
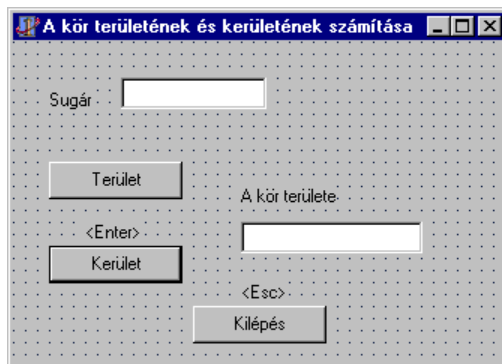


3.3.2 Vezérlőgombok

1. <Enter> és <Esc> nyomógombok használata: [nyomo1](#)
2. Kétállapotú jelölőnégyzet alkalmazása: [jelolo1](#)
3. Két- és háromállapotú jelölőnégyzet kezelése: [jelolo2](#)
4. Választógombok alkalmazása: [valaszt1](#)
5. **GroupBox** vezérlőbe helyezett választógombok kezelése: [valaszt2](#)
6. Fontstílus beállítása jelölőnégyzetekkel [Fontstil](#)
7. Alapszínek megadása választógombokkal [Alapszin](#)

 Készítsünk programot, amely a nyomógomb használatával kiszámítja a kör területét és kerületét! Alkalmazzunk **Default** (<Enter>) és **Cancel** (<Esc>) típusú nyomógombokat is! (*nyomol*)

Készítsük el az alkalmazást úgy, hogy egy szövegszerkesztőbe lehessen beírni a kör sugarát! A program nyomógombok megnyomására számítsa ki a kerületet és a területet! Az alkalmazás megtervezett megjelenése:



Ennek megfelelően a *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Sugar: TEdit;
    Terulet: TButton;
    Kerulet: TButton;
    Kilepes: TButton;
    Label2: TLabel;
    Eredmeny: TEdit;
    Label3: TLabel;
    Label4: TLabel;
    procedure TeruletClick(Sender: TObject);
    procedure SugarChange(Sender: TObject);
    procedure KeruletClick(Sender: TObject);
    procedure KilepesClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    R : real;
  end;
```

Az eseménykezelő eljárások:

A form betöltésekor meghívódó eseménykezelő eljárásban beállítjuk a nyomógombok tulajdonságait: a *Kerulet* nyomógomb az <Enter> billentyű megnyomására is működjön, ezért a **Default** tulajdonságát **true** értékre állítjuk. A *Kilepes* nyomógomb pedig az <Esc> billentyű megnyomására is működjön, ezt a **Cancel** tulajdonság **true** értékre állításával érjük el. Az *Eredmeny* szövegmező csak olvasható, és a tartalmát töröljük.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  Kerulet.Default := true; // Return billentyűre is működik
  Kilepes.Cancel := true; // az Esc billentyűre is működik
  Eredmeny.ReadOnly := true; // a szövegmező csak olvasható
  Eredmeny.Text := '';
end;
```

A "Sugár" szövegmező tartalmának változásakor hívódik meg a *SugarChange* eseménykezelő eljárás, amelyben *Label2* és az *Eredmeny* szövegmező tartalmát töröljük.

```
procedure TForm1.SugarChange(Sender: TObject);
begin
  Label2.Caption := '';
  Eredmeny.Clear;
end;
```

Szükségünk lesz arra, hogy szövegből számot készítsünk, még hozzá úgy, hogy a tizedes pontot és a tizedes vesszőt is kezeljük. Ha a beírt szöveg nem szám, akkor az *Str2Dbl* függvény 0-t ad eredményként, egyébként a számot.

```
function Str2Dbl(const s: string):double;
var dc : char;
begin
  dc := decimalseparator;
  if pos(',',s) <> 0 then
    decimalseparator := ','
  else
    decimalseparator := '.';
  try
    Str2Dbl := strToFloat(s);
  except
    Str2Dbl := 0;
  end;
  decimalseparator := dc;
end;
```

