

Chapter 04

변수와 연산자



목차

1. 변수와 데이터형
2. 연산자

1. 변수와 데이터형

■ 변수의 개념

■ 변수

- 프로그램이 실행되는 동안 사용자가 입력이나 계산한 출력값 등을 저장하는 기억 공간

■ 변수 선언

- 프로그래머가 프로그램을 작성하다가 기억 공간이 필요하면 변수의 이름과 크기를 컴퓨터에 알림
- 컴퓨터는 요청받은 크기만큼 기억 공간을 확보하고 시작 주소를 프로그램에 알림
- 프로그램은 할당된 시작 주소와 선언된 변수명을 매칭하여 사용



그림 4-1 변수의 개념

1. 변수와 데이터형

■ 변수의 선언

- 변수의 크기를 선언할 때는 직접 몇 바이트인지 입력하지 않고 해당 데이터형을 입력
- 변수 선언 형식

형식 | Dim 변수명 As 데이터형(크기)

예 | Dim score As integer
Dim name As string

1. 변수와 데이터형

■ 변수의 선언

■ 변수명 작성 규칙

- 영문자, 한글, 숫자, 밑줄(_)을 조합하여 사용할 수 있으며 다른 특수 기호 X
- 첫 글자는 반드시 영문자나 한글(기호나 숫자 X)
- 첫 글자가 예약어(Sub, Private, End 등) X
- 255자 초과 불가
- 대·소문자 구분 없음

표 4-1 변수 선언의 예

올바른 변수 선언	잘못된 변수 선언
<pre>Dim student_name As String Dim score As Byte Dim 학과 as String Dim kor, mat, eng As Integer</pre>	<pre>Dim name\$ As String Dim 123_address As String Dim Private As Integer</pre> <p>→ 불가한 특수 기호(\$) 사용</p> <p>→ 숫자로 시작</p> <p>→ 예약어 사용</p>

1. 변수와 데이터형

■ 데이터형

- 변수의 데이터형은 변수가 몇 바이트의 기억 공간을 차지해야 하는지, 어떤 형태로 저장되는지 결정

표 4-2 비주얼 베이직의 데이터형

데이터형	크기 (바이트)	표현 범위와 설명
정수	Byte	0~255(부호 없는 정수)
	UShort	0~65,535(부호 없는 정수)
	UInteger	0~4,294,967,295(부호 없는 정수)
	ULong	0~18,446,744,073,709,551,615(부호 없는 정수)
	SByte	-128~127
	Short	-32,768~32,767
	Integer	-2,147,483,648~2,147,483,647
	Long	-9,223,372,036,854,775,808~9,223,372,036,854,775,807
정수/ 실수	Decimal	0~+/-79,228,162,514,264,337,593,543,950,335(정수) 0~+/-7.9228162514264337593543950335E28(실수)

1. 변수와 데이터형

데이터형		크기 (바이트)	표현 범위와 설명
실수	Single	4	음수 : $-3.4028235E38 \sim -1.401298E-45$ 양수 : $1.401298E-45 \sim 3.4028235E38$
	Double	8	음수 : $-1.79769313486232E308 \sim -4.94065645841247E-324$ 양수 : $1.79769313486232E308 \sim 4.94065645841247E-324$
문자	Char	2	한 유니코드 문자 표현(0~65535의 매칭되는 문자 표현 가능)
	String	4	문자열 표현(약 20억 개의 유니코드 문자)
기타	Boolean	2	True 또는 False 표현, 기본 값은 False
	Date	8	1년 1월 1일 0:00:00~9999년 12월 31일 11:59:59 PM
	Object	4	모든 형식 저장 가능

1. 변수와 데이터형

■ 변수의 동시 선언

① `Dim num1, num2, num3 As Integer`

→ num1, num2, num3 모두 Integer 형이다.

② `Dim kor, eng As Integer, ave as Single`

→ kor, eng는 Integer 형이고 ave는 Single 형이다.

■ 데이터형 미지정

- 기본값으로 Object 형이 할당
- Object 형은 모든 데이터형을 나타낼 수 있어 편리하지만 처리 속도가 느리기 때문에 알맞은 데이터형을 선언하는 것이 더 효율적

③ `Dim num1, num2, num3`

→ num1, num2, num3는 기본형인 Object 형이 자동 할당된다.

1. 변수와 데이터형

■ 변수에 값 대입

- 연산자 = 오른쪽에 있는 값을 왼쪽 변수명에 대입(저장)

형식 | 변수명 = 값 or 다른 변수명 or 컨트롤의 속성값

- 변수 선언을 할 때 다음과 같이 바로 값을 대입 가능

```
Dim score As Integer = 100
```

```
Dim name As string = "VisualBasic"
```

```
Dim isSelected As Boolean = (10>5)
```

표 4-3 변수에 값 대입의 예

변수의 종류	대입 연산자 적용의 예	변수의 종류	대입 연산자 적용의 예
숫자인 경우	price = 1000 amount = 47 total = price*amount	기타	isSelected = True isSelected = (10>5)
문자인 경우	firstname = "a" name = "VisualBasic" secondname = firstname age = TextBox1.text		

1. 변수와 데이터형

■ 명시적 선언과 묵시적 선언

- 명시적 선언 - 변수를 선언하고 사용
- 묵시적 선언 - 변수를 선언하지 않고 사용
- 묵시적 선언의 경우, 첫 번째로 저장되는 값에 따라 데이터형이 자동으로 결정
- 변수를 선언하지 않으면 실수로 오류가 발생할 가능성이 크기 때문에 명시적으로 선언하는 것이 안전

표 4-4 명시적 선언과 묵시적 선언

명시적 선언	묵시적 선언
<code>Dim score As integer</code> <code>score = 100</code>	<code>score = 100</code>

1. 변수와 데이터형

■ 명시적 선언과 묵시적 선언

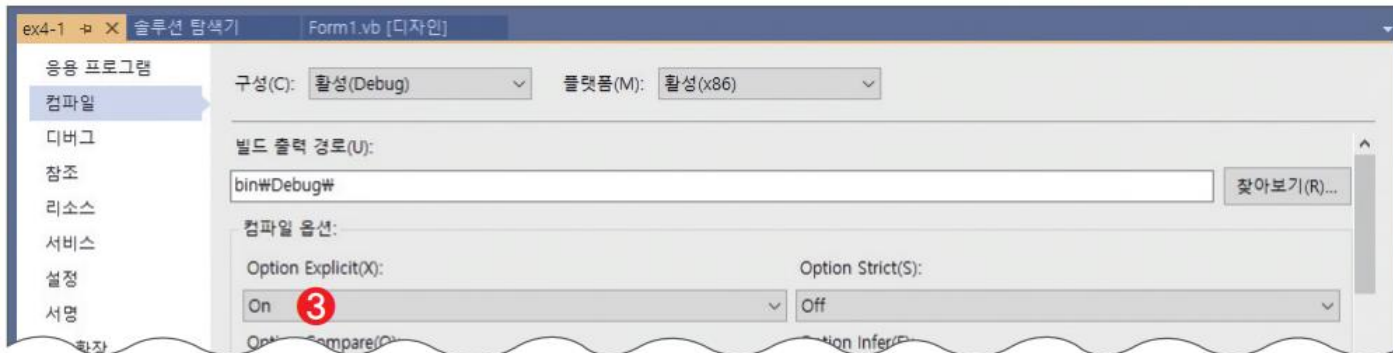
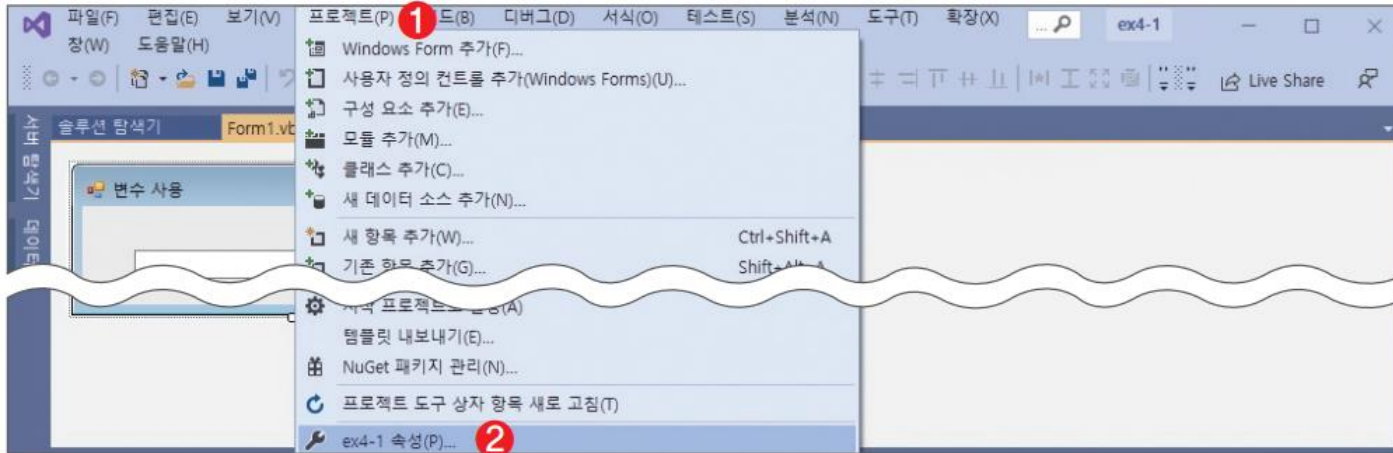


