

MEdytor 1.1

(dokument jeszcze w trakcie opracowania)

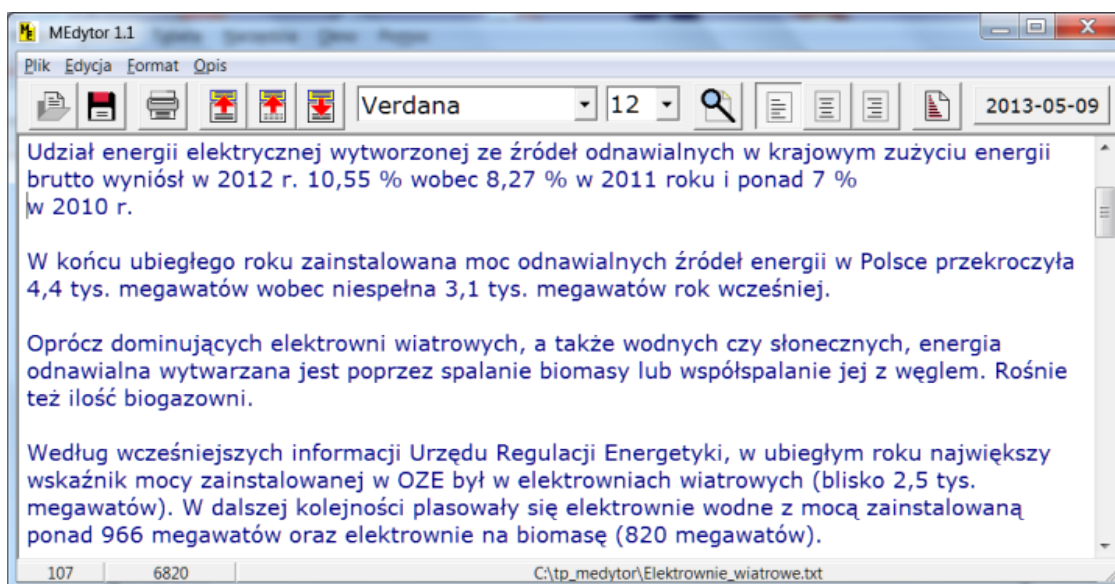
Opis tworzenia programu

Na przykładzie środowiska programowania Delphi przedstawiono proces tworzenia prostej aplikacji przeznaczonej do redagowania tekstu tzw. edytora, a właściwie mini edytora, tu w skrócie o nazwie MEdytor. W programowaniu wykorzystano podstawowe narzędzia dostępne w tym środowisku. Funkcjonuje ono w oparciu o model programowania obiektowego.

W aplikacji wykorzystano gotowe komponenty wizualne należące do tzw. biblioteki VCL (ang. *Visual Component Library*) i komponenty niewizualne, które podczas projektowania aplikacji są zaznaczane na formatkach w postaci ikon.

Proces tworzenia aplikacji obejmuje pięć etapów:

1. Planowanie programu.
2. Projektowanie interfejsu.
3. Definiowanie właściwości i zdarzeń oraz określanie opcji.
4. Testowanie programu.
5. Tworzenie pliku wykonywalnego (ang. *Executable file*).



