모듈 6

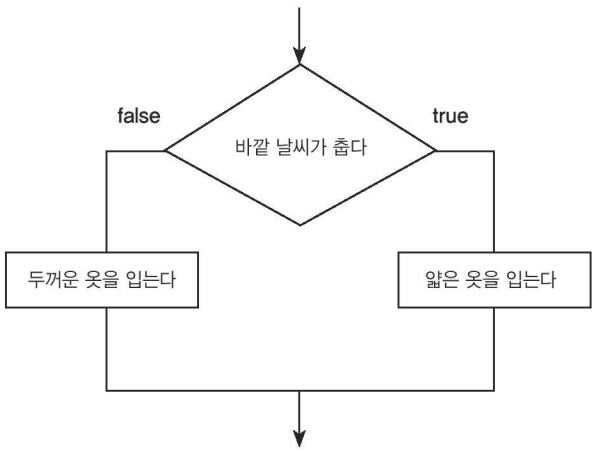
프로그램에서 조건에 따라 실행. IF…THEN…ELSE 문

참과 거짓을 판단하는 논리 표현들을 사용할 수 있게 됩니다.
크고 작은 것을 비교할 수 있게 됩니다.
조건에 따라 다른 행위를 할 수 있도록 프로그램을 만들 수 있게 됩니다.

지금까지는 작동할 행위의 순서가 정해진 프로그램을 작성하였다. 이런 프로그램은 작성된 순서대로 진행된다. 하나의 행위가 실행되고 나서 그 다음이 실행되는 방식이다. 이런 프로그램 구조를 선형linear 프로그램 구조라고 한다.

실생활에서, 우리는 종종 다양한 선택을 하게 된다. 우리는 바깥 날씨가 춥다면 두꺼운 옷을 입고, 그렇지 않으면 얇은 옷을 입는다.

이 상황을 플로우차트flowchart, 흐름도로 그려보면,



다이아몬드 모양의 상자에 적힌 "바깥 날씨가 춥다" 구문이 참이나 거짓이냐에 따라 우리는 "두꺼운 옷을 입는다" 또는 "얇은 옷을 입는다"라는 행위를 한다.

이런 구조를 브랜칭branching, 가지치기 구조라고 한다.

대안이 2 개 이상인 상황에서, 하나를 선택하는 경우는 프로그램에서 매우 일반적이다. 델파이 언어에서는 *if… then…else* 구문을 사용하여 이런 선택 상황을 처리한다.

문장 구조

```
if <논리 표현식> then
   <구문 1>
else
   <구문 2>;
```

*if…then…else*를 실행하는 순서는,

논리 표현식(조건 표현식)의 값을 먼저 도출하고,
만약 논리 표현식의 결과가 `True` 이면, 구문 1을 실행하고,
그렇지 않으면(논리 표현식의 결과가 `False`이면), 구문 2를 실행한다.

예문 1.

정수 2 개가 주어지면, 그 중 가장 큰 값을 계산해보자.

해법

```
procedure TfrmMy.btnMaxClick(Sender: TObject);
var
   a, b, m: Integer;
begin
   a:= StrToInt(edtA.Text);
   b:= StrToInt(edtB.Text);
   if a>b then
      m:= a
   else
```

```
        m:= b;
    lblMax.Caption:= IntToStr(m)
end;
```

잘 보면 *if…then…else* 구문에서 *else* 앞에는 ; 세미콜론이 없다. *if…then…else* 는 복잡하지만 "한 문장"이기 때문에 *else* 앞에서 문장이 그냥 끝나버리면 안 된다. 만약 *else* 앞에 ; 세미콜론을 넣고(문장을 끝내고), 이어서 *else*에서부터 시작하면, *if*가 없는 *else*문장이 되어버린다.

델파이 언어로 작성할 때, 하나의 행위만을 허용되는 상황에서 여러 가지 행위를 실행해야 한다면 복합compound문을 사용하면 된다. 복합문은 여러 개의 문장을 *begin…end*(연산자 괄호) 안에 넣는다.

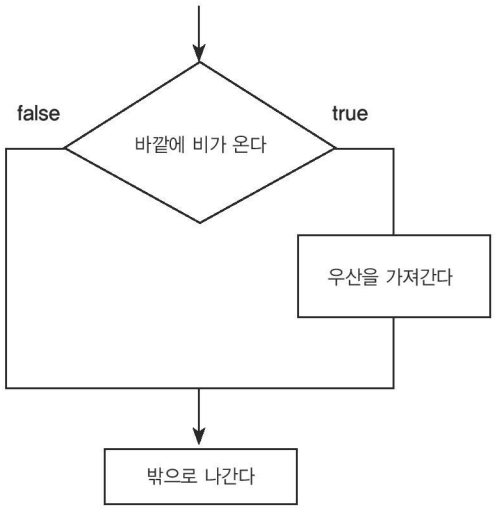
예를 들어 *then* 또는 *else*에 의해 갈라지게 되는 곳에 하나 이상의 문장을 실행해야 한다면, *begin…end*(연산자 괄호) 안에 이 문장들을 넣으면 된다.

```
…
if x<0 then
begin
   y:= 1;
   k:= k+1;
end
else
begin
   y:= 2*x;
   k:= x;
end;
```

복합문 compound statement

복합문

가끔 선택 방식이 다를 수도 있다.



이런 구조는 *else*가 없기 때문에, 더 짧게 *if…then* 구문으로 작성한다.

문장 구조

```
if <논리 표현식> then
    <구문 1>;
```

이와 같이 짧은 버전의 조건conditional 문에서도 긴 버전과 실행 방식은 같다. 만약, 논리 표현식에서 도출된 값이 `True`이면, 구문 1이 실행된다. 그렇지 않다면 *if…then* 문장에서는 실행되는 문장이 전혀 없다.

실습

Exercise 1.

2 개의 정수 m과 n을 사용자로부터 입력 받는다.
만약 n을 m으로 나누어 정수가 딱 떨어질 수 있다면, n을 m으로 나눈 값을 표시한다. 그렇지 않으면 "n은 m으로 나눌 수 없습니다"라는 메시지를 표시한다.

Exercise 2.

세 자리 정수를 받아서, 바로 읽든 거꾸로 읽든 같은 수(예: 373)가 되는지를 알아내보자.

Exercise 3.

서로 다른 3개의 실수 중에서 최대값과 최소값을 찾아 내보자.

Exercise 4.

집이 하나 있는데 이 집에는 방이 n개가 있고 각 방에는 순서대로 방 번호가 붙어 있다. 번호는 a 부터 시작한다. 이 집에 있는 모든 방의 번호를 더하면 짝수인지 아닌지를 알아내보자.