

3.3.4 Szöveg megjelenítők

1. **Label** vezérlő egyszeri illetve kettős kattintás eseménykezelése: [label2](#)
2. Jelszó beolvasása **Edit** vezérlővel: [jelszo](#)
3. **Memo** vezérlő többsoros szövegszerkesztő bemutatása: [memo](#)
4. **Memo** vezérlő állománykezeléssel: [MemoFile](#)



Készítsünk egy alkalmazást, amely bemutatja a címkéken az egyszeri és a kettős kattintás eseménykezelését! (*label2*)

A **Label** vezérlők reagálnak a bal egérgombbal való egyszeri kattintásra – *OnClick* illetve a dupla kattintásra – *OnDblClick* esemény keletkezik. A kiírt szöveg tájékoztat az esemény bekövetkeztéről. A példaprogram induló képe, egyszeri illetve dupla kattintás esetén a kijelzés:



A *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    procedure Label1Click(Sender: TObject);
    procedure Label2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Label1DblClick(Sender: TObject);
    procedure Label2DblClick(Sender: TObject);
  end;
```

Eseménykezelő eljárások:

A **FormCreate** eseménykezelő eljárásban a két címkemezőben a program alapállapotára vonatkozó szöveget írjuk ki.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Label1.Caption := 'Label1 - egérekattintásra vár';
  Label2.Caption := 'Label2 - egérekattintásra vár';
end;
```

Az egyszeri kattintás eseménykezelő eljárásai, melyek a megfelelő címkemezőben jelzik az *OnClick* eseményt.

```
procedure TForm1.Label1Click(Sender: TObject);
begin
  Label1.Caption := 'Egér - OnClick esemény';
end;

procedure TForm1.Label2Click(Sender: TObject);
begin
  Label2.Caption := 'Egér - OnClick esemény';
end;
```

A kettős kattintás eseménykezelő eljárásai, melyek a megfelelő címkemezőben jelzik az *OnDblClick* eseményt.

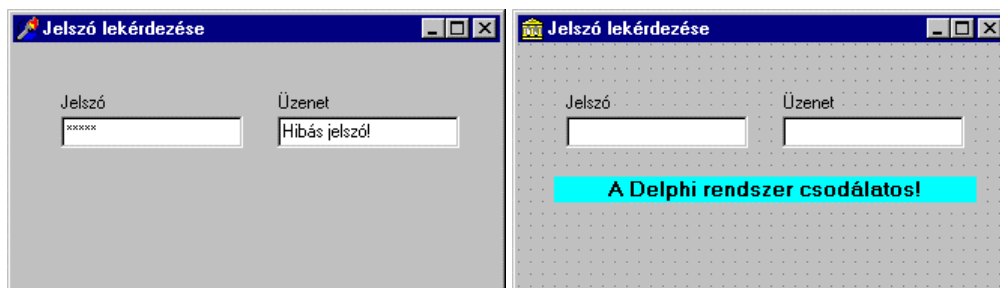
```
procedure TForm1.Label1DblClick(Sender: TObject);
begin
  Label1.Caption := 'Egér - OnDblClick esemény';
end;

procedure TForm1.Label2DblClick(Sender: TObject);
begin
  Label2.Caption := 'Egér - OnDblClick esemény';
end;
```



Olvassunk be egy nagybetűs 5 karakteres jelszót, majd jelezzük az elfogadást, illetve ha hibás a megadás! A jelszó ablakban csillag karaktereket írjunk vissza! (*jelszo*)

A JELSZO alkalmazás futási ablakai:



A *Form1* osztálya:

```
type
  TForm1 = class(TForm)
    Jelszo: TEdit;
    Uzenet: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
  procedure FormCreate(Sender: TObject);
  procedure JelszoChange(Sender: TObject);
  private
    titok : string;
  end;
```

Eseménykezelő eljárások:

A form betöltésekor meghívódó *FormCreate* eseménykezelő eljárásban a *titok* sztringet feltöltjük a jelszóval, valamint beállítjuk a *Jelszo* szövegmező tulajdonságait. A ***PasswordChar*** jellemző értékét '*' karakterre állítjuk. A ***MaxLength*** a begépelhető szöveg maximális hosszát határozza meg. Az *Uzenet* szövegmező ***ReadOnly*** tulajdonságának ***True*** értékére állításával az ablak tartalmát a felhasználó nem változtathatja meg. A *Label3* vezérlőt láthatatlanná tesszük.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Jelszo.PasswordChar := '*';
  Jelszo.MaxLength := 5;
  Jelszo.CharCase := ecUpperCase;
  Uzenet.ReadOnly := true;
  titok := 'TITOK';
  Label3.Visible := false;
end;
```

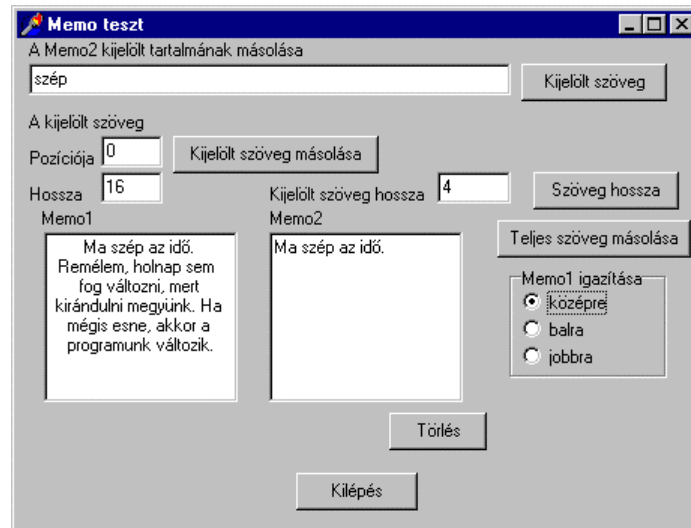
Már a *Jelszo* ablakba beírt első karakter hatására is meghívódik a *JelszoChange* eseménykezelő eljárás, amely a *Jelszo* szövegmezőben megadott jelszót összehasonlítja a *titok* változó tartalmával, majd az összehasonlítás eredményét megjeleníti az *Uzenet* ablakban. Ha a jelszó hibátlan, a *Label3* vezérlő láthatóvá válik, különben láthatatlan marad.