Rys. 1. Ekran główny aplikacji MEdytor

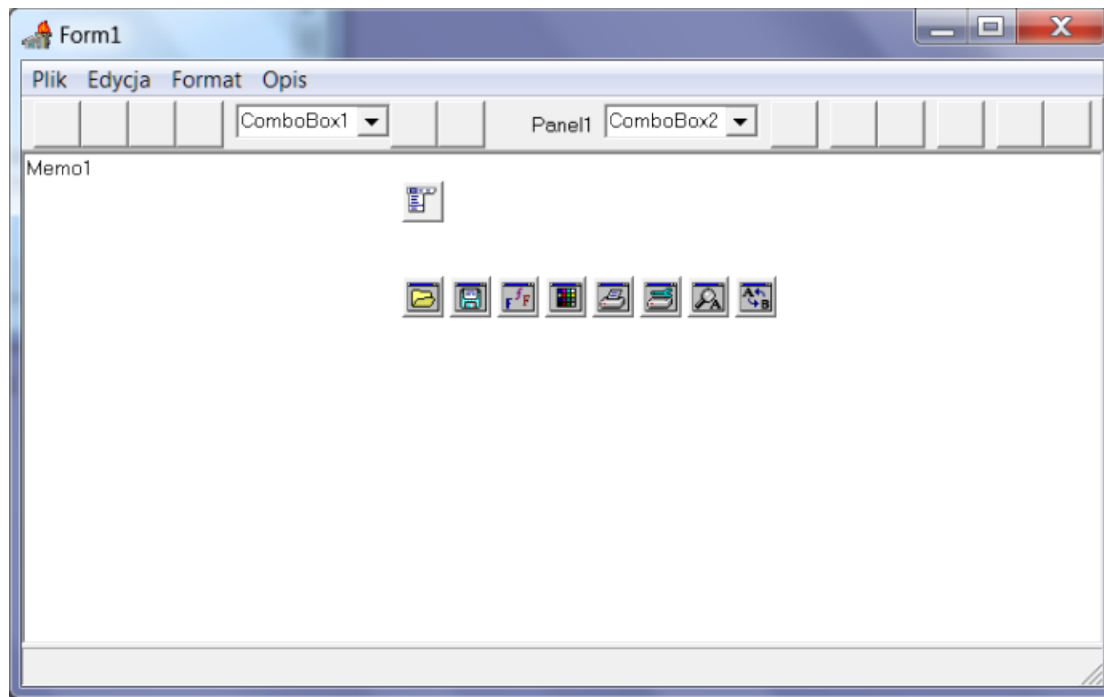
Planowanie programu

1. Aplikacja ma obsługiwać podstawowe funkcje programu do redagowania tekstu zapisanego w kodach rozszerzonego ASCII (ang. *American Standard Code for Interchange Information*) i i operacje odczytu oraz zapisu plików tekstowych, a także ich wydruku.
2. W centralnym polu będzie rozmieszczany redagowany tekst z opcjami wyrównywania do krawędzi lewej (wstępne ustawienie), prawej albo wypośrodkowania zapisów.
3. W ramach wybranych i zaznaczonych fragmentów zapisu będą dostępne operacje kopiowania, wycinania i doklejania fragmentów tekstu poprzez bufor pamięci tzw. schowek.
4. Wszystkie opcje będą dostępne z menu rozwijanego, a niektóre – częściej używane – będą dostępne za pomocą przycisków szybkiego reagowania.
5. Na pasku stanu, przy dolnym brzegu aplikacji, będą podawane informacje o wielkości zapisu (liczba wierszy i znaków) oraz bieżącej nazwie pliku.
6. Należy przewidzieć opcję sortowania wierszy w polu tekstu.
7. Zapis programu powinien mieć charakter dydaktyczny i opierać się na najprostszych rozwiązaniach w środowisku Delphi oraz dostarczać jedynie podstaw do ew. rozwinięcia w innych aplikacjach.

Projektowanie interfejsu

Jak pokazuje rysunek 1 na podstawowej formatce (*Form1*) rozmieszczono szereg komponentów.

Z zakładki Standard dodano menu główne (*MainMenu1*), panel (*Panel1*) na którym umieszczono dwa pola złożone (*ComboBox1* i *ComboBox2*). Z tej samej zakładki wprowadzono na formatkę główny komponent aplikacji tj. pole do redagowania tekstu (*Memo1*).

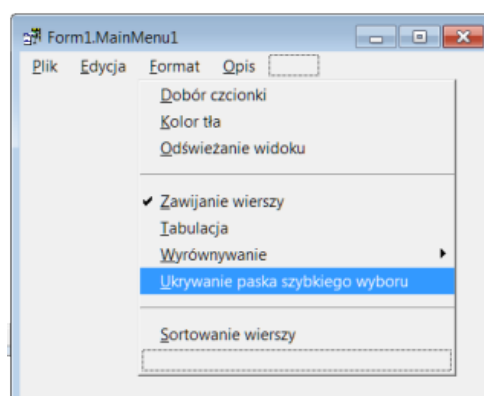


Rys. 2. Wstępne rozmieszczenie komponentów

Z zakładki *Additional* wybrano klawisze szybkiego wyboru (od *SpeedButton1* do *SpeedButton12*), a z zakładki *Win32* (w starszych wersjach *Win95*) wprowadzono pasek stanu (*StatusBar1*).

Ponadto z zakładki *Dialogs* zostały wykorzystane komponenty: *OpenDialog1*, *SaveDialog1*, *PrinterSetupDialog1*, *PrintDialog1*, *FindDialog1*, *ReplaceDialog1*, *FontDialog1* i *ColorDialog1*. Komponenty te są na formacie zaznaczone symbolicznie przez osiem ikon zestawionych w jednym rzędzie na rysunku 2.