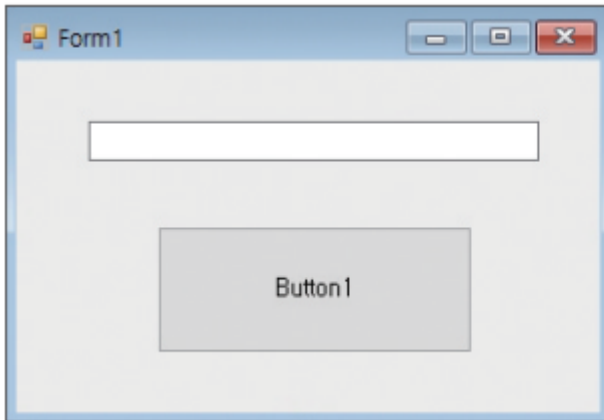
그림 4-2 강제로 명시적 선언을 사용하도록 설정하는 방법

1. 변수와 데이터형

실습하기 4-1 변수 선언하고 사용하기

- Score라는 변수에 90점을 저장 후 <출력>을 클릭하면 텍스트박스에 출력하는 프로그램

1) 윈도우 폼 디자인, 속성 설정



컨트롤	속성	속성값
Form1	Text	변수 사용
TextBox1	Text	(빈칸)
Button1	Text	출력

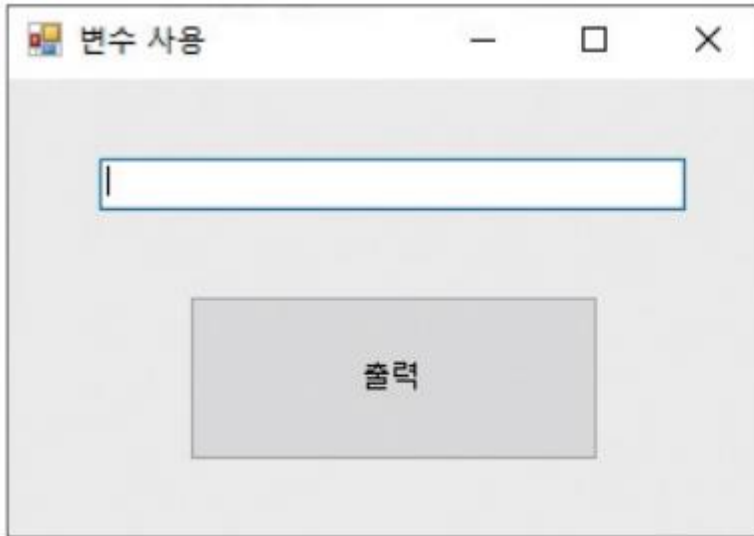
2) <출력>을 더블클릭하여 코드 작성

```
01 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.  
    Click  
02     Dim Score As Integer  
03     Score = 90  
04     TextBox1.Text = Score  
05 End Sub
```

1. 변수와 데이터형

실습하기 4-1 변수 선언하고 사용하기

3) 실행 결과 확인

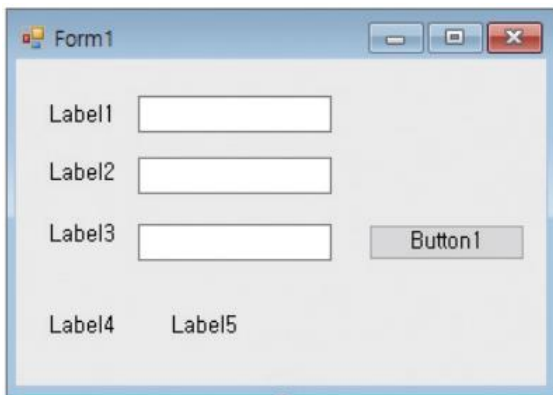


1. 변수와 데이터형

실습하기 4-2 성적 합계 구하기(명시적 선언)

- 국어, 영어, 수학 과목의 점수를 입력받아 총점을 구하고 출력하는 프로그램 작성

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값	컨트롤	속성	속성값
Form1	Text	성적 계산하기	TextBox2	Text	(빈칸)
	StartPosition	CenterScreen	TextBox3	Text	(빈칸)
Label1	Text	국어	Label4	Text	총점 :
Label2	Text	영어	Label5	Text	0
Label3	Text	수학	Button1	Text	총점 구하기
TextBox1	Text	(빈칸)			

1. 변수와 데이터형

실습하기 4-2 성적 합계 구하기(명시적 선언)

3) <총점 구하기>를 더블클릭하여 코드 작성

```
01 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
02     Dim Kor, Eng, Mat, Total As Integer
03     Kor = TextBox1.Text
04     Eng = TextBox2.Text
05     Mat = TextBox3.Text
06     Total = Kor + Eng + Mat
07     Label5.Text = Total
08 End Sub
```

4) 실행 결과 확인

성적 계산하기

국어 100

영어 100

수학 100

총점 구하기

총점 : 300

1. 변수와 데이터형

실습하기 4-3 성적 합계 구하기(묵시적 선언)

