

제어문 - 조건문

HyoJoon Han
동국대학교
han6343@dongguk.edu



제어문의 개요

if 조건문

if else 조건문

if else if 조건문

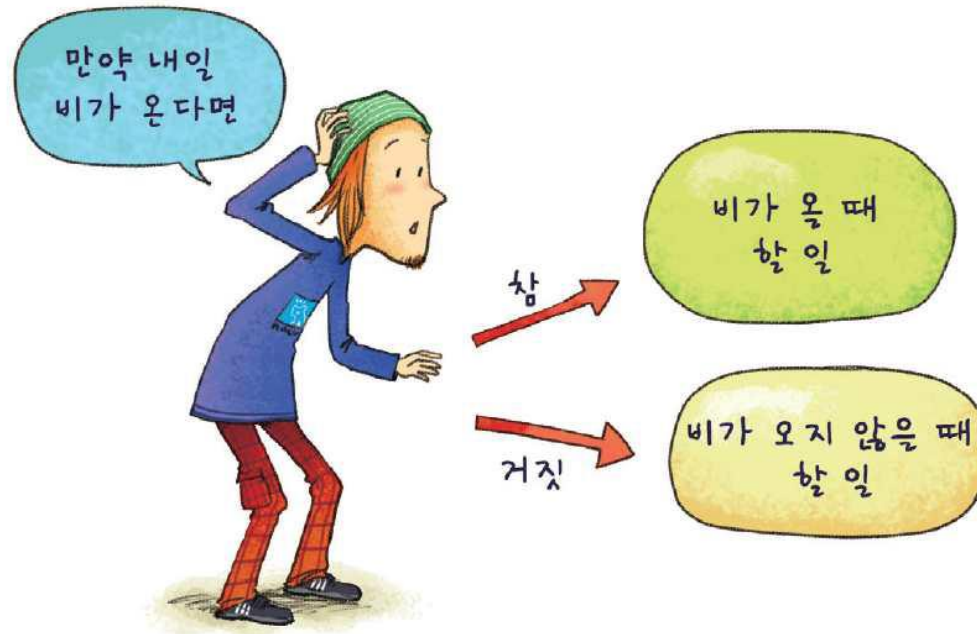
중첩 조건문

switch 조건문

삼항 연산자

- 제어문의 개념
 - 제어문의 정의
 - 프로그램의 실행을 인위적으로 조작하는 명령문
 - 프로그램이 실행될 때 처리할 순서를 결정한다.

그림 5-1 제어문의 의미



- 제어문의 종류
 - 조건문 : 조건을 검사하여 그 결과에 따라 선택적인 작업을 수행
 - 반복문 : 조건에 따라 원하는 명령문을 반복 수행

구분	제어문	설명
조건문	If 조건문	조건을 검사하여 그 결과에 따라 선택적인 작업을 수행
	Switch 조건문	
반복문	While 반복문	조건에 따라 특정 명령문을 반복 수행
	do while 반복문	
	for 반복문	
	foreach 반복문	

- IF 조건문

- 조건이 참이면 if{ } 사이의 명령문을 수행하고 if문을 종료한다. 조건이 거짓이면 수행하지 않고 다음 명령문으로 바로 넘어간다.