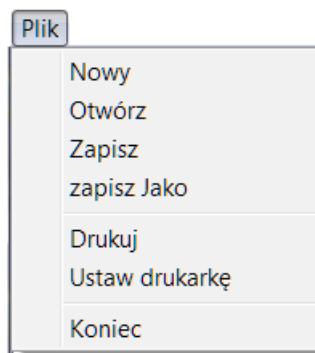
Po dwukrotnym szybkim pstryknięciu myszką na ikonie menu głównego (*MainMenu1*) otwiera się okno edytora (rysunek 3), za pomocą którego są wpisywane opcje menu.



Rys. 3. Okno edytora głównego menu

W programie utworzono cztery grupy opcji: *Plik*, *Edycja*, *Format* i *Opis*. Duże litery pozwalają szybciej rozwinąć wybraną grupę po naciśnięciu klawiszy odpowiednio *Alt+P*, albo *Alt+E*, albo *Alt+F* i podobnie *Alt+O*.

Rysunek 4 przedstawia rozwinięte menu *Plik*. Poszczególne opcje także można wybierać przez naciśnięcie klawisza *Alt* i odpowiedniej dużej litery.



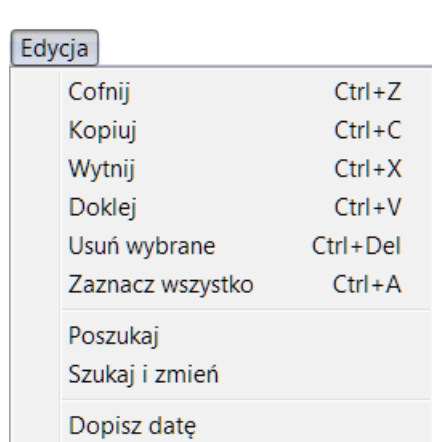
Rys. 4. Menu Plik

Druga grupa opcji *Edycja* (rysunek 5) zawiera dodatkowo możliwość szybkiego wyboru przez klawisze Ctrl i podaną literę, zgodnie ze standardami wyboru stosowanymi w systemie Windows.

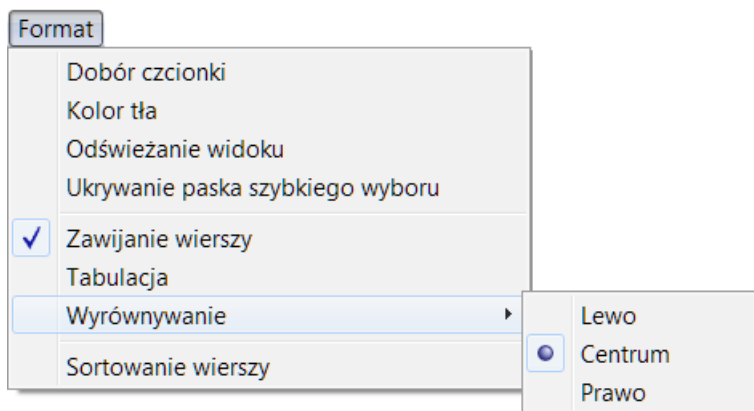
W grupie opcji *Format* zestawiono możliwości zmiany wyglądu pola redagowania tekstu (rysunek 6), a także dołączono możliwość porządkowania narastającego wierszy (sortowania). Na tym samym rysunku pokazano grupę opcji podmenu *Wyrównywanie*. Grupa ta jest zorganizowana jako współzależna (ang. *RadioGroup*). Aktualnie aktywną opcję oznacza niewielkie kółko (tu: *Centrum*).

Niektóre z opcji są przełącznikami dwustanowymi włączone-wyłączone, jak np. opcje *Zawijanie wierszy*, *Ukrywanie paska szybkiego wyboru* lub *Tabulacja*.

Stan włączenia jest zaznaczany w menu jak na rysunku 6 przy opcji *Zawijanie wierszy*.

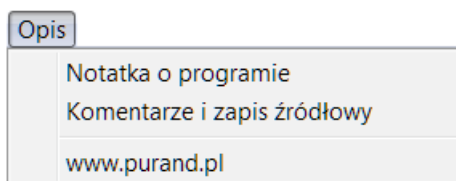


Rys. 5. Rozwinięte menu Edycja



Rys. 6. Grupa opcji menu Format

Ostatnia grupa opcji jest związana z przedstawieniem i opisem samego programu. Dodano także łącze do witryny internetowej o adresie www.purand.pl, z której program można pobrać.



