

Kierunek: **ETI**

Przedmiot: **Programowanie w środowisku RAD - Delphi**

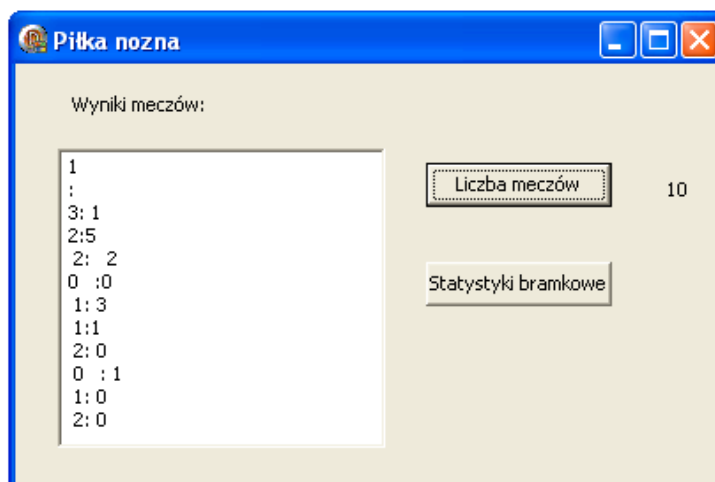
Rok **III** Semestr 5

## Ćwiczenie 6 – Aplikacja wielo-okienkowa cd

Opracuj aplikację realizującą obliczenia na podstawie danych wpisywanych w komponencie **Memo**. Program powinien umożliwiać wprowadzanie i edycję danych w komponencie Memo oraz wyświetlanie wyników obliczeń wykonanych na podstawie tych danych. Dane dotyczą wyników meczów piłkarskich. W kolejnych wierszach komponentu Memo należy wpisywać:

- 1 wiersz: nr kolejki rozgrywek (1 liczba)
- 2 wiersz: znak separatora używanego przy wpisywaniu wyniku meczu (dwukropek)
- kolejne wiersze: wyniki bramkowe poszczególnych meczów (np.: 2:1)

Wyniki obliczeń wykonywanych przez program powinny być wyświetlane na oddzielnych formatkach zgodnie z instrukcją wykonania. Proponowaną postać formatki głównej przedstawia poniższy rysunek:



WYKONANIE:

- Rozmieść na formatce komponent Memo oraz komponenty przycisków i etykiet. Przycisk "Statystyki bramkowe" powinien być wstępnie nieaktywny (Enabled= false).

W sekcji **public** typu **TForm1** wpisz deklarację składowych:

```
tytuł: String;  
N: integer;  
x, y: array[1..20] of real;
```

Składowe będą widoczne we wszystkich metodach typu TForm1.

- Dołącz metodę obsługującą zdarzenie **OnClick** przycisku górnego. Proponowana zawartość procedury:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    i, x1, y1: Integer;
```

```

    separator: string;
begin
{Wycięcie z tekstu wyników kolejnych meczów i wpisanie ich do tablic x i y}
N:=0;
with Memol do
    if Lines.Count>1 then
        try
            tytul:='Wyniki kolejki nr: '+ Lines[0];
            separator:=Lines[1][1];
            for i:= 2 to Lines.Count-1 do
                begin
                    x1:=pos(separator, lines[i])-1;
                    x1:=StrToInt(Trim(copy(Lines[i], 1, x1)));
                    y1:= pos(separator, Lines[i])+1;
                    y1:=StrToInt(Trim(copy(Lines[i], y1, length(Lines[i])-y1+1)));
                    x[i-1]:=x1;
                    y[i-1]:=y1;
                    N:=N+1;
                end;
            Label2.Caption:=IntToStr(N);
            Button1.Enabled:=true;
        except
            MessageDlg('Wpisz poprawne dane!', mtWarning, [mbOk], 0);
            Memol.SetFocus;
        end;
    end;
end;

```

W zapisie procedury wykorzystano następujące funkcje obsługi łańcuchów:

**function** Pos(Substr: string; S: string): Integer;

Funkcja ustala pozycję łańcucha *Substr* w łańcuchu *S*. Zwraca nr znaku łańcucha *S* od którego zaczyna się zapis *Subsstr* lub 0 jeśli łańcucha *Subsstr* nie znaleziono.

**function** Copy(S: string; Index, Count: Integer): string;

Zwraca łańcuch wycięty z łańcucha *S* zawierający *Count* znaków począwszy od znaku *Index*.

**function** Trim(const S: string): string;

Zwraca łańcuch *S* bez początkowych i końcowych spacji.

Zapisana wyżej metoda przepisuje wyniki bramkowe meczów do pól *x* i *y* obiektu *Form1*.

W etykiecie *Label2* wyświetlana jest liczba rozegranych meczów.

- Wypróbuj działanie programu z metodą **Button1Click** zapisaną następująco:

