
(: pp. 32~49)

1. 가

(1) (object): (

: ,)

() : , , ()

가 , ()

(2) (form): 가

(3) (control): 가

(4) (property): 가 가 ,

(5) (method): 가

(6) (class):

(7) (event) (event procedure)

1) :

2) : 가

(8) :

(GUI)

2. : (*.frm) -> (compile) -> (*.exe)

(p. 535)

3. (IDE (Integrated Development Environment)): ,

,

(1) ():

(2) (): (main) -> (sub) -> ('...'
)

(3)

(4) :

1) , ,

(5) :

(6) :

(7) : 가

1)

2) , ,

3) 가

4)

(8) :

1) ()-

()- 가

2) (.) 가 (

: (T)-> -> -> ->)

3)

-

-

가 :

-

: underscore(_), underscore

-

(')

, Rem

가

(9) :

(: pp. 60~62; pp. 92~109)

1. (variable): ,

() a=5

b=6

c=a+b

2.

(1)

1) 256

2) , , underscore(_) 가

3) underscore

4) 가 underscore

5) (reserved word) 가 , [] OK!

()

- NumOfPeople, Crow1234, Do_Draw, _BeginEnd, [public] (O.K!)
- Variable*, 4Crow, private, _ (Not Allowed!)

(2)

1)

3. ,

(1) :

1) 가

(2) 'Option Explicit'

$$(3) \quad ,$$

(4) ‘=’

4.

(1) (local variable): 가

1) 'Dim'

2) ‘Static’ – Permanent, 가
가 가

(2) (module variable) (global variable): , ,

()

1) : 'Private' 'Dim'

() Module myModule

Private strMyMsg As String

Sub Initialize()

strMyMsg= “ ”

End Sub

Sub Display()

Debug.WriteLine(strMyMsg)

End Sub

End Module

2) : ‘Public’ ‘Friend’

() Module myModule1

Public strMyMsg As String

End Module

Module myModule2

Sub Initialize()

strMyMsg= “ ”

End Sub

Sub Display()

Debug.WriteLine(strMyMsg)

End Sub

End Module

Module myModule2

5. (data type)

(1) (Byte)

1) : 1 byte

2) : 0~255()

3)

() Dim byCount As Byte

(2) Short , (Integer), Long

1) : 2/4/8 bytes

2) : -32,678~32,677/-2,147,483,648~2,147,483,647/p. 67

3)

() Dim nMyVariable/iMyVariable/lMyVariable As Short/Integer/Long

(3) :

1)Single ()

- : 4 bytes

- : 1.401298E-45 ~ 3.402823E38 (가)

-

() Dim sHeight As Single

-

6

(:

123456750.12 => 1.234568E08)

2)Double ()

- : 8 bytes

- : 4.94065645841247E-324 ~ 1.79769313486232E308(가)

-

() Dim dblVolum As Double

-

14

(4)Decimal :

1) : 12 bytes

2) : p.71

3)

() Dim dMyVariable as Decimal

(5) (Char):

1) : 2 bytes

2) : 0~65535

3)

() Dim cMyVariable as Char

(6) (String)

1) : $10+2*($) , 1 byte

2) 가 , ASCII

3)

```
( ) Dim strMyName As String
```

4)

- '+'

```
( ) Dim strTitle, strName, strNumber As String
```

```
strTitle= Queen
```

```
strName=Elizabeth
```

```
strNumber=II
```

```
Debug.write(strTitle + strName + strNumber)
```

- '&' :

```
( ) Dim iA As Integer
```

```
Dim strB As String
```

iA=5

strB= “men”

Debug.write(iA & strB)

5) (pp. 74~76)

(7) (boolean)

1) : 1 byte

2) : True, False

3)

() Dim bIsFirst As Boolean

(8)Date

1) : 8 bytes

2)

- 1 1 1 ~ 9999 12 31 ()

- 0:00:00 ~23:59:59 ()

3)

() Dim dtToday As Date

4) ‘#’ (: D=#96/11/14/ 11:20:35 PM#)

5) (pp. 79~81)

(9)Object : 가

1) : 4 bytes

2)

() Dim objMyVariable1, objMyVariable2 as Object

objMyVariable1=10

objMyVariable2= “Hello”

6. : , (p. 85)

1)

() Dim iIntegerValue As Integer

Dim byByteValue As Byte

byByteValue=123

iIntegerValue=byByteValue

2) CType :

- : CType(,)

7. : VarType

- : VarType()

() iVar=123 ; strVar= "Hello"

MyType=VarType(iVar) "Integer"

MyType=VarType(strVar) "String"

8. (p. 86~87)

9. :

1) : Option Strict On/Off

2) ,

10. :

1)

[Public| Private] Const [As] =

() Const conPi=3.141592

2) 가

() Const conPi2=conPi*3

11.