A "Terület" nyomógomb megnyomásakor meghívódó *TeruletClick* eseménykezelő eljárásban, ha a *Sugar* adatbeviteli mező hossza nagyobb nullánál, akkor *Sugar* mező tartalmát az *Str2Dbl* függvénnyel átalakítva az *R* változóba írjuk. Ha a kör sugara nagyobb nullánál, akkor beállítjuk az eredmény fejlécét, kiszámítjuk a kör területét és megjelenítjük az *Eredmeny* szövegmezőben.

```
procedure TForm1.TeruletClick(Sender: TObject);
begin
  if length(Sugar.Text) > 0 then
  begin
    R := Str2Dbl(Sugar.Text);
    if R > 0 then
    begin
      Label2.Caption := 'A kör területe';
      Eredmeny.Text := FloatToStr(sqr(R)*pi);
    end;
    Kerulet.SetFocus;
  end;
end;
```

A "Kerület" nyomógomb megnyomásakor meghívódó *KeruletClick* eseménykezelő eljárásban, ha a *Sugar* adatbeviteli mező hossza nagyobb nullánál, akkor *Sugar* mező tartalmát az *Str2Dbl* függvénnyel átalakítva az *R* változóba írjuk. Ha a kör sugara nagyobb nullánál, akkor beállítjuk az eredmény fejlécét, kiszámítjuk a kör kerületét és megjelenítjük az *Eredmeny* szövegmezőben.

```
procedure TForm1.KeruletClick(Sender: TObject);
begin
  if length(Sugar.Text) > 0 then
  begin
    R := Str2Dbl(Sugar.Text);
    if R > 0 then
    begin
      Label2.Caption := 'A kör kerülete';
      Eredmeny.Text := FloatToStr(2*R*pi);
    end;
  end;
end;
```

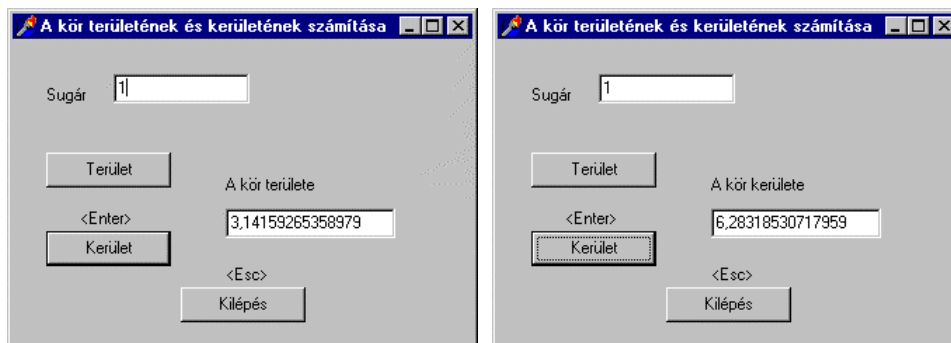
A *Kilépés* nyomógomb, vagy az <Esc> billentyű megnyomásakor meghívódó *KilepesClick* eseménykezelő eljárásban a program befejezi a működését.

```
procedure TForm1.KilepesClick(Sender: TObject);
begin
  Application.Terminate;
end;
```

A *FormCreate* eseménykezelő eljárásban megadjuk, hogy a futó alkalmazás ablaka hol jelenjen meg a képernyőn és mekkora legyen a magassága és szélessége (ezt a **Left**, **Top**, **Height** és **Width** tulajdonságoknál tehetjük meg) az ablak elhelyezkedését és méretét beállító tulajdonságok értékét:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    // az ablak bal felső sarkának koordinátái
    Left := 200;
    Top := 100;
    // az ablak magassága és szélessége
    Height := 240;
    Width := 380;
end;
```

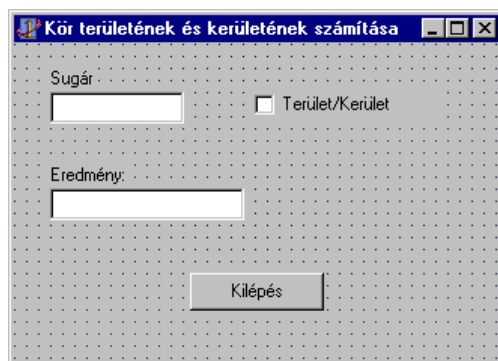
A NYOMO1 alkalmazás két futási képe:





Írjunk programot, amely kiszámítja a kör területét és kerületét! A számításhoz használjunk kétállapotú jelölőnégyzetet! (*jelolo1*)

Készítsük el az alkalmazást úgy, hogy egy szövegszerkesztőbe lehessen beírni a kör sugarát, választógomb döntse el, hogy mit számolunk! Az alkalmazás megtervezett megjelenése:



A szöveg számmá alakítására a [Nyomo1](#) példa a *Str2Dbl* függvényét használjuk.

A *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Sugar: TEdit;
    KerTer: TCheckBox;
    Label2: TLabel;
    Eredmeny: TEdit;
    Kilepes: TButton;
    procedure KilepesClick(Sender: TObject);
    procedure SugarChange(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    R : real; // a kör sugara
    { Private declarations }
  public
    { Public declarations }
  end;
```