1) 폼 디자인과 컨트롤 속성 설정은 [실습하기 4-2]와 동일

2) <총점 구하기>를 더블클릭하여 코드 작성

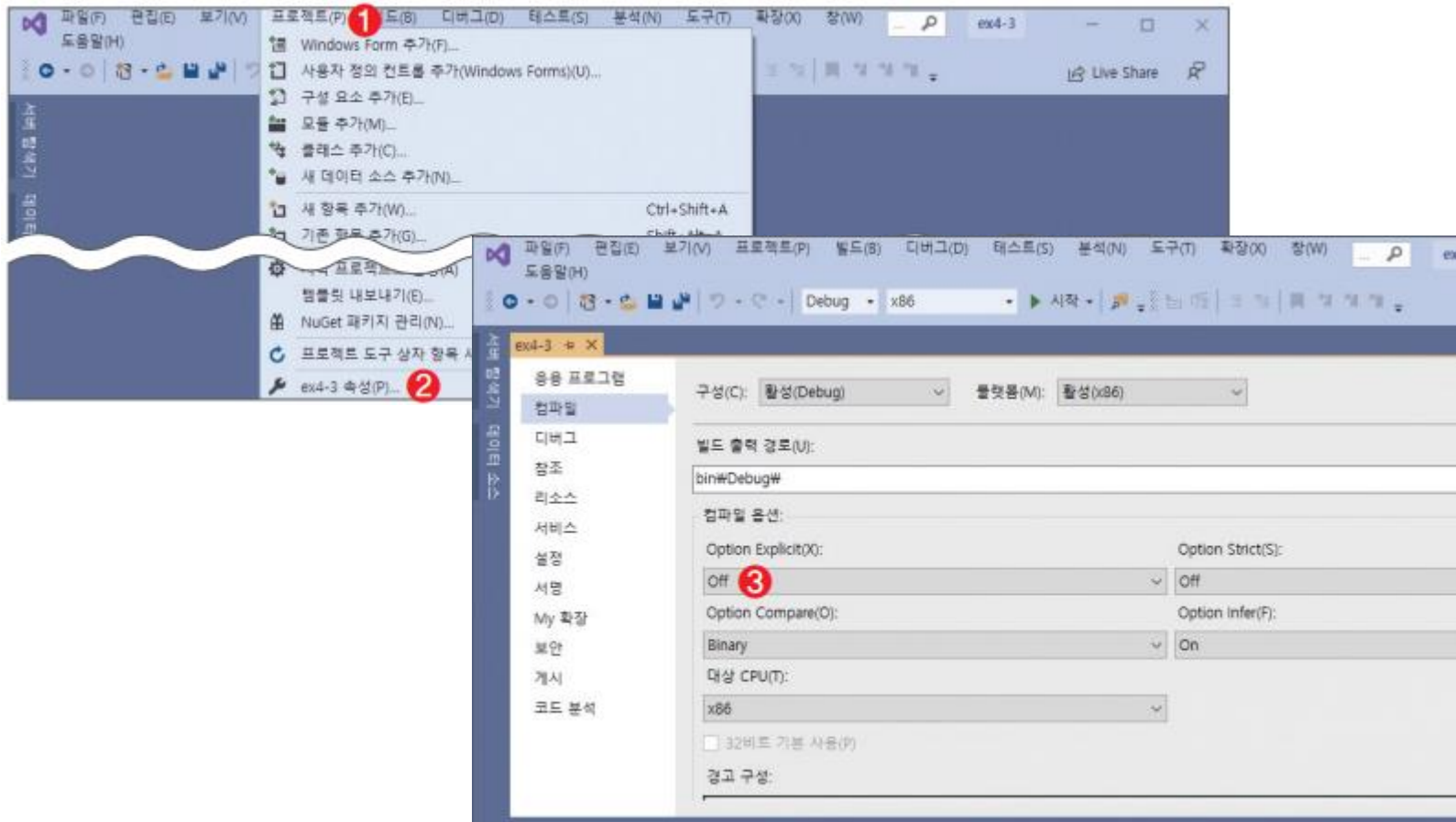
```
01 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
02     Kor = TextBox1.Text
03     Eng = TextBox2.Text
04     Mat = TextBox3.Text
05     Total = Kor + Eng + Mat
06     Label5.Text = Total
07 End Sub
```


1. 변수와 데이터형

실습하기 4-3 성적 합계 구하기(묵시적 선언)

3) 묵시적 선언도 허용하도록 변경

- [프로젝트]-[ex4-3 속성]-[컴파일]을 선택하여 'Option Explicit'를 'Off'로 설정



1. 변수와 데이터형

실습하기 4-3 성적 합계 구하기(묵시적 선언)

4) 실행 결과 확인

- 확인 후 다시 Option Explicit'를 'On'으로 설정해 놓는다.

성적 계산하기

국어 100

영어 100

수학 100

총점 구하기

총점 : 100100100

1. 변수와 데이터형

■ 변수의 수명과 유효 범위

- 변수는 자신을 선언하는 Dim 문을 포함한 코드 범위 안에서만 값을 가질 수 있음
- 유효 범위
 - 변수가 메모리를 차지하고 값을 유지하는 구간

표 4-5 변수의 유효 범위

유효 범위		설명
지역 변수	블록	<ul style="list-style-type: none">• 선언 시작과 종료문 사이를 말함 : If~EndIf, For~Next, Do~Loop, While~EndWhile, Select~EndSelect 등• 블록 내에 선언된 변수는 해당 블록 내에서만 사용 가능• 예 <pre>If score > 60 Then Dim pass As String pass = "ok" End If</pre>
전역 변수	포함된 모든 모듈	<ul style="list-style-type: none">• 모듈 수준에서 Public으로 선언된 변수로 모든 모듈 내에서 사용 가능• 여러 개의 폼으로 구성된 경우에 모든 폼에서 사용 가능• 예 <pre>Public notice_all As String</pre>

1. 변수와 데이터형

유효 범위		설명
지역 변수		<ul style="list-style-type: none">• 프로시저 내에서 선언된 변수는 해당 프로시저 내에서만 사용 가능• 이 수준의 변수를 '지역변수'라 함• 예 <pre>Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click Dim sum As Integer Dim pass As String If sum > 60 Then pass = "ok" End If TextBox1.Text = sum TextBox2.Text = pass End Sub</pre>
모듈 변수	모듈	<ul style="list-style-type: none">• 모듈 수준에서 Private으로 선언된 변수로 모듈 내의 모든 프로시저에서 사용 가능• 예 <pre>Private total As Integer Sub total_plus() total = total + 1 End Sub Sub total_minus() total = total - 1 End Sub</pre>

1. 변수와 데이터형

- 지역변수의 사용

- 지역변수는 메모리 반납 여부에 따라 Dim과 Static(정적) 변수로 나뉨
- Dim 변수 - 프로시저 실행이 종료되면 메모리를 반납하고 값 삭제
- Static 변수 - 메모리를 반납하지 않고 저장된 값을 그대로 유지하므로, 프로시저가 종료되더라도 그 값을 유지해야 할 경우에 사용

1. 변수와 데이터형

▪ Dim 지역변수 선언

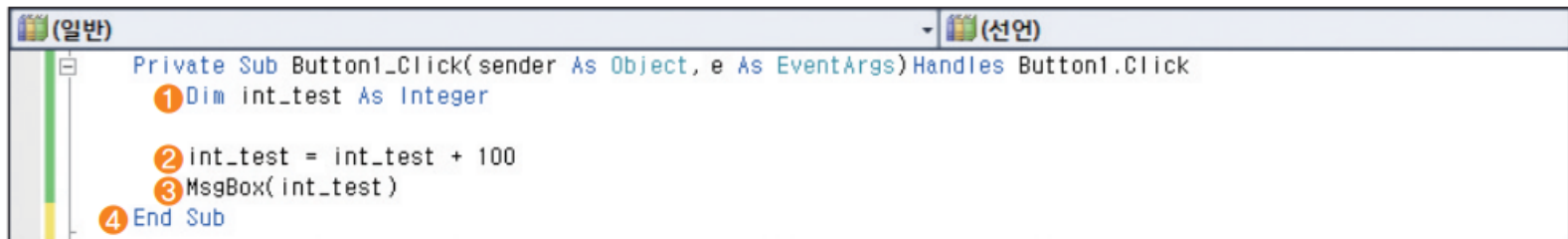
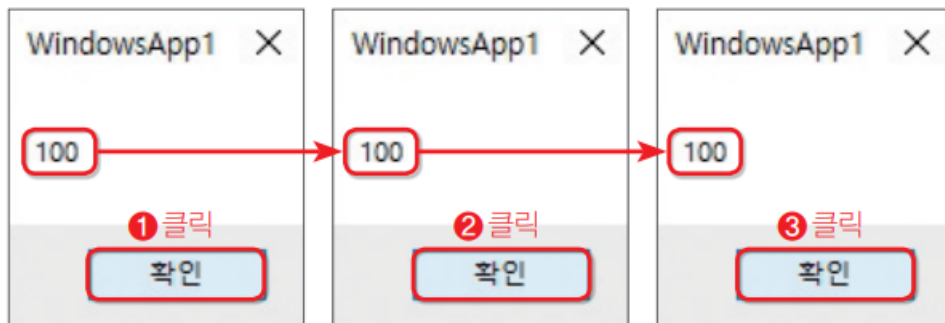


