

## 반복<sup>Loop</sup>들

같은 처리를 일정한 조건에서 계속 반복할 수 있는 코드를 작성할 수 있게 됩니다.

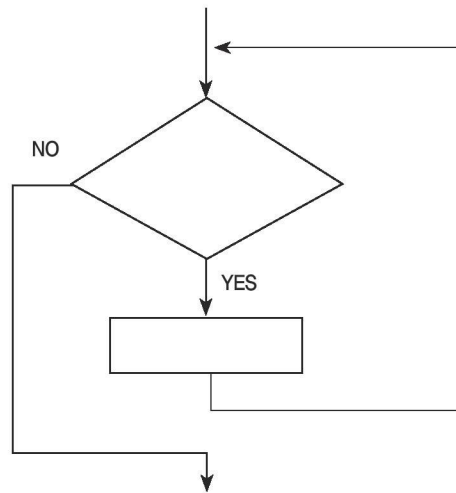
**Loop**<sup>루프, 반복</sup>는 정해진 순서대로 구문을 실행하기를 반복한다.

루프에는 두 가지 종류가 있다. 조건을 ‘먼저 확인’하고 나서 실행하는 루프(*while...do... 루프*)와 실행 ‘후’에 조건을 확인하는 루프(*Repeat...Until... 루프*)이다.

조건을 ‘먼저 확인’하고 나서 실행하는 루프(*while...do... 루프*)의 델파이 표현

```
while <논리 표현식> do
    <구문>;
```

*while...do* 루프는 논리 표현식 즉 조건 표현식이 `True`라면, 같은 구문을 다시 반복한다. 루프 안의 구문이 실행되면 대체로 조건 표현식의 결과에도 영향을 끼치기 때문에, 다시 반복하기 전에 조건 표현식은 항상 다시 계산된다. 만약, 맨 처음 계산된 조건 표현식에서 값이 `False`라면 구문은 한번도 실행되지 못한다.



while...do 플로우차트

여러 개의 구문을 반복해야 한다면 do 다음에 begin... end 블록을 사용한다.

```

while <논리 표현식> do
begin
  <구문 1>;
  <구문 2>;
  <구문 3>;
end;
  
```

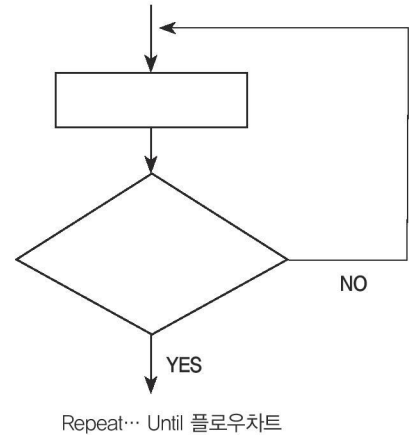
실행 '후'에 조건을 확인하는 루프(Repeat...Until... 루프)의 델파이 표현 형식

```

repeat
  <구문 1>;
  <구문 2>;
  ...
  <구문 n>;
until <논리 표현식>;
  
```

이 루프는 루프의 본문(*Repeat*와 *Until* 사이에 있는 부분)이 먼저 실행된다. 그리고 나서 논리 표현식의 값을 확인한다. 만약 *False*이면 이 구문들을 다시 반복해서 실행한다. 그렇지 않으면(논리 표현식이 *True*이면) 루프는 종료된다. 따라서 *Repeat...Until* 루프의 본문은 최소한 한번은 실행된다. 논리 표현식은 루프를 '종료'할 것인지를 결정하는 조건이다.

만약 본문을 반복할 횟수를 알고 있다면, 횟수를 셀 수 있는 *for* 루프를 사용할 수 있다.



## 형식

```
for <루프 카운터>:= <최초 값> To <최종 값> do
    <구문>;
```

또는

```
for <루프 카운터>:= <최초 값> DownTo <최종 값> do
    <구문>;
```

〈루프 카운터〉 — *Integer*<sup>정수</sup> 타입의 변수이다. 최초 값에서 시작하여 *To* 인 경우에는 1씩 '증가'하고, *DownTo* 인 경우에는 1씩 '감소'하면서, 자동으로 변한다.

〈최초 값〉 — 정수 표현식: 반복을 시작할 때 루프 카운터에 이 값이 들어간다.

〈최종 값〉 — 정수 표현식: 루프 카운터에 이 값이 들어가면 마지막 실행을 한다.

〈구문〉 — 반복하여 실행될 구문이다. 루프 카운터가 최종 값을 넘게 되면 반복을 중단한다. 여러 구문이 필요하다면 *do* 다음에 *begin... end*를 사용하여 감싼다.

최초 값과 최종 값은 루프가 시작할 때 단 한번만 계산된다.

*for* 루프 역시 조건을 '먼저 확인'하는 루프이다.

## 실습

### Exercise 1.

일정한 간격으로 떨어져 있는 수평선을  $N$  개 그려보자.  $N$ 은 텍스트박스에서 받는다.

### Exercise 2.

한 변의 길이가  $s$ 인 정사각형  $N$ 개를 일정한 간격으로 떨어뜨려서 그려보자. 숫자  $N$ 과  $s$ 는 텍스트박스에서 받는다.

### Exercise 3.

$N$ 개의 가로 줄과  $M$ 개의 세로 줄이 있는 체스 판을 그려보자. 숫자  $N$ 과  $M$ 은 텍스트박스에서 받는다.

### Exercise 4.

중심점이 같은 원들을  $N$ 개 그린다. 가장 작은 원의 반지름은  $r$ 이고 가장 큰 원의 반지름은  $R$ 이다. 숫자  $N$ ,  $r$ ,  $R$ 은 텍스트박스에서 받는다.

### Exercise 5.

선 3개의 길이가 될 숫자 3개를 받는다. 이 3개의 선으로 삼각형을 만들 수 있다면 삼각형을 그린다. 만약 삼각형을 만들 수 없다면, 알맞은 에러 메시지를 표시한다.

### Exercise 6.

크기가 다른 정사각형  $N$  개를 포개지도록 그린다. 선(즉 사각형의 각 변)을 그려서 정사각형을 완성하는 프로시저를 작성한다.

### Exercise 7.

간단한 아날로그 시계를 그린다. 텍스트박스에 시<sup>hour</sup>와 분<sup>minute</sup>을 입력하고 버튼을 누르면 시계의 시침과 분침이 그 시간을 가리키도록 해보자.