```
procedure TForm1.JelszoChange(Sender: TObject);
begin
  if Jelszo.Text = titok then
  begin
    Uzenet.Text := 'Elfogadva';
    Label3.Visible := true;
  end
  else
  begin
    Uzenet.Text := 'Hibás jelszó!';
    Label3.Visible := false;
  end;
end;
```



Memo felhasználásával tároljunk többsoros szöveget! A kijelölt szöveget másoljuk *Memo*, illetve *TEdit* típusú szövegmezőbe! Jelezzük ki a tárolt szövegek hosszát, valamint igazítsuk a *Memo* szövegét a lehetséges háromféle módon! (*memo*)

A MEMO alkalmazás futási képe:



A *TForm1* osztálya:

```
type
TForm1 = class(TForm)
    Memo1: TMemo;
    Memo2: TMemo;
    Label1: TLabel;
    Hossz: TEdit;
    KijeloltSzoveg: TButton;
    TeljesMasolas: TButton;
    Kilepes: TButton;
    Honnan: TEdit;
    Mennyit: TEdit;
    Leolvas: TButton;
    Torles: TButton;
    Edit2: TEdit;
    RadioGroup1: TRadioGroup;
    SzovegHossz: TButton;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure KilepesClick(Sender: TObject);
    procedure TeljesMasolasClick(Sender: TObject);
    procedure LeolvasClick(Sender: TObject);
    procedure TorlesClick(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
    procedure SzovegHosszClick(Sender: TObject);
    procedure KijeloltSzovegClick(Sender: TObject);
end;
```

Eseménykezelő eljárások:

A *Form1* objektum létrejöttkor megívódó *FormCreate* eseménykezelő eljárásban beállítjuk a *Memo1* kezdőértékét.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    // a Memo1 kezdőértékének beállítása
    Memo1.Text := 'Ma szép az idő. Remélem, holnap sem '+
        ' fog változni, mert kirándulni megyünk.'+
        ' Ha mégis esne, akkor a programunk változik.';
end;
```

A "*Teljes szöveg másolása*" nyomógomb megnyomásakor meghívódó *TeljesMasolasClick* eseménykezelő eljárásban a *Memo1* teljes szövegét átmásoljuk a *Memo2* szövegmezőjébe.

```
procedure TForm1.TeljesMasolasClick(Sender: TObject);
begin
    Edit2.Clear;
    // a "Szöveg hossza" szövegmező törlése
    Hossz.Clear;
    // a Memo1-ben lévő teljes szöveg Memo2-be másolása
    Memo2.Text := Memo1.Text;
end;
```

A "*Kijelölt szöveg másolása*" nyomógomb megnyomásakor meghívódó *LeolvasClick* eseménykezelő eljárásban a kijelölt szöveg kezdőpozícióját a "*Pozíciója*", a kijelölt szöveg hosszát pedig a "*Hossza*" szövegmezőben írjuk ki. A *CopyToClipboard* metódussal a *Memo1* kivágott szövegét a vágólapra írjuk és a *PasteFromClipboard* metódussal a *Memo2*-be másoljuk.

```
procedure TForm1.LeolvasClick(Sender: TObject);
begin
    Memo2.Clear;
    Edit2.Clear;
    Hossz.Clear;
    // Memo2
    Memo2.Alignment := taLeftJustify;
    // a kijelölés kezdőpozíciója
    Honnan.Text := IntToStr(Memo1.SelStart);
    // a kijelölt szöveg hossza
    Mennyit.Text := IntToStr(Memo1.SelLength);
    if StrToInt(Mennyit.Text) > 0 then
    begin
        // ha van kijelölt szöveg, akkor
        // a Vágóasztalra írása
        Memo1.CopyToClipboard;
        // másolás a Vágóasztalról a Memo2-be
        Memo2.PasteFromClipboard;
    end;
end;
```