그림 4-3 프로시저 수준 Dim 변수 선언의 예

- ① 정수형 변수 int_test가 생성되고 초깃값으로 0이 저장됨
- ② int_test 값에 정수 100을 더함
- ③ int_test 값을 출력함
- ④ int_test 변수가 소멸됨



1. 변수와 데이터형

■ Static 지역변수 선언

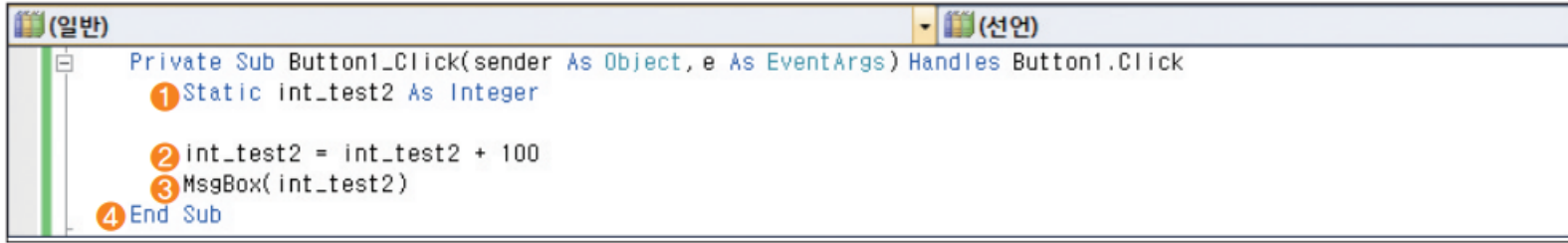
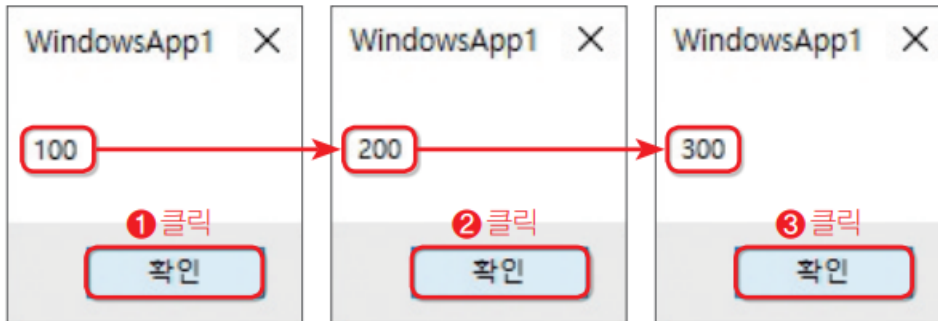


그림 4-4 프로시저 수준 Static 변수 선언의 예

- ① 정수형 변수 int_test2가 생성되고 초기값으로 0이 저장됨
- ② int_test2 값에 정수 100을 더함
- ③ int_test2 값을 출력함
- ④ int_test2 변수가 소멸되지 않고 값이 유지됨



1. 변수와 데이터형

■ 모듈변수의 선언

- 프로시저 내부가 아닌 코드 편집 창 상단의 '(일반)' 부분에서 Dim 또는 Private으로 선언
- 선언된 모듈변수는 모듈 내에 기술된 모든 프로시저에서 이용할 수 있음

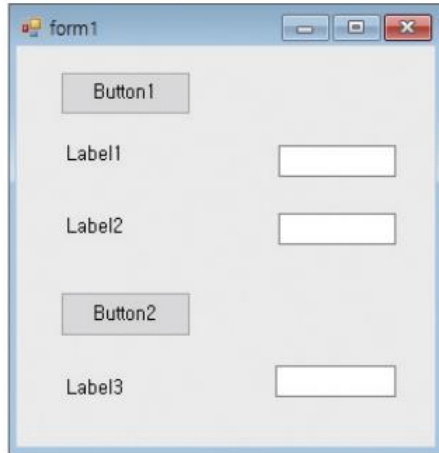
■ 전역변수의 선언

- 프로젝트 내의 모든 프로시저에서 변수를 공유하고자 할 때는 Public 전역변수 선언자를 사용
- 한 번 선언하면 프로젝트가 종료될 때까지 메모리를 유지

1. 변수와 데이터형

실습하기 4-4 지역변수와 모듈변수 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Button1	Text	지역변수
Button2	Text	모듈변수
Label1	Text	지역변수 var_pro 값
Label2	Text	모듈변수 var_mod 값
Label3	Text	모듈변수 var_mod 값
TextBox1, 2, 3	Text	(빈칸)

1. 변수와 데이터형

실습하기 4-4 지역변수와 모듈변수 사용하기

3) 코드 작성

```
01 Private var_mod As Integer
02 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.
    Click
03     Dim var_pro As Integer

04     var_pro = var_pro + 1
05     var_mod = var_mod + 10
06     TextBox1.Text = var_pro
07     TextBox2.Text = var_mod
08 End Sub

09 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.
    Click
10     var_mod = var_mod + 100
11     TextBox3.Text = var_mod
12 End Sub
```

4) 프로그램 실행. 버튼을 클릭하여 지역변수와 모듈변수의 사용 결과를 비교

1. 변수와 데이터형

실습하기 4-4 지역변수와 모듈변수 사용하기

5) <지역변수>를 세 번 클릭하면 지역변수 var_pro 값은 같지만 모듈변수 var_mod 값은 증가

form1

지역변수 ① 클릭

지역변수 var_pro 값 1

모듈변수 var_mod 값 10

모듈변수

모듈변수 var_mod 값

form1

지역변수 ② 클릭

지역변수 var_pro 값 1

모듈변수 var_mod 값 20

모듈변수

모듈변수 var_mod 값

form1

지역변수 ③ 클릭

지역변수 var_pro 값 1

모듈변수 var_mod 값 30

모듈변수

모듈변수 var_mod 값

6) <모듈변수>를 두 번 클릭하면 var_mod 값이 '130', '230'으로 증가. 다시 <지역변수>를 한번 클릭하면 지역변수 var_pro는 역시 '1'이지만 모듈변수 var_mod는 '240'으로 증가

form1

지역변수

지역변수 var_pro 값 1

모듈변수 var_mod 값 30

① 클릭

모듈변수

모듈변수 var_mod 값 130

기존값 +100

form1

지역변수

지역변수 var_pro 값 1

모듈변수 var_mod 값 30

② 클릭

모듈변수

모듈변수 var_mod 값 230

기존값 +10

form1

지역변수 ③ 클릭

지역변수 var_pro 값 1

모듈변수 var_mod 값 240

모듈변수

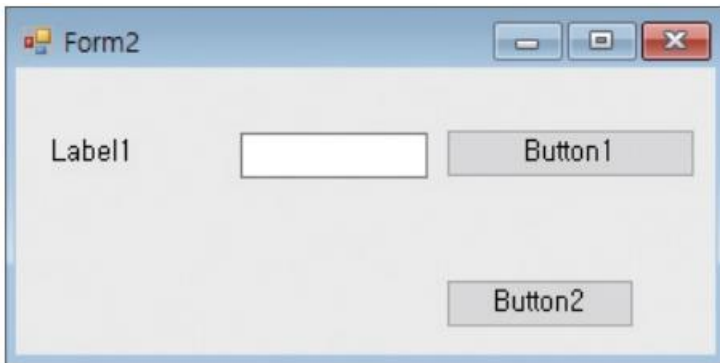
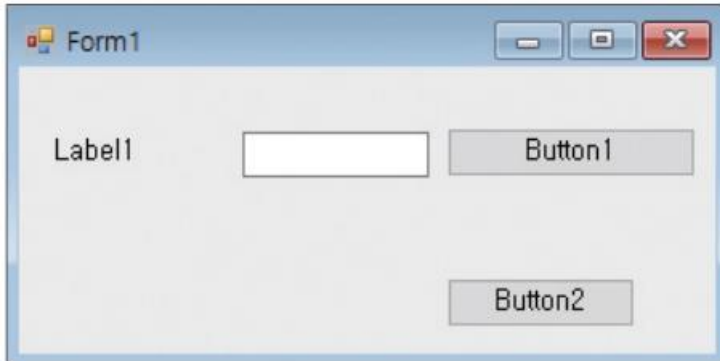
모듈변수 var_mod 값 230

