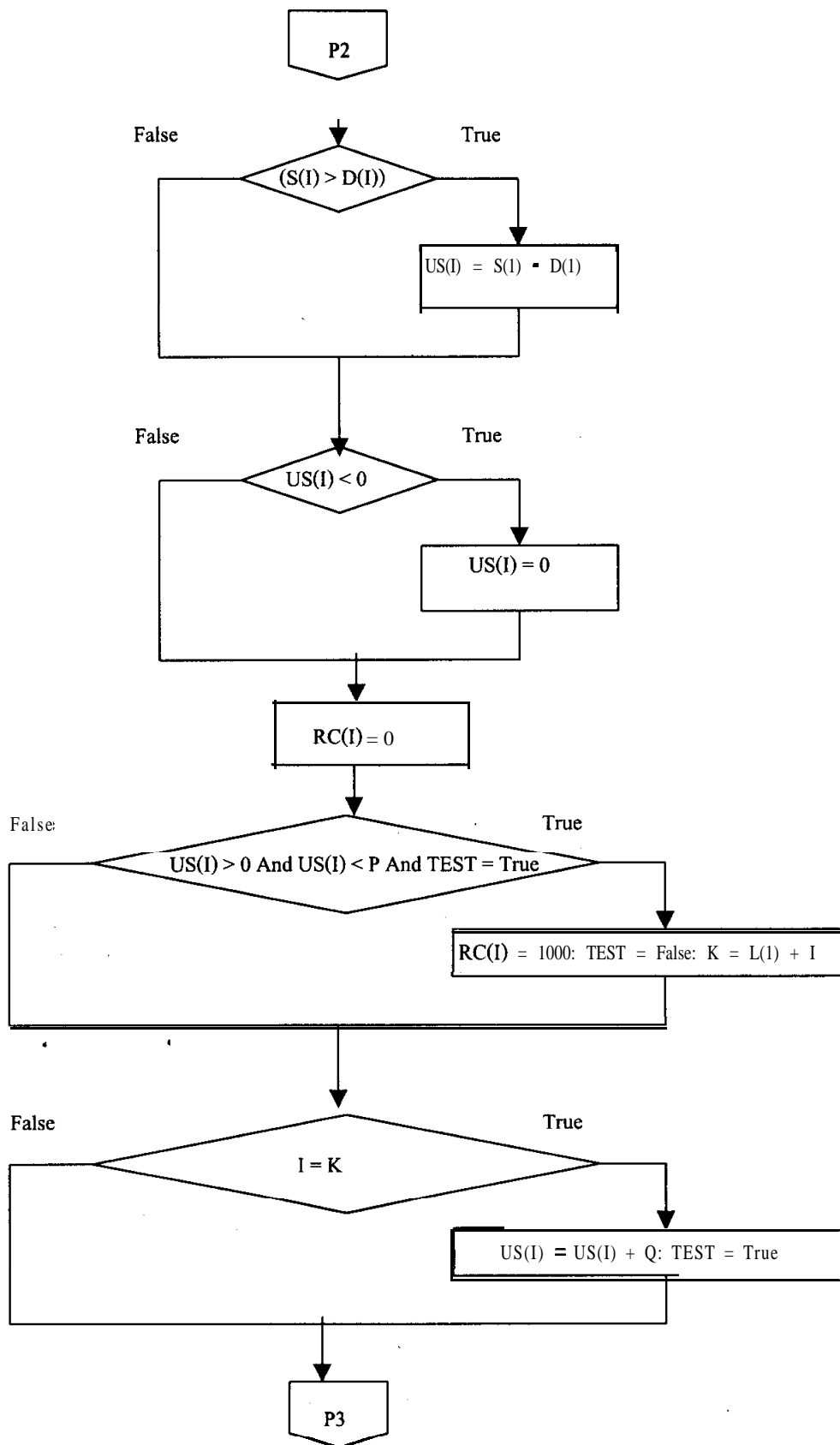
```
Form1.Caption = " INFORMATION ON STOCK
DECISION FOR PRODUCT
Form1.FillColor = white
Command1.Enabled = FLASE
Text1.Visible = False
Text2.Visible = False
Text3.Visible = False
Label1.Visible = False
Label2.Visible = False
Label3.Visible = False
P = Val(Text2.Text)
Q = Val(Text3.Text)
```

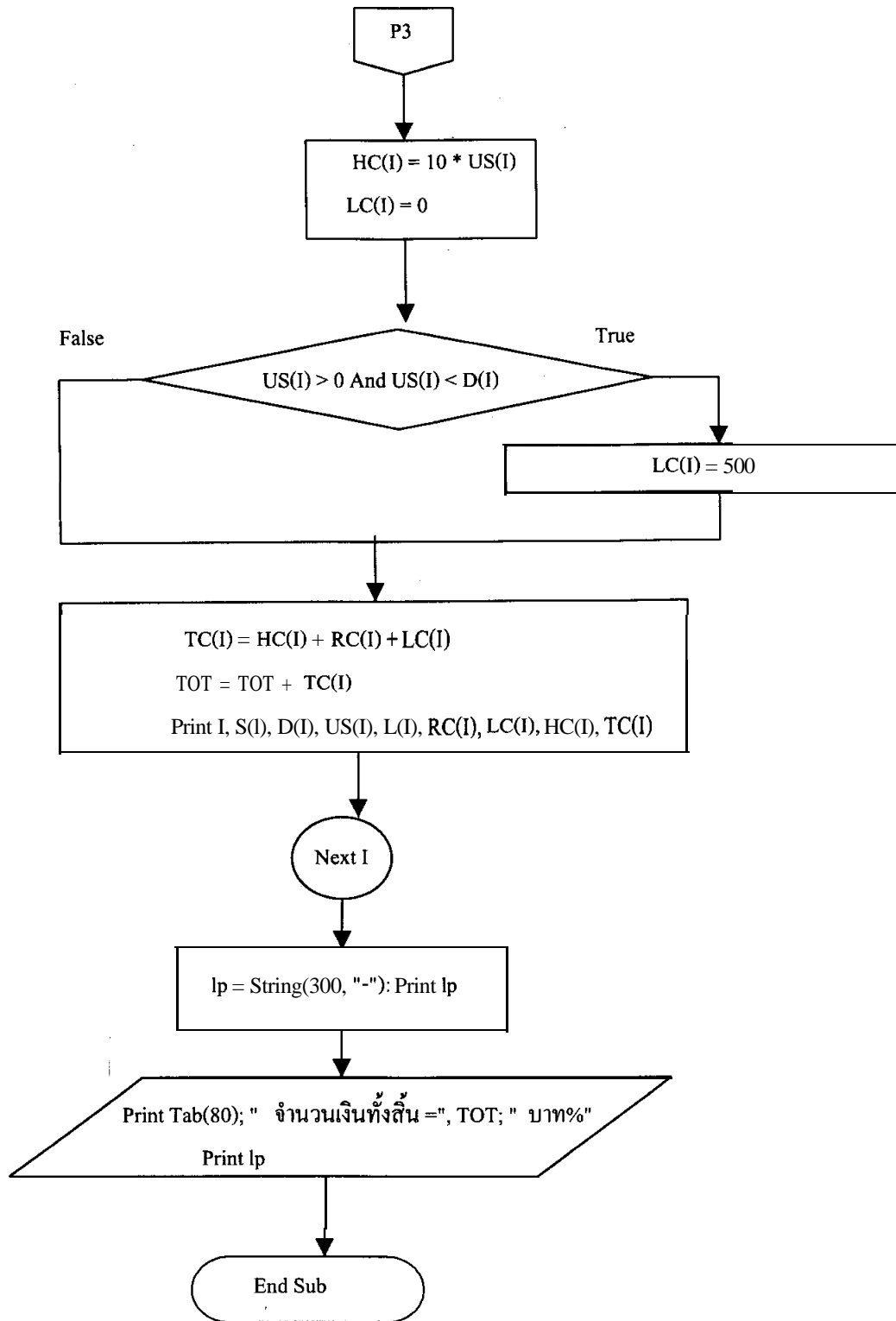
```
I = 1
S(I) = Val(Text1.Text)
D(I) = 0
US(I) = S(I) * D(I)
L(I) = 0
RC(I) = 0
HC(I) = 10 * US(I)
LC(I) = 0
TC(I) = HC(I) + RC(I) * LC(I)
TOT = TOT + TC(I)
TEST = True
K = 0
```

