

StringGrid^{스트링그리드} 실습

표 안에 있는 데이터를 자유롭게 다루고 활용할 수 있게 됩니다.

찾고 있는 값이 들어있는 셀이 몇 열 몇 행에 있는지 인덱스번호^{index}를 알아내 보자.

그리드에서 찾고 있는 값이 들어있는 셀의 인덱스번호를 찾는 문제는 인덱스번호 2개 즉, 행과 열을 모두 찾아야 한다는 점만 다른 뿐 *TMemo*나 1차원 배열에서 사용했던 방식으로 풀 수 있다. 그저 첫 번째 인덱스번호를 찾는 루프 안에 두 번째 인덱스번호를 찾는 루프를 하나 더 추가하면 된다.

Exercise 1.

정수 행렬이 있다. 이 행렬의 각 행마다 음수가 몇 개씩 들어 있는지 찾아서 적어보자.

해법

Random 함수를 사용하여 *StringGrid*에 숫자를 채운다. 그리드의 인덱스번호는 0에서부터 시작 한다는 것을 기억하자. 결과는 *TMemo*에 표시하기로 한다.

텔파이 프로그램을 적어보자.

폼 안에 텍스트박스 2개를 두어서 행과 열의 개수를 지정하도록 한다. *StringGrid*와 *TMemo*, 그리고 버튼 2개도 폼에 올려 둔다.

첫 번째 버튼을 클릭하면 *StringGrid*를 만들고 채운다.

```
procedure TForm1.Button1Click(Sender: TObject);
var
    i, j, n, m: Integer;
begin
    n:=StrToInt(edit1.Text);
    m:=StrToInt(edit2.Text);
    sgdEx.RowCount:= n;
    sgdEx.ColCount:= m;
    for i:=0 to n-1 do
        for j:=0 to m-1 do
            sgdEx.Cells[j,i]:= IntToStr(Random(100)-50);
        end;
    end;
```

두 번째 버튼을 클릭하면 각 줄에서 음수인 요소의 개수를 찾아서 *TMemo*에 표시한다.

```
procedure TForm1.Button2Click(Sender: TObject);
var
    i, j, n, m, k: Integer;
begin
    n:=sgdEx.RowCount;
    m:=sgdEx.ColCount;
    for i:=0 to n-1 do
        begin
            k:=0;
            for j:=0 to m-1 do
                if StrToInt(sgdEx.Cells[j, i])<0 then
                    k:=k+1;
            end;
            memol.Lines.Append(IntToStr(k));
        end;
    end;
```

Exercise 2.

음수인 요소들 모두를 음수에서 양수로 교체해보자.

가능한 해법들:

- 음수에 (-1) 을 곱한다.
- 음수의 절대값을 찾아낸다(*Abs* 함수를 사용한다).
- 음수 앞에 단항 연산자 $-$ 를 붙인다.

프로그램 1

```
procedure TForm1.Button2Click(Sender: TObject);
var
  i, j, n, m: Integer;
begin
  n:=sgdEx.RowCount;
  m:=sgdEx.ColCount;
  for i:=0 to n-1 do
    begin
      for j:=0 to m-1 do
        if StrToInt(sgdEx.Cells[j, i])<0 then
          sgdEx.Cells[j, i]:=IntToStr((-1)*StrToInt(sgdEx.Cells[j, i]));
        end
      end;
    end;
  end;
```

프로그램 2

```
procedure TForm1.Button2Click(Sender: TObject);
var
  i, j, n, m: Integer;
begin
  n:= sgdEx.RowCount;
  m:= sgdEx.ColCount;
```

```

for i:= 0 to n-1 do
begin
    for j:= 0 to m-1 do
        sgdEx.Cells[j,i]:= IntToStr(Abs(StrToInt(sgdEx.Cells[j, i])));
    end
end;

```

프로그램 3

```

procedure TForm1.Button2Click(Sender: TObject);
var
    i, j, n, m: Integer;
begin
    n:= sgdEx.RowCount;
    m:= sgdEx.ColCount;
    for i:= 0 to n-1 do
    begin
        for j:= 0 to m-1 do
            if StrToInt(sgdEx.Cells[j, i])<0 then
                sgdEx.Cells[j, i]:= IntToStr(-StrToInt(sgdEx.Cells[j, i]));
        end
    end;
end;

```

Exercise 3.