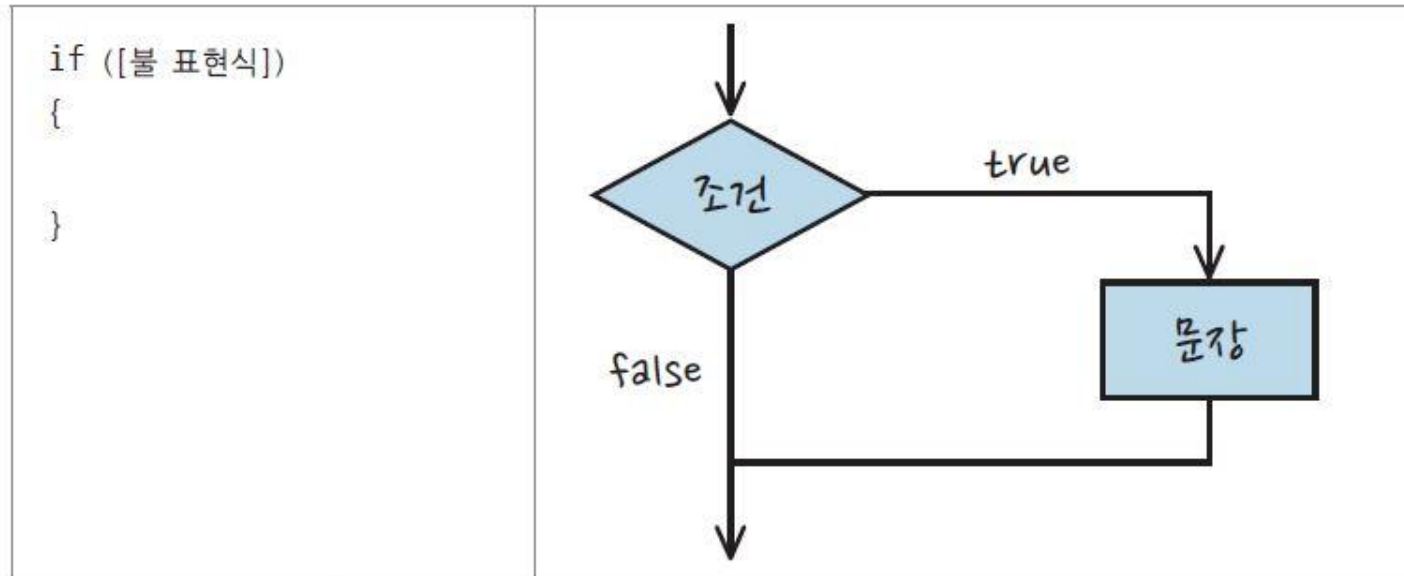


그림 3-1 if 조건문 형식과 순서도

- 코딩 규칙

코드 3-2 중괄호 사용 방식

/3장/Conditions

```
if(true)
{

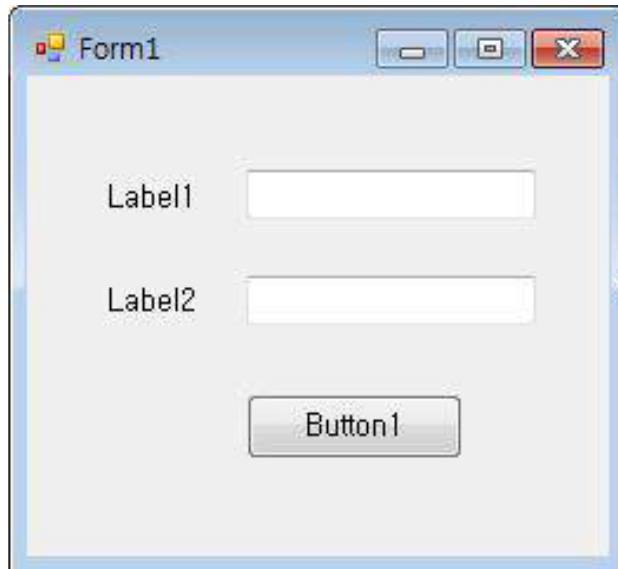
}
```

```
if(true) {

}
```

- 실습 – IF 문을 이용하여 합격 여부 출력하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Form1	Text	학점 계산
Label1	Text	이름
Label2	Text	점수
TextBox1, TextBox2	TextAlign	Center
	Text	(빈칸)
Button1	Text	결과 확인

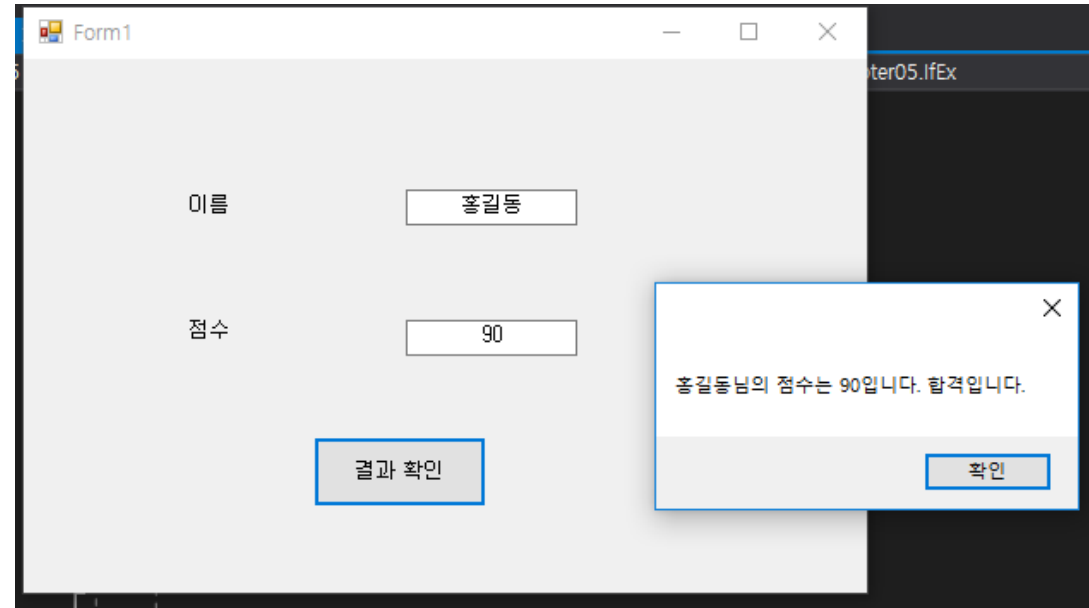
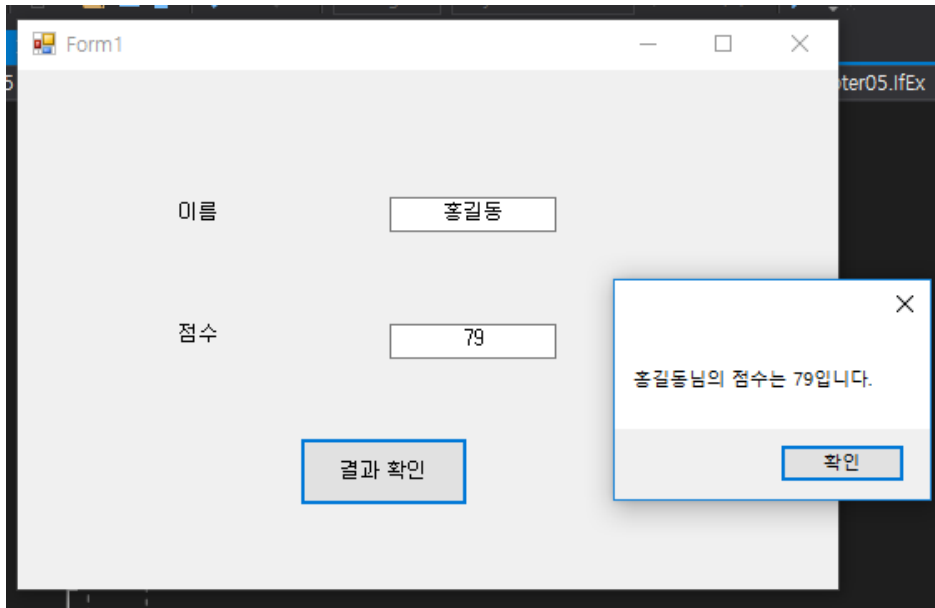
3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int score = int.Parse(textBox2.Text);
    String result = "";

    if (score >= 80) {
        result = "합격입니다.";
    }

    MessageBox.Show(textBox1.Text + "님의 점수는 " +
        score.ToString() + "입니다. " + result);
}
```


4) 실행 결과



- 논리 연산자와 조건문