(1) : (+, -, *, /), ^(), \ (), Mod(), &()

() MyValue=10 Mod 3 '1

MyValue=11\4 '2

(2) :

- <, <=, >, >=, =, <>(), Like

- 가

- Like

a) : 1 Like 2

b) 1 2 , 'True'

'False'

c)

. *: (: a*a)

. ? : (: a?a)

. # : (: a#a)

. [1- 2] : (: [a-z])

. [! 1- 2]: 가 (: [!a-z])

. [] : (: a[*]a)

d)

. "ABCD" Like "AB*" -> True

. "BA" Like "?A" -> True

. "1A" Like "#A" -> True

. "A" Like "[A-Z]" -> True

. "A" Like "[!A-Z]" -> False

. “a*a” Like “a[*]a” -> True

(3) : Not(), And(), Or(), Xor(), AndAlso(), OrElse() (p. 107)

(4) :

- : =

- : +=, -=, *=, /=, \=, &=

() iCount=iCount+2

iCount+=2

strMyString &= “Hello”

(: pp. 248~268)

1. ?

(=> =>)

2.

1) (Debugging)

2) 가

3.

1)Sub :

2)Function : ,

3) :

4. Sub

(1)

1)

[Public| Protected| Friend| Protected Friend| Private] Sub ([
])

End Sub

(2) Private : Sub 가 가 가

(3) Public : Sub 가 가

(4) Protected : 가

(5) : Sub 가 Sub

1)

[Optional][ByVal][ByRef][ParamArray] [()] [As] [=]

- ByVal : 가

(a)

(b) Sub

(c)

(d)() (p. 261)

Sub Main_Procedure()

Dim n As Integer

n=3

Call A_Procedure(n)

Debug.WriteLine(n)

End Sub

Sub A_Procedure(ByVal n As Integer)

n=n+1

End Sub

- **ByRef** : (Default)

(a) 가 가

(b) Sub

(c) ‘()’ ByVal

(d)() (p. 262)

Sub Main_Procedure()

Dim A,B As Integer

A=3 : B=5

Call Swap(A,B)

Debug.WriteLine(A & “ ” & B)

End Sub

Sub Swap(A, B As Integer)

Dim Temp As Integer

Temp=A: A=B: B=Temp

End Sub

- Optional : 가

(a)

(b) [=] 가

(c)ParamArray

(d)() (p. 266)

A_Procedure(“ ”, “ ”)

A_Procedure(“ ”,)

Sub A_Procedure(X As String, Optional Y As String= “ ”)

Debug.WriteLine(X & Y)

End Sub

- ParamArray :

(a) ()

(b)ByRef, Optional

(c) For Each ~ Next

object

(d) () (p. 267)

Debug.WriteLine(Max(1,2))

Debug.WriteLine(Max(1,2,3,4))

Function Max(ByVal ParamArray Nums()) As Object

Dim i As Object

Max=Nums(0)

For Each i In Nums

If i > Max Then Max=i

Next i

End Function

5. Sub

(1) : 가 Sub

1) Private Sub _ ()

End Sub

(2) : Sub

6. Sub

1) [Call] ([])

7. Function

(1)

1)

[Public| Protected| Friend| Protected Friend| Private] Function ([

)] [As]

- Sub (Default: Public)

- [As]: Function ,

가 (Default: object)

(2)

\

(3)Function

1) [Call] ([])

(4) () (p. 255)

Debug.WriteLine(Sum(5,3))

Debug.WriteLine(5+Sum(2,3))

Function Sum(ByVal A As Integer, ByVal B As Integer) As Integer

Sum=A+B

End Function

(5) : 'System.Math' (p. 256)

8. / /

(1)

1)

[]. (())

2) (p. 260)

Public Module MyModule1

Sub MyProcedure()

MyModule2.MyProcedure()

End Sub

End Module

Public Module MyModule2

Sub MyProcedure()

...