기존값 +10

1. 변수와 데이터형

실습하기 4-5 전역변수 사용하기

1) 윈도우 폼 디자인



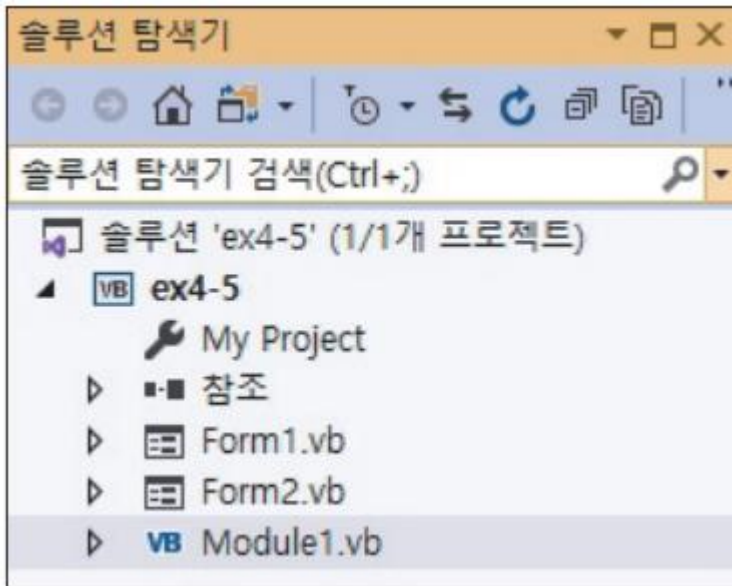
2) 속성 설정

폼	컨트롤	속성	속성값
Form1	Button1	Text	전역변수+1증가
	Button2	Text	Form2 보기
	Label1	Text	전역변수 Sum값
	TextBox1	Text	(빈칸)
Form2	Button1	Text	전역변수+100증가
	Button2	Text	Form1 보기
	Label1	Text	전역변수 Sum값
	TextBox1	Text	(빈칸)

1. 변수와 데이터형

실습하기 4-5 전역변수 사용하기

3) [프로젝트]-[클래스 추가] 메뉴에서 '모듈'을 선택하여 다음과 같이 Module1.vb 파일을 추가



1. 변수와 데이터형

실습하기 4-5 전역변수 사용하기

4) 코드 작성

- **Module1**

```
01 Module Module1
02     Public sum As Integer = 0
03 End Module
```

- **form1**

```
01 Public Class Form1
02     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.
        Click
03         sum = sum + 1
04         TextBox1.Text = sum
05     End Sub

06     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.
        Click
07         Form2.Show()
08     End Sub
09 End Class
```

1. 변수와 데이터형

실습하기 4-5 전역변수 사용하기

4) 코드 작성

- **form2**

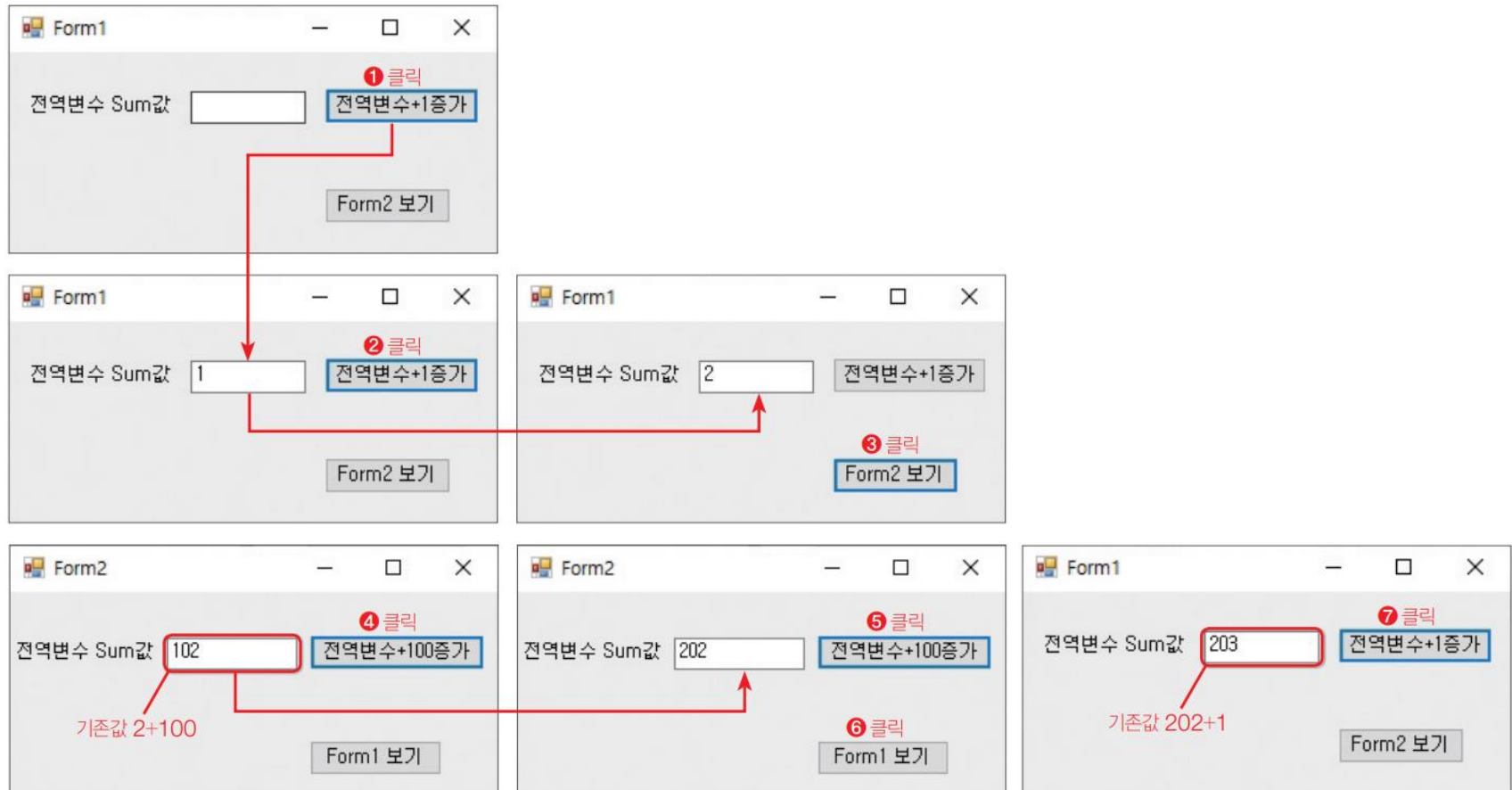
```
01 Public Class Form2
02     Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.
        Click
03         sum = sum + 100
04         TextBox1.Text = sum
05     End Sub

06     Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.
        Click
07         Form1.Show()
08     End Sub
09 End Class
```