```
double score = 2.0;  
if (4.0 < score < 4.5) { }
```

'<' 연산자는 'bool' 및 'double' 형식의 피연산자에 적용할 수 없습니다.

```
// 학점 변수  
double score = 3.6;  
  
// 조건을 구분합니다  
if(score == 4.5)  
    Console.WriteLine("신");  
else if (4.2 <= score && score < 4.5)  
    Console.WriteLine("교수님의 사랑");  
else if (3.5 <= score && score < 4.2)  
    Console.WriteLine("현 체제의 수호자");  
else if (2.8 <= score && score < 3.5)  
    Console.WriteLine("일반인");
```

명령문이 한 줄로 끝나면
중괄호 생략 가능

- 두 가지로 분명하게 나눌 때 사용 용이
- 문장이 여러 줄이면 중괄호 반드시 사용
- 조건이 참이면 문장1, 거짓이면 문장2를 수행

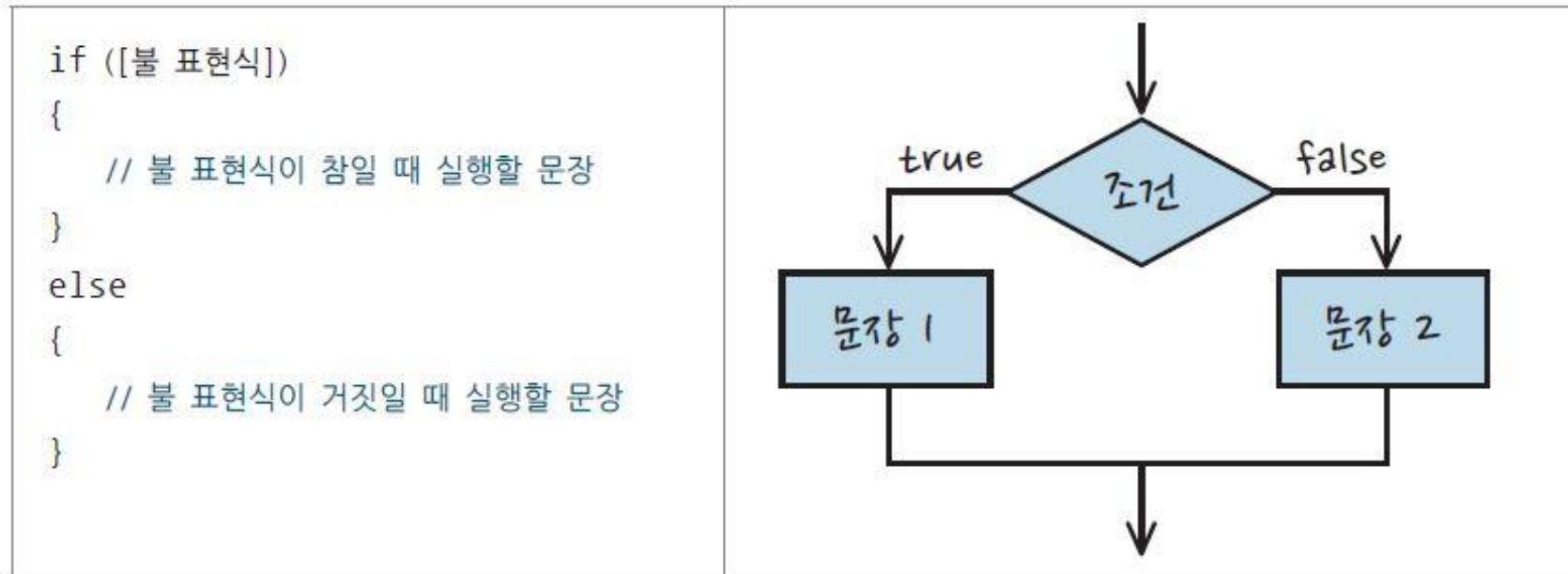


그림 3-2 if else 조건문 형식과 순서도

- 실습 – If ~ else 문을 이용하여 합격/불합격 판정 메시지 출력하기
 - IF 조건문 실습 코드를 다음과 같이 변경

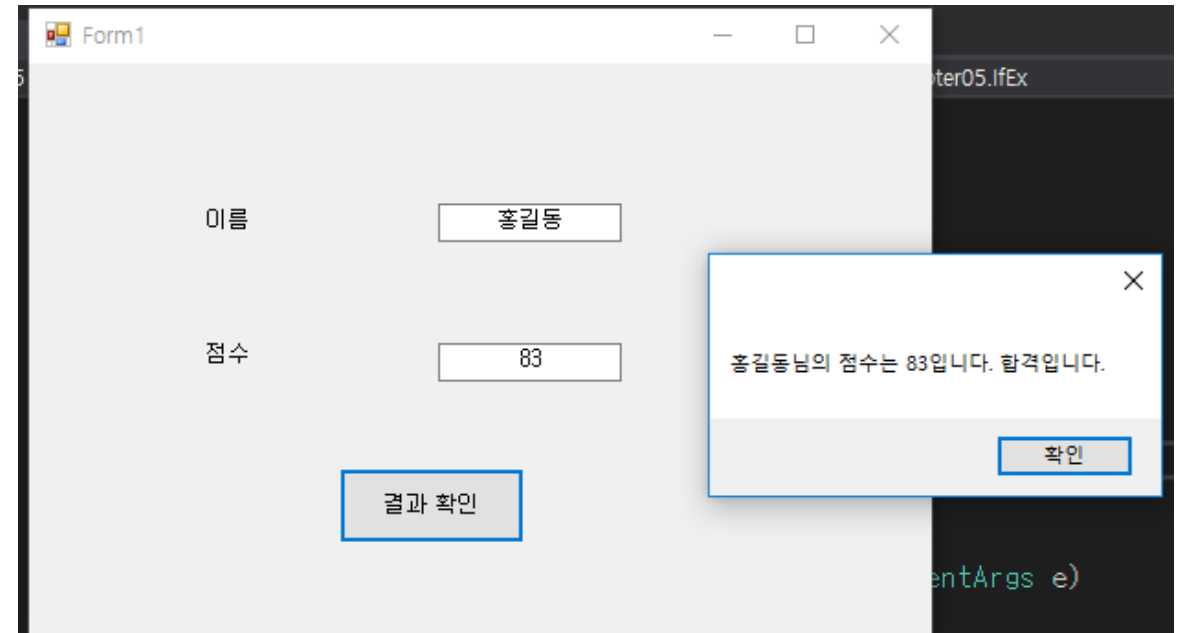
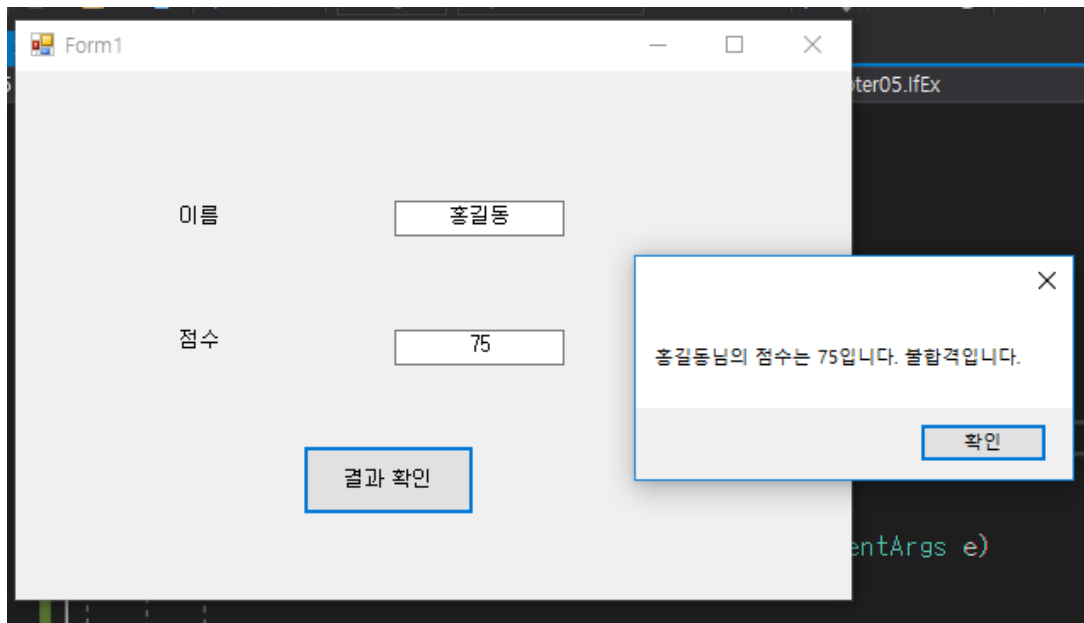
1) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int score = int.Parse(textBox2.Text);
    String result = "";

    if (score >= 80)
    {
        result = "합격입니다.";
    }
    else
    {
        result = "불합격입니다.";
    }

    MessageBox.Show(textBox1.Text + "님의 점수는 " +
        score.ToString() + "입니다. " + result);
}
```