Eseménykezelő eljárások:

A jelölőnégyzet és a szövegmező alapállapotát a form betöltésekor meghívódó *FormCreate* eseménykezelő eljárásban állítjuk be:

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  KerTer.State := CbChecked;
  Eredmeny.ReadOnly := true; // a szövegmező csak olvasható
  Eredmeny.Text := '';
end;
```

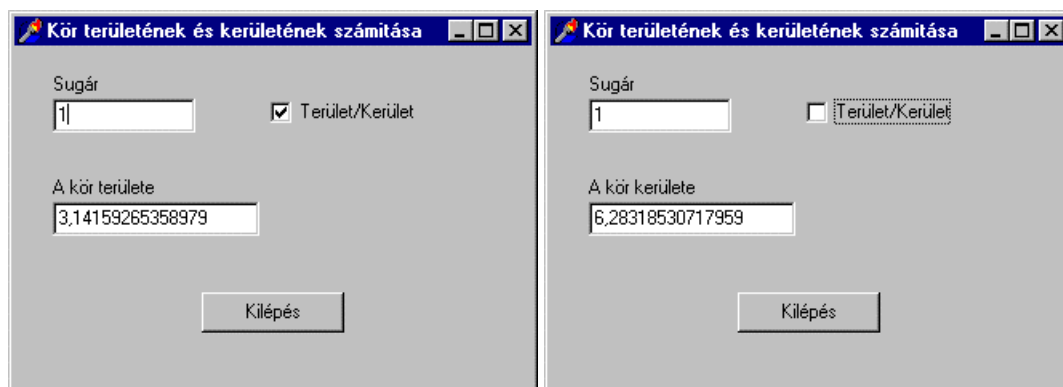
A "Sugár" szövegmező tartalmának megváltozásakor és a "Terület/Kerület" jelölőnégyzet kiválasztásakor hívódik meg a *SugarChange* eseménykezelő eljárás. Ha az *R* adatmezőbe konvertált érték pozitív, akkor a jelölőnégyzet állapotának megfelelően kijelezzük a kiszámított eredményt és a hozzá tartozó szöveget, egyébként az *Eredmeny* szövegmező tartalmát töröljük.

```

procedure TForm1.SugarChange(Sender: TObject);
begin
    R := Str2Dbl(Sugar.Text);
    if R > 0 then
    begin
        if KerTer.State = CbChecked then
        begin
            Label2.Caption := 'A kör területe';
            Eredmeny.Text := FloatToStr(sqr(R)*pi);
        end
        else
        begin
            Label2.Caption := 'A kör kerülete';
            Eredmeny.Text := FloatToStr(2*R*pi);
        end;
        else
            Eredmeny.Text := '';
        end;
    end;

```

A JELOLO1 alkalmazás két futási ablaka:





Írjunk programot, amely kiszámítja a kocka felszínét, térfogatát, az oldalatlót, a testátlót valamint az oldallapok területét. A műveleteket két- és háromállapotú jelölőnégyzetek jelöljék ki! (*jelolo2*)

Készítsük el az alkalmazás felületét! Az alkalmazás megtervezett megjelenése:

A *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Oldale1: TEdit;
    GroupBox1: TGroupBox;
    FT: TCheckBox;
    LTL: TCheckBox;
    GroupBox2: TGroupBox;
    Kilepes: TButton;
    Label2: TLabel;
    Eredmeny1: TEdit;
    Label3: TLabel;
    Eredmeny2: TEdit;
    procedure KilepesClick(Sender: TObject);
    procedure Oldale1Change(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FTClick(Sender: TObject);
    procedure LTClick(Sender: TObject);
  private
    a: double; // a kocka oldaléle
  end;
```

A szöveg számmá alakítására [Nyomol](#) példa a *Str2Dbl* függvényét használjuk.