1. 변수와 데이터형

실습하기 4-5 전역변수 사용하기

5) 실행 결과 확인



2. 연산자

- 연산자

- $+$, $-$, $*$, $/$ 과 같이 이미 정의된 연산을 수행하는 기호

- 피연산자

- 연산의 대상 또는 연산자의 동작을 받는 변수나 값

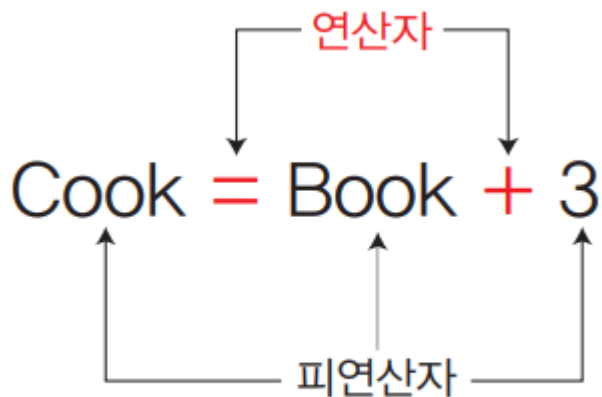


그림 4-5 연산자와 피연산자

2. 연산자

■ 산술 연산자

표 4-6 산술 연산자의 종류

연산자	일반 형식	의미	사용 예
+	$x+y$	덧셈	$2+5$
-	$x-y$	뺄셈	$5-3$
*	$x*y$	곱셈	$4*6$
/	x/y	나눗셈	$8/3$
^	x^y	거듭제곱	3^4
\	$x \setminus y$	x를 y로 나눈 몫의 정수를 구함	$8 \setminus 3$
Mod	$x \text{ Mod } y$	x를 y로 나눈 나머지를 구함	$8 \text{ Mod } 3$

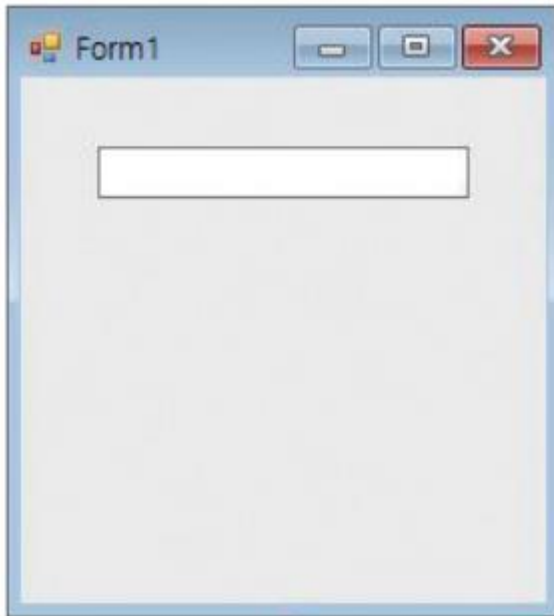
$$\begin{array}{r} 7 \longleftarrow \text{몫} \\ 2 \overline{) 15} \\ \underline{14} \\ 1 \longleftarrow \text{나머지} \end{array}$$

그림 4-6 Mod 연산의 예

2. 연산자

실습하기 4-6 산술 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Form1	Text	산술 연산자
	MultiLine	True
TextBox1	Size	150, 150
	Text	(빈칸)

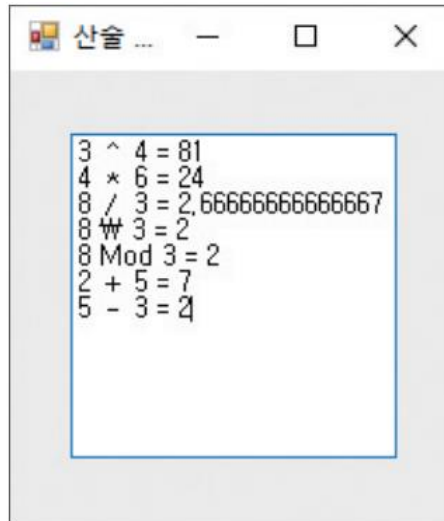
2. 연산자

실습하기 4-6 산술 연산자 사용하기

3) 코드 작성

```
01 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.  
    Load  
02     TextBox1.Text = "3 ^ 4 =" & 3 ^ 4 & vbCrLf &  
03         "4 * 6 =" & 4 * 6 & vbCrLf &  
04         "8 / 3 =" & 8 / 3 & vbCrLf &  
05         "8 \ 3 =" & 8 \ 3 & vbCrLf &  
06         "8 Mod 3 =" & 8 Mod 3 & vbCrLf &  
07         "2 + 5 =" & 2 + 5 & vbCrLf &  
08         "5 - 3 =" & 5 - 3  
09 End Sub
```

4) 실행 결과 확인



2. 연산자

■ 관계(비교) 연산자

- 두 피연산자의 대소 관계를 비교하여 참이면 True, 거짓이면 False 값을 반환

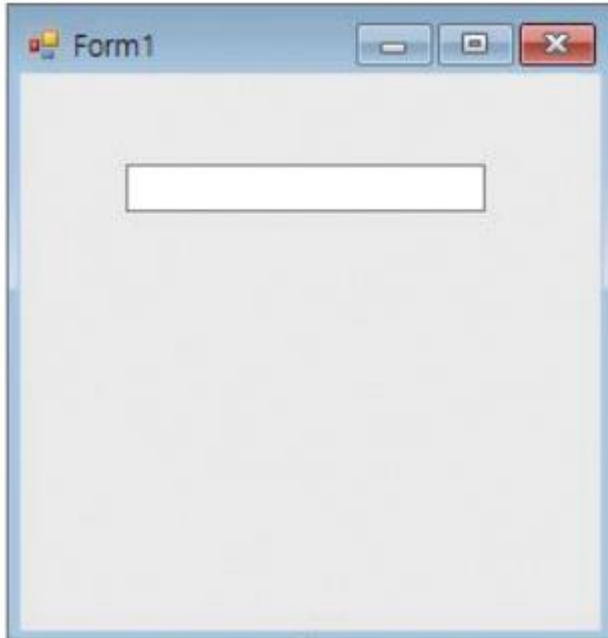
표 4-7 관계 연산자의 종류

연산자	일반 형식	의미	사용 예
<	$x < y$	작다(x가 작으면 True).	$9 < 15 \rightarrow \text{True}$
<=	$x \leq y$	작거나 같다(x가 작거나 같으면 True).	$21 \leq 21 \rightarrow \text{True}$
>	$x > y$	크다(x가 크면 True).	$31 > 33 \rightarrow \text{False}$
>=	$x \geq y$	크거나 같다(x가 크거나 같으면 True).	$11 \geq 10 \rightarrow \text{True}$
=	$x = y$	같다(x와 y가 같으면 True).	$23 = 25 \rightarrow \text{False}$
< >	$x < > y$	같지 않다(x와 y가 같지 않으면 True).	$24 < > 24 \rightarrow \text{False}$
Is	$x \text{ Is } y$	두 개체가 가리키는 값이 같은지 알아낸다 (x와 y 두 개체가 가리키는 변수가 같으면 True, 다르면 False).	set Form_ob = Form1 set Label_ob = Label1 Form_ob Is Label_ob $\rightarrow \text{False}$
Like	$x \text{ Like } y$	두 문자열을 비교한다(x와 y를 비교하여 같으면 True, 다르면 False).	"Basic" Like "basi" $\rightarrow \text{False}$

2. 연산자

실습하기 4-7 관계 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Form1	Text	관계 연산자
	MultiLine	True
TextBox1	Size	150, 150
	Text	(빈칸)

