

1. A PASCAL program szerkezete

Programfej
Definíciós, illetve deklarációs rész
Programtörzs

2. Első PASCAL program

```
Program legelso;  
Begin  
  Write('Rendben');  
End.
```

Mentés
Futtatás
Javítás

```
Program legelso;  
Begin  
  Write('Rendben');  
  Readln;  
End.
```

Mentés
Futtatás

3. A változók deklarációja

```
Program masodik;  
Var szoveg: string;  
Begin  
  szoveg:='Rendben';  
  Write(szoveg);  
  Readln;  
End.
```

Fontosabb változótípusok

Integer	-	egész szám -32768..32767 között
Real	-	valós szám 2.939E-39..1.701E38 között
Char	-	egy karakter
String	-	legfeljebb 255 karakterből álló szöveg (karakter sorozat)
Boolean	-	logikai (értéke igaz-true, vagy hamis-false)

Értékadó utasítás

Változó:=új érték (a szöveges értéket '' jelek közé kell tenni)

Harmadik program

```
Program harmadik;  
Var i,j : integer;  
    x : real;  
Begin  
    i:=9;  
    j:=6;  
    x:=i/j;  
    Write(x);  
    Readln;  
End.
```

Miért lett az x változó valós típusú?

Negyedik program (beolvasás billentyűzetről)

```
Program negyedik;  
Var i,j: integer;  
    x: real;  
Begin  
    Readln(i);  
    j:=6;  
    Write(x);  
    Readln;  
End.
```

Readln(i) az i változó értékét a billentyűzetről olvastatja be.

4. Kiírás a képernyőre

Write(változó)	-	kiírás
Writeln(változó)	-	kiírás + soremelés

```
Program kiiras;  
Var x:real;  
Begin  
    x:=3.14;  
    Writeln(x);  
    Readln;  
End.
```

A valós számokban tizedespont szerepel.
A valós számok alértelmezetten normálalakban íródnak ki.

Javítás

```
Program kiiras;  
Var x:real;  
Begin  
    x:=3.14;  
    Writeln(x:6:2);
```

```
Readln;  
End.
```

Writeln(x:6:2) - a valós számot 6 karakteren ábrázoljuk, amelyből 2 lesz a tizedesek hossza.

```
Program pozicionalas;  
Var x:string;  
Begin  
  x:='Rendben';  
  Writeln(x:12);  
  Readln;  
End.
```

A Rendben szó előtt megjelenik 5 szóköz.

```
Program hibasiras;  
Var a,b : integer;  
Begin  
  a:=11; b:=22;  
  Writeln(a,b);  
  Readln;  
End.
```

Javítás

```
Program hibasiras;  
Var a,b : integer;  
Begin  
  a:=11; b:=22;  
  Writeln(a:3,b:3);  
  Readln;  
End.
```

„Interaktív program”

```
Program irolvas;  
Var i,j:integer;  
    x:real;  
Begin  
  Write('Az osztando:');  
  Readln(i);  
  j:=6;  
  Writeln('Az osztando: 6');  
  x:=i/j;  
  Writeln('Az eredmény: ',x:5:2);  
  Readln;  
End.
```

Elemezzük a programot!

5. Számítás

Értékkadás - változó:=kifejezés;

```
Program hiba1;  
Var i,j,k : integer;  
Begin  
  Write('i:');  
  Readln(i);  
  Write('j:');  
  Readln(j);  
  k:=i/j;  
  Writeln('k=',k:5);  
  Readln;  
End.
```

Javítás: - k:real; vagy k:=round(i/j); (kerekítés)

Elemezzük a következő értékkadó utasítást! a:=a/2+b;

6. Az egyszerű elágazás

Kérjünk be egy számot!
Ha a szám pozitív, akkor írassuk ki a '+' szöveget!

```
Program elojel;  
Var x:integer;  
Begin  
  Write('Add meg a számot:');  
  Readln(x);  
  If x>0 then writeln('+');  
  Readln;  
End.
```

Elágazás - if feltétel then utasítás;
A then utáni utasítás csak akkor hajtódik végre, ha az if utáni feltétel igaz.

Módosítsuk a programot úgy, hogy nem pozitív szám esetén 'nem +' szöveget írassunk ki!

```
If x>0 then writeln('+') else writeln('nem +');
```

Elágazás - if feltétel then utasítás1 else utasítás2;
Az utasítás1 csak akkor hajtódik végre, ha a feltétel igaz, az utasítás2 csak akkor hajtódik végre, ha a feltétel hamis.

Módosítsuk a programot úgy, hogy negatív szám esetén a '-' szöveget, 0 esetén a 'nulla' szöveget írja ki!

```
If x>0 then writeln('+') else if x<0 then writeln('-') else writeln('0');
```

7. Többszörös elágazás

Kérjünk be egy hónap sorszámát!
Írjuk ki a hónap nevét!

```
Program honap;  
Var x:integer;  
Begin  
  Write('A hónap sorszama:');  
  Readln(x);  
  Case x of  
    1:writeln('Januar');  
    2:writeln('Februar');  
    3:writeln('Marcius');  
    4:writeln('Aprilis');  
    5:writeln('Majus');  
    6:writeln('Junius');  
    7:writeln('Julius');  
    8:writeln('Augusztus');  
    9:writeln('Szeptember');  
    10:writeln('Oktober');  
    11:writeln('November');  
    12:writeln('December');  
    Else writeln('A sorszam nem megfelelo.')  End;  
  Readln;  
End.
```

Általános szerkezet

```
Case kifejezés of  
  Érték1:utasítás1;  
  Érték2:utasítás2;  
  ...  
  Értékn:utasításn;  
  Else utasítás  
End.
```

Az az utasítás hajtódik végre, amely előtti értéket a kifejezés felveszi.

8. A tömbök

Kérjünk be 10 egész számot, írjuk ki az összegüket!

Ha azonos típusú változókra van szükség, akkor célszerű tömböt (sorozatot) használni!
A változókra azonos azonosítóval hivatkozhatunk, csak a sorszámuk különbözteti meg őket,
amelyet [] jelek közé rakunk.

```
Deklaráció - var szam : array [1..10] of integer;  
Ezzel létrehozuk a szam[1], szam[2], ..., szam[10] változókat.
```

```

Program osszegzes;
Var szam:array[1..10]of integer;
    osszeg:integer;
begin
  write('Az 1. szám:');
  readln(szam[1]);
  write('A 2. szám:');
  readln(szam[2]);
  write('A 3. szám:');
  readln(szam[3]);
  write('A 4. szám:');
  readln(szam[4]);
  write('Az 5. szám:');
  readln(szam[5]);
  write('A 6. szám:');
  readln(szam[6]);
  write('A 7. szám:');
  readln(szam[7]);
  write('A 8. szám:');
  readln(szam[8]);
  write('A 9. szám:');
  readln(szam[9]);
  write('A 10. szám:');
  readln(szam[10]);
  osszeg:=szam[1]+ szam[2]+ szam[3]+ szam[4]+ szam[5]+ szam[6]+ szam[7]+ szam[8]+
  szam[9]+ szam[10];
  writeln('Az osszeg: ', osszeg);
  readln;
end.

```

Hát ez így nagyon csúnya!

9. A FOR ciklus

For ciklusváltozó:=kifejezés1 to kifejezés2 do utasítás;

A ciklusváltozó felveszi a kifejezés1 értékét, végrehajtja az utasítást, majd a ciklusváltozó értéke eggyel nő, újra végrehajtott az utasítás, és ez így megy mindaddig, míg a ciklusváltozó el nem éri a kifejezés2 értékét.

Illetve

For ciklusváltozó:=kifejezés1 downto kifejezés2 do utasítás;

Az előzőtől annyiban tér el, hogy minden lépés után eggyel csökken a ciklusváltozó értéke.