End Sub

End Module

(2) /

1)

 / . (I I)

2) (pp. 259~260)

- **Dim MyClassA As New ClassA**

MyClassA.MyProcedure()

- Dim MyStruc As StructA

MyStruc.MyProcedure()

- : ,
- ():

(: pp. 224~247)

1.IF

(1) IF :

1)

If Then

End If

If Then

2)

If Number >100 Then

Number=100

Debug.WriteLine(Number)

End If

(2)If ... Then ... Else

- 가

1)

If 1 Then

1

ElseIf 2 Then

2

```

Else
    3
End If

- ElseIf
    1 , 1 True 1
    False 3

- Else
    2 , 1 True 1
    2 가 True 2 , False

```

3

End If

- **Elself**

1 ,

1 True

1

False

3

- Else

2 ,

1 True

1

,

2 가 True

2

,

False

2)

If Jumsu >=95 Then

Debug.WriteLine("A+")

ElseIf Jumsu>=90 And Jumsu <95 Then

Debug.WriteLine("A")

```
ElseIf Jumsu>=85 And Jumsu <90 Then
Debug.WriteLine("B+")
ElseIf Jumsu>=80 And Jumsu <85 Then
Debug.WriteLine("B")
ElseIf Jumsu>=75 And Jumsu <80 Then
Debug.WriteLine("C+")
ElseIf Jumsu>=70 And Jumsu <75 Then
Debug.WriteLine("C")
Else
Debug.WriteLine("D")
Endif
```

2.Select Case :

(1)

Select Case

Case 1

1

Case 2

2

...

Case Else

End Select

- Case

1)Case 1 To 4: To

2)Case 1,3,5:

3)Case Is > 10: Is

(2)

Dim Number As Integer

Number=8

Select Case Number

Case 1 To 5

Debug.Print “Between 1 and 5”

Case 6,7,8

Debug.Print “Between 6 and 8”

Case Is >8 And Number <11

Debug.Print “Greater than 8”

Case Else

Debug.Print “Not Between 1 and 10”

End Select

3.Try ... Catch ... Finally :

(1)

Try

Catch

가

Finally

Catch

End Try

() Try

a=b\c

Catch

a=0

Finally

Debug.WriteLine(a)

End Try

(2) Finally

(3) Catch

() Catch When (Catch When Err.number=11)

Catch As Exception (Catch er As Exception)

Catch As Exception When (Catch er As Exception When

Err.number=11)

(4) 'Exit Try' : Try ; Finally

Try

(5) 'Throw' :

() Throw()

4. For ... Next :

(1)

For = To [Step]

Next

- < , :

, <= >

, Next 가

- > , :

$$- \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{y}} - \frac{\partial \mathcal{L}}{\partial \mathbf{z}} \right) / \frac{\partial \mathbf{z}}{\partial \mathbf{w}} + \mathbf{1} \left(\frac{\partial \mathcal{L}}{\partial \mathbf{w}} \right)$$

Debug.WriteLine(Count)

For b=1 To 10

If a=b Then

Debug.WriteLine(a*b)

Exit For

End if

Next b

Debug.WriteLine(“ ”)

Next a

5.For Each ... Next

-

(1)

For Each In

Next

- **object**

- **collection**

(2)

Dim strData(1 To 3) As String

Dim vData As Object

Dim I As Integer

For I=1 To 3

StrData(I)= “ ” + CStr(I)

Next I

For Each vData In strData

Debug.WriteLine(vData)

Next vData

6.Do

-

(1)Do While ... Loop (While ... End While)

-

True

False

1)

Do While

Loop

2)

Count=1

Do While Count <= 10

Debug.WriteLine(Count)

Count=Count+1

Loop

(2)Do Until ... Loop

-

True 가

True 가

1)

Do Until

Loop

2)

Count=1

Do Until Count > 10

Debug.WriteLine(Count)

Count=Count+1

Loop

(3)Do ... Loop While

-

While

, True

False 가

1)

Do

Loop While

2)

Count=1

Do

Print Count

Count=Count+1

Loop While Count <= 10

(4)Do ... Loop Until

-

Until

, True 가

True 가

1)

Do

Loop Until

2)

Count=1

Do

Debug.WriteLine(Count)

Count=Count+1

Loop Until Count > 10

(5)Exit Do

- Do

1)

Dim Check, Count

Check=True: Count=0

Do

Do While Count < 20

Count=Count + 1

If Count = 10 Then

Check=False

Exit Do

End If

Loop

Loop Until Check=False

7.Goto

-

1)

Goto

...