Rys. 7. Menu z opisem programu

Wybranie opcji *Komentarze i zapis źródłowy* otwiera ten opis zachowany w osobnym pliku w formacie pdf.

Przy dolnej ramie okna aplikacji znajduje się pasek stanu (*StatusBar1*). Na rysunku 2 pasek ten nie jest jeszcze podzielony na trzy panele, na których są pokazywane liczby wierszy i znaków w polu *Memo1* oraz nazwa pliku. Podział ten jest widoczny na rysunku 1.

Definiowanie właściwości i zdarzeń oraz określanie opcji

(w przygotowaniu)

Zapis źródłowy programu

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Printers, ComCtrls, Buttons, ExtCtrls, ShellAPI, Menus;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    Plik1: TMenuItem;
    Edycja1: TMenuItem;
    Nowy1: TMenuItem;
    Otworz1: TMenuItem;
    Zapisz1: TMenuItem;
    N1: TMenuItem;
    Koniec1: TMenuItem;
    Kopiuuj1: TMenuItem;
    Wytnij1: TMenuItem;
    Doklej1: TMenuItem;
    UsunWybrane1: TMenuItem;
    Format1: TMenuItem;
    Panel1: TPanel;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    StatusBar1: TStatusBar;
    Memo1: TMemo;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    FontDialog1: TFontDialog;
    OpenFileDialog1: TOpenDialog;
    ZawijanieWierszy1: TMenuItem;
    SaveDialog1: TSaveDialog;
    N2: TMenuItem;
    Drukuj1: TMenuItem;
```

```
UstawDrukarke1: TMenuItem;
ZaznaczWszystko1: TMenuItem;
Tabulacja1: TMenuItem;
Wyrownywanie1: TMenuItem;
SpeedButton9: TSpeedButton;
SpeedButton10: TSpeedButton;
zapiszJako1: TMenuItem;
Lewo1: TMenuItem;
Centrum1: TMenuItem;
Prawo1: TMenuItem;
PrinterSetupDialog1: TPrinterSetupDialog;
PrintDialog1: TPrintDialog;
ColorDialog1: TColorDialog;
Cofnij1: TMenuItem;
N3: TMenuItem;
Poszukaj1: TMenuItem;
FindDialog1: TFindDialog;
ReplaceDialog1: TReplaceDialog;
SzukajZmien1: TMenuItem;
SpeedButton1: TSpeedButton;
SpeedButton11: TSpeedButton;
SpeedButton12: TSpeedButton;
DoborCzcionki1: TMenuItem;
KolorTla1: TMenuItem;
OdnawianieWidoku1: TMenuItem;
N4: TMenuItem;
N5: TMenuItem;
SortowanieWierszy1: TMenuItem;
Opis1: TMenuItem;
NotaProgramu1: TMenuItem;
purand1: TMenuItem;
Komentarze1: TMenuItem;
N6: TMenuItem;
N7: TMenuItem;
NotujDate1: TMenuItem;
UkrywaniePaska1: TMenuItem;
procedure FormCreate(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure Memo1Change(Sender: TObject);
procedure OdswiezZapis1Click(Sender: TObject);
procedure Czcionka1Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure Otworz1Click(Sender: TObject);
procedure ZawijanieWierszy1Click(Sender: TObject);
procedure Zapisz1Click(Sender: TObject);
procedure Nowy1Click(Sender: TObject);
procedure zapiszJako1Click(Sender: TObject);
procedure UstawDrukarke1Click(Sender: TObject);
procedure Drukuj1Click(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure Kopiuj1Click(Sender: TObject);
procedure Wytnij1Click(Sender: TObject);
procedure Doklej1Click(Sender: TObject);
procedure UsunWybrane1Click(Sender: TObject);
procedure ZaznaczWszystko1Click(Sender: TObject);
procedure KolorTla1Click(Sender: TObject);
procedure Prawo1Click(Sender: TObject);
procedure Lewo1Click(Sender: TObject);
procedure Centrum1Click(Sender: TObject);
```