Az előző program for ciklussal

```
Program osszegzes;
Var szam:array[1..10]of integer;
    osszeg,i:integer;
begin
    writeln('Kérem a 10 számot!');
    for i:=1 to 10 do readln(szam[i]);
    osszeg:=0;
    for i:=1 to 10 do osszeg:=osszeg+szam[i];
    writeln('Az osszeg: ',osszeg);
    readln;
end.
```

FELADAT: Készíts olyan programot, amely bekér 15 valós számot, majd kiírja az átlagukat!

10. A WHILE (elől tesztelő) ciklus

While feltétel do utasítás;

A program az utasításhoz érve megvizsgálja a feltételt. Ha értéke igaz, akkor végrehajtja az utasítást, és újra vizsgálja a feltételt. Ezt ismétli mindaddig, míg a kifejezés értéke hamis nem lesz.

Az összegző feladat while ciklussal

```
Program osszegzes;
Var szam:array[1..10]of integer;
    osszeg,i:integer;
begin
    writeln('Kérem a 10 számot!');
    i:=1;
    osszeg:=0;
    while i<=10 do
        begin
            readln(szam[i]);
            osszeg:=osszeg+szam[i];
            i:=i+1;
        end;
    writeln('Az osszeg: ',osszeg);
    readln;
end.
```

FELADAT: Az átlagoló feladatot készítsd el WHILE ciklussal!

11. A REPEAT (hátral tesztelő) ciklus

```
Repeat
    Utasítás1;
    Utasítás2;
```

```
...
  Utasításn;
Until feltétel;
```

Először végrehajtnak az utasítások, majd megvizsgálja a feltételt. Ha a feltétel hamis, akkor ismét végrehajtja az utasításokat, majd ismét a feltétel vizsgálata következik. Ez történik mindaddig, míg a feltétel igaz nem lesz.

Az összegző feladat repeat ciklussal

```
Program osszegzes;
Var szam:array[1..10]of integer;
    osszeg,i:integer;
begin
  i:=1;
  osszeg:=0;
  writeln('Kérem a 10 számot!');
  repeat
    readln(szam[i]);
    osszeg:=osszeg+szam[i];
    i:=i+1;
  until i>10;
  writeln('Az osszeg: ',osszeg);
  readln;
end.
```

FELADAT: Készítsd el az átlagoló feladatot REPEAT ciklus segítségével is!

12. A PASCAL program szerkezete

```
{programfej}
{globális hatású fordítási direktívák}
  Program programnév;

{deklarációs rész}
  {lokális hatású fordítási direktívák}
  Uses {a használt unit könyvtárak felsorolása};
  Label {címkék deklarációja};
  Const {konstansok deklarálása};
  Type {típusok deklarálása};
  Var {változók deklarálása};

  Procedure eljárásnév(paraméterek);
  {deklarációk}
  Begin
    {az eljárás törzsének utasításai}
  End;
```



```

Function függvéynév(paraméterek):típus;
{deklarációk}
Begin
  {a függvény törzsének utasításai}
  Függvéynév:=utasítás; {a függvény eredménye}
End;

```

```

{a főprogram blokkja}
Begin
  {a főprogram törzsének utasításai}
End.

```

13. Egyszerű adattípusok

Egész típusok

Shortint	-128..127
Integer	-32768..32767
Longint	-2147483648..2147483647
Byte	0..255
Word	0..65535

Valós típusok

Real	2.9E-39..1.7E38
Single	1.5E-45..3.4E38
Double	5.0E-324..1.7E308
Extended	3.4E-4932..1.1E4932
Comp	$-2^{63}+1..2^{63}-1$

Logikai típus boolean

Szöveges típusok

Char	egy karakter
String	max. 255 karakterből álló karaktersorozat

14. Struktúrált adattípusok

Tömb

y:array[1..10]of integer	egydimenziós egészekből álló tömb, elemei: y[1], y[2], ..., y[10]
x:array[1..2,1..3]of string	kétdimenziós szövegekből álló tömb, elemei: x[1,1], x[1,2], x[1,3], x[2,1], x[2,2], x[2,3]
a:array[1..2,1..2,1..3]of real	háromdimenziós valós számokból álló tömb, elemei: a[1,1,1], a[1,1,2], a[1,1,3], a[1,2,1], a[1,2,2], a[1,2,3], a[2,1,1], a[2,1,2], a[2,1,3], a[2,2,1], a[2,2,2], a[2,2,3]

Rekord

```
Deklaráció:      var datum:record
                  Ev:1000..2000;
                  Honap:1..12;
                  Nap:1..31;
                  Megjegyzes:string;
                  End;
```

```
Értékkadás:     datum.ev:=1974;
                  Datum.honap:=3;
                  Datum.nap:=13;
                  Datum.megjegyzes:='Zoli szuletesnapja';
```

File

```
Var adat:file of integer;
```

```
Var datumok:file of datum;
```

15. Logikai műveletek

Not	tagadás
And	és
Or	vagy
Xor	kizáró vagy

16. Összehasonlító műveletek

=	egyenlő
<>	nem egyenlő
<	kisebb
>	nagyobb
<=	kisebb vagy egyenlő
>=	nagyobb vagy egyenlő

17. Sztringek összekapcsolása

```
a:='Kiskun';
b:='lacháza';
c:=a+b;
```

A c változó tartalma 'Kiskunlacháza'.

18. Szabványos függvények

Matematikai függvények

Abs(x)	x abszolút értéke
Cos(x)	a radiánban megadott x koszinusza

Frac(x)	x törtrésze
Int(x)	x egészrésze valós típusként
Pi	pi értéke
Random	0 <= véletlenszám < 1 valós érték
Random(n)	n egész, 0 <= egész véletlenszám < n
Randomize	a véletlenszám generátornak véletlen kezdőértéket ad
Round(x)	x egészre kerekített értéke
Sin(x)	a radiánban megadott x szinusza
Sqr(x)	x^2
Sqrt(x)	x négyzetgyöke
Trunc(x)	x egészrésze egész típusként

Szövegkezelő függvények

Concat(s1,s2,..)	sztringek összefűzése
Copy(s,i,c)	az s sztring i. sorszámú karakterétől számítva c darab karaktert kimásol
Delete(s,i,c)	az s sztring i. sorszámú karakterétől számítva c darab karaktert töröl
Insert(f,s,i)	az f sztringet az s sztringbe szúrja az i. karakter elé
Length(s)	az s sztring karaktereinek száma (hossza)
Pos(x,s)	az x sztringet keresi az s sztringben
Str(x,s)	az x egész számot az s sztringbe konvertálja
Str(x:w:d,s)	az x valós számot w hosszúságú, d tizedest ábrázoló s sztringbe konvertálja
Uppcase(c)	a c kisbetűt nagybetűvé alakítja
Val(s,v,code)	az s sztringet, ha lehet v számmá konvertálja, ha nem sikerül, akkor a code változó az első hibás karakter pozícióját fogja tartalmazni.
Inttostr(x)	az x egész számot szöveggé alakítja.
Strtoint(s)	az s szöveget egész számmá alakítja

19. A Pascal nyelv utasításai

Üres utasítás

;

Értékadó utasítás

x:=3.5;

s:='szoveg';

y:=w;

z:=x*y+sqrt(x);

x:=x+1;

Elemezzük őket!

FELADAT: Készítsünk olyan programot, amely bekér két számot, majd kiírja a számtani és Mértani közepüket!

Két változó (x, y) tartalmának a felcserélése

HIBÁS MEGOLDÁS

x:=y; y:=x;

Miért nem jó így?