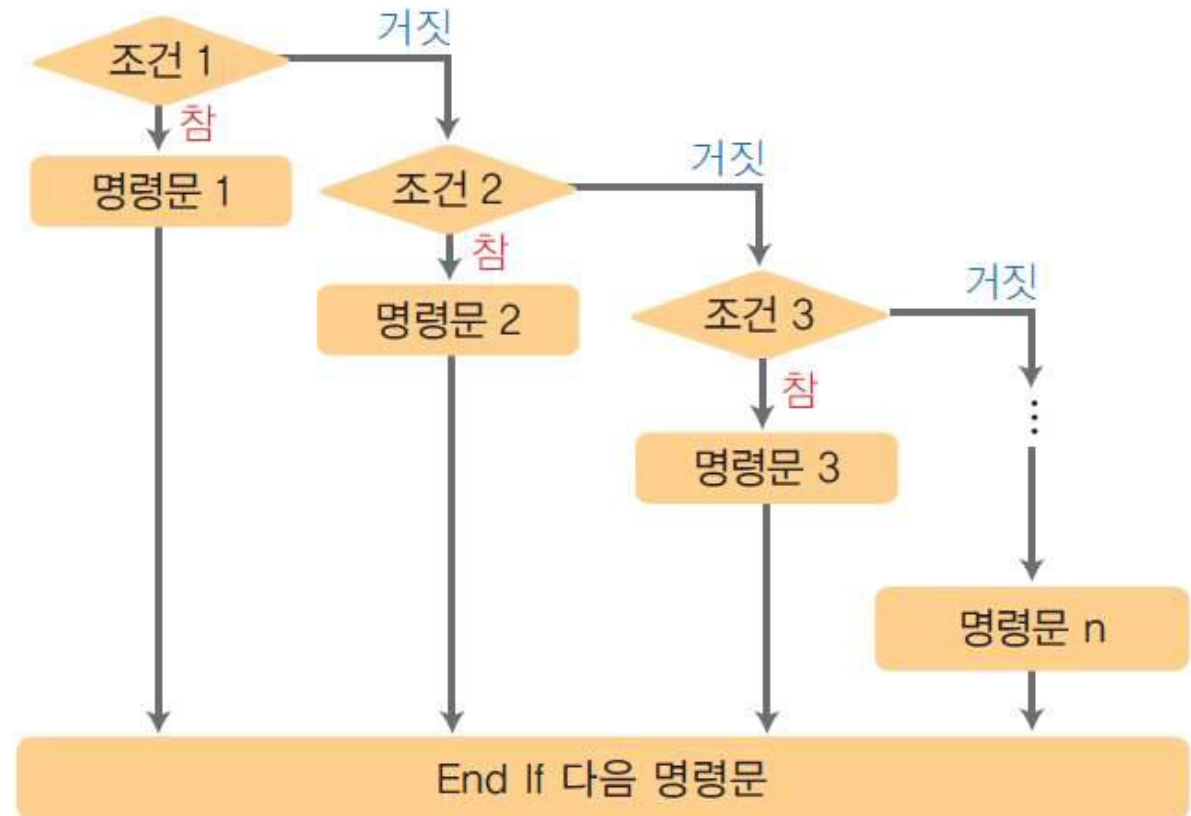
2) 실행 결과 확인



- 중복되지 않는 조건이 세 가지 이상일 때 사용하는 조건문

```
if ([불 표현식])  
{  
    문장;  
}  
else if ([불 표현식])  
{  
    문장;  
}  
else if ([불 표현식])  
{  
    문장;  
}  
else  
{  
    문장;  
}
```

그림 3-4 if else if 조건문 형식



- 실습 – if else if 문을 이용하여 학점 출력하기
 - if else 조건문 실습 코드를 다음과 같이 변경

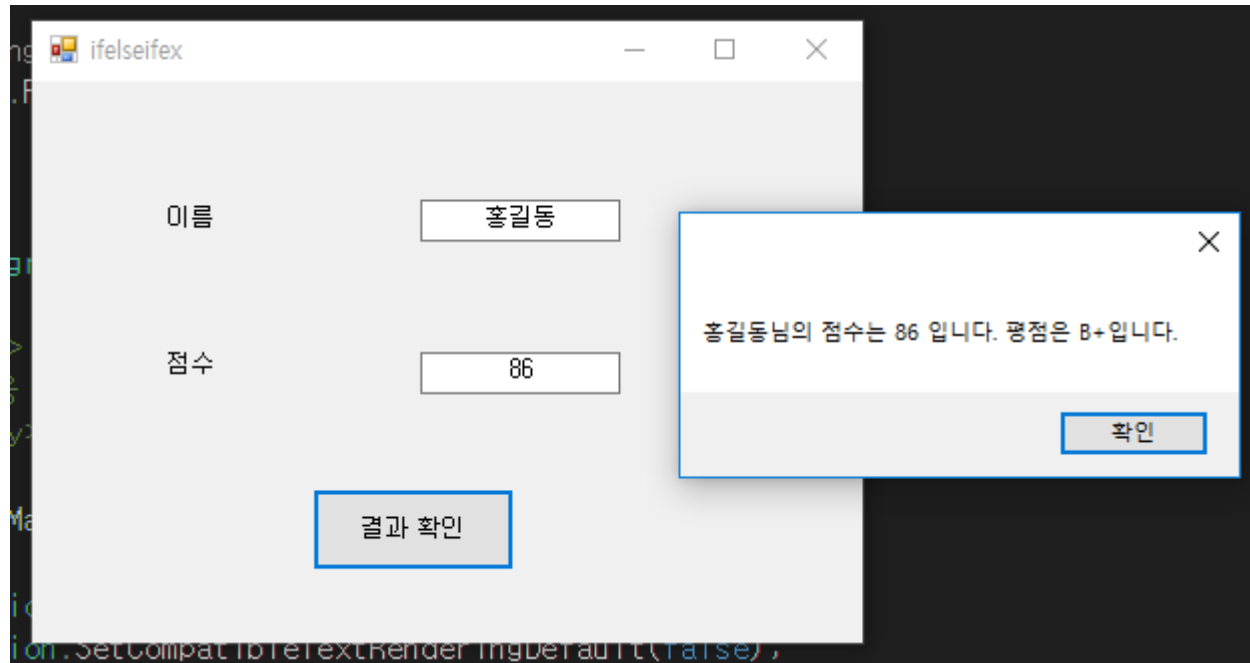
1) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int score = int.Parse(textBox2.Text);
    String result = "";

    if (score >= 95)
        result = "A+";
    else if (score >= 90)
        result = "A";
    else if (score >= 85)
        result = "B+";
    else if (score >= 80)
        result = "B";
    else if (score >= 75)
        result = "C+";
    else if (score >= 70)
        result = "C";
    else
        result = "F";

