

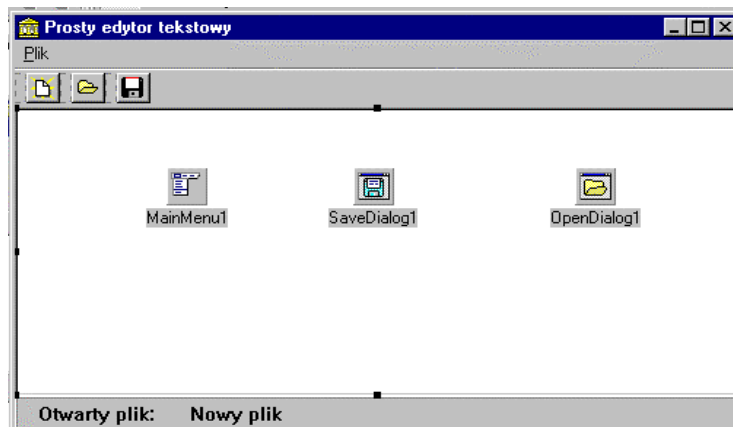
Kierunek: **ETI**

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

Rok **III** Semestr 5

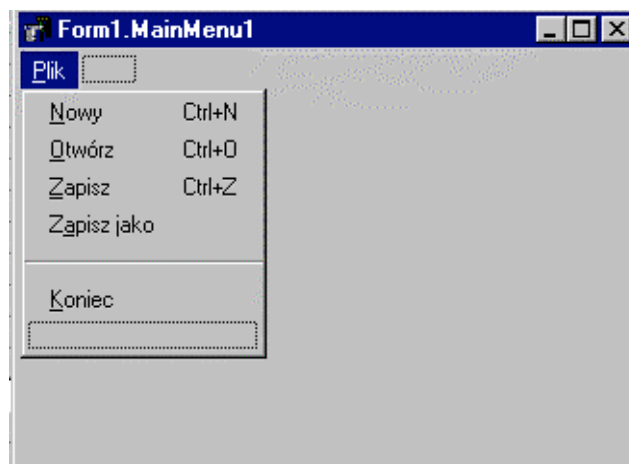
### Ćwiczenie 4 - Edytowanie plików tekstowych

1. Uruchom Windows, a następnie Delphi.
2. Opracuj aplikację realizującą edycję pliku tekstowego. Wygląd okienka aplikacji w fazie projektowania przedstawiono poniżej:



WYKONANIE:

- Zredaguj menu programu wykorzystując edytor menu (**Menu Designer**). Poniżej przedstawiono okienko edytora menu po zdefiniowaniu opcji:



- Dołącz komponenty:
  - **Memo** z palety **Standard**,
  - **OpenDialog** i **SaveDialog** z palety **Dialogs**.
- Ustaw początkową wartość właściwości **Lines** komponentu **Memo** korzystając z okienka **String List Editor** na "" (Łańcuch pusty).

- Dołącz metodę obsługującą opcję menu **Otwórz**. Proponowana zawartość procedury:

```

procedure TForm1.Otworz1Click(Sender: TObject);
begin
if OpenFileDialog1.Execute then
    begin
        Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
        SaveDialog1.FileName := OpenFileDialog1.FileName;
    { Label2.Caption := OpenFileDialog1.FileName;}
    end;
end;

```

UWAGA 1: Wywoływana w procedurze metoda (funkcja o wyniku *Boolean*) **Execute** wyświetla okienko dialogowe pozwalające użytkownikowi na wybranie otwieranego pliku. Potwierdzenie wybranej nazwy pliku klawiszem **OK** spowoduje zakończenie funkcji **Execute** z wartością *True*. Jednocześnie właściwość **FileName** komponentu **OpenDialog** uzyskuje wartość równą nazwie pliku.

UWAGA 2: Stosowany w przykładzie komponent **Memo** przechowuje swoją zawartość tekstową we właściwości **Lines** typu *TString*. Każda wartość typu *TString* posiada m. in. właściwości:

**Strings** - tablica, której elementy zawierają kolejne wiersze tekstu (numeracja od 0),  
**Count** - liczba elementów (wierszy tekstu),

oraz metody:

**procedure LoadFromFile**(Const FileName: String); - ładowanie zawartości pliku o nazwie *FileName*  
**procedure SaveToFile**(Const FileName: String); - zapis zawartości *Lines* do pliku *FileName*

Kolejne wiersze tekstu zapisanego we właściwości **Lines** dostępne są także jako *Lines[1]*, *Lines[2]* itd.

- Dołącz metodę obsługującą opcję menu **Zapisz jako**. Procedura powinna zawierać wyświetlenie okienka dialogowego dla wyboru zmienianej nazwy pliku (metoda **Execute** komponentu **SaveDialog**) oraz zapis do pliku treści wyedytowanych w komponencie **Memo** (metoda **SaveToFile** komponentu **Memo**).

