

## TMemo<sup>메모</sup> 컨트롤(계속)

TMemo 컨트롤에 있는 문장을 더 자유자재로 다룰 수 있게 됩니다.

유용한 프로퍼티와 메소드를 몇 개를 더 살펴보자.

### Clear<sup>비우기</sup>

TMemo의 내용을 완전히 지운다.

```
memEx.Lines.Clear;
```

### Append<sup>덧붙이기</sup> 와 Add<sup>더하기</sup> 메소드

TMemo가 비어있거나, Clear가 호출된 상태라면, Append<sup>덧붙이기</sup> 메소드를 호출하여 TMemo에 글을 넣을 수 있다. Append 메소드는 TMemo의 맨 뒤에 새 줄을 덧붙인다.

#### 예문

TMemo의 각 줄마다 숫자 하나씩 1~10까지 순서대로 넣는다.

```
...
memEx.Lines.Clear;
for i:= 1 to 10 do
begin
    memEx.Lines.Append(IntToStr(i));
end;
```

**Add**<sup>더하기</sup> 메소드도 *Append*와 하는 일이 같다.

```
memEx.Lines.Add('이 예문은 문자열 목록을 사용한다.');
```

위 예문처럼 이 메소드 2개 모두 파라미터는 문자열이다.

*Add*는 새로 문자열이 추가된 줄의 인덱스번호를 반환한다.

```
b:=memEx1.Lines.Add(edtEx1.Text);
lblEx1.Caption:= IntToStr(b);
```

위 예문을 실행하면 텍스트박스 *edtEx1*에 있는 글이 메모 *memEx1*에 추가된다. 그리고 나서 레이블 *lblEx1*에는 추가된 줄의 번호가 표시된다.

*Clipboard*<sup>클립보드</sup>를 사용하면 *TMemo*에 있는 글의 일부를 제거, 저장, 복구 할 수도 있다.

## Insert<sup>끼워넣기</sup> 메소드

지정된 인덱스번호<sup>index</sup>에 새 줄을 만들고 문자열을 끼워 넣는다. 그 뒤의 줄들은 자동으로 뒤로 밀려난다.

```
memEx.Lines.Insert(2,''); //2번 줄에 빈줄을 하나 끼워넣는다. 원래 있던 2번 줄은 삭제되지 않고 뒤로 밀린다.
```

## 메모의 Lines 줄 소팅<sup>sorting</sup> 하기

줄을 소팅(Sorting, 일정한 순서가 되도록 재배열)하려면 아래의 알고리즘을 쓸 수 있다.

최소값을 가지는 *String*을 찾아낸다. (문자열 비교는 한 문자씩 한다. A<B<C ..<Z ...)

첫 번째 줄과 최소값 줄을 맞바꾼다.

위 구문을 (N-1) 번 반복한다. (N은 TMemo에 있는 줄의 전체 개수)

이것을 프로그램으로 적어보자.

```
procedure TfrmEx1.btnEx1Click(Sender: TObject);
begin
    //TMemo의 각 i 번 줄마다 아래를 반복한다.
    //i 번호 줄에서 시작하여 마지막 줄까지 String을 서로 비교하여 최소값 문자열의 줄 번호 (이하 '최소
    값 줄의 번호')를 찾아낸다.
    //최소 줄 번호를 i 번호 줄과 맞바꾼다.
end;
```

이 프로그램에 필요한 변수, 연산자, 함수(최소값 줄의 번호를 찾아 내는 함수)를 적어보자.

만들어 진 프로그램은 아래와 같다.

```
function PlMin(numStr: Integer): Integer;
var
    k, m: Integer;
begin
    m:=NumStr; //우선, 비교를 시작할 줄을 최소값 줄이라고 하자 (시작하는 줄 번호를 최소값의 줄 번호에
    넣는다).
    for k:=m+1 to frmEx1.memEx1.Lines.Count-1 do //NumStr 번에서 시작하여 그 이하의
    모든 줄마다 반복한다.
        if frmEx1.memEx1.Lines[k]< frmEx1.memEx1.Lines[m] then //만약 현재 줄
        이 최소값 줄의 번호이면
            m:=k; //m 값 (최소값 줄의 번호)을 현재 줄 번호로 바꾼다.
    PlMin:=m; // 찾아 낸 최소값의 줄 번호를 함수의 반환 값에 넣는다.
end;

procedure TfrmEx1.btnEx1Click(Sender: TObject);
```

```

var
    i, j: Integer;
begin
    for i:= 0 to memEx1.Lines.Count-1 do //TMemo 전체 줄 반복
    begin
        j:=PlMin(i); //i 번 줄에서 마지막 줄까지 중에서, 최소값 줄의 번호를 찾아낸다.
        memEx1.Lines.Exchange(i, j); //서로 자리를 맞바꾼다.
    end;
end;

```

## 실습

### Exercise 1.

TMemo의 각 줄마다 숫자 하나씩을 넣는다.

이 숫자들 중에서 0보다 큰 숫자들은 모두 두 번째 TMemo에 넣고, 0보다 작은 숫자들(0 포함)은 모두 세 번째 TMemo에 넣는 프로그램을 만들어보자.

### Exercise 2.

친구들 이름 목록을 TMemo에 넣는다. 이 목록을 정렬하는 프로그램을 만들어보자.

### Exercise 3.

TMemo 2 개에 내용을 넣어둔다.

이 TMemo 2 개에서 내용이 같은 줄이 있는지를 서로 비교하여 찾아낸다. 찾아낸 줄 모두 그 내용을 세 번째 TMemo에 넣는다. 내용이 같은 줄이 전혀 없으면 없다는 메시지를 넣는다.

### Exercise 4.

TMemo의 각 줄마다 숫자 하나씩을 넣는다. 숫자를 오름차순으로 정렬하는 프로그램을 작성해보자.