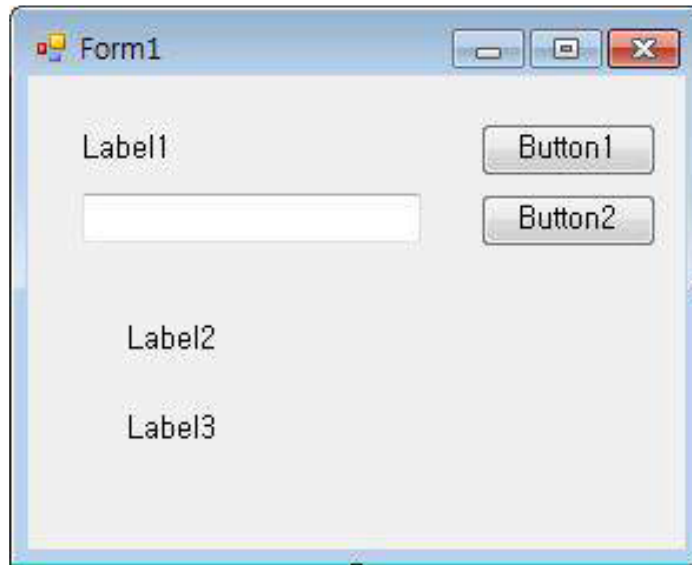
    MessageBox.Show(textBox1.Text + "님의 점수는 " +
        score.ToString() + " 입니다. 평점은 " + result + "입니다.");
}
```

2) 실행 결과



- 실습 – if else if 문을 이용하여 숫자 맞추는 게임 만들기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
Form1	Text	숫자 맞추기 게임
Label1	Text	정답을 입력하세요(1~100)
Label2	Text	준비
Label3	Text	정답
TextBox1	Text	(빈칸)
Button1	Text	새 게임
Button2	Text	확인

3) 코드 작성

```
namespace Chapter05
{
    public partial class randNum : Form
    {
        int i;

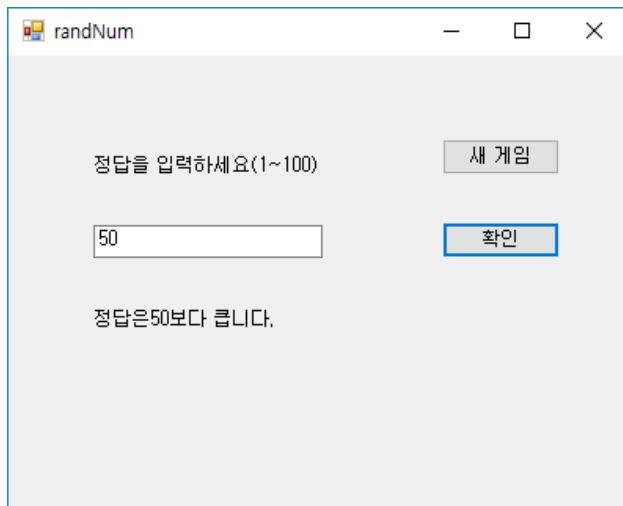
        public randNum()
        {
            InitializeComponent();
        }

        private void randNum_Load(object sender, EventArgs e)
        {
            Random rnd = new Random();
            i = rnd.Next(1, 100);
            label3.Visible = false;
            label3.Text = i.ToString();
        }
    }
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    i = rnd.Next(1, 100);
    label2.Text = "준비";
    label3.Visible = false;
    label3.Text = i.ToString();
    textBox1.Text = "";
}

private void button2_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("값을 입력하세요");
        return;
    }
    int num = int.Parse(textBox1.Text);
    if (num > i)
        label2.Text = "정답은 " + num.ToString() + "보다 작습니다.";
    else if (num < i)
        label2.Text = "정답은 " + num.ToString() + "보다 큼니다.";
    else
    {
        label2.Text = "정답입니다.";
        label3.Visible = true;
    }
}
```

4) 실행 결과



randNum

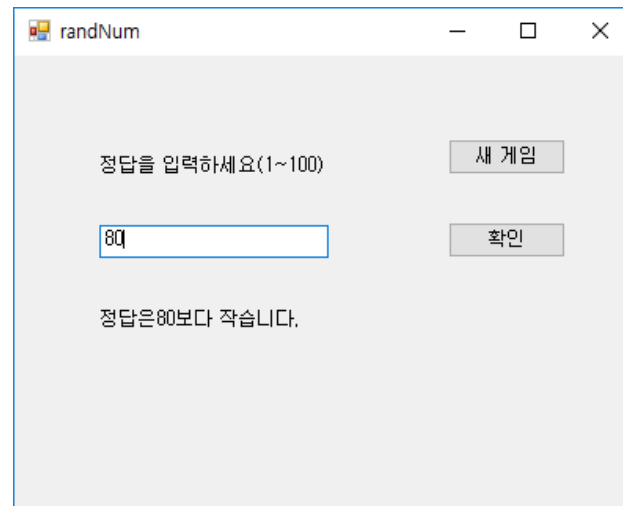
정답을 입력하세요(1~100)

50

새 게임

확인

정답은 50보다 큽니다.



randNum

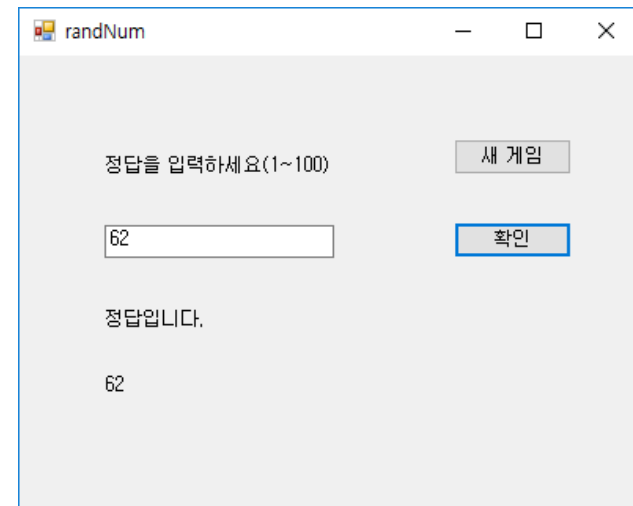
정답을 입력하세요(1~100)

80

새 게임

확인

정답은 80보다 작습니다.



randNum

정답을 입력하세요(1~100)

62

새 게임

확인

정답입니다.

62

- 조건문 안에 조건문을 중첩해 사용
- 여러 번 더 중첩 가능
- A 이면서 B 인 상황 같이 두 가지 조건 이상의 조건 만족이 필요할 때 사용 가능