Eseménykezelő eljárások:

A form betöltésekor a *FormCreate* eseménykezelő eljárás hívódik meg, amelyben különféle alapállapotokat állítunk be:

- a jelölőnégyzetek állapotát bejelöltre állítjuk
- a szövegmezők csak olvashatók, mivel a **ReadOnly** tulajdonságot **true** értékűre állítjuk

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  // a jelölőnégyzetek alapértékei
  FT.State := cbChecked;
  FT.AllowGrayed := false; // a jelölőnégyzet kétállapotú,
                          // ez az alapértelmezés
  LTL.State := cbChecked;
  LTL.AllowGrayed := true; // a jelölőnégyzet háromállapotú
  LTL.State := cbChecked;
  // a szövegmezők csak olvashatók
  Eredmeny1.ReadOnly := true;
  Eredmeny2.ReadOnly := true;
end;
```

A "Kocka oldaléle" szövegmező tartalmának megváltozásakor meghívódó *OldalElChange* eseménykezelő eljárásban töröljük az eredményeket és a szövegeket kijelző szövegmezők tartalmát.

```
procedure TForm1.OldalElChange(Sender: TObject);
begin
    Label2.Caption := '';
    Label3.Caption := '';
    Eredmeny1.Clear;
    Eredmeny2.Clear;
end;
```

A "Felszin/Térfogat" jelölőnégyzet kiválasztásakor meghívódó *FtClick* eseménykezelő eljárásban a jelölőnégyzet állapotának megfelelően kiszámítjuk és kijelezzük az eredményt és a hozzá tartozó szöveget.

```
procedure TForm1.FtClick(Sender: TObject);
begin
    if (length(OldalEl.Text) > 0) then
    begin
        a := Str2Db1(OldalEl.Text);
        if a > 0 then
        begin
            if FT.State = cbChecked then
            begin
                Label2.Caption := 'Kocka felszine';
                Eredmeny1.Text := FloatToStr(6*a*a);
            end
            else
            begin
                Label2.Caption := 'Kocka térfogata';
                Eredmeny1.Text := FloatToStr(a*a*a);
            end;
        end;
    end;
end;
```

A "Lapátló/Testátló/Lapterület" jelölőnégyzet kiválasztásakor meghívódó eseménykezelő eljárásban a jelölőnégyzet állapotának megfelelően kiszámítjuk és kijelezzük az eredményt és a hozzá tartozó szöveget.