A "*Törlés*" nyomógomb megnyomásakor meghívódó *TorlesClick* eseménykezelő eljárásban töröljük a "*Kijelölt szöveg hossza*" szövegmezőt és a *Memo2* ablakot.

```
procedure TForm1.TorlesClick(Sender: TObject);
begin
    // a "Szöveg hossza" szövegmező törlése
    Hossz.Clear;
    // a Memo2 törlése
    Memo2.Clear;
end;
```

A "*Memo1 igazítása*" választógomb-csoport gombjaival középre, balra és jobbra igazíthatjuk a *Memo1* szövegét.

```
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
    // a Memo2 szöveg pozícionálása
    case RadioGroup1.ItemIndex of
        // középre
        0: Memo1.Alignment:= taCenter;
        // balra
        1: Memo1.Alignment:= taLeftJustify;
        // jobbra
        2: Memo1.Alignment:= taRightJustify;
    end;
end;
```

A "*Szöveg hossza*" nyomógomb megnyomásakor a *Memo2*-ben kijelölt szöveg hosszát írjuk ki a "*Kijelölt szöveg hossza*" szövegmezőben.

```
procedure TForm1.SzovegHosszClick(Sender: TObject);
begin
    // a "Szöveg hossza"
    Hossz.Text := IntToStr(Memo2.SelLength);
end;
```

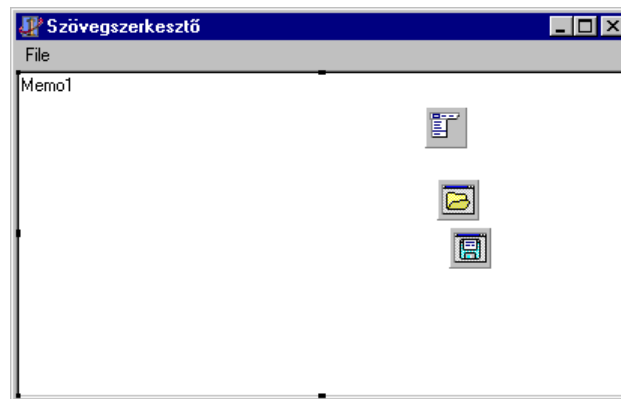
A "*Kijelölt szöveg*" nyomógomb megnyomásakor a *KijeloltSzovegClick* eseménykezelő eljárásban a *Memo2*-ben kijelölt szöveget a "*A Memo2 kijelölt tartalmának másolása*" szövegmezőbe írjuk.

```
procedure TForm1.KijeloltSzovegClick(Sender: TObject);
begin
    Edit2.Text:= Memo2.SelText;
end
```

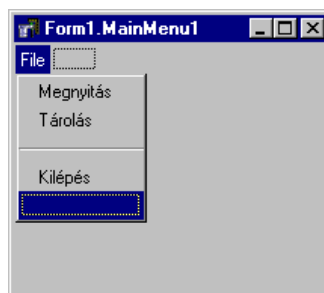


Készítsünk olyan programot, amely szövegszerkesztőként működik és a szövegeket állományba képes menteni, illetve beolvasni onnan! (*MemoFile*)

A feladat megoldásához használjuk a fájlok megnyitására és tárolására szolgáló általános párbeszédablakokat. Anélkül, hogy a részletekbe merülnénk, helyezzük a formra a **Dialogs** palettalap **OpenDialog** és **SaveDialog** komponenseit! Mivel menüvezérelten szeretnénk az állományokat nyitni, illetve tárolni menüre is szükségünk lesz.



A menütervező:



A TForm1 osztály a menüesemények kezelőivel:

```
type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;      // A főmenü
    Megnyits1: TMenuItem;  // A Megnyitás menü
    Trols1: TMenuItem;     // A Tárolás menü
    N1: TMenuItem;
    Kilps1: TMenuItem;     // A Kilépés menü
    Memo1: TMemo;
    OpenDialog1: TOpenDialog; // Az OpenDialog komponens
    SaveDialog1: TSaveDialog; // A SaveDialog komponens
    procedure Megnyits1Click(Sender: TObject);
    procedure Trols1Click(Sender: TObject);
    procedure Kilps1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

A *Megnyitás* menüesemény kezelője:

```
procedure TForm1.Megnyits1Click(Sender: TObject);
begin
    // Az általános dialógussal meghatározott TXT kiterjesztésű
    // beolvassuk a szöveget.
    if OpenDialog1.Execute then
        Memo1.Lines.LoadFromFile(OpenDialog1.FileName);
end;
```

A *Tárolás* menüesemény kezelője:

```
procedure TForm1.Trols1Click(Sender: TObject);
begin
    // Tároljuk a Memo1 szövegét az általános dialógussal meghatározott
    // néven TXT kiterjesztéssel
    if SaveDialog1.Execute then
        Memo1.Lines.SaveToFile(SaveDialog1.FileName+'.TXT');
end;
```

A *Kilépés* menüesemény kezelője:

```
procedure TForm1.Kilps1Click(Sender: TObject);
begin
    // Kilépés a programból
    Application.Terminate;
end;
```