```

procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton7Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure Cofnij1Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure ComboBox1Exit(Sender: TObject);
procedure Tabulacja1Click(Sender: TObject);
procedure Poszukaj1Click(Sender: TObject);
procedure FindDialog1Find(Sender: TObject);
procedure FormResize(Sender: TObject);
procedure SzukajZmien1Click(Sender: TObject);
procedure ReplaceDialog1Find(Sender: TObject);
procedure ReplaceDialog1Replace(Sender: TObject);
procedure Koniec1Click(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure SortowanieWierszy1Click(Sender: TObject);
procedure SpeedButton12Click(Sender: TObject);
procedure purand1Click(Sender: TObject);
procedure NotaProgramu1Click(Sender: TObject);
procedure Komentarze1Click(Sender: TObject);
procedure UkrywaniePaska1Click(Sender: TObject);
procedure NotujDate1Click(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure ComboBox2Change(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

```

```

var
  Form1: TForm1;

```

implementation

```

{$R *.DFM}
//-----
procedure TForm1.FindDialog1Find(Sender: TObject);
var
  iFoundStart: longint; //początek znalezionej zapisu
  iSearchStart: longint; //start pierwszego i kolejnych poszukiwań
begin
  with Memo1 do begin
    if SelLength>0 then iSearchStart:=SelStart+SelLength
    else iSearchStart:=0;

    //Opcja: Uwzględnij wielkość liter
    if FindDialog1.Options*[frMatchCase]=[frMatchCase] then
      iFoundStart:=Pos(FindDialog1.FindText, copy(Text,iSearchStart,Length(Text)))-1
    else
      iFoundStart:=Pos(UpperCase(FindDialog1.FindText),
        copy(UpperCase(Text),iSearchStart,Length(Text)))-1;

    if iFoundStart<0 then

```

```

begin
  ShowMessage('Koniec szukania!');
  Exit
end;
if iFoundStart>=0 then
begin
  SetFocus;
  if iSearchStart=0 then SelStart:=iFoundStart
  else SelStart:=iFoundStart+iSearchStart-1;
  SelLength:=Length(FindDialog1.FindText)
end;
end;
end;
//-----
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Koniec1Click(Sender)
end;
//-----
procedure TForm1.FormCreate(Sender: TObject);
var
  i: byte;
begin
  Application.HintColor:=clWhite; //tło wyskakujących komentarzy

  ComboBox1.Clear; //oczyszczenie komponentu z nazwami czcionek
  ComboBox1.Items:=Screen.Fonts; //wprowadzenie dostępnych nazw czcionek
  ComboBox1.ItemIndex:=ComboBox1.Items.IndexOf('Verdana'); //ustawienie Verdany
  ComboBox2.Clear; //oczyszczenie komponentu z rozmiarami czcionek
  for i:=6 to 39 do //zakres zmian od 6 do 2*39-14=64
    if i<14 then ComboBox2.Items.Add(IntToStr(i))
    else ComboBox2.Items.Add(IntToStr(i*2-14)); //wpisanie rozmiarów
  ComboBox2.ItemIndex:=ComboBox2.Items.IndexOf('12'); //ustawienie rozmiaru 12
  Memo1.Font.Name:= ComboBox1.Items[ComboBox1.ItemIndex]; //ustawienie indeksu
  Memo1.Font.Size:= StrToInt(ComboBox2.Items[ComboBox2.ItemIndex]); //jw.
  Memo1.Color:=clWhite; //kolor tła
  Memo1.Font.Color:=clNavy; //kolor czcionki
  Memo1.Font.Style:=[]; //styl czcionki normalny
  OpenFileDialog1.FileName:='notatka.txt'; //wstępna nazwa pliku
  SaveDialog1.FileName:=OpenDialog1.FileName; //nazwa pliku do zapisu
  Caption:=' MEdytor 1.1 '; //nazwa programu na pasku tytułu

  Memo1.ScrollBars:=ssVertical; //tylko pionowy pasek przesuwu
  SpeedButton8.Caption:=DateToStr(Now); //na przycisku data
end;
//-----
procedure TForm1.FormResize(Sender: TObject);
//Wyznacza liczbę wierszy po zmianie rozmiarów formatki i pola Memo
begin
  StatusBar1.Panels[0].Text:=IntToStr(Memo1.Lines.Count)
end;
//-----
procedure TForm1.FormShow(Sender: TObject);
begin
  Memo1.SetFocus; //skupienie działań w polu Memo1
end;
//-----

```