행과 열의 개수가 같은 배열이 있다. 이 배열이 주 대각선을 따라 대칭 배열인지 아닌지를 알아내 보자.

```

0 1 0 0 1
1 0 0 1 1
0 0 1 1 0
0 1 1 0 0

```

해법

이 규칙을 사용하자: 행렬이 대칭이라면, $i = 1, 2, \dots, n$ 이고 $j = 1, 2, \dots, n$ 일때 $i > j$ 인 모든 경우에 `StringGrid1.Cells[i, j] = StringGrid1.Cells[j, i]` 이어야 한다. 따라서, 프로그램을 이렇게 적으면 된다.

```
procedure TForm1.Button1Click(Sender: TObject);
var
    i, j, n: Integer;
begin
    n:= StrToInt(edit1.Text);
    sgdEx.RowCount:= n;
    sgdEx.ColCount:= n;
    for i:=0 to n-1 do
        for j:=0 to n-1 do
            sgdEx.Cells[j, i]:= IntToStr(Random(2));
        label2.Caption:= '';
    end;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
    i, j, n, m: Integer;
    simm: boolean;
begin
    n:= sgdEx.RowCount;
    simm:= True;
    {일단 이 행렬이 대칭이라고 가정한다}
    i:=1;
    while simm and (i < n) do
    begin
        j:=0;
        while (j < i) and (sgdEx.Cells[j, i]= sgdEx.Cells[i, j]) do
            j:= j+1;
        simm:= (j=i);
        i:= i+1;
    end;
end;
```

```

if simm then
    label2.Caption:= '이 행렬은 대칭입니다' ;
else
    label2.Caption:= '이 행렬은 대칭이 아닙니다' ;
end;

```

Exercise 4.

n개의 열과 m개의 행을 가진 행렬을 다음과 같이 적어보자.

```

12345 ... .. n
2n ... n+2 n+1
2n+1 ... .. 3n

```

해법

이런 식으로 행렬을 채울 수 있도록 규칙을 정해야 한다. 이 경우 규칙은 다음과 같다.

만약 홀수 행이면, $\text{sgdEx.Cells}[j, i] := \text{IntToStr}(i*m + j+1)$ 이다. 그렇지 않으면, (짝수 행이라면) $\text{sgdEx.Cells}[j, i] := \text{IntToStr}((i+1)*m-j)$ 이다.

이 규칙을 프로그램으로 적어보자.

```

procedure TForm1.Button1Click(Sender: TObject);
var
    i, j, n, m: Integer;
begin
    n:=StrToInt(edit1.Text);
    m:=StrToInt(edit2.Text);
    sgdEx.RowCount:= m;
    sgdEx.ColCount:= n;
    for i:=0 to m-1 do
        for j:=0 to n-1 do
            if i mod 2 =0 then
                sgdEx.Cells[j, i]:= IntToStr(i*n + j+1);
            end;
        end;
    end;
end;

```

```

else
    sgdEx.Cells[j, i] := IntToStr((i+1)*n-j);
end;

```

실습

Exercise 1.

행과 열이 모두 n 개인 다음과 같은 구조가 되도록 행렬을 0과 1로 채운다.
(예문은 5×5 배열인 경우):

```

1 1 1 1 1
0 1 1 1 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0

```

Exercise 2.

행과 열이 모두 n 개인 다음과 같은 구조가 되도록 행렬을 0과 1로 채운다.
(예문은 5×5 배열인 경우):

```

0 0 0 0 0
0 0 0 0 0
0 0 1 0 0
0 1 1 1 0
1 1 1 1 1

```

Exercise 3.

특정 범위 안의 무작위 값들로 m 개의 열과 n 개의 행을 가진 행렬을 채운다. 범위의 최대 값과 최소값은 키보드에서 입력 받는다.

각 행에서 최소값을 가진 요소를 찾은 후에, 주 대각선에 있는 요소와 맞바꾼다.

Exercise 4.

특정 범위 안의 무작위 값들로 m 개의 열과 n 개의 행을 가진 행렬을 채운다. 범위의 최대 값과 최소값은 키보드에서 입력 받는다.

각 열에서 홀수 행에 있는 값들을 찾아서 표시한다.

그리고, 홀수 열에 있는 모든 양수 값을 찾아서 표시한다.