:

-

가 ,

8.On Error GoTO

- 가
(Resume Next 가)

1)

On Error GoTo

•
•
()
•
•
:
()

Resume Next

-

- : ,

(: pp. 269~281)

1. ()

(1)

: Dim () As

- :

- :

(2)

-

- ()

(a) **Dim ave(3) As integer**

ave(0)=10 : ave(1)=20 : ave(2)=30

Dim ave() As Integer = {10,20,30}

(b) **Dim student(2, 2) As integer**

ave(0,0)=10 : ave(0,1)=20 : ave(1,0)=30 : ave(1,1)=40

Dim student(,) As Integer={{10,20},{30,40}}

2.

-

-

-

가 가

(1)

- : Dim () As

(2) Redim

-

-

- ()Dim mat() As Integer

Sub A_Procedure()

...

Redim mat(3,4)

...

End Sub

- Redim

- **Preserve**

() **Redim Preserve** mat(3,5)

()

() **Ubound** –

: Ubound([,])

: Ubound(mat,1), Ubound(mat,2)

3.

- ,

(1)

: Erase

(: pp. 112~221)

1.

(1) : Visual Basic ,

(2) 가 :

(3) X : Microsoft

2.

(1) (Label)

-

- Text

1) 가

- **Text** : , ‘&’

- **BorderStyle** :

- **AutoSize** :

- **UseMnemonic** : ‘&’

2)

- 가

(2) (LinkLabel)

-

1) 가

- **LinkArea** : 가

- **LinkColor** :

1) 가

- Locked :
 - Text :
 - Multiline : ,
Ctrl+Return
- ```
() Sub Form_Load()
 Text1.Text= “ .” & vbCrLf & _
 “ .”
End Sub
```
- ScrollBars : , None/Horizontal/Vertical/Both
  - SelectionStart :
  - SelectionLength :
  - WordWrap : 가

3.

(1) (Button)

1) 가

- **AcceptButton** : 가 **Enter**

- **CancelButton** : 가 **Esc**

- **BackgroundImage** /Image :

- **ImageAlign** : , / / / /

2)

-

- space bar enter

-

## (2) (CheckBox)

-

### 1) 가

- **Checked** :
- **CheckState** : **Checked/UnChecked/Indeterminate( )**
- **ThreeState** : **Indeterminate**
- **AutoCheck** : 'V'

## (3) (RadioButton)

-

### 1) 가

- **Checked** :
- **Enabled** :

**(4) (GroupBox)**

- 
- 

**(5) (Panel)**

- 

**(6) (ListBox) (ComboBox)**

- 

**1)**

- 가 (            );
- 가 (            )
- :       ,           (            가   ),           (            가   )

## 2) 가

- **Items.Add(item) :** 가
- **Items.Insert(index,item):**
- **Items.Remove(item)/Items.RemoveAt(index):**
- **Items.Clear():**

## 3) 가

- **Sorted :**
- **SelectedItem :**
- **SelectedIndex :** index ,
- **Items.Count :**
- **MultiColumn :** , True(가 )/False( )

- **SelectionMode** : ,  
None/One/MultiSimple/MultiExtended

**(7) (CheckedListBox)**

- 1) 가  
- **Items()** :  
2) 가  
- **GetItemChecked(index):** 가  
- **SetItemChecked(index, True/False):** /

**(8) (DomainUpDown)**

- /

**(9) (NumericUpDown)**

-

**(10) (Scroll Bar)**

**1) 가**

- **Change** : 가
- **Scroll** :

**2) 가**

- **Value** : 가
- **Min, Max** : Value 가 ( )
- **SmallChange** :
- **LargeChange** :

**(11) (Timer)**

-

**(12) (PictureBox)**

-

, bmp,ico,gif,jpg

- Image           FromFile

(    ) Image.FromFile(“C:\MyCar.bmp”)

#### 4.    가

##### (1)           (ProgressBar)

-

##### 1)    가

- Value       :            가

- Min(Max)     :            (    )    가

##### (2)           (ToolBar)

-

##### 1)    가

- **ImageList** : ,  
**'Image** ,
- **Appearance** : (Flat/Normal)
- **Buttons** : 가 , **'ToolBarButton** ,
- Style** : (PushButton/ToggleButton/Seperator/  
**DropDownButton)**
- ImageIndex** :
- Text** :
- **ButtonClick** : 'e' **Button**  
, (Text ),  
**(ImageIndex )**