```

procedure TForm1.ComboBox2Change(Sender: TObject);
begin
    Memo1.Font.Size:=StrToInt(ComboBox2.Items[ComboBox2.ItemIndex]);
end;
//-----
procedure TForm1.Centrum1Click(Sender: TObject);
begin
    Memo1.Alignment:=taCenter;
    SpeedButton9.Down:=true;
    Centrum1.Checked:=true
end;
//-----
procedure TForm1.ComboBox1Change(Sender: TObject);
begin
    Memo1.Font.Name:=ComboBox1.Items[ComboBox1.ItemIndex];
end;
//-----
procedure TForm1.ComboBox1Exit(Sender: TObject);
begin

end;
//-----
procedure TForm1.Memo1Change(Sender: TObject);
begin
    Memo1.Modified:=true;
    StatusBar1.Panels[0].Text:= IntToStr(Memo1.Lines.Count);
    StatusBar1.Panels[1].Text:= IntToStr(Length(Memo1.Text))
end;
//-----
procedure TForm1.OdswiezZapis1Click(Sender: TObject);
begin
    Memo1.Refresh
end;
//-----
procedure TForm1.Czcionka1Click(Sender: TObject);
begin
    if FontDialog1.Execute then Memo1.Font:=FontDialog1.Font;
    ComboBox1.ItemIndex:=ComboBox1.Items.IndexOf(FontDialog1.Font.Name);
    ComboBox2.ItemIndex:=ComboBox2.Items.IndexOf(IntToStr(FontDialog1.Font.Size));
end;
//-----
procedure TForm1.SpeedButton8Click(Sender: TObject);
//Wprowadza do zapisu aktualną datę z systemu
begin
    SpeedButton8.Caption:=DateToStr(Now);
    Memo1.Lines.Add(SpeedButton8.Caption)
end;
//-----
procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
    Centrum1Click(Sender);
    SpeedButton9.Down:=true
end;
//-----
procedure TForm1.SzukajZmien1Click(Sender: TObject);
begin
    ReplaceDialog1.Execute
end;
//-----

```

```

procedure TForm1.Poszukaj1Click(Sender: TObject);
begin
    FindDialog1.Execute
end;
//-----
procedure TForm1.ReplaceDialog1Find(Sender: TObject);
begin
    FindDialog1.FindText:=ReplaceDialog1.FindText;
    FindDialog1.Options:=ReplaceDialog1.Options;
    ReplaceDialog1.Options:=ReplaceDialog1.Options-[frReplaceAll];
    FindDialog1Find(Sender)
end;
//-----
procedure TForm1.ReplaceDialog1Replace(Sender: TObject);
begin
    if (Memo1.SelLength=0) or (Memo1.SelText<>ReplaceDialog1.FindText) then
        if Memo1.SelLength=0 then Exit; //koniec procedury gdy nic nie znaleziono

    if (ReplaceDialog1.Options*[frReplace])=[frReplace] then
        begin //zamiana tylko raz
            Memo1.SelText:=ReplaceDialog1.ReplaceText;
            Memo1.SelLength:=Length(ReplaceDialog1.ReplaceText);
            Memo1.SetFocus
        end
    else
        if (ReplaceDialog1.Options*[frReplaceAll])=[frReplaceAll]) then
            begin //zamiana wszystkich znalezionych zapisów
                Memo1.SelText:=ReplaceDialog1.ReplaceText;
                Memo1.SelLength:=Length(ReplaceDialog1.ReplaceText);
                Memo1.SelStart:=Memo1.SelStart+Memo1.SelLength; //nowy początek szukania
                ReplaceDialog1Find(Sender); // kolejne szukanie
                ReplaceDialog1Replace(Sender) //samouruchomienie procedury zamiany
            end;
        end;
end;
//-----
procedure TForm1.Tabulacja1Click(Sender: TObject);
begin
    Tabulacja1.Checked:= not Tabulacja1.Checked;
    Memo1.WantTabs:= Tabulacja1.Checked;
end;
//-----
procedure TForm1.Otworz1Click(Sender: TObject);
begin
    try
        if OpenFileDialog1.Execute then begin
            Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
            StatusBar1.Panels[2].Text:=OpenDialog1.FileName;
            SaveDialog1.FileName:=OpenDialog1.FileName;
            Memo1.Modified:=false //plik odczytany jest już zapisany
        end;
    except
        ShowMessage('Błąd otwierania pliku!')
    end;
end;
//-----

```