```
procedure TForm1.LTLClick(Sender: TObject);
begin
    if length(OldalEl.Text) > 0 then
    begin
        a := Str2Db1(OldalEl.Text);
        if a > 0 then
        begin
            case LTL.State of
                cbChecked :
                begin
                    Label3.Caption := 'Lapátló';
                    Eredmeny2.Text := FloatToStr(sqrt(sqr(a)+sqr(a)));
                end;
                cbUnchecked:
                begin
                    Label3.Caption := 'Testátló';
                    Eredmeny2.Text := FloatToStr(a*sqrt(3));
                end;
                cbGrayed :
                begin
                    Label3.Caption := 'Lapterület';
                    Eredmeny2.Text := FloatToStr(a*a);
                end;
            end;
        end;
    end;
end;
```


A JELOLO2 alkalmazás két futási ablaka:

The image displays two side-by-side screenshots of the JELOLO2 application interface. Both windows have the title bar 'Kocka: felszín, térfogat, lapátló, testátló, lapterület számítása'. Each window contains a 'Kocka oldaléle' input field with the value '1'. Below this is a 'Számítások kijelölése' section with two checkboxes: 'Felszín/Térfogat' and 'Lapátló/Testátló/Lapterület'. A 'Kilépés' button is located at the bottom of each window.

Left Window:

- 'Kocka oldaléle': 1
- 'Számítások kijelölése':
 - ☒ Felszín/Térfogat
 - ☒ Lapátló/Testátló/Lapterület
- 'Eredmény' section:
 - 'Kocka felszine': 6
 - 'Lapátló': 1,4142135623731

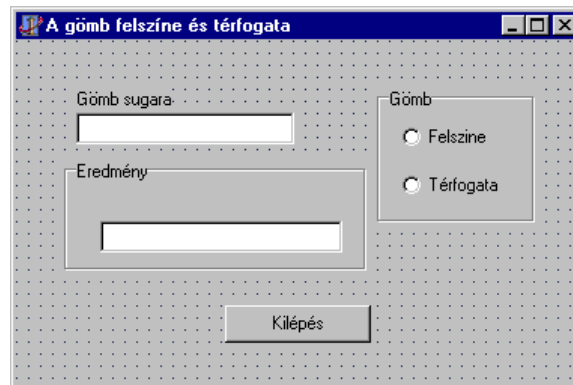
Right Window:

- 'Kocka oldaléle': 1
- 'Számítások kijelölése':
 - ☐ Felszín/Térfogat
 - ☐ Lapátló/Testátló/Lapterület
- 'Eredmény' section:
 - 'Kocka térfogata': 1
 - 'Testátló': 1,73205080756888



Írjunk programot a gömb felszínének és térfogatának kiszámítására! A számítások kijelölését választógombokkal oldjuk meg! (*valasztl*)

A megtervezett alkalmazás:



A *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    Felszine: TRadioButton;
    Terfogata: TRadioButton;
    Label1: TLabel;
    GSugar: TEdit;
    GroupBox2: TGroupBox;
    Eredmeny: TEdit;
    Label2: TLabel;
    Kilepes: TButton;
  procedure KilepesClick(Sender: TObject);
  procedure FelszineClick(Sender: TObject);
  procedure TerfogataClick(Sender: TObject);
  procedure GSugarChange(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  private
    R : real; // a gömb sugara
  end;
```

A szöveg számmá alakítására [Nyomol](#) példa a *Str2Dbl* függvényét használjuk.

Eseménykezelő eljárások:

A form betöltésekor meghívódó *FormCreate* eseménykezelő eljárásban az *Eredmeny* adatmezőt csak olvashatóra állítjuk.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  // a szövegmező csak olvasható
  Eredmeny.ReadOnly := true;
end;
```

A "Sugár" beviteli mező tartalmának változásakor meghívódó *GSugarChange* eseménykezelő eljárásban az eredményt kijelző szövegmezők tartalmát töröljük.

```
procedure TForm1.GSugarChange(Sender: TObject);
begin
  Eredmeny.Clear;
  Label2.Caption := '';
end;
```

A "Felszín" választógombra való kattintáskor a *FelszineClick* eseménykezelő eljárásban leolvassuk a "Sugár" adatmező tartalmát az *R* adatmezőbe. Ha az *R* értéke pozitív, akkor kiszámítjuk a gömb felszínét és az eredményt a szöveggel együtt kijelezzük.