(3) **(StatusBar)**

-

### 1) 가

- **ShowPanels** : , 가
- **Text** : , ShowPanels False 가
- **Panels** : 가 , 'StatusBarPanel',  
Item : 가

### (4) (TabControl)

-

### 1) 가

- **TabPage** : , 'TabPage',
- **Alignment** : (Top/Bottom/Left/Right)

(5) (ListView)

1) `getLargeIcon()` : `LargeIcon` , `SmallIcon` , `List` , `Details`

2) `getView()` :

`View` :

`Items` : `convertView` , `'ListViewItem'` ,

`Columns` : `LinearLayout` , `LinearLayout` (Left/Center/Right)

`ColumnHeader` , `'ColumnHeader'` ,

`SmallImageList` : `ImageList`

(`SmallIcon/List/Details` )

`LargeImageList` : `ImageList`

(`LargeIcon` )

3) `AfterLabelEdit` :

, 'e' Item  
 (Integer ) Label (String )  
 (6) (TreeView) : 가  
 1) 가  
 Nodes : , 'TreeNode' , 가  
 TopNode : 가  
 SelectedNode : 가  
 ● FirstNode( ), LastNode( ), NextNode( ), PrevNode( )  
 ● Parent( )  
 2) AfterSelect : , 'e'  
 Node Index , Text  
 (7) (NotifyIcon):

1) 가

**Icon** : **Icon**

**Text** :

(8) (Tooltip)

1) 가

**Active** :

**InitialDelay** / **Reshowdelay** / **AutoPopDelay**

2) **SetToolTip** :

(9) (TrackBar) :

1) 가

**TickFrequency** :

**TickStyle** : (None/TopLeft/BottomRight/Both)

(10) (DateTimePicker)

1) 가

**Format** : (Long/Short/Custom)

**CustomFormat**

● :

**MyDateTimePicker.Format=Custom**

**DateTimePicker.CustomFormat= “yyyy MMM dd ddd”**

**DateTimePicker.CustomFormat= “ ‘ : yy MM d – HH:mm:s ddd”**

5. X

(1) (MaskedTextBox) :

1) 가

**Mask** : ‘#’ ,

**PromptInclude : PromptChar**

**CtlText :**

**(2)MS (MSChart) : 2 3 , ,**

**1) 가**

**Title.Text :**

**Plot.Axis(MSChart20Lib.VtChAxizId.VtChAxisIdX).AxisTitle.Text : X**

**Plot.Axis(MSChart20Lib.VtChAxizId.VtChAxisIdY).AxisTitle.Text : Y**

**Legend.Location.Visible :**

**Legend.Location.LocationType :**

가

**ChartData** :

---

( : pp. 284~316)

1.

(1) :

**Class**

/ / /

.....

**End Class**

( ) **Class CCar**

**End Class**

(2) :

( (instance) )

**Dim                    As New                    (                    )**

**Dim                    As                    =New                    (                    )**

**(                    ) Dim My\_Car As New CCar()**

2.                    :                    ,

(1)

(2)‘                    .                    ’

(3)

**Class CCar**

**Dim m\_x, m\_y As Integer**

**End Class**

**Dim My\_Car As New CCar()**

**My\_Car.m\_x=100**

**Dim iCurX As Integer=My\_Car.m\_x**

**3.**

**(1)**

**Property            () As**

**Set (ByVal value As            )**

**End Set**

**Get**

**End Get**

**End Property**

**(2)**

**,**

**가**

**(3)**

**Class CCar**

### Private m\_x As Integer

## Property CarX() As Integer

## Set (ByVal x As Integer)

$$\mathbf{m}_x = \mathbf{x}$$

## End Set

## Get

**Return** `m_x`

## End Get

## End Property

## End Class

## Dim My\_Car As New CCar()

|                                |       |     |
|--------------------------------|-------|-----|
| My_Car.CarX=100 ‘Set...End Set | , m_x | 100 |
|--------------------------------|-------|-----|

```
Dim iCurX As Integer=My_Car.CarX 'Get...End Get , iCurX
```

(4)ReadOnly WriteOnly : (Get...End Get ) ,  
(Set...End Set )