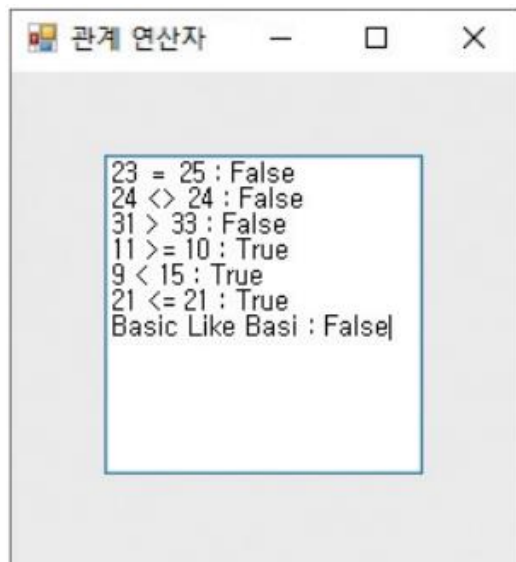
2. 연산자

실습하기 4-7 관계 연산자 사용하기

3) 코드 작성

```
01 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.  
    Load  
02     TextBox1.Text = "23 = 25 : " & CStr(23 = 25) & vbCrLf &  
03         "24 <> 24 : " & CStr(24 <> 24) & vbCrLf &  
04         "31 > 33 : " & CStr(31 > 33) & vbCrLf &  
05         "11 >= 10 : " & CStr(11 >= 10) & vbCrLf &  
06         "9 < 15 : " & CStr(9 < 15) & vbCrLf &  
07         "21 <= 21 : " & CStr(21 <= 21) & vbCrLf &  
08         "Basic Like Basi : " & CStr("Basic" Like "Basi")  
09 End Sub
```

4) 실행 결과 확인



2. 연산자

■ 논리 연산자

- 논리식의 참과 거짓을 판정하여 True/False를 반환

표 4-8 논리 연산자의 종류

연산자	일반 형식	의미
And	x And y	x, y 둘 다 True일 때만 True
Or	x Or y	x, y 둘 다 False일 때만 False
Xor	x Xor y	x와 y가 다르면 True, 같으면 False
Not	Not x	x가 True이면 False, False이면 True
Andalso	x Andalso y	x와 y가 같으면 True, 다르면 False
OrElse	x OrElse y	x가 True이고 y가 False이면 False, 그 외는 True

2. 연산자

실습하기 4-8 논리 연산자 사용하기

1) 앞서 작성한 프로그램에서 컨트롤의 속성을 다음과 같이 일부 변경

컨트롤	속성	속성값
Form1	Text	논리 연산자
	MultiLine	True
TextBox1	Size	150, 150
	Text	(빈칸)

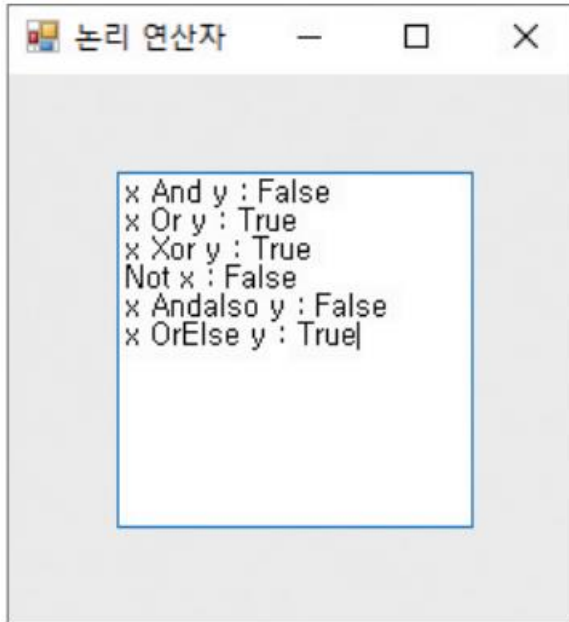
2) 다음과 같이 논리 연산자를 사용하는 코드로 변경

```
01 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.  
    Load  
02     Dim x, y As Boolean  
03     x = True  
04     y = False  
05     TextBox1.Text = "x And y : " & CStr(x And y) & vbCrLf &  
06                     "x Or y : " & CStr(x Or y) & vbCrLf &  
07                     "x Xor y : " & CStr(x Xor y) & vbCrLf &  
08                     "Not x : " & CStr(Not x) & vbCrLf &  
09                     "x AndAlso y : " & CStr(x AndAlso y) & vbCrLf &  
10                     "xOrElse y : " & CStr(x OrElse y)  
11 End Sub
```

2. 연산자

실습하기 4-8 논리 연산자 사용하기

3) 실행 결과 확인



```
x And y : False
x Or y : True
x Xor y : True
Not x : False
x AndAlso y : False
x OrElse y : True
```

2. 연산자

■ 대입 연산자

표 4-9 대입 연산자의 종류

연산자	일반 형식	의미
=	$x = y$	<ul style="list-style-type: none">· y의 값을 x에 저장한다.· 오른쪽의 값을 왼쪽에 대입한다.

형식 | (①) = (②)

- ①의 자리에는 값을 저장할 수 있는 변수 등이 와야 하고, ②의 자리에는 값 자체나 값을 담고 있는 변수 등이 올 수 있음

$x = 10$ → 올바른 표현

$x = y + 10$ → 올바른 표현

$10 = x$ → 잘못된 표현

$x + y = 15$ → 잘못된 표현

2. 연산자

■ 연결 연산자

- 문자열들을 결합하여 하나의 문자열을 반환
- + 연산자는 문자열만 연결할 수 있음

표 4-10 연결 연산자의 종류

연산자	일반 형식	의미	사용 예
&	"x"&"y"	두 식을 연결(결과는 "xy")	"Cook"&"Book"→ CookBook
+	"문자열x"+"문자열y"	두 문자열 연결(결과는 "문자열x문자열y")	"Cook"+"Book"→ CookBook

표 4-11 + 연산자 사용 규칙

연산자	일반 형식	의미
x + y	x와 y 모두 수식	덧셈(산술 연산)
	x와 y 모두 문자열	문자열 연결(연결 연산)
	x는 수식, y는 문자열 또는 그 반대	(오류 발생)

2. 연산자

■ 연산자 우선순위

- 연산자는 크게 괄호→산술 연산자→연결 연산자→관계 연산자→논리 연산자 순으로 계산
- 비교 연산자끼리는 우선순위가 같아 왼쪽에서 오른쪽으로 진행
- 산술 연산자와 논리 연산자는 [표 4-12]와 같은 우선순위에 따라 계산

표 4-12 연산자 우선순위

연산자	일반 형식	우선순위
산술 연산자	\wedge (거듭제곱)	높음 ↑
	$*$, $/$ (곱셈과 나눗셈)	
	\backslash (몫의 정수)	
	Mod (나머지)	
	$+$, $-$ (덧셈과 뺄셈)	
연결 연산자	$\&$ (문자열 연결)	↓ 낮음
관계 연산자	$<$, $<=$, $>$, $=>$, $=$	
논리 연산자	Not	
	And	
	Or	
	Xor	
	AndAlso	
	OrElse	