```

procedure TForm1.FelszineClick(Sender: TObject);
begin
  if length(GSugar.Text) <> 0 then
    begin
      R := Str2Dbl(GSugar.Text);
      if R > 0 then
        begin
          Label2.Caption := 'Gömb felszíne';
          Eredmeny.Text := FloatToStr(4*sqr(R)*pi);
        end;
      end;
    end;
end;

```

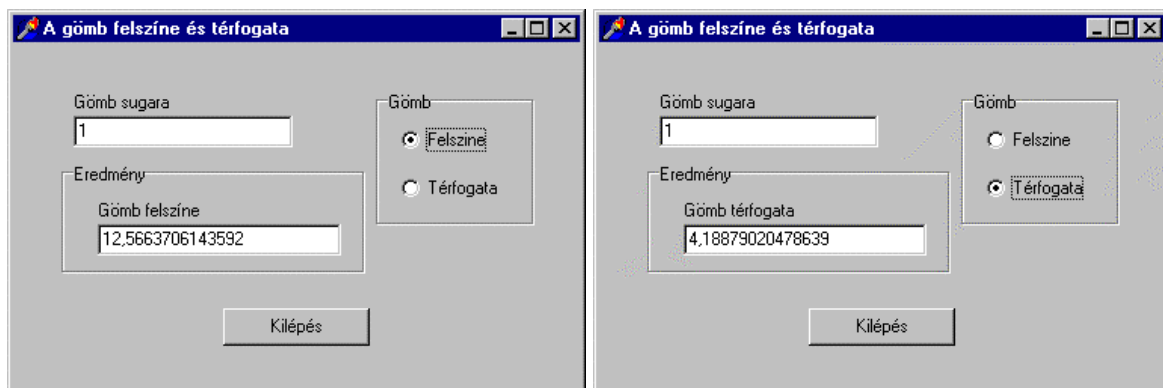
A "Térfogat" választógombra való kattintáskor a *TerfogatClick* eseménykezelő eljárásában leolvassuk a "Sugár" adatmező tartalmát az *R* adatmezőbe. Ha az *R* értéke pozitív, akkor kiszámítjuk a gömb térfogatát és az eredményt a szöveggel együtt jelenítjük meg.

```

procedure TForm1.TerfogataClick(Sender: TObject);
begin
  if length(GSugar.Text) <> 0 then
    begin
      R := Str2Dbl(GSugar.Text);
      if R > 0 then
        begin
          Label2.Caption := 'Gömb térfogata';
          Eredmeny.Text := FloatToStr(4*sqr(R)*R*pi/3);
        end;
      end;
    end;
end;

```

A VALASZT1 alkalmazás futási ablakai:





Írjunk programot a gömb felszínének és térfogatának kiszámítására! A számítások kijelölését a *RadioGroup* vezérlőben létrehozott választógombokkal oldjuk meg! (*valaszt2*)

A megtervezett alkalmazás:



A *TForm1* osztálya:

```
type
  TForm1 = class(TForm)
    Label1: TLabel;
    GSugar: TEdit;
    GroupBox2: TGroupBox;
    Eredmeny: TEdit;
    Label2: TLabel;
    Kilepes: TButton;
    RadioGroup1: TRadioGroup;
    procedure KilepesClick(Sender: TObject);
    procedure GSugarChange(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  end;
```

A szöveg számmá alakítására [Nyomol](#) példa a *Str2Dbl* függvényét használjuk.

Eseménykezelő eljárások:

A *FormCreate* eseménykezelő eljárásban az *Eredmeny* szövegmezőt csak olvashatóra állítjuk.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  // a szövegmező csak olvasható
  Eredmeny.ReadOnly := true;
end;
```

A "Sugár" beviteli mező tartalmának megváltozásakor a *GSugarChange* eseménykezelő eljárásban az *Eredmeny* szövegmezőt és a *Label2* címkemezőt töröljük.

```
procedure TForm1.GSugarChange(Sender: TObject);
begin
  Eredmeny.Clear;
  Label2.Caption := '';
end;
```

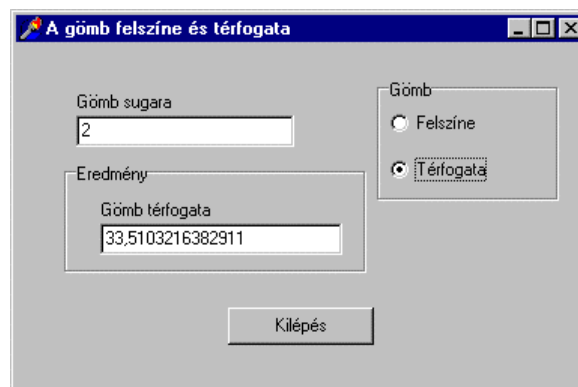
A választógombokon való kattintáskor meghívódó eseménykezelő eljárásokban a **RadioGroup** vezérlő **ItemIndex** tartalma dönti el, hogy melyik választógomb került kiválasztásra. Az index 0-tól kezdődik. A választógomb kiválasztásának megfelelően megjelenítjük a kiszámított eredményt és a hozzá tartozó szöveget.

```

procedure TForm1.RadioGroup1Click(Sender: TObject);
var
    R : integer;
begin
    if length(GSugar.Text) <> 0 then
        begin
            R := StrToInt(GSugar.Text);
            if R > 0 then
                begin
                    case RadioGroup1.ItemIndex of
                        0 : begin
                            Label2.Caption := 'Gömb felszíne';
                            Eredmeny.Text := FloatToStr(4*sqr(R)*pi);
                        end;
                        1 : begin
                            Label2.Caption := 'Gömb térfogata';
                            Eredmeny.Text := FloatToStr(4*sqr(R)*R*pi/3);
                        end;
                    end;
                end;
            end;
        end;
    end;
end;