## ReadOnly(      WriteOnly)

(5)Default : ,  
가 ,

## Default

**1)**

## Class CCar

### Private m\_x(2) As Integer

## Default Property CarX(ByVal index As Integer) As Integer

### Set (ByVal value As Integer)

**m\_x(index)=value**

## End Set

**Get**

**Return m\_x(Index)**

**End Get**

**End Property**

**End Class**

**Dim My\_Car As New CCar()**

**My\_Car(0)=100: My\_Car(1)=150**

**iCurX1=My\_Car(0): iCurX2=My\_Car(1)**

4. : 가 가 , Sub Function

(1)

**Class CCar**

**Private m\_x As Integer**

**Sub SetX(ByVal x As Integer)**

**m\_x=x**

**End Sub**

**Fuction GetX() As Integer**

**Return m\_x**

**End Function**

**End Class**

**Dim My\_Car As New CCar()**

**Dim iCurX As Integer**

**My\_Car.SetX(5)**

**iCurX=My\_Car.GetX()**

5. : 가

(1) : 'WithEvents'

1)

**Private WithEvents My\_Car As New CCar**

(2)

1) : Event ( )

**: Public Event Event\_Draw(ByVal iPos As Integer)**

2) : RaiseEvent ( )

**: Class CCar**

**Public Event Event\_Draw(ByVal iPos As Integer)**

**Sub Draw(ByVal iPos As Integer)**

**RaiseEvent Event\_Draw(0)**

**End Sub**

**End Class**

```
(3) : Sub 'Handles'
 ; 'AddHandler'
```

1) ('Handles' )

```
Private Sub OnDraw(ByVal iPos As Integer) Handles My_Car.Event_Draw
Debug.WriteLine(iPos)
End Sub
```

## 2) ('AddHandler' )

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs)
 Handles MyBase.Load
 AddHandler My_Car.Event_Draw, AddressOf OnDraw
End Sub
```

### Private Sub OnDraw(ByVal iPos As Integer)

## Debug.WriteLine(iPos)

**End Sub**

6. ( ) ( )

(1) (constructor): 가 , (overloading)

가

1)

## Class

**Sub New(                    )**

**End Sub**

## End Class

2)

## Class CCar

## Sub New()

```
Debug.WriteLine(“ 가 .”)
```

```
End Sub
```

```
Sub New(ByVal iNum As Integer)
```

```
Debug.WriteLine(“ ”& iNum & “ 가 .”)
```

```
End Sub
```

```
End Class
```

```
Dim My_Car As New CCar()
```

```
Dim My_Car As New CCar(1)
```

**(2) (destructor): 가**

**1)**

```
Class
```

```
Protected Override Sub Finalize()
```

```
End Sub
```

## End Class

2) **‘Nothing’** , 가 가  
( ) **‘Finalize’** 가 가 ,  
**‘System.GC.Collect()’** 가

---

( : pp. 317~354)

1. : 가 가 가

가

2. : (sub class) ‘Inherits’

(base class)

(1)

**Class MyParentClass**

**Public iMyParentValue As Integer**

**End Class**

**Class MySubClass**

**Inherits MyParentClass**

**Public iMySubValue As Integer**

**End Class**

**Dim obj As New MySubClass()**

**obj.iMyParentClass=5**

**obj.iMySubClass=10**

(2) **MustInherit** : **‘MustInherit’**

1)

**MustInherit Class MyParentClass**

**End Class**

3. **(overriding):**

가 ;

**‘overridable’** , **‘overrides’**

**(1)**

**Class MyParentClass**

**Public Overridable Sub Hello()**

**End Sub**

**End Class**

**Class MySubClass**

**Inherits MyParentClass**

**Public Overrides Sub Hello()**

**End Sub**

**End Class**

**Dim obj As New MySubClass()**

**Dim obj2 As New MyParentClass()**

**obj.Hello() ‘Sub Class                   가**

obj2.Hello() 'Base Class 가

(2) : p. 324 ; 'New'

가

(3)  $\vdots$

;**‘Shadows’**

**1)**

## Class MyParentClass

## Public Sub Hello()

**End Sub**

## End Class

## Class MySubClass

## Inherits MyParentClass

## Public shadows Sub Hello()

**End Sub**

## End Class

**Dim obj As MySubClass = New MySubClass()**

## obj.Hello() 'Sub Class 가

2) : p. 330 ; 'As'

가

(4) 가 : ‘MustOverride’

**4.**

(1) : (가)

(2) 가 : 가

**‘MyBase’**

**1)**

**Class MyParentClass**

**Public Sub New()**

**End Sub**

**Public Sub New(ByVal strName As String)**

**End Sub**

**End Class**

**Class MySubClass**

**Inherits MyParentClass**

**Public Sub New(ByVal strName As String)**

**{MyBase.New(strName)}**

**End Sub**

**End Class**

**Dim obj As MySubClass = New MySubClass("Hi.") ‘**

**가**

**, { ..... } 가**

**가**

**5.**

**(1)**

**:**

**,**

**(2)**

**Class MyParentClass**

**Public Event MyParentEvent()**

**Protected Sub DoEvent() ‘**

**RaiseEvent MyParentEvent()**

**End Sub**

**End Class**

**Class MySubClass**

**Inherits MyParentClass**

**Public Sub Hello()**

**DoEvent()**

**End Sub**

**End Class**

**Public Class Form1**

**Private Sub OnMyEvent()**

**End Sub**

**Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)**

**Handles Button1.Click**