```

procedure TForm1.ZawijanieWierszy1Click(Sender: TObject);
begin
  ZawijanieWierszy1.Checked := not ZawijanieWierszy1.Checked;
  Memo1.WordWrap:=ZawijanieWierszy1.Checked;
  if Memo1.WordWrap then Memo1.ScrollBars:=ssVertical
  else Memo1.ScrollBars:=ssBoth;

end;
//-----
procedure TForm1.Zapisz1Click(Sender: TObject);
begin
try
  Memo1.Lines.SaveToFile(SaveDialog1.FileName);
  Memo1.Modified:=false;
except
  ShowMessage('Błąd zapisu pliku!')
end;
end;
//-----
procedure Abovo; //Pisanie od nowa
begin
  Form1.Memo1.Clear; //usuwa wszystkie zapisy
  //Wprowadza standardową nazwę pliku
  Form1.OpenDialog1.FileName:='notatka.txt';
  Form1.SaveDialog1.FileName:=Form1.OpenDialog1.FileName;
  Form1.StatusBar1.Panels[2].Text:=Form1.OpenDialog1.FileName;

  Form1.Memo1.ScrollBars:=ssVertical; //tylko przesuw w pionie
  Form1.Memo1.Modified:=false; //nie trzeba zapisywać

  //Zerowanie paneli na pasku stanu
  Form1.StatusBar1.Panels[0].Text:='0';
  Form1.StatusBar1.Panels[1].Text:='0';
end;
//-----
procedure TForm1.Nowy1Click(Sender: TObject);
begin
  if Memo1.Modified then //gdy były zmiany w zapisach
    Case Application.MessageBox('Usuwa wszystkie zapisy?','Nowy zapis',
      MB_OkCancel+MB_DefButton2+MB_IconQuestion) of
      IdOk: Abovo;
      IdCancel: Abort //wybór bez konsekwencji
    End
  else Abovo
end;
//-----
procedure TForm1.zapiszJako1Click(Sender: TObject);
begin
try
  if SaveDialog1.Execute then begin
    Memo1.Lines.SaveToFile(SaveDialog1.FileName);
    StatusBar1.Panels[2].Text:=SaveDialog1.FileName;
    Memo1.Modified:=false;
  end;
except
  ShowMessage('Błąd zapisu pliku!')
end;
end;
//-----

```

```

procedure TForm1.UkrywaniePaska1Click(Sender: TObject);
begin
    UkrywaniePaska1.Checked:= not UkrywaniePaska1.Checked;
    if UkrywaniePaska1.Checked then Panel1.Visible:=false
    else Panel1.Visible:=true
end;
//-----
procedure TForm1.UstawDrukarke1Click(Sender: TObject);
begin
    PrinterSetupDialog1.Execute;
end;
//-----
procedure TForm1.Drukuj1Click(Sender: TObject);
var
    i: integer; //nr wiersza
    tekst: TextFile; //zmienna pliku
begin
    try
        if PrintDialog1.Execute then begin
            AssignPrn(tekst); //przypisuje drukarkę plikowi
            Rewrite(tekst); //otwarcie drukarki
            Printer.Canvas.Font:=Memo1.Font; //taka sama czcionka
            for i:=0 to Memo1.Lines.Count-1 do
                writeln(tekst, Memo1.Lines[i]);
            writeln(tekst, #12); //FormFeed, wysuwa kartkę
            CloseFile(tekst)
        end;
    except
        ShowMessage('Błąd wydruku!')
    end;
end;
//-----
procedure TForm1.SortowanieWierszy1Click(Sender: TObject);
var
    TekstLista: TStringList; //lista z wierszy Memo
    i: integer; //zmienna sterująca
begin
    if Memo1.Lines.Count>1000 then begin //przyjęte ograniczenie
        ShowMessage('Liczba wierszy sortowanych'+#13#10+
            'jest ograniczona do 1000!');
        exit
    end;
    //Ostrzeżenie o możliwości zmian nieodwracalnych
    case Application.MessageBox('Posortowane wiersze zastąpią dotychczasowe!'+#10#13+
        'Czy pomimo to wykonać operację?', 'Akceptacja sortowania',
        MB_YESNO+MB_IconExclamation) of
        IDNO: Exit
    end;

    TekstLista:=TStringList.Create; //utworzenie obiektu
    for i:= 0 to Memo1.Lines.Count - 1 do
        TekstLista.Add(Memo1.Lines[i]); //wiersze Memo do TekstListy
    TekstLista.Sort; //sortowanie listy
    Memo1.Clear; //oczyszczenie Memo
    for i:= 0 to TekstLista.Count - 1 do
        Memo1.Lines.Add(TekstLista.Strings[i]); //lista do Memo
    TekstLista.Clear; //wykasowanie listy wierszy
    Memo1.Modified:=true //odnotowanie zmian
end;

```