```

A VALASZT2 alkalmazás futási ablaka:





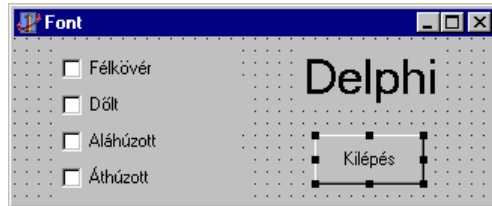
Készítsünk programot, amely jelölőnégyzetekkel szabályozza a „Delphi” felirat betűtípusának stílusát (dőlt, kövér, aláhúzott stb.)! (*Fontstíl*)

A *Label1* címke **Font** jellemzőjének **Style** (*TFontStyles*) halmaz típusú mezeje határozza meg a betűk megjelenését.

```
TFontStyle = (fsBold, fsItalic, fsUnderline, fsStrikeOut);
```

```
TFontStyles = set of TFontStyle;
```

Tegyük a formra a halmazelemeknek megfelelő jelölőnégyzeteket! Ha a jelölőnégyzet bekapcsolt, akkor a fontstílus tartalmazza a megfelelő halmazelemet.



A program indulásakor felvesszük az alapbeállításokat.

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Bold.Checked:=true; // az alap betűtípus félkövér
end;
```

Mindegyik jelölőnégyzet a félkövérnek megfelelően működik.

```
procedure TForm1.BoldClick(Sender: TObject);
begin
    if Bold.Checked then
        Label1.Font.Style:=Label1.Font.Style+[fsBold]
    else
        Label1.Font.Style:=Label1.Font.Style-[fsBold];
end;
```

A *Kilepes* gomb segítségével szakíthatjuk meg a program futását.

```
procedure TForm1.KilepesClick(Sender: TObject);
begin
    Application.Terminate;
end;
```

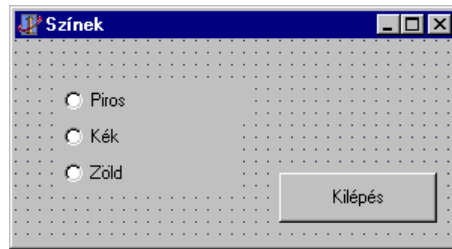
A futási képből jól látszik, hogy a jelölőnégyzetek állapota egymástól független.





Írjunk olyan programot, amelyben a form háttérszínét választógombokkal állítjuk be! (*Alapszín*)

A program jól mutatja, hogy a rádiógombok közül csak egy lehet kiválasztva. Az alkalmazás megjelenése:



Az alapszínnek *cl* előtagú konstansokkal definiálhatók (*clRed*, *clBlue* és *clGreen*). A program indításához be kell állítani az alaphelyzetet

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    Form1.Color:=clRed;
    Piros.Checked:=true;
end;
```

A *Kilepes* gomb segítségével szakíthatjuk meg a program futását.

```
procedure TForm1.KilepesClick(Sender: TObject);
begin
    Application.Terminate;
end;
```

Például a *Zöld* rádiógomb választása

```
procedure TForm1.ZoldClick(Sender: TObject);
begin
    Form1.Color:=clGreen;
end;
```

Érdemes megjegyezni, hogy elegendő a *Form1 Color* tulajdonságát beállítani, a *ParentColor* logikai változó *true* állapota miatt a választógombok öröklik a szülő színét.