```
Dim obj As MySubClass=New MySubClass()
AddHandler obj.MyParentEvent, AddressOf OnMyEvent
obj.Hello()
End Sub
End Class
```

## **6. Me/MyBase/MyClass**

**(1) Me :**

**1)**

```
Class CCar
Private strName As String
Sub Hello()
Dim strName As String
```

**strName= 'A'**

**Me.strName= 'B'**

**End Sub**

**End Class**

**(2) MyBase :**

**1)**

**Class MyParentClass**

**Public Overridable Sub Hello()**

**End Sub**

**End Class**

**Class MySubClass**

**Inherits MyParentClass**

**Public Overrides Sub Hello()**

**MyBase.Hello()**

**End Sub**

**End Class**

**Dim obj As MySubclass=New MySubClass()**

**obj.Hello() ‘ Hello 가**

**(3) Myclass :**

**1)**

**Class MyParentClass**

**Public Sub DoSomething()**

**Hello() {MyClass.Hello()}**

**End Sub**

**Public Overridable Sub Hello()**

**End Sub**

## End Class

## Class MySubClass

## Inherits MyParentClass

## Public Overrides Sub Hello()

**End Sub**

## End Class

## Dim obj As MySubclass=New MySubClass()

`obj.DoSomething() 'Hello`      `가`      `, { ..... }`

Hello 가

---

( : pp. 355~398)

1. 가? 'is-a' -
2. ? 'is-a' 'act-as' 'A B (A act-as B)'

3.

(1)

**Interface**

**End Interface**

(2) , , , 가

(3) ,

(4)

**Public Interface IObject**

**Function GetMyValue() As Integer**

**End Interface**

**Class MyParentClass**

**Implements IObject**

**Private MyValue As Integer=5**

**Public Function GetMyValue() As Integer Implements IObject.GetMyValue**

**Return MyValue**

**End Fuction**

**End Class 'MyParentClass act-as IObject**

**4.                ?**

**(1)            (early)/            (late)**

1) : 가

:

2)

```
Public Class CEarlyBindClass
```

```
Public Sub hello
```

```
End Sub
```

```
End Class
```

```
Dim obj As New CEarlyBindClass
```

```
obj.hello() 'hello() 가
```

```
Option Strict Off
```

```
Public Class CLateBindClass
```

**Public Sub hello**

**End Sub**

**End Class**

**Public Sub DoIt(ByVal obj As Object)**

**obj.hello() ‘**

**End Sub**

**(Option Strict On )**

**Public Sub DoIt(ByVal obj As Object)**

**Dim localObj As CLateBindClass**

**localObj=CType(obj,CLateBindClass) ‘obj**

**CLateBindClass**

**localObj.hello() ‘hello() 가**

**End Sub**

(2) 가

1)

2)

3)

5. ?

(1) Private: 가

(2) Protected: 가

(3) Friend: 가

(4) Public: 가

6. ? 가 ,

, 'shared' ,

(1) , , ,

7. : , 'Delegate'

(1)

```
Public Class Form1
```

```
Public Delegate Sub MyDelegate(ByVal message As String)
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
```