```
if (불 표현식)
{
    if (불 표현식)
    {
        문장;
    }
    else
    {
        문장;
    }
}
else
{
    if (불 표현식)
    {
        문장;
    }
    else
    {
        문장;
    }
}
```

그림 3-3 중첩 조건문 형식

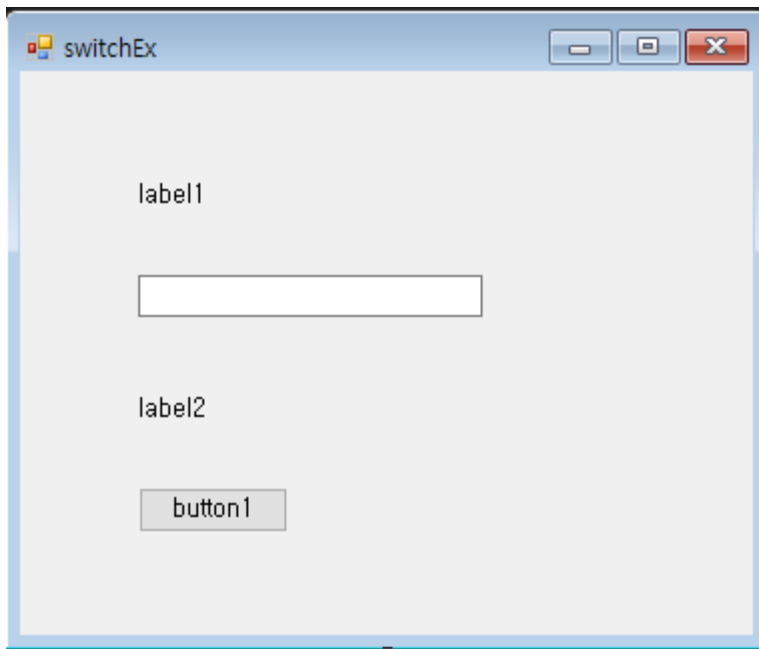
- 선택 사항이 많으면 다중 if 문의 구조가 복잡해져서 switch 문을 사용하여 단순한 구조로 작성 가능
- 조건을 주고 그 조건이 가질 수 있는 값을 각 case 목록에 나타낸다.
- 해당하는 case를 찾아 연결된 명령문을 수행하고 switch 문을 빠져나가는 구조. 조건에 해당하는 값이 없을 때는 default 다음 명령문을 수행
- break 키워드 : switch 조건문 또는 반복문을 빠져나갈 때 사용
- switch 조건문 괄호 안에는 비교할 값 입력, 입력값을 기준으로 특정 코드 실행