```

//-----
procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
    Prawo1Click(Sender);
    SpeedButton10.Down:=true
end;
//-----
procedure TForm1.SpeedButton11Click(Sender: TObject);
begin
    Drukuj1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton12Click(Sender: TObject);
begin
    Sortowaniewierszy1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    Poszukaj1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    Zapisz1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    Otworz1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
    Kopiuj1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
    Wytnij1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
    Doklej1Click(Sender)
end;
//-----
procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
    Lewo1Click(Sender);
    SpeedButton7.Down:=true
end;
//-----
procedure TForm1.Kopiuj1Click(Sender: TObject);
begin
    Memo1.CopyToClipboard;
end;
//-----

```

```

procedure TForm1.NotaProgramu1Click(Sender: TObject);
begin
  ShowMessage('  MiniEdytor, czyli MEdytor,'+#13#10+ #13#10+
' jest niewielkim programem dydaktycznym z dołączonymi komentarzami'+#13#10+
' i zapisem źródłowym (Medytor_Opis.pdf).'+#13#10+
' Program realizuje podstawowe funkcje aplikacji przeznaczonej'+#13#10+
' do redagowania tekstu. Jednocześnie został napisany w sposób'+#13#10+
' bardzo oszczędny z ograniczeniem się tylko do podstawowych narzędzi'+#13#10+
' dostępnych w środowisku programowania Delphi.'+#13#10+#13#10+
#9#9#9#9#9#9#9+' PurAnd')
end;
//-----
procedure TForm1.Wytnij1Click(Sender: TObject);
begin
  Memo1.CutToClipboard;
end;
//-----
procedure TForm1.Doklej1Click(Sender: TObject);
begin
  Memo1.PasteFromClipboard;
end;
//-----
procedure TForm1.NotujDate1Click(Sender: TObject);
begin
  SpeedButton8Click(Sender)
end;

//-----
procedure TForm1.UsunWybrane1Click(Sender: TObject);
begin
  Memo1.ClearSelection;
end;
//-----
procedure TForm1.ZaznaczWszystko1Click(Sender: TObject);
begin
  Memo1.SelectAll;
end;
//-----
procedure TForm1.KolorTla1Click(Sender: TObject);
begin
  if ColorDialog1.Execute then Memo1.Color:=ColorDialog1.Color;
end;
//-----
procedure TForm1.Komentarze1Click(Sender: TObject);
begin
  ShellExecute(handle,'open','Medytor_Opis.pdf','',SW_NORMAL)
end;
//-----
procedure TForm1.Koniec1Click(Sender: TObject);
begin
  if Memo1.Modified then
    case Application.MessageBox('Zapisy mogą być utracone!'+#10#13+
' Czy pomimo to zakończyć pracę?', 'Koniec pracy aplikacji',
MB_YESNO+MB_IconStop) of
      IDYES: halt;
      IDNO: Abort
    end
  else halt
end;
end;

```

```
//-----
procedure TForm1.Prawo1Click(Sender: TObject);
begin
  Memo1.Alignment:=taRightJustify;
  SpeedButton10.Down:=true;
  Prawo1.Checked:=true
end;
//-----
procedure TForm1.Cofnij1Click(Sender: TObject);
begin
  with Memo1 do
    if HandleAllocated then SendMessage(Handle,EM_UNDO,0,0)
end;
//-----
procedure TForm1.Lewo1Click(Sender: TObject);
begin
  Memo1.Alignment:=taLeftJustify;
  SpeedButton7.Down:=true;
  Lewo1.Checked:=true
end;

end.
```

Literatura

1. Norton P., Przewodnik po Delphi 2, Wyd. PLJ, Warszawa 1997
2. Kwasowicz W., Wprowadzenie do Object Pascal i Delphi, Wyd. MIKOM, Warszawa 2002
3. Osier D., Grobman S., Batson S., Delphi 2, Wyd. Helion, Gliwice 1997
4. Biernat J., Ćwiczenia z podstawowych komponentów Delphi, Wyd. MIKOM, Warszawa 2002
5. Boduch A., Barszcz Ł., Delphi 2005, Wyd. Helion, Gliwice 2005
6. Urbański A.P., Zastosowanie biblioteki VCL pakietu Delphi, Wyd. Nakom, Poznań 2001
7. Snarska A., Ćwiczenia z Delphi 3.0, 4.0, 5.0, Wyd. MIKOM, Warszawa 2000
8. Strzałkowski K., Podstawy Delphi, Wyd. Stachurski, Kielce 2000