2. 연산자

실습하기 4-9 연산자 우선순위에 따른 결과 값 비교하기

1) 앞서 작성한 프로그램에서 컨트롤의 속성을 다음과 같이 변경

컨트롤	속성	속성값
Form1	Text	연산자 우선순위

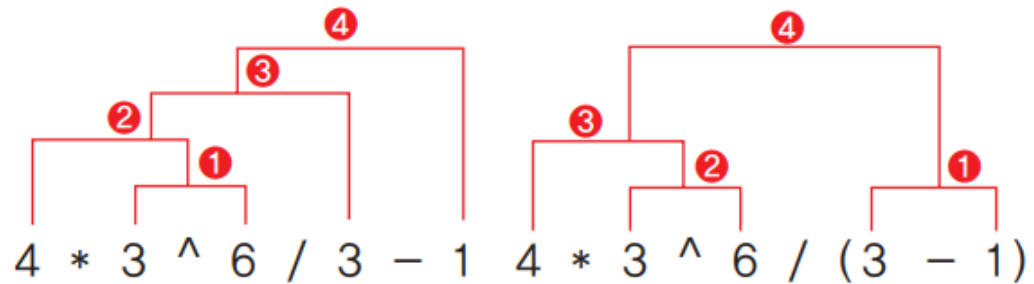
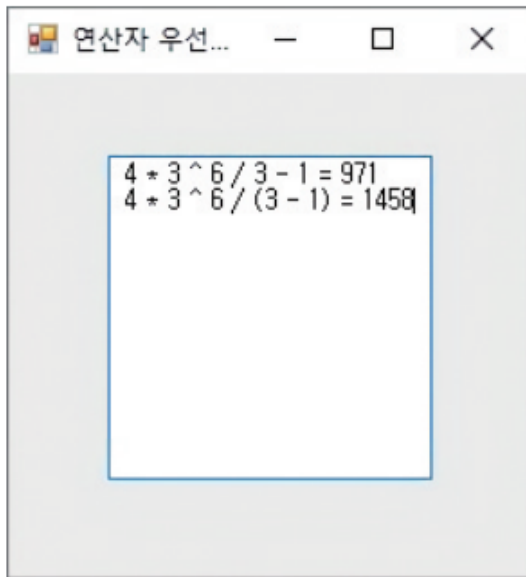
2) 코드 변경

```
01 Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.  
    Load  
02     TextBox1.Text = " 4 * 3 ^ 6 / 3 - 1 = " & 4 * 3 ^ 6 / 3 - 1 & vbCrLf &  
03         " 4 * 3 ^ 6 / (3 - 1) = " & 4 * 3 ^ 6 / (3 - 1)  
04 End Sub
```

2. 연산자

실습하기 4-9 연산자 우선순위에 따른 결과 값 비교하기

3) 실행 결과 확인



2. 연산자

실습하기 4-10 간단한 사칙연산 계산기 만들기

1) 윈도우 폼 디자인

2) 속성 설정

컨트롤	속성	속성값
Form1	Text	사칙연산 계산기
Label1	Text	입력값1
Label2	Text	입력값2
Label3	Text	결과값
TextBox1	Text	(빈칸)
TextBox2	Text	(빈칸)
TextBox3	Text	(빈칸)
Button1	Text	+
Button2	Text	-
Button3	Text	*
Button4	Text	/
Button5	Text	지우기

2. 연산자

실습하기 4-10 간단한 사칙연산 계산기 만들기

3) 코드 작성

```
01 Private intX, intY, Result As Integer
02 Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.
    Click
03     intX = Val(TextBox1.Text)
04     intY = Val(TextBox2.Text)
05     Result = intX + intY
06     TextBox3.Text = Result
07 End Sub
08 Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.
    Click
09     intX = Val(TextBox1.Text)
10     intY = Val(TextBox2.Text)
11     Result = intX - intY
12     TextBox3.Text = Result
13 End Sub
```

2. 연산자

실습하기 4-10 간단한 사칙연산 계산기 만들기

3) 코드 작성

```
14 Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.  
    Click  
15     intX = Val(TextBox1.Text)  
16     intY = Val(TextBox2.Text)  
17     Result = intX * intY  
18     TextBox3.Text = Result  
19 End Sub  
20 Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.  
    Click  
21     intX = Val(TextBox1.Text)  
22     intY = Val(TextBox2.Text)  
23     Result = intX / intY  
24     TextBox3.Text = Result  
25 End Sub
```

2. 연산자

실습하기 4-10 간단한 사칙연산 계산기 만들기

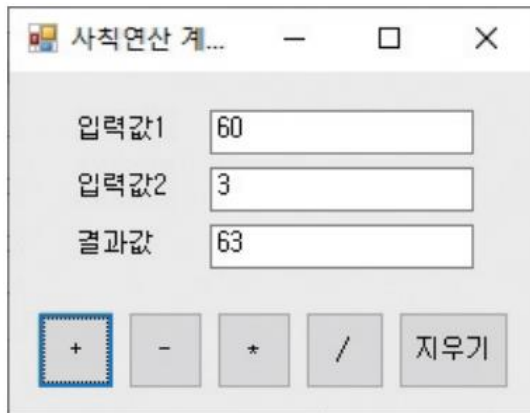
3) 코드 작성

```
26 Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.  
    Click  
27     TextBox1.Text = ""  
28     TextBox2.Text = ""  
29     TextBox3.Text = ""  
30     TextBox1.Focus()  
31 End Sub
```

2. 연산자

실습하기 4-10 간단한 사칙연산 계산기 만들기

4) 실행 결과 확인



사칙연산 계...

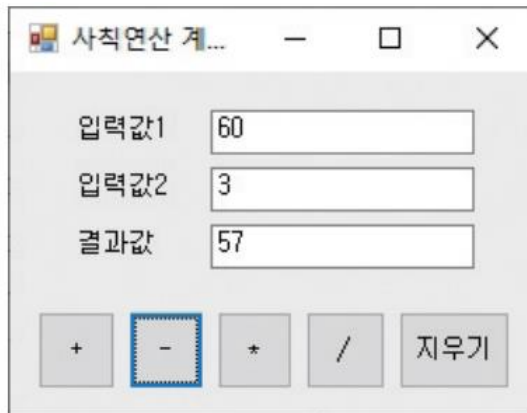
입력값1 60

입력값2 3

결과값 63

+ - * / 지우기

▲ + 버튼을 클릭했을 때



사칙연산 계...

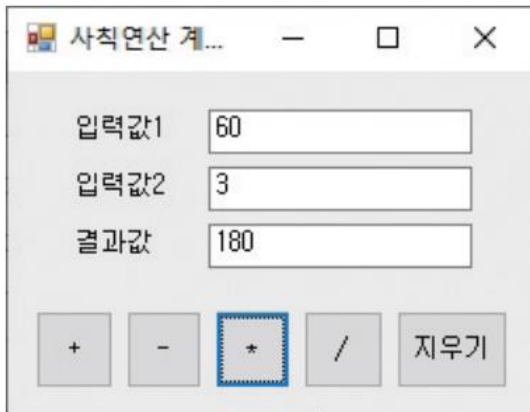
입력값1 60

입력값2 3

결과값 57

+ - * / 지우기

▲ - 버튼을 클릭했을 때



사칙연산 계...

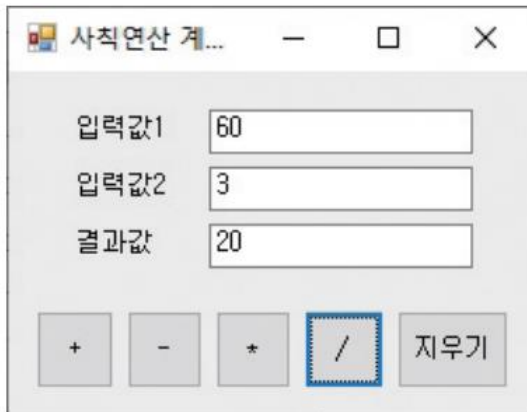
입력값1 60

입력값2 3

결과값 180

+ - * / 지우기

▲ * 버튼을 클릭했을 때



사칙연산 계...

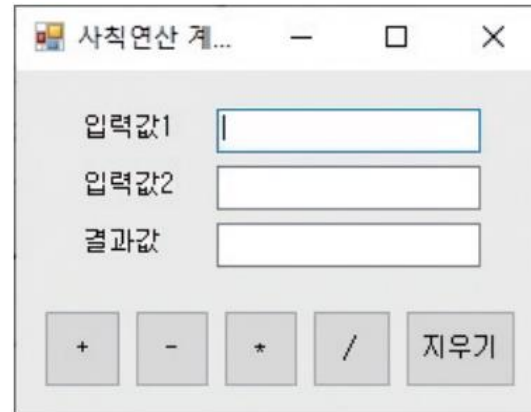
입력값1 60

입력값2 3

결과값 20

+ - * / 지우기

▲ / 버튼을 클릭했을 때



사칙연산 계...

입력값1 |

입력값2

결과값

+ - * / 지우기

▲ 지우기 버튼을 클릭했을 때



Thank You