```
switch ([비교할 값])  
{  
    case [값]:  
        문장  
        break;  
    case [값]:  
        문장  
        break;  
    default:  
        문장  
        break;  
}
```

그림 3-6 switch 조건문의 형식

- 실습 – switch 조건문으로 홀수와 짝수 구분

1) 윈도우 폼 디자인



2) 속성 설정

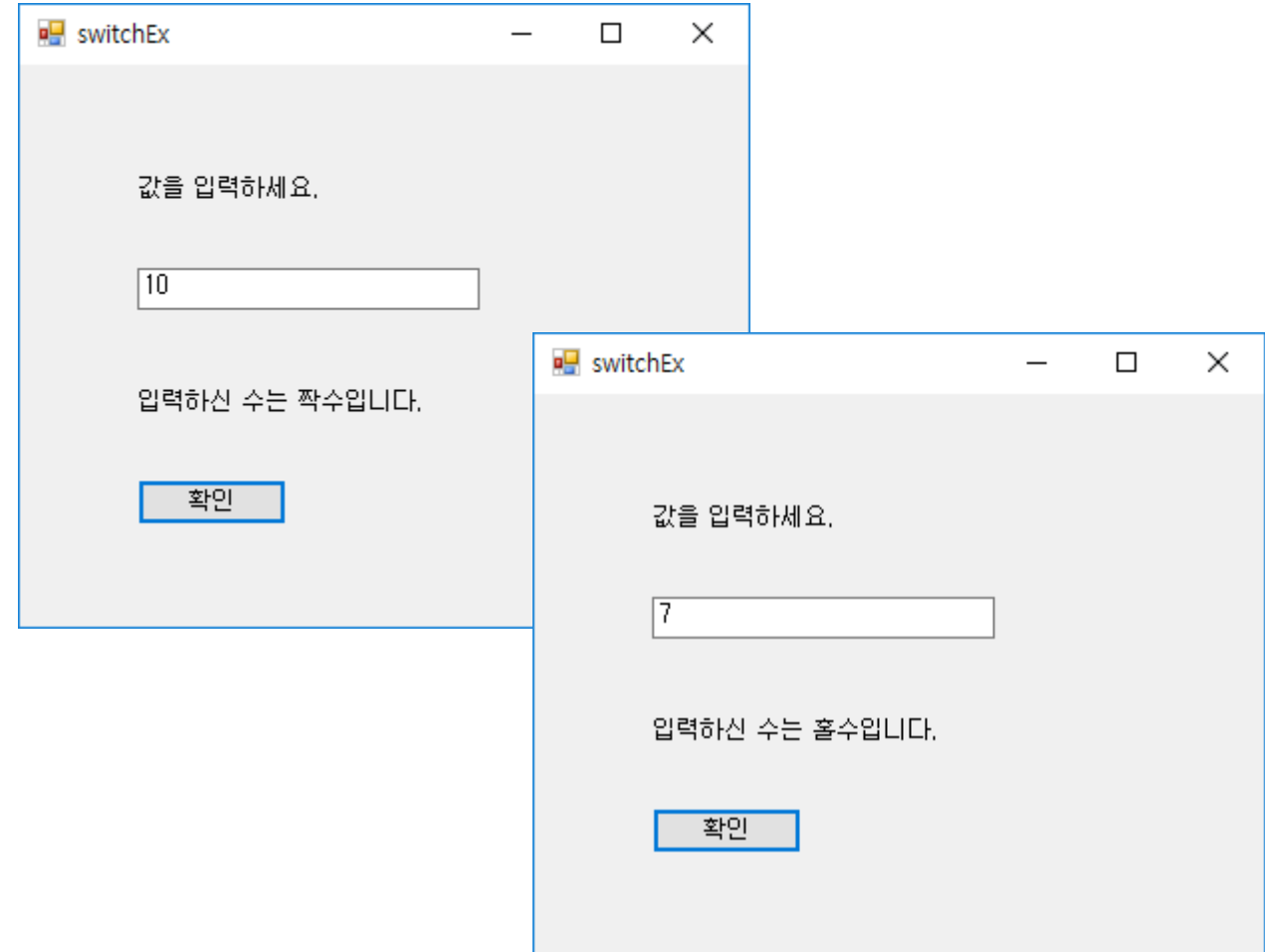
컨트롤	속성	속성값
label1	Text	값을 입력하세요.
label2	Text	(빈칸)
textBox1	Text	(빈칸)
button1	Text	확인

3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int num = int.Parse(textBox1.Text);
    switch (num%2)
    {
        case 0:
            label2.Text = "입력하신 수는 짝수입니다.";
            break;

        case 1:
            label2.Text = "입력하신 수는 홀수입니다.";
            break;
    }
}
```

4) 실행 결과

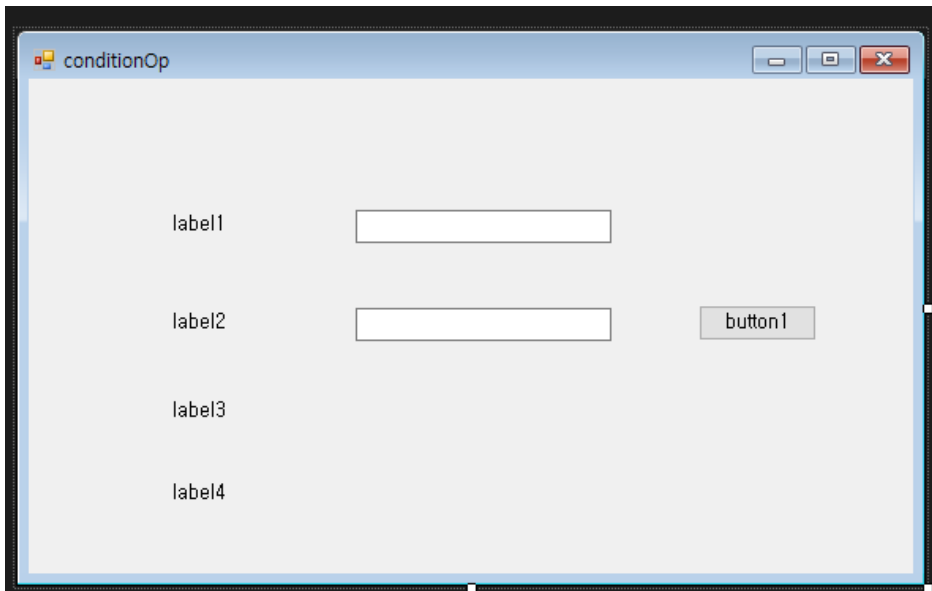


- 삼항 연산자
 - 어떤 조건에 의해 값이 결정되는 연산자

연산자	일반 형식	의미
t?x:y	max=x>y?x:y	? 앞의 조건문이 True 이면 : 앞의 값이, False 이면 : 뒤의 값이 대입

- 실습 – 삼항 연산자 사용하기

1) 윈도우 폼 디자인



2) 속성 설정

컨트롤	속성	속성값
label1	Text	입력값1
label2	Text	입력값2
label3	Text	(빈칸)
label4	Text	(빈칸)
textBox1	Text	(빈칸)
textBox2	Text	(빈칸)
button1	Text	확인

3) 코드 작성

```
private void button1_Click(object sender, EventArgs e)
{
    int x = int.Parse(textBox1.Text);
    int y = int.Parse(textBox2.Text);
    int max = x > y ? x : y;
    int min = x < y ? x : y;
    label3.Text = "두 입력 값 중 큰 값은 = " + max.ToString();
    label4.Text = "두 입력 값 중 작은 값은 = " + min.ToString();
}
```

4) 실행 결과

conditionOp

입력값1 10

입력값2 15

확인

두 입력 값 중 큰 값은 = 15

두 입력 값 중 작은 값은 = 10