JÓ MEGOLDÁS

z:=x; x:=y; y:=z;

Kell egy z segédváltozó

Logikai értékadás

b1:=true;

b2:=false;

b3:=i<j;

b4:=b1 and b2;

b5:=b2 or b1 and b2;

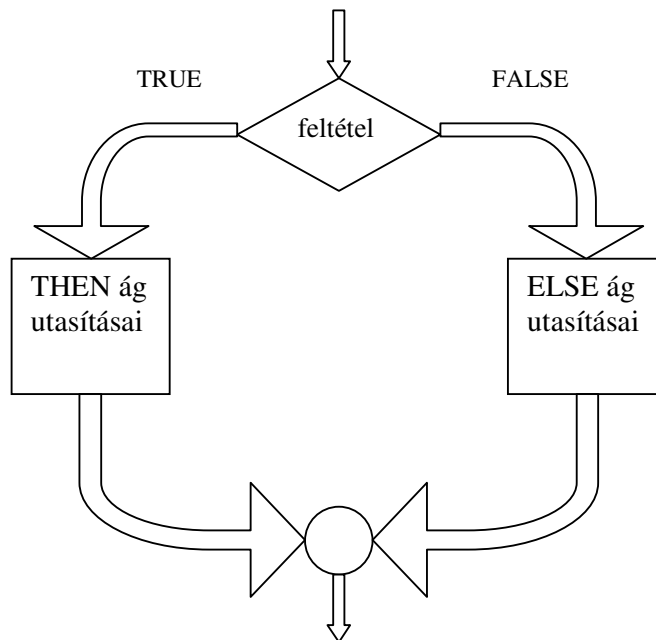
b6:=b1;

b7:=(i=3) or (j>3);

b8:=(b1 or b2) and (j+2<i*3);

20. Feltételes utasítások

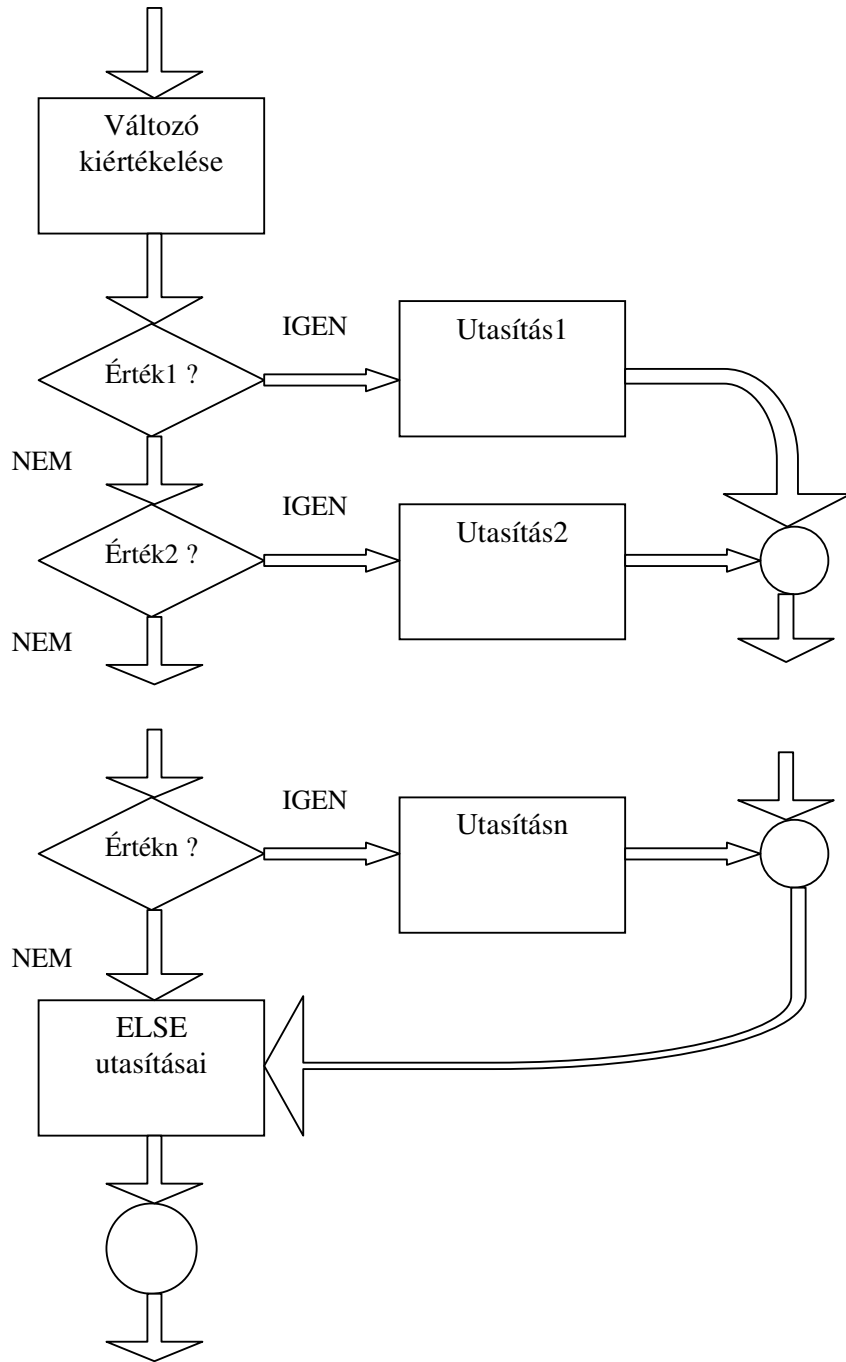
If feltétel then utasítás1 else utasítás2;



FELADAT: Írj olyan programot, amely bekéri a víz hőmérsékletét celsius fokokban, majd kiírja a víz halmazállapotát!

FELADAT: Egy dolgozat értékelése a következő volt. 0..49 elégtelen, 50..59 elégséges, 60..69 közepes, 70..89 jó, 90..100 jeles. Írj olyan programot, amely bekéri az elért pontszámot, majd kiírja az elért osztályzatot szöveggel!

Case változó of
Érték1:utasítás1;
Érték2:utasítás2;
...
Értékn:utasításn;
Else utasítás;
End;



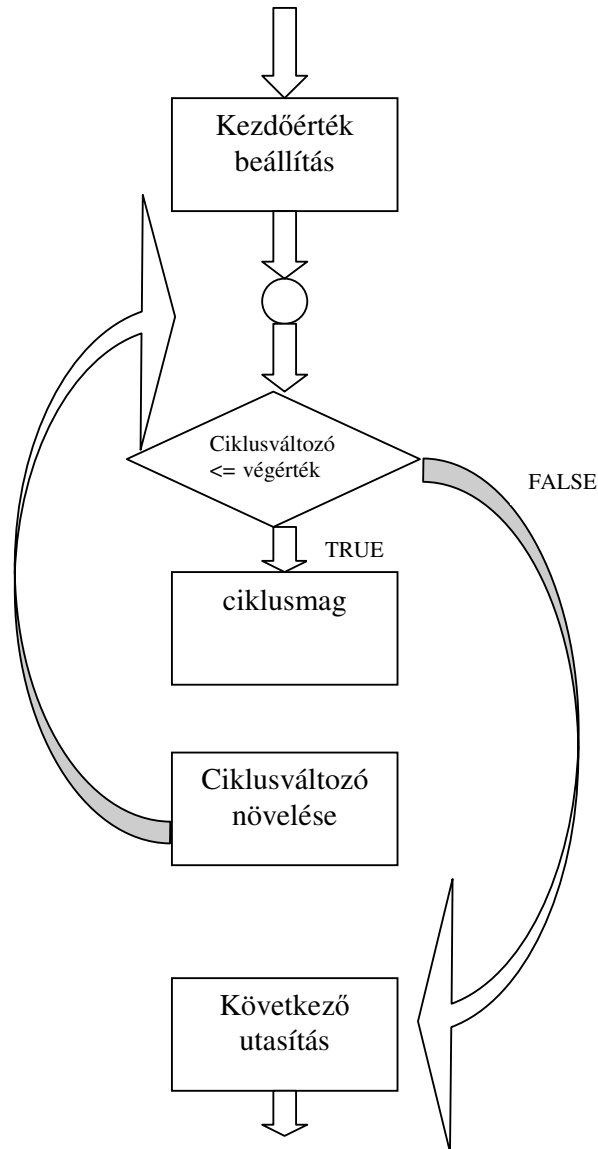
FELADAT: Az osztályozó programot készítsd el case utasítással is!

21. Ciklusutasítások

For ciklusváltozó:=kezdőérték to végérték do ciklusmag;

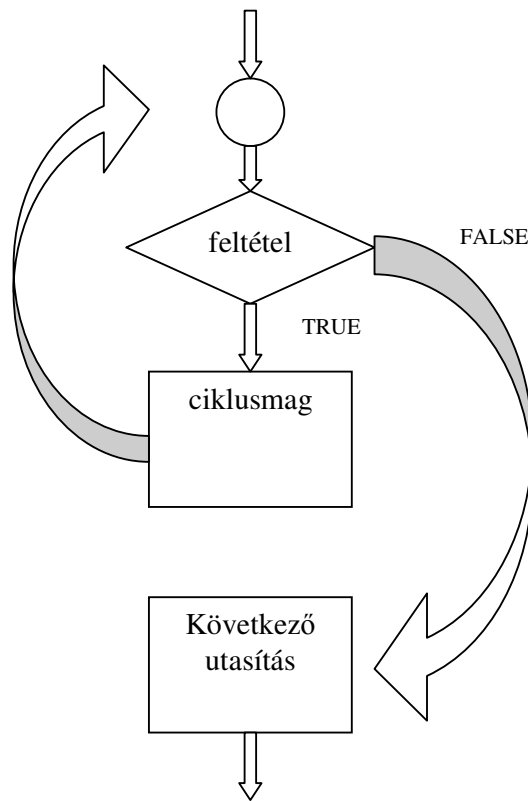
Illetve

For ciklusváltozó:=kezdőérték downto végérték do ciklusmag;



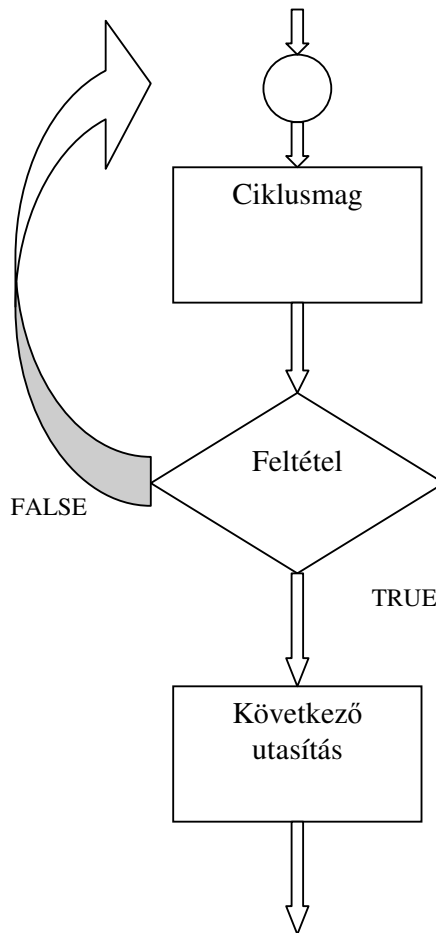
FELADAT: Add össze az első húsz egész számot FOR ciklus segítségével!

While feltétel do ciklusmag;



FELADAT: Add össze az első húsz egész számot while utasítás segítségével!

```
Repeat  
  Ciklusmag;  
Until feltétel;
```



FELADAT: Add össze az első 20 egész számot a repeat utasítás segítségével!

22. Feladatok

- I. Olvasd be a téglatest adatait és számítsd ki a téglatest felszínét és térfogatát!
- II. Számítsd ki az első n négyzetszám összegét! Az n értékét a program indulásakor kelljen megadni!
- III. N darab egész számot egy tömbbe olvass be, számold meg, hogy hány eleme negatív, nulla és pozitív!
- IV. N darab egész számot egy tömbbe olvass be, számold meg hány páros és páratlan eleme van a tömbnek, és hány eleme osztható hárommal!
- V. N darab egész számot egy tömbbe olvass be, számold meg hány eleme egyjegyű, kétjegyű, háromjegyű, több mint három jegyű!
- VI. Írj programot, amely bekéri n darab tanuló nevét és megtakarított pénzét! Az adatokat táblázatos formában írja vissza, valamint számítsa ki a megtakarított pénzek összegét is!
- VII. Készíts egy egyszerű számológép programot, amely az adatokat `szam1`, `szam2` alakban várja, a műveletet `+`, `-`, `*`, `/` valamelyikeként. A programból az `e` betűvel lehessen kilépni.
- VIII. Olvasd be egy egész számot, majd írd ki fordítva!

- IX. Írj egy programot, amely 0 és 100 között generál egy véletlenszámot. A program párbeszédés formában kérdezzen a számra, és segítségül jelezze, hogy a kapott szám kevés vagy sok a kitalálandó számhoz képest. A lépéseket számlálja, amelyet a szám kitalálásakor írja vissza!
- X. N darab kockadobásból számold meg, mennyi 1,2,3,4,5 és 6 volt a dobás értéke!
- XI. Milliméterben megadott mennyiséget számíts át méter, centiméter és milliméterre!
- XII. Másodpercben megadott időt számíts át óra, perc, másodpercre!
- XIII. Egy tetszőleges bevitt szövegben a szóközöket cseréld *-ra! Pl.: 'Ma szép idő van.', 'Ma*szép*idő*van.'
- XIV. Írj programot, amely egy évszámról megmondja, hogy szökőév-e?
- XV. Írj programot egy bekért egész szám római alakra való átírásához!
- XVI. Írj ötös lottó kitöltéséhez generáló programot!
- XVII. Személyi adatok tárolására készíts rekordot. Írj olyan programot, amely az ilyen típusú elemekből álló tömböt feltölti, majd adott elemét listázza!
- XVIII. Személyi adatok tárolására készíts rekordot. Írj olyan programot, amely az ilyen típusú elemekből álló tömböt feltölti, majd összes elemét listázza táblázatos formában!

23. Függvények

Készítsünk programot, amely egy adott egész szám harmadik hatványát (köbét) írja ki! A megoldáshoz készítsünk saját függvényt!

Program harmhatv;

Var szam:integer;

Function KOB(x:integer):integer;

Begin

KOB:=x*x*x;

End;

Begin

Writeln('Kerem a szamot!');

Readln(szam);

Writeln('A szám harmadik hatványa: ',KOB(szam));

Readln;

End.

A program áttekinthetőbb lett. A főprogram törzse csak az i/o műveleteket tartalmazza.

Készítsünk programot, amely egy háromszög oldalából előállítja a területét!

Program haromszogter;

Var x,y,z:real;

Function terület(a,b,c:real):real;

Var s:real;

Begin

s:=(a+b+c)/2;

terület:=sqrt(s*(s-a)*(s-b)*(s-c));

end;

begin

write('a oldal:');readln(x);

```

write('b oldal:');readln(y);
write('c oldal:');readln(z);
writeln('A háromszög területe:',terulet(x,y,z):6:2);
readln;
end.

```

24. Eljárások

Írjunk programot, amely bekéri egy háromszög oldalait, majd kiírja a területét!

```

Program haromszog;
Var x,y,z,t:real;
  Procedure terulet(a,b,c:real; var ter:real);
  Var s:real;
  Begin
    s:=(a+b+c)/2;
    ter:= sqrt(s*(s-a)*(s-b)*(s-c));
  end;
begin
write('a oldal:');readln(x);
write('b oldal:');readln(y);
write('c oldal:');readln(z);
terulet(a,b,c,t);
writeln('A háromszög területe:',t:6:2);
readln;
end.

```

Írjunk olyan programot, amely kiszámítja egy háromszög területét, kerületét, ellenőrzi az adatok helyességét, a beolvasást és kiírást is eljárás végzi!

```

Program haromszog;
Var x,y,z,k,t:real;
  Procedure beolvas(var a,b,c:real);
  Var b:boolean;
  Begin
    Repeat
      b:=false;
      write('a oldal:');readln(a);
      write('b oldal:');readln(b);
      write('c oldal:');readln(c);
      if (a+b>c) and (a+c>b) and (b+c>a) then b:=true else writeln('Hibás adatok!');
    until b;
  end;
  function terulet(a,b,c:real):real;
  var s:real;
  begin
    s:=(a+b+c)/2;
    terulet:= sqrt(s*(s-a)*(s-b)*(s-c));
  end;
  function kerulet(a,b,c:real):real;

```

```

begin
    kerulet:=a+b+c;
end;
procedure kiir(ker,ter:real);
begin
    writeln(' A haromszog terulete: ', ter:6:2);
    writeln(' A haromszog kerulete: ', ker:6:2);
end;
begin
    beolvas(x,y,z);
    t:=terulet(x,y,z);
    k:=kerulet(x,y,z);
    kiir(k,t);
    readln;
end.

```

A program így nagyon jól áttekinthető, az esetleges hibák könnyebben megtalálhatók.

25. Feladatok

- I. Írj programot eljárással és függvénnyel a téglalap kerületének és területének kiszámítására!
- II. Írj programot eljárással és függvénnyel a Pitagorasz-tétel alkalmazására!
- III. Írj programot eljárással és függvénnyel a Celsius fokok és a Kelvin oda-visszaalakítására!
- IV. Írj programot eljárással és függvénnyel, amely az alábbi n tagú sor összegét számítja ki: $h=1+1/2+1/3+\dots+1/n$
- V. Írj programot eljárással, amely egy adott mondatot szavaira bont és a szavakat sztring tömbben adja vissza!
- VI. Írj programot eljárással és függvénnyel, amely adott karakterből adott hosszúságú sztringet állít elő. Pl. ha a bemenő paraméterek: 'a', 5, akkor az eredmény: 'aaaaa'
- VII. Írj programot eljárással és függvénnyel, amely meghatározza két egész szám legnagyobb közös osztóját!
- VIII. Írj programot eljárással és függvénnyel, amely beolvas egy n elemű tömböt, és kiírja a tömb elemeinek átlagát!
- IX. Írj programot eljárással és függvénnyel, amely kiszámítja 1-től n-ig az egész számok összegét!

26. Szöveges fájlok kezelése

A fájl változónak textfile típusúnak kell lennie.

```
Var f:textfile;
```

A fájlnev hozzárendelése a változóhoz:

```
Assignfile(f, 'c:\konyvtar\szoveg.txt');
```

A szövegfájl megnyitása:

Reset(f); csak olvasásra

Rewrite(f); új szövegfájlt hoz létre, a már létezőt újraírja

Append(f); hozzáírásra

I/O műveletek:

Write(f, szoveg1); a szoveg1 sztring beírása a fájlba

Writeln(f, szoveg2); a szoveg2 sztring soremeléssel kerül be a fájlba

Read(f, szoveg1); a fájlból a soron következő szövegrész a szoveg1 változóba kerül.

Readln(f, szoveg2); a fájlból a soron következő sor a szoveg2 változóba kerül;

Eof(f) : igaz, ha a fájl végére értünk, egyébként hamis.

A szövegfájl lezárása:

Closefile(f);

Készítsünk programot, amely bekéri egy dolgozatot író tanulók eredményeit, és azok érdemjegyeit, majd ezt egy szöveges fájlba tárolja!

Program szovegkiiras;

```
var f:textfile;
```

```
    letszam:integer;
```

```
    nev:array[1..50]of string;
```

```
    jegy:array[1..50]of integer;
```

```
procedure beolvas;
```

```
var i:integer;
```

```
begin
```

```
    write('A tanulok szama: '); readln(letszam);
```

```
    for i:=1 to letszam do
```

```
        begin
```

```
            write('A tanulo neve: '); readln(nev[i]);
```

```
            write('Osztalyszat: '); readln(jegy[i]);
```

```
        end;
```

```
end;
```

```
procedure kiir;
```

```
var i:integer;
```

```
    szoveg:string;
```

```
begin
```

```
    assignfile(f, 'dolgozat.txt');
```

```
    rewrite(f);
```

```
    for i:=1 to letszam do writeln(f, nev[i]:40+jegy[i]);
```

```
    closefile(f);
```

```
end;
```

```
begin
```

```
beolvas;  
kiir;  
end;
```

A keletkezett szövegfájl tartalmát ellenőrizzük Jegyzetömbbel!

Készítsünk egy szöveges fájlt Jegyzetömbbel! Készítsünk programot, amely a tartalmát a képernyőre listázza!

```
Program lista;  
Var f:textfile;  
    Filename, szoveg:string;  
Begin  
    Write('Kérem a fájl nevét! '); readln(filename);  
    Assignfile(f, filename);  
    Reset(f);  
    While not eof do  
        Begin  
            Readln(f, szoveg);  
            Writeln(szoveg);  
        End;  
    Readln;  
End.
```

27. Típusos fájlok kezelése

Készítsünk olyan típusos fájlt, ahol a dolgozat eredményeit úgy tároljuk, hogy a név szöveggént, az osztályzat számként kerül tárolásra. (Takarékosabb helyfoglalás.)

Program dolgozat;

```
Type dogaered=record  
    Nev:string[40];  
    Jegy:integer;  
End;
```

```
Var letszam,i:integer;  
    Adat:doгаered;  
    f: file of doгаered;
```

```
Begin  
    Write('Kérem a tanuló letszámát! '); readln(letszam);  
    Assignfile(f, 'dolgozat.dat');  
    Rewrite(f);  
    For i:=1 to letszam do  
        Begin  
            Write('A tanuló neve: '); readln(adat.nev);  
            Write('A tanuló osztályzata: '); readln(adat.jegy);  
            Write(f, adat);  
        End;  
    End;
```

```
Closefile(f);  
End;
```

A fájl tartalma legegyszerűbben a Total Commander nézőkéjével tekinthető meg.

Készítsünk olyan programot, amely az előbb letárolt adatokat a képernyőre listázza!

Program adatlista;

```
    Type dogared=record  
        Nev:string[40];  
        Jegy:integer;  
    End;  
  
    Var Adat:dogared;  
        f: file of dogared;  
  
begin  
    assignfile(f, 'dolgozat.dat');  
    reset(f);  
    while not eof(f) do  
        begin  
            read(f, adat);  
            writeln(adat.nev+adat.jegy);  
        end;  
    closefile(f);  
    readln;  
end.
```

28. Feladatok

I. Írj programot, amely az

```
    Type rt=record  
        Tipus:string[12]  
        Ear:real;  
        Db:integer;  
    End;
```

rekordokat tartalmazó fájlt létrehozza, kiírja annak tartalmát, miközben az ear mezőket összegzi!

II. Egy szöveges fájl egész számokat tartalmazzon a soraiban! Írj programot, amely a szövegfájl sorait egy egész típusú tömbbe olvassa, majd kiszámítja az összegüket!

III. Írj egy programot, amely egy szöveges fájl első 4 sorát egy másik szöveges fájlba másolja!

29. A Windows alkalmazások szerkezete

TApplication

A Delphi a windows alkalmazások készítéséhez a TApplication osztályt definiálja.

Belépéskor, vagy új alkalmazás létrehozásakor (File > New Application), automatikusan elkészül egy alap forrásnyelvű főprogram, amely a következőképpen néz ki. (View>Units)

```
program Project1;
```

```
uses
```

```
  Forms,
```

```
  Unit1 in 'Unit1.pas' {Form1};
```

```
{ $R *.RES }
```

```
begin
```

```
  Application.Initialize;
```

```
  Application.CreateForm(TForm1, Form1);
```

```
  Application.Run;
```

```
end.
```

A főprogram DPR kiterjesztéssel kerül mentésre.

A projekthez csatlakozó erőforrások RES kiterjesztést kapnak.

Az ablakkezelő modulok PAS kiterjesztésűek.

Az ablakok adatai DFM kiterjesztésű fájlba mentődnek.

30. Az alkalmazások ablaka

Amikor új projektet nyitunk, akkor az ablak modellje látható a jobb oldalon és az objektum-felügyelő a bal oldalon.

A Properties fülön állítható a tulajdonságok értéke, az Events fülön az eseménykezelő eljárások.

Az ablakkezelő modul forráskódja alapértelmezetten a következőképpen néz ki:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
```

```
type
```

```
  TForm1 = class(TForm)
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    { Public declarations }
```

```

end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

end.

```

Ha futtadjuk, akkor a program már működőképes, igaz, hogy csak egy üres ablak, de a gombjai működnek, mozgatható, méretezhető.

Módosítsuk a programot úgy, hogy induláskor az ablak színe piros legyen! Erre két lehetőségünk van:

- a) a properties fülön a form1 color tulajdonságát clRed-re állítjuk.
- b) Az events fülön módosítjuk az onCreate eljárást (amely az ablak létrejöttkor fut le) a következőképpen:

```

procedure TForm1.FormCreate(Sender: TObject);
begin
  color:=clred; {értékadó utasítás}
end;

```

Módosítsuk a programot úgy, hogy a bal egérgombbal kattintva az ablak sárga színű legyen!

Az onClick eljárást kell a következőképpen módosítanunk:

```

procedure TForm1.FormClick(Sender: TObject);
begin
  color:=clyellow;
end;

```

Módosítsuk a programot úgy, hogy újra kattintva az ablakon, az ablak újra piros színű legyen, és a két szín váltakozzon!

Az onClick eljárást kell módosítani.

```

procedure TForm1.FormClick(Sender: TObject);
begin
  if color=clyellow then color:=clred else color:=clyellow;
end;

```

Elemezzük az utasítást!

Módosítsuk a programot úgy, hogy bezáráskor azt hagyassuk jóvá a felhasználóval!

Az onClose eljárást kell módosítanunk.


```

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  if messagedlg('Zárjuk az ablakot ?', mtconfirmation, [mbytes,mbno],0)=mryes
  then action:=cafree else action:=canone;
end;

```

Elemezzük az utasítást!

Módosítsuk a programot úgy, hogy az egér mozgásokor jelezze ki a kurzor pozícióját az ablak címsorában!

Az onMouseMove eljárást kell módosítanunk.

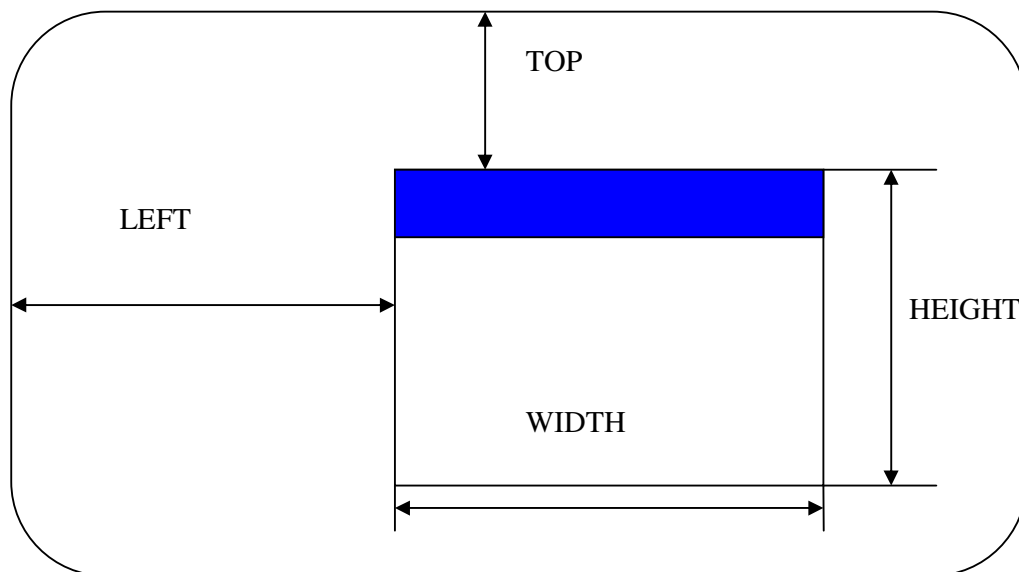
```

procedure TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  caption:=inttostr(x)+' , '+inttostr(y);
end;

```

Elemezzük az eljárást!

31. Az ablakok tulajdonságai



Windowstate	wsnormal	-	normál
	Wsmimized	-	minimalizált
	Wsmaximized	-	maximalizált
Formstyle	fsnormal	-	szokásos
	Fsstayontop	-	mindig a legfölső

Caption	-	felirat
Bordericons	-	a rendszermenü gombjai közül melyik kerüljön az ablakra
Borderstyle	-	bsnone - nincs keret, nem méretezhető át Bssingle - egyvonalas keret, nem méretezhető át Bsdiallog - párbeszédablak keret, nem méretezhető Bssizeable - szabványos, méretezhető Bstoolwindow - a bssingle kisebb betűs címmel Bssizetoolwin - a bssizeable kisebb betűs címmel

Többféle kurzormutató közül választhatunk, ezek egész számmal (0..-22) adhatók meg.

Írjunk programot, amely kattintásra változtatja az egérmutatót, és a címsorba kiírja a megfelelő értéket!

```

procedure TForm1.FormClick(Sender: TObject);
const eger:integer=0;
begin
  eger:=eger-1;
  caption:=inttostr(eger);
  cursor:=eger;
  if eger<-22 then eger:=0;
end;

```

32. Feladatok

- I. Készítsünk programot, amelyből menü és nyomógomb segítségével is ki lehet lépni! Kilépéskor kérdezzen rá, hogy a felhasználó valóban ki akar-e lépni!
- II. Írjunk alkalmazást, amelynek ablakában a DELPHI felirat egérrel felemelhető, elmozgatható és letehető!
- III. Készítsünk olyan alkalmazást, amelynek címsora a billentyűzetről begépelve módosítható! (A backspace billentyű virtuális kódja vk_back, a chr függvény tér vissza a kódnak megfelelő karakterrel.) (onkeydown)
- IV. Készítsünk alkalmazást, amely a billentyűzetről begépelte szöveget jeleníti meg címkékben, az egyikben normál, a másikban fordított irányban! (onkeypress)
- V. Készíts programot, amelyen egy nyomógomb mindig ott jelenik meg, ahová kattintunk!
- VI. Készíts programot, amely gombnyomásra módosítja az egérkurzor alakját!

33. Menük létrehozása

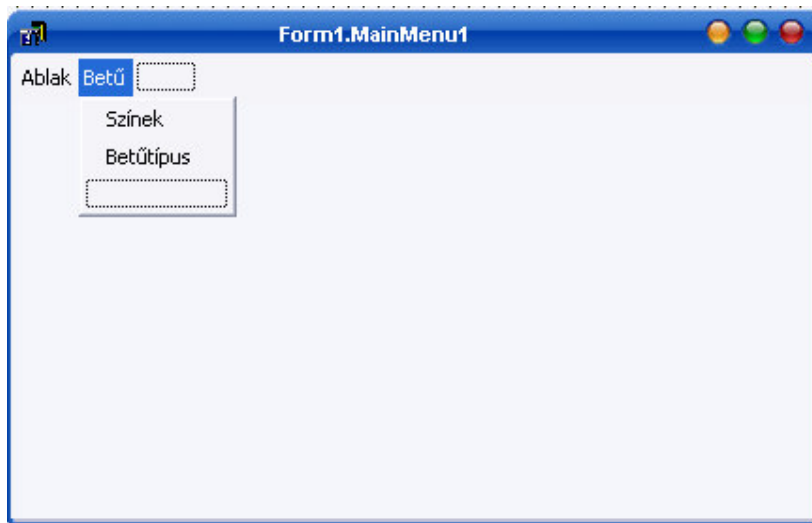
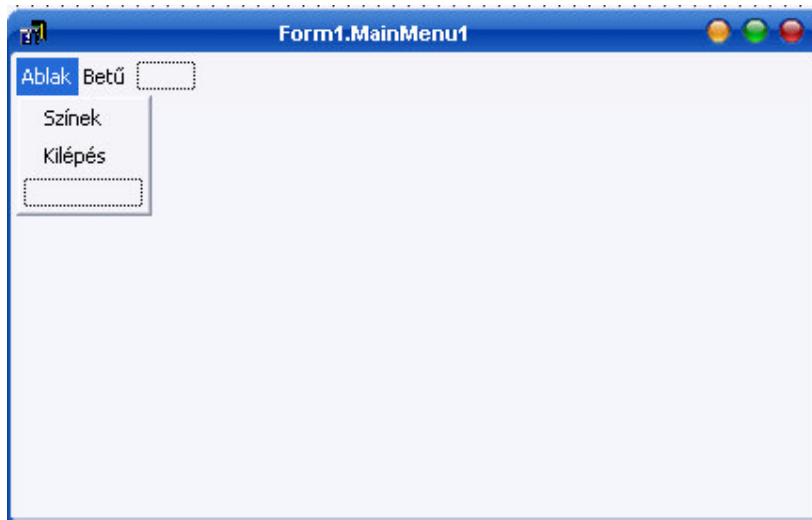
Készítsünk menüvezérelt programot, amely módosítja az ablak színét, az ablakon lévő betű színét, típusát. A menühierarchia legyen a következő:

```

Ablak > Színek, Kilépés
Betű > Színek, Betűtípus

```

A mainmenu komponenset kell használnunk.



Szükség van még egy címkére, valamint a colordialog és a fontdialog komponensre
A szükséges eljárások:

```
procedure TForm1.Sznek1Click(Sender: TObject);  
begin  
    if colordialog1.Execute then color:=colordialog1.Color;  
end;
```

```
procedure TForm1.Kilps1Click(Sender: TObject);  
begin  
    close;  
end;
```

```
procedure TForm1.Sznek2Click(Sender: TObject);  
begin  
    if colordialog1.Execute then label1.Color:=colordialog1.Color;  
end;
```

```
procedure TForm1.Bettpus1Click(Sender: TObject);
begin
  if fontdialog1.Execute then label1.Font:=fontdialog1.Font;
end;
```

34. Feladatok

- I. Készíts olyan programot, amely a következő menürendszerrel rendelkezik:
Menü > Szöveg > Szöveg megjelenítése F1; Szöveg eltüntetése F2; Kilépés
A megfelelő eljárásokat egy címkével végezze el.
- II. Készíts olyan menüvezérelt programot amely az előzőt úgy módosítja, hogy a szöveg színét be lehet állítani!
- III. Az előző programot módosítsuk úgy, hogy rendelkezzen egy színes szöveg kipipálható menüponttal. Ha nincs kipipálva, akkor a szöveg fekete legyen és ne lehessen beállítani a színét!
- IV. Az előző programot módosítsuk úgy, hogy hangjelzés történjen, ha valaki a színes szöveg kipipálása nélkül próbálja a szöveg színének átállítását kezdeményezni!
- V. Készítsünk olyan programot, melyben felbukkanó menü segítségével választhatunk a 2 gyökének, négyzetének megjelenítése, valamint az eredmény törlése között!
- VI. Módosítsuk az előző programot úgy hogy a funkciók menüből is legyenek elérhetők!
- VII. Készítsünk olyan programot, amely felbukkanó menü segítségével állítja be a betűszínt és a betűtípust!

35. Vezérlőgombok

Nyomógomb

Készítsünk programot, amely három nyomógombot tartalmaz Piros, Sárga, Kék felirattal, és a feliratnak megfelelően módosítja az ablak színét!

A Piros gomb eljárása példaként:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  color:=clred;
end;
```

A Jelölőnégyzet (Checkbox)

Az előző programot módosítsuk úgy, hogy helyezzünk el két jelölőnégyzetet is Színezhető és Hangjelzés felirattal. Az ablak csak akkor legyen színezhető, ha az első jelölőnégyzet kiválasztott, ha nem választották ki, és mégis módosítani szeretnénk a színt, akkor a Hangjelzés bekapcsolt állapotában hangjelzést hallasson a program!

A Piros gomb eljárása példaként:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  if checkbox1.checked then color:=clred else
  if checkbox2.checked then beep;
end;
```

A Radiobutton

Az előző programot módosítuk úgy, hogy nyomógombok helyett rádiógombok szerepeljenek!

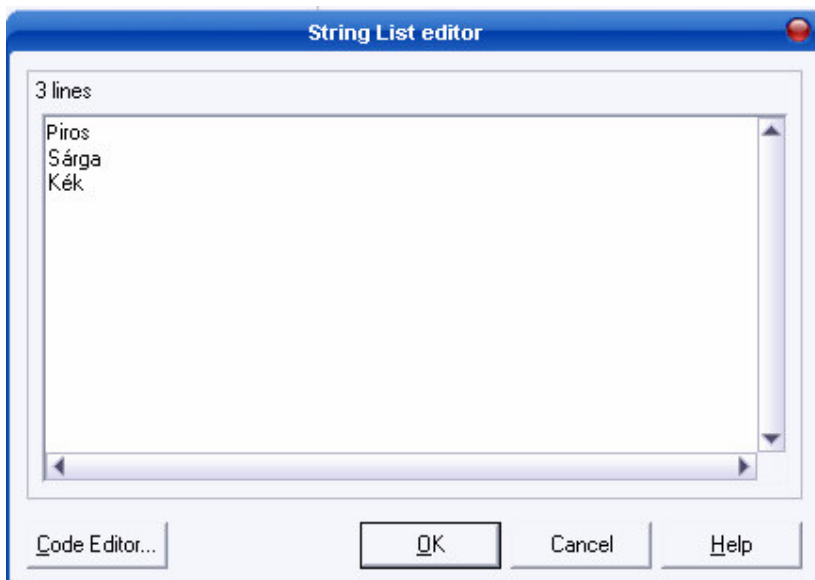
A Piros radiobutton eljárása példaként:

```
procedure TForm1.RadioButton1Click(Sender: TObject);
begin
  if checkbox1.checked then color:=clred else
    if checkbox2.checked then beep;
end;
```

A Radiogroup

Az előző programot módosítuk úgy, hogy különálló radiobuttonok helyett radiogroupot használunk.

A bejegyzéseket az Items tulajdonsághoz kell felvenni!



```
procedure TForm1.Radiogroup1Click(Sender: TObject);
begin
  case radiogroup1.ItemIndex of
    0:color:=clred;
    1:color:=clyellow;
    2:color:=clblue;
  end;
end;
```

36. Feladatok

- I. Készíts programot, amely nyomógombok használatával kiszámítja a kör területét, illetve területét!
- II. Módosítsuk az előző programot úgy, hogy csak egy nyomógombja legyen. Azt, hogy mit számítunk, rádióbuttonnal lehessen kiválasztani!
- III. Írjunk olyan programot, ahol rádiógombbal dönthetünk arról, hogy egy kocka felszínét, térfogatát, oldalatlóját vagy testatlóját számítsuk ki!
- IV. Írjunk programot a gömb felszínének, illetve térfogatának kiszámításához, a választást mindenképp radiogroup használatával oldjuk meg!
- V. Készítsünk programot, amely jelölőnégyzetek segítségével befolyásolja egy címke betűstílusát! (dőlt, félkövér, aláhúzott)
(if checkbox1.checked then label1.Font.style:=label1.Font.style+[fsbold]
else label1.Font.style:=label1.Font.style-[fsbold];
if checkbox2.checked then label1.Font.style:=label1.Font.style+[fsitalic]
else label1.Font.style:=label1.Font.style-[fsitalic];
if checkbox3.checked then label1.Font.style:=label1.Font.style+[fsunderline]
else label1.Font.style:=label1.Font.style-[fsunderline];)

37. Scrollbar

Min, max	-	a csúszka szélső helyzetei
Position	-	aktuális pozíció
Kind	-	sbHorizontal: vízszintes, sbVertical: függőleges

Írjunk olyan programot, amely a címsorba kiírja a csúszka pozícióját!

```
procedure TForm1.ScrollBar1Change(Sender: TObject);  
begin  
  caption:=inttostr(scrollbar1.position);  
end;
```

38. Feladat

- I. Készíts olyan programot, amely egy vízszintes és egy függőleges gördítősávot tartalmaz, és úgy mozognak, hogy a pozícióik összege mindig 101!
- II. Készíts görgetősávos színkeverőt!
(RGB(piros, zöld, kék); a színek intenzitása 0..255)

39. Szöveg megjelenítők

Label (Címke)

Caption	-	a címke szövege
Transparent	-	átlátszó
Wordwrap	-	többsoros
Font	-	betűtípus

Edit

Maxlength	-	maximális hossz (0 esetén bármekkora)
ReadOnly	-	nem szerkeszthető
Seltext	-	a kiválasztott szöveg
Sellength	-	a kiválasztott szöveg hossza
Selstart	-	a kiválasztott szöveg első karakterének pozíciója
Passwordchar	-	jelszókarakter

Készíts olyan programot, amelyben két szövegbeviteli mező van, az egyik nem szerkeszthető, viszont mindig azt a szöveget tartalmazza, amit a másik!

```
procedure TForm1.Edit1Change(Sender: TObject);
begin
  edit2.Text:=edit1.Text;
end;
```

clear	-	törli a tartalmát
clearselection	-	törli a kijelölt szöveget
copytoclipboard	-	a kiválasztott szöveget vágólappra másolja
cuttoclipboard	-	a kiválasztott szöveget vágólappra másolja, az eredetit törli
pastefromclipboard	-	a vágólapon található szöveget a beviteli mezőbe másolja
selectall	-	a teljes szöveget kijelöli

Memo

Lines	-	a sorok bevitel, szerkesztése
Scrollbar	-	hol legyen a gördítősáv

Add, delete, clear, insert

Nagyon sok vezérlő ugyanaz, mint az edit vezérlőnél.

40. Feladatok!

- I. Készíts alkalmazást, amely egy címkére való kattintáskor pirosra állítja annak szövegét, dupla kattintáskor pedig feketére!
- II. Olvassunk be egy jelszót, ami 5 betűből állhat maximum! A karakterek helyén * jelenjen meg! A program jelezze az elfogadást, illetve a hibás megadást!
- III. A memo felhasználásával készíts olyan programot, amellyel txt fájlokat lehet menteni, illetve beolvasni (egyszerűsített Jegyzetomb)!

41. Listás vezérlők

Listbox

Items	-	lista elemei
Multiselect	-	több elem is kiválasztható
Itemindex	-	a kiválasztott elem sorszáma (0 az első, stb. Ha nincs, akkor -1)
Sorted	-	rendezett

Készíts olyan programot, amely az ablak címsorába írja az elem sorszámát!

```
procedure TForm1.ListBox1Click(Sender: TObject);
begin
  caption:=inttostr(listbox1.itemindex);
end;
```

Módosítsd úgy a programot, hogy a kiválasztott elem kerüljön a címsorba!

```
procedure TForm1.ListBox1Click(Sender: TObject);
begin
  caption:=listbox1.items[listbox1.itemindex];
end;
```

Készíts olyan programot, amelyben a listadoboz a piros, sárga, kék szavakat tartalmazza, és ennek megfelelően képes megváltoztatni az ablak színét!

```
procedure TForm1.ListBox1Click(Sender: TObject);
begin
  case listbox1.itemindex of
    0:form1.Color:=clred;
    1:form1.Color:=clyellow;
    2:form1.Color:=clblue;
  end;
end;
```

Combobox

Egy edit és egy listbox összeépítéséből keletkezik, tulajdonságait ezektől örökli.

Dropdowncount - a megjelenített tételek száma

42. Feladatok

- I. Készíts programot, amely képes egy edit mezőbe beírt szöveget egy listboxba írni, majd gombnyomásra a listbox tartalmát rendezni!
- II. Módosítsd az előző programot úgy, hogy másik gombot lenyomva törlésre is legyen lehetőség!
- III. Készítsünk olyan programot, amely két listaablakot tartalmaz, mindegyikhez lehet új elemeket felvenni, és az egyikben kijelölt elemeket a másikba át lehet pakolni!
- IV. Készíts olyan programot, amely egy combobox-ban kiválasztott fájlnevű bittérképes grafikát jelenít meg!

43. Az időzítő

Készítsünk olyan programot, amely másodpercenként növel egy megjelenített értéket!

```
procedure TForm1.FormCreate(Sender: TObject);
begin
    timer1.Interval:=1000; //millisecundumban
    timer1.Enabled:=0;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    label1.Caption:=inttostr(strtoint(label1.caption)+1);
end;
```

44. Feladat

- I. Készíts digitális órát!
(az időt a time függvény kérdezi le, az időt timetostr függvény alakítja szöveggé!)
- II. Készíts olyan alkalmazást, amely 5 másodpercenként váltogatja a megjelenített bittérképes grafikát!

45. Több ablak

Form2.showmodal;

Készíts olyan programot, amely a bekért két szám szorzatát egy új ablakban jeleníti meg!

```
procedure TForm1.Button1Click(Sender: TObject);
begin
    form2.label1.caption:=inttostr(strtoint(form1.edit1.text)*strtoint(form1.edit2.text));
    form2.showmodal;
end;
```

45. Gyakorlófeladatok

- I. Készíts programot, amely kiszámítja egy beolvasott szám négyzetgyökét!
- II. Készíts programot, amely kiszámítja két szám szorzatát, a számokat vízszintes és függőleges Trackbar vezérlőkkel lehessen beállítani!
- III. Készítsünk olyan programot, amelyben menüből választhatjuk ki, hogy kör, vagy téglalap legyen látható, az alakzatok színének beállításához használjunk colordialogot!
- IV. Készíts olyan programot, amely megjeleníti a dátumot és az időt!
- V. Készíts ébresztőóra programot!
- VI. Készíts olyan programot, amely téglalapot rajzol, a paramétereit egy új ablakon kérjük be! A színezés colordialoggal történjen!
- VII. Készíts szorzótábla gyakorlító programot, amely közli, hogy jó e a kiszámolt szorzat, vagy sem, valamint végül a teljesítményt is értékeli!

VÉGE