

1차원 정적^{Static} 배열^{Array}

타입이 동일한 요소들을 모아서 배열로 다룰 수 있게 됩니다.

배열^{Array}은 같은 요소들(타입이 같은 값이나 변수들)을 모아놓은 모음이다. 배열^{Array} 안에 있는 요소들은 각각 자신들의 인덱스번호^{index} 즉 키^{key}(정해지는 이름)가 있어서 다른 요소와 구분될 수 있다. 예를 들어 ‘학급 출석부’ 이라는 목록이 있다면, 이 목록에 있는 각 요소의 타입^{Type}은 ‘학생’이다. 여기에 있는 학생 각자는 자신이 몇 번째로 불리는 지를 가리키는 번호^{Index}를 가지고 있다. 그리고, 각 요소의 값은 ‘학생의 실제 이름’이다.

배열 형식

```
<배열 이름>: Array [<시작 인덱스번호>...<마지막 인덱스번호>] of <요소의 타입>;
```

배열 형식에 들어가는 것들을 살펴보자.

- 배열 이름
- Array(키워드)

– 시작 인덱스번호와 마지막 인덱스번호

인덱스번호의 범위 (예: 1..40, 2..2, 0..10)이다. 앞뒤의 숫자는 각각 인덱스 번호가 될 수 있는 가장 작은 숫자와 가장 큰 숫자이다. 최소 번호가 최대 번호보다 클 수는 없다. 이 두 숫자 사이에는 마침표 2개를 넣는다. 인덱스 번호의 타입은 반드시 열거형 *enumeration* 즉 자신의 앞 뒤에 있는 요소들(의 값)이 미리 정의되어 있는 타입이어야 한다. 우리는 열거형 타입 몇 개를 이미 알고 있다. 모든 정수형 타입들과 *Boolean* 타입은 열거형 타입이다. 인덱스번호의 범위를 정할 때는 반드시 상수^{Constant}만 사용할 수 있다. 즉, 배열의 크기는 프로그램을 작성할 때 결정되는 것이지, 프로그램이 실행되는 중에 크기를 변경할 수 없다.

– 요소의 타입

배열^{Array} 예문

```
A: Array[1..10] of Real;
```

배열 A에는 요소가 10개 있다. 각 요소에는 *Real*^{실수}이 들어간다. 배열의 특정한 요소에 접근하려면 인덱스번호를 사용한다. 인덱스번호와 같은 타입이 나오는 표현식을 써도 좋다.

예문

```
x:= 5;
A[x+2]:= 4.5;
```

배열의 7번 (5+2) 요소에 4.5 라는 값이 들어간다.

인덱스번호를 통해 배열 요소에 접근하면, 작업을 더욱 효율적으로 하게 된다. 예를 들어 배열의 모든 요소에 0을 넣고 싶으면, 아래와 같이 루프를 사용하면 된다.

```
for i:= 1 to 10 do
  A[i]:= 0;
```

배열을 프로시저의 파라미터로 사용하려면, 만들어진 배열을 먼저 사용자 타입으로 만들어야 한다. 예를 들어,

```
type
  TMyArray = Array[1..30] of Integer;
  .....
  .....
var
  A: TMyArray;
```

요소가 N 개 들어있는 배열을 무작위 숫자로 채우는 작업을 해보자. 무작위 숫자의 범위를 정하기 위해 두 개의 텍스트박스에서 숫자를 입력 받는다. 배열 요소의 값은 이 범위 안의 숫자이다. 배열의 값들을 메모에 표시하고 가장 값이 큰 요소를 찾아보자.

이 임무를 완수하는 알고리즘^{algorithm} 전체를 주석으로 적어보자.

```
begin
  //배열 값이 될 수 있는 숫자의 범위를 정한다.
  //명시된 범위 안의 숫자들을 가지고 배열을 채운다.
  //채워진 배열을 TMemo에 표시한다.
  //가장 값이 큰 요소를 찾아서 AMax라고 한다.
  //AMax를 표시한다.
end.
```

위에서 주석으로 기록해 둔 알고리즘 전체를 구문으로 적어보자.

```
begin
  SetArrayRange(rMin, rMax); //배열 값의 범위를 지정한다.
  FillArray(a, rMin, rMax); //명시된 범위 안의 숫자들을 가지고 배열을 채운다.
  OutputArray(a, n); //채워진 배열을 TMemo에 표시한다.
  AMax:=Max(a); //가장 큰 요소를 찾아서 AMax라고 한다.
  lblAMax.Caption:=IntToStr(AMax); //AMax를 표시한다.
end.
```

각 프로시저와 함수들의 내용을 작성하기에 앞서, 배열을 파라미터parameter로 사용하려면 사용자 타입으로 만들어 놓아야 한다는 점을 기억하자.

```

const
    n=20;
type
    Array_n_elements=Array[1..n] of Integer;
procedure SetArrayRange(var ch1, ch2: Integer);
begin
    ch1:=StrToInt(frmArr.edtCh1.Text);
    ch2:=StrToInt(frmArr.edtCh2.Text);
end;
procedure FillArray(var a: Array_n_elements; rMin, rMax: Integer);
var
    i: Integer;
begin
    Randomize;
    for i:=1 to n do
        a[i]:=Random(rMax-rMin+1)+rMin;
    end;
procedure OutputArray(a: Array_n_elements; n: Integer);
var
    i: Integer;
begin
    frmArr.memIsh.Lines.Clear;
    for i:=0 to n-1 do
        frmArr.memIsh.Lines.Append(IntToStr(a[i+1]));
    end;
function Max(a: Array_n_elements): Integer;
var
    i, m: Integer;
begin
    m:=a[1];
    for i:=1 to n do
        if a[i]>m then
            m:=a[i];
    Max:=m;

```

```

end;
procedure TfrmArr.btnClick(Sender: TObject);
var
    rMin, rMax: Integer;
    AMax: Integer;
    A: Array_n_elements;
begin
    SetArrayRange(rMin, rMax); //배열 값의 범위를 지정한다.
    FillArray(a, rMin, rMax); //명시된 범위 안의 숫자들을 가지고 배열을 채운다.
    OutputArray(a, n); //TMemo에 배열을 표시한다.
    AMax:=Max(a); //가장 큰 요소를 찾아서 AMax라고 한다.
    lblAMax.Caption:=IntToStr(AMax); //AMax를 표시한다.
end;

```

실습

Exercise 1.

배열을 하나 만든다. 배열에 들어갈 값의 범위는 텍스트박스로 받는다.

- 배열을 *TMemo*에 표시한다.
- 레이블 몇 개를 만들고 다음 내용을 표시한다.
 - 요소들의 합
 - 요소들의 평균
 - 값이 양수인 요소들의 개수와 음수인 요소들의 개수
 - 값이 가장 큰 요소와 가장 작은 요소
- 배열을 두 개로 쪼갬다 — 하나는 양수 값을 가진 요소들의 배열이고 나머지 하나는 0 또는 음수 값을 가진 요소들의 배열이다. *TMemo*를 2개 만들어서 그 안에 각 배열을 넣는다.