```

procedure TForm1.Button3Click(Sender: TObject);
var i: integer;
begin
N:=0;
with Memol do
    if Lines.Count>1 then
        try
            tytul:='Wyniki kolejki nr: '+ Lines[0];
            Lines.NameValueSeparator:=Lines[1][1];
            for i:= 2 to Lines.Count-1 do
                begin
                    x[i-1]:=StrToInt(Trim(Lines.Names[i]));
                    y[i-1]:=StrToInt(Trim(Lines.ValueFromIndex[i]));
                end;
        except
            MessageDlg('Wpisz poprawne dane!', mtWarning, [mbOk], 0);
            Memol.SetFocus;
        end;
    end;
end;

```

```

        N:=N+1;
    end;
Label2.Caption:=IntToStr(N);
Button1.Enabled:=true;
except
    MessageDlg('Wpisz poprawne dane!', mtWarning, [mbOk], 0);
    Memo1.SetFocus;
end;
end;

```

W tej wersji metoda wykorzystuje bardziej zaawansowane narzędzia komponentu Memo które pozwalają na sprawną obsługę zapisów postaci:

```

Nazwa1=Wartosc1
Nazwa2=Wartosc2

```

Tego typu zapisy można obrabiać wyłuskując z wiersza łańcuch wpisany przed znakiem równości przy pomocy metody **Names[nr]** a napis umieszczony za znakiem równości przy pomocy metody **ValueFromIndex[nr]**. Znak równości jest separatorem domyślnym. Można go zmienić metodą **NameValueSeparator**.

Do dalszego wykorzystania pozostaw wersję metody która wydaje Ci się lepsza.

- Dopisz samodzielnie metodę obsługującą zdarzenie **OnClick** przycisku dolnego Metoda powinna wyświetlać nowe okienko, a w nim ustalony tytuł kolejki oraz informacje o:
  - liczbie bramek strzelonych we wszystkich meczach
  - liczbie bramek strzelonych we wszystkich meczach przez zawodników gospodarzy
  - liczbie bramek strzelonych we wszystkich meczach przez zawodników gości
  - największej liczbie bramek strzelonych w jednym meczu
  - największej różnicy bramek w meczu
- Dodaj na formatce głównej nowy przycisk powodujący wyświetlenie jeszcze jednego okienka. W okienku tym należy wyświetlać następujące informacje:
  - liczba meczów wygranych przez gospodarzy
  - liczba meczów wygranych przez gości
  - średnia liczba bramek strzelonych w meczach zakończonych różnicą bramek większa niż 3
  - wynik meczu w którym padło najwięcej bramek

Okienko z tymi informacjami powinno być niemodalne. Okienko nie powinno być kreowane automatycznie w czasie uruchamiania programu, lecz dopiero przy okazji wyświetlania.

**UWAGA** W związku z zapisanym wyżej wymogiem należy po dołączeniu nowej formatki (Form3) usunąć z pliku projektu linię kreującą okienko. Metoda obsługująca wyświetlanie okienka powinna zawierać instrukcję kreującą okienko np. postaci:

```
Form3:=TForm3.Create(Application)
```

Należy także zadbać o usuwanie obiektu okienka każdorazowo przy jego zamykaniu. W tym celu należy obsłużyć zdarzenie **OnClose** formatki dodając metodę:

```

procedure TForm3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action:=caFree;
  { ustawienie parametru Action równego caFree powoduje, że okienko jest
  usuwane z pamięci przy zamykaniu }
end;

```

- Opracuj nową aplikację która będzie wykonywała te same obliczenia jak program poprzedni lecz do wprowadzania danych będzie wykorzystywała zamiast komponentu Memo dwa komponenty StringGrid.

### Szczegóły wykonania:

Dla każdego z dwu komponentów StringGrid ustaw odpowiednio szerokość 1 kolumny oraz właściwości:

```

      ColCount = 1
      RowCount = 20
      FixedCols = 0
      FixedRows = 0
      Options.goEditing = true

```

Wygeneruj metodę obsługi zdarzenia OnGetEditMask oraz wpisz jej treść:

```
Value := '99';
```

Metoda ustanawia dla wszystkich komórek StringGrida maskę wprowadzania dopuszczającą wpisywanie liczb całkowitych co najwyżej dwucyfrowych.

W sekcji **public** typu **TForm1** wpisz deklarację składowych:

```

N: integer;
x, y: array[1..20] of real;

```

Składowe będą widoczne we wszystkich metodach typu TForm1.

Wygeneruj metodę obsługi zdarzenia OnClick dla przycisku pierwszego, a następnie wpisz treść metody:

```

procedure TForm1.Button1Click(Sender: TObject);
var pom1, pom2:string;
      i: integer;
begin
  i:=0;
repeat
  pom1:=StringGrid1.Cells[0, i]; pom2:=StringGrid2.Cells[0, i];
  if (pom1 <> '') and (pom2 <> '') then
  begin
    i:=i+1;
    x[i]:=StrToFloat(pom1);
    y[i]:=StrToFloat(pom2);
  end;
until ( pom1 = '' ) or ( pom2 = '' ) or ( I = StringGrid1.RowCount );
  N:=i;
  Label2.Caption:=IntToStr(N);
  Button2.Enabled:=N<>0;
end;

```

Samodzielnie rozwiąż pozostałe problemy.