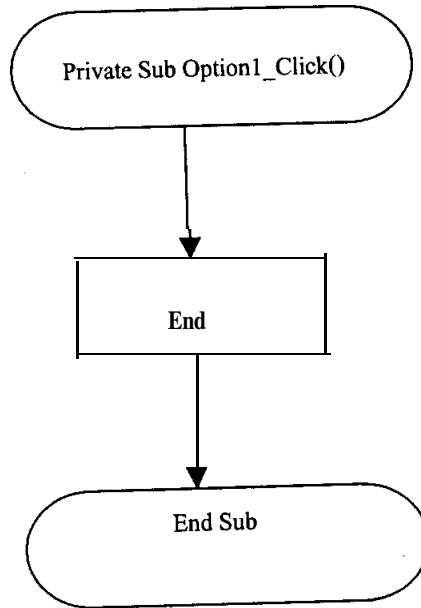












```
Module  
  
Public D(1 To 20) As Integer  
Public L(1 To 20) As Integer
```

## คำถามท้ายบท

1. จงออกแบบระบบงานในการบันทึกข้อมูล (Data Entry) ของเพิ่มข้อมูลของพนักงานในบริษัท โดยที่ข้อมูลของพนักงานแต่ละคนจะประกอบด้วย
  - รหัสประจำตัว
  - ชื่อ นามสกุล
  - บ้านเลขที่ที่อยู่
  - อายุ
  - ตำแหน่ง
  - เงินเดือนปัจจุบัน
2. จงออกแบบระบบเพื่อทำงานในการรับข้อมูลคือ เงินต้น อัตราดอกเบี้ย จำนวนปีที่ฝากเงิน แล้วทำการคำนวณหาเงินรวม
3. จงออกแบบระบบเพื่อรับจำนวนเงินจำนวนหนึ่ง แล้วนำไปแตกเป็น ธนบัตรใบละ 1000 บาท 500 บาท 100บาท 50บาท 20 บาท 10 บาท 5 บาท และเหรียญ 1 บาท เพื่อแสดงผลออกที่จอภาพ
4. จงออกแบบระบบในการทำการคิด โปรแกรมค่าฝึกอบรมในรายการต่อไปนี้ที่จะเลือก
  - MS WORD 800 บาท
  - EXCEL 1200 บาท
  - WINDOW97 1000 บาท
  - ACCESS 1500 บาท
  - AUTOCAD 1800 บาท
  - POWERPOINT 1450 บาท
5. จากระบบประสานต่อไปนี้ให้ นักศึกษาออกแบบโครงสร้างข้อมูล เลือกเหตุการณ์(Event) แล้วนำไปออกแบบผังโปรแกรมของวัตถุแต่ละส่วนที่จะปฏิบัติงาน โดยเหตุการณ์ที่กำหนด การรับข้อมูล ให้รับเป็นตัวเลขเช่น วันที่เกิด 5 เดือนเกิด 1 ปีพ.ศ. 2489  
ผลที่ได้จะแสดงออกมาว่า ชื่อตั้งกล่าวเกิด  
วันที่ 5 มกราคม พ.ศ.2489