Przykładowa zawartość metody:

```

procedure TForm1.Zapiszjako1Click(Sender: TObject);
begin
if SaveDialog1.Execute then
    begin
        Memo1.Lines.SaveToFile(SaveDialog1.FileName);
        OpenFileDialog1.FileName := SaveDialog1.FileName;
    { Label2.Caption := SaveDialog1.FileName;}
    end;
end;

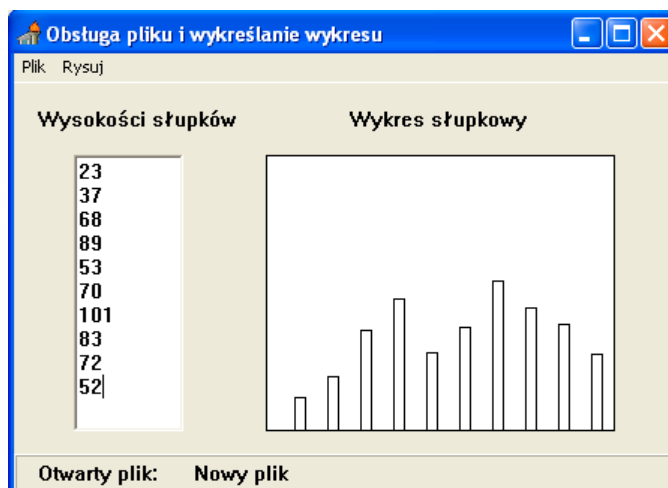
```

- Ustal dla komponentów *OpenDialog1* i *SaveDialog1* wartość właściwości *DefaultExt* równej *txt* oraz właściwości *Filter* jako listy zawierającej filtry: *Pliki tekstowe (\*.txt)* i *Wszystkie pliki (\*.\*)*.
- Dołącz metodę obsługującą opcję menu **Zapisz**. Procedura powinna zawierać jedynie wywołanie metody zapisującej do pliku treść wpisaną w komponencie **Memo** lub, gdy żaden plik nie jest jeszcze otwarty (nie ustalono nazwy wpisywanej we właściwości **FileName** komponentu **SaveDialog**) wywołanie metody odpowiedzialnej za obsługę odpowiedniego okienka dialogowego.  
Dołącz metodę obsługującą opcję menu **Nowy**.
- Dołącz pasek stanu. W panelu rozmieszczonym na dole ekranu wyświetlaj nazwę edytowanego pliku. W przypadku edycji nowego pliku wyświetlaj napis: Nowy plik.
- Dołącz pasek narzędziowy. Rozmieść na komponencie **Panel** komponenty **SpeedButton** wyposażając je w odpowiednie obrazki ikon (właściwość **Glyph**). Odpowiednie pliki bitmapowe można znaleźć w zasobach Delphi przeszukując katalogi zgodnie z bieżącą instalacją pakietu. Podłącz do zdarzeń **OnClick** przycisków odpowiednie metody obsługi. Pasek narzędziowy możesz także zorganizować wykorzystując komponenty *ToolBar* i *ImageList*
- Przetestuj działanie programu.

## ZADANIE DOMOWE

Opracuj aplikację realizującą wyświetlanie wykresu słupkowego. Program powinien umożliwiać edycję i zapis pliku tekstowego oraz wyświetlanie w okienku graficznym prostokątnych słupków o wysokości wyznaczonej wartościami liczbowymi zapisanymi w pliku i wyświetlanymi w komponencie Memo.

Proponowaną postać ekranu przedstawia poniższy rysunek:



Wykonanie zadania:

- Skopiuj do nowego foldera projekt edytora opracowany w ćwiczeniu, a następnie dostosuj i uzupełnij wygląd formatki.

- Dołącz komponent **Image** z palety **Additional**,
- Dołącz metodę obsługującą zdarzenie **OnCreate** komponentu **Form**. W metodzie ustal kolor tła komponentu **Image** oraz narysuj ramkę wokół komponentu,

UWAGA: Każdy komponent typu **TImage** posiada m.in. właściwości:

- **Width** - szerokość komponentu,
- **Height** - wysokość komponentu,
- **Canvas** - zawartość komponentu. Właściwość typu obiektowego **TCanvas** grupująca właściwości m.in.:
  - **Brush** - właściwość typu obiektowego **TBrush** grupująca cechy tła komponentu m.in. cechę **Color**,
  - **Pen** - właściwość typu obiektowego **TPen** grupująca cechy linii kreślonych na komponentcie,

oraz metody m.in.:

- procedure **MoveTo**(X, Y: Integer); - umieszczenie kursora graficznego w punkcie X, Y
- procedure **LineTo**(X, Y: Integer); - wykreślenie linii do punktu X, Y.

- Dołącz metodę obsługującą opcję menu **Rysuj**. Proponowana zawartość procedury:

```

procedure TForm1.Rysuj1Click(Sender: TObject);
var i, krok, wysokosc, kod: integer;
begin
if Memo1.Lines.Count>0 then
  begin
    krok:=(Image1.Width-10) div Memo1.Lines.Count;
    for i:=1 to Memo1.Lines.Count do
      begin
        val(Memo1.Lines[i-1], wysokosc, kod);
        if kod=0 then
          begin
            Image1.Canvas.MoveTo(i*krok-3, Image1.Height);
            Image1.Canvas.LineTo(i*krok-3, Image1.Height - wysokosc);
            Image1.Canvas.LineTo(i*krok+4, Image1.Height - wysokosc);
            Image1.Canvas.LineTo(i*krok+4, Image1.Height);
          end;
        end;
      end;
    end;
  end;

```

- Uzupełnij program i przetestuj jego działanie. Uzupełnij listwę narzędziową o przyciski rysowania i kasowania wykresu oraz przycisk rysujący linię łączącą wierzchołki słupków.