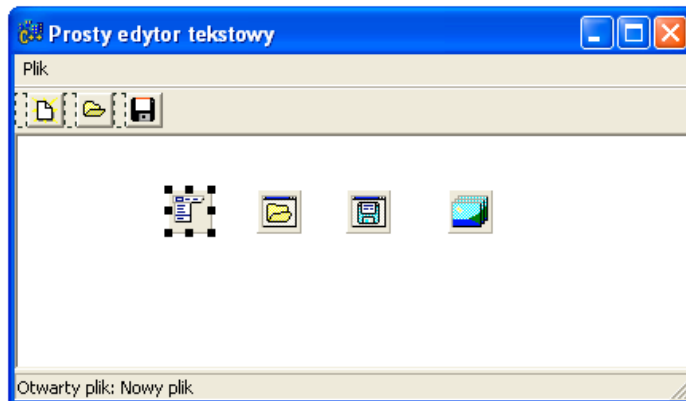


Wydział: **Zarządzania i Modelowania Komputerowego**
Kierunek: **Inżynieria Danych**
Przedmiot: **Programowanie w środowisku RAD-C++**
Rok 2 Semestr 3

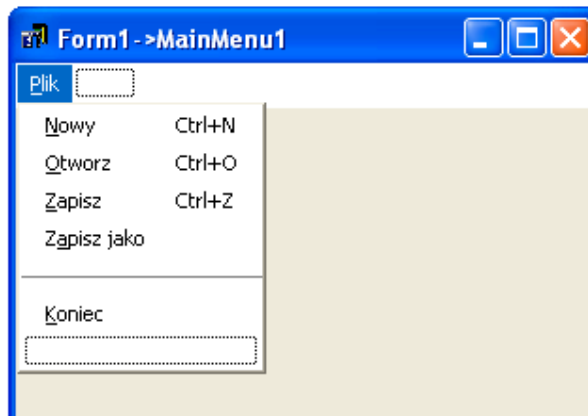
Ćwiczenie 9 - Edytowanie plików tekstowych

Opracuj aplikację realizującą edycję pliku tekstowego. Wygląd okienka aplikacji w fazie projektowania przedstawiono poniżej:



WYKONANIE:

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



- Dołącz komponenty:
 - **Memo** z palety **Standard**,
 - **OpenDialog** i **SaveDialog** z palety **Dialogs**.
- Komponent Memo rozmieść w na całej powierzchni formatki przez odpowiedni wybór właściwości Align. 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:

```
void __fastcall TForm1::Otworz1Click(TObject *Sender)
{
    if (OpenDialog1->Execute())
        Memo1->Lines->LoadFromFile(OpenDialog1->FileName);
        SaveDialog1->FileName = OpenDialog1->FileName;
    // StatusBar1->SimpleText = "Otwarty plik :   " + OpenDialog1->FileName;
}
}
```

UWAGA 1: Wywoływana w procedurze metoda (funkcja o wyniku *Bool*) **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:

- virtual void __fastcall **LoadFromFile**(const AnsiString FileName); - ładowanie zawartości pliku o nazwie *FileName*
- virtual void __fastcall **SaveToFile**(const AnsiString FileName); - zapis zawartości *Lines* do pliku o nazwie *FileName*

- 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:

```
void __fastcall TForm1::ZapiszJako1Click(TObject *Sender)
{
    if (SaveDialog1->Execute())
    {
        Memo1->Lines->SaveToFile(SaveDialog1->FileName);
        OpenDialog1->FileName = SaveDialog1->FileName;
    // StatusBar1->SimpleText = "Otwarty plik :   " + SaveDialog1->FileName;
    }
}
```

- 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. Powinien zawierać powtórzenie działań Nowy, Otwórz i Zapisz.
- Przetestuj działanie programu. Zauważ, że program ma jeszcze szereg usterek. Na przykład na życzenie użytkownika usuwa nawet duże zmiany dokonane w edytowanym tekście wyświetlając bez dodatkowego pytania nowy pusty plik lub otwarty plik inny. Dokonaj ulepszeń w programie poprzez wyświetlanie dodatkowego pytania w przypadku wyboru przez użytkownika wyświetlania nowego, pustego pliku lub otwierania istniejącego pliku gdy w poprzednio wyświetlanym tekście dokonano jakiś zmian. Wykorzystaj właściwość Memo o nazwie Modified która przyjmuje wartość True jeśli w tekście Lines wprowadzono zmiany. Zastosuj pytanie:

```

if (Memo1->Modified)
    { if (!(MessageDlg("Czy porzucic zmiany?", mtInformation,
                      TMsgDlgButtons() << mbYes << mbNo, 0) == mrYes))
      return;
    }

```

Pytanie powinno być wyświetlane jeśli użytkownik próbuje usunąć tekst w którym dokonano zmian.

Przeczytaj w Helpie C-Buidera opis funkcji MessageDlg. Pamiętaj, że twój program powinien uaktualniać właściwość Modified jeśli tekst z Memo jest zapisywany.

ZADANIE DOMOWE

Opracuj, poprzez dostosowanie poprzednio realizowanej aplikacji, program obliczający sumę i sumę kwadratów liczb zapisywanych w komponencie Memo.