```
Dim NewHello As New MyDelegate(AddressOf CMyClass.Hello)
```

```
NewHello(“ ”)
```

```
End Sub
```

```
End Class
```

```
Class CMyClass
```

```
Public Shared Sub Hello(ByVal message As String)
```

**End sub**

## End Class

**8.**

(1)  $\vdash$  , , , ,

**1)**

## Structure

## End Structure

2)

## Structure Person

## Public name As String

### Public score As Integer

## End Structure

(2) : 가

1)

**Enum** [As ]

1 [= ]

2 [= ]

...

**End Enum**

---

( : pp.400~411)

1. : 가

2. :

(1) 가

(2)

(3) Text (-)

(4) ( )

, Alt

(5) ( ) , Ctrl ,

(6) 5 가

(7) Gray Enabled False

(8) 가 Checked

True

(9) RadioCheck

True

(10) (...) 가 가

가

3. : 가 가

(1)

**(2)**

**ContextMenu**

**(3)**

---

( : pp. 412~442)

1. :

2.

(1)

1) ,

2)

**MyForm.ShowDialog()**

( : ShowDialog(Hide)

**Visual**

**True(False)**

( ) )

(2)

1)

가

2)

**MyForm.Show()**

3.

**(1)InputBox, MsgBox**

(2)

(3)

**4. InputBox, MsgBox**

:

**(1)InputBox**

1) : InputBox ( l, ll, l )

- < >, < >

2)

**FileName=InputBox(“ ”, “ ”)**

**FileName=InputBox(“ ”, “ ”, “\*.txt”)**

## **(2)MsgBox**

**1) : MsgBox ( [, ] [, ] )**

**(p. 414**

**)**

**2)**

**Msg = “ 가 가 .”**

**Style = MsgBoxStyle.OKOnly**

**Title = “ ”**

**Response = MsgBox(Msg, Style, Title)**

**5.**

**(1)**

1) **Windows** 가

2)

(2) 가

1)Text :

2)FromBorderStyle : (FixedSingle)

3)MaxButton : (False)

4)MinButton : (False)

5)ControlBox : (False)

(3)

1) :

**Dim MyForm As frmMain = New frmMain()**

2) : Close

(4) : ,

6.

(1) ( ) : , , ,  
 , OpenFileDialog(SaveFileDialog) ShowDialog

1) 가 (p. 431 )

2)

cdgOpen.Filter = “ (\*.\*) | \*.\* | (\*.txt) | \*.txt”

(2) : , ColorDialog ShowDialog

(3) : , , , , FontDialog  
ShowDialog

- (4) : , **FolderBrowserDialog**  
**ShowDialog**
- (5) : , **PrintDialog** **ShowDialog**  
, **PrintPreviewDialog, PageSetupDialog, PrintDocument**

---

( : pp. 508~530)

1. : / , ANSI (American National Standards Institute)

(1) :

1) : FileOpen ( , , [ , ] [ , ] )  
: , FreeFile

: OpenMode.Input/Output/Append

: , OpenAccess.Read/Write/ReadWrite

: ,

**OpenShare.LockRead/LockReadWrite/Shared**

2)

**Dim iFileNum As Integer**

**iFileNum=FreeFile()**

**FileOpen(iFileNum,“MyFile.Fil”,OpenMode.Output)**

(2) :

1) : FileClose( )

(3) : OpenMode.Input

1) : Input( , )

“ ”

“ ”

2)

“ ”, “1”, “ ” ‘

**Input(iFileNum,strName): Input(iFileNum,strNumber): Input(iFileNum,strDept)**

**(4) : OpenMode.Output OpenMode.Append**

**1) : WriteLine( , 1, 2, ... )**

**“ ”**

**(,)**

**2)**

**Dim strString As String, iNumber As Integer**

**strString= “Hello”**

**iNum=12345**

**WriteLine(iFileNum,strString,iNum)**

**2. : ( ) 가**

**(1)**

1)           : FileOpen(           ,           ,           l,           ll,           ll,           l)  
                   : OpenMode.Random  
                   :

2)

**Public Structure SPerson**

**<VBFixedString(10)> Public strName As String ‘**

**<VBFixedString(10)> Public strNum As String**

**<VBFixedString(10)> Public strDept As String**

**End Structure**

**Private m\_iFileNum As Integer**

**Private Person As SPerson**

**iFileNum=FreeFile()**

**FileOpen(iFileNum, ‘MyFile.Fil’,OpenMode.Random, , ,Len(Person))**

(2)

1) : FileGet( , [, ])

2)

**FileGet(iFileNum,Employee,iPosition)**

(3)

1) : FilePut( , [, ])

2)

**LastRecord=LastRecord+1**

**FilePut(iFileNum,Emplotee,LastRecord)**

3. : 가 ,  
가  
 ,

(1)

1) : **FileOpen**( , , [ , ], )  
: **OpenMode.Binary** \

(2)

,