

## 2.4. Object Pascal kifejezések

1. Logikai műveletek bemutatása: [Bool](#)
2. Bitenkénti logikai műveletek: [Bit](#)
3. Halmazműveletek bemutatása: [Halmaz1](#)
4. A hét napjaival végezhető halmazműveletek: [Halmaz2](#)
5. Számokkal végezhető halmazműveletek: [Halmaz3](#)



A program a logikai értékeket („*true*”, „*false*”) kisbetűs szöveg formájában olvassa be, majd átalakítja logikai értéké és elvégzi a logikai műveleteket.

```
program bool;
{$APPTYPE CONSOLE}
uses
  SysUtils;
var
  A, B, C, D : boolean;
  R1, R2, R3, R4 : boolean;
  S : string;
begin
  writeln('Logikai muveletek');
  writeln;
  writeln('Az adatok erteke true vagy false lehet!');
  writeln;

  write('A = '); readln(s);
  if s = 'true' then A := true else A := false;
  write('B = '); readln(s);
  if s = 'true' then B := true else B := false;
  write('C = '); readln(s);
  if s = 'true' then C := true else C := false;
  write('D = '); readln(s);
  if s = 'true' then D := true else D := false;
  writeln;
  R1 := A and B or C;
  R2 := A or B and C or D;
  R3 := (A or not B) and (C or B);
  R4 := (A or B) and (C or D);

  writeln('A and B or C = ',R1);
  writeln('A or B and C or B = ',R2);
  writeln('(A or not B) and (C or B) = ',R3);
  writeln('(A or B) and (C or D) = ',R4);
  readln;
end.
```



Tervezzünk egy olyan programot, amely a beolvasott adatokon bitenkénti logikai műveleteket hajt végre. (*bit*)

A program bemutatja a *byte* típusú adatokon bitenként elvégezhető: **not**, **and**, **or**, **xor**, valamint az **shl** és **shr** logikai műveleteket.

Felhívjuk a figyelmet arra, hogy az alábbi programban nem ellenőrizzük azt, hogy a felhasználó valóban *byte* típusú számot ad-e meg bemeneti adatként.

```
program bit;
{$APPTYPE CONSOLE}
uses
  SysUtils;
var
  a,b,c : byte;
begin
  writeln('Bitenkenti logikai muveletek');
  writeln;
  writeln('-- not a --');
  write(' a = '); readln(a);
  writeln(' not ': 21, a:3, ' = ', not a);
  writeln;

  writeln('-- a and b --');
  write(' a = '); readln(a);
  write(' b = '); readln(b);
  writeln(a:16, ' and ', b:3, ' = ', a and b);
  writeln;

  writeln('-- a or b --');
  write(' a = '); readln(a);
  write(' b = '); readln(b);
  writeln(a:16, ' or ', b:3, ' = ', a or b);
  writeln;

  writeln('-- a xor b --');
  write(' a = '); readln(a);
  write(' b = '); readln(b);
  writeln(a:16, ' xor ', b:3, ' = ', a xor b);
  writeln;

  writeln('-- a shl b --');
  write(' a = '); readln(a);
  write(' b = '); readln(b);
  writeln(a:16, ' shl ', b:3, ' = ', a shl b);
  writeln;

  writeln('-- a shr b --');
  write(' a = '); readln(a);
  write(' b = '); readln(b);
  writeln(a:16, ' shr ', b:3, ' = ', a shr b);
  writeln;
  readln;
end.
```



Tervezzünk egy olyan programot, amely az *abc* betűiből halmazokat képez és a különböző, a halmazok között végzett műveletek eredményeit megjeleníti! (*Halmaz1*)

A program definiálja az *abc* felsorolt típust. A *hh* típus az *abc* felsorolt típus elemeinek halmaza. az *AH*, *BH*, *CH*, *DH* és az *RH* *hh* típusú halmazok, amelyeket különböző módon definiáljuk és ezekkel végezzük a műveleteket. A műveletek eredményeinek megjelenítésére a *hkar* karaktertömböt hívjuk segítségül. Kihasználjuk, hogy a felsorolt típus elemeinek sorszáma megegyezik a megfelelő karakter sorszámaival a karaktertömbben.

```
program halmaz1;
{$APPTYPE CONSOLE}
uses
  SysUtils;

type
  abc = (a,b,c,d,e,f,g,h);
  hh = set of abc;

const
  { A halmaz elemeinek megfelelő karakter
    konstans tömbje: }
  hkar : array[abc] of char = 'abcdefgh';
var
  AH, BH, CH, DH, RH : hh;
  i : abc;
begin
  AH := [a,b,c,d];
  BH := [a,c,e,f];
  CH := [a,g,h];
  DH := [b,f];
  RH := AH + BH * CH - DH;
  write('Az eredményhalmaz elemei: ');
  for i:=a to h do
    if i in RH then write(hkar[i]);
  writeln;
  readln;
end.
```



Készítsünk egy olyan programot, amely a hét napjaira vonatkozó vizsgálatokat végez halmazműveletek segítségével! (*Halmaz2*)

A programban a *napok* felsorolt-, a *het* ezekből képzett halmaz típus. A *teljes\_het*, *pihenonapok*, *munkanap*, *rendel*, *nem\_rendel* *het* típusú halmazok, melyeket a programban definiálunk és aztán vizsgálatainkat a különféle halmazműveletek segítségével végezzük.

```
program halmaz2;
{$APPTYPE CONSOLE}
uses
  SysUtils;
type
  napok = (hetfo, keddi, szerda, csutortok,
           pentek, szombat, vasarnap);
  het = set of napok;
var
  munkanap, pihenonapok, teljes_het : het;
  rendel, nem_rendel, szabad_vagyok : het;

begin
  teljes_het := [hetfo..vasarnap];
  pihenonapok := [szombat, vasarnap];
  szabad_vagyok := [szerda];
  if teljes_het >= pihenonapok then
    writeln('A teljes_het a pihenonapokat is tartalmazza');
  munkanap := teljes_het - pihenonapok;
  if munkanap <= teljes_het then
    writeln('A munkanap reszhalmaz a teljes_hetnek');
  rendel := [hetfo, szerda, pentek];
  nem_rendel := munkanap - rendel;
  if csutortok in nem_rendel then
    writeln('Csutortok szabadnap');
  if szerda in rendel then
    writeln('Szerdan van rendeles');
  if rendel * szabad_vagyok <> [] then
    writeln('Rendelesre mehetek');
  readln;
end.
```



A programban definiáljuk a *szamok* típust, mely az egész számok 1-10.-ig terjedő részhalmaza. Az *sz10* a *szamok* típusú elemek halmaza. A *paratlan*, *paros*, *ketto\_oszthato*, *harom\_oszthato*, *negy\_oszthato*... halmazok az oszthatóság szempontjából elkülöníthető részhalmazok. Ezeket értékadással és halmazműveletekkel töltjük fel, majd kiírjuk a tartalmukat.

```
program halmaz3;
{$APPTYPE CONSOLE}
uses
  SysUtils;
type
  szamok = 1..10;
  sz10 = set of szamok;

var
  paratlan, paros, ketto_oszthato, harom_oszthato,
  negy_oszthato, ketto_harom_oszt,
  ketto_negy_oszt, teljes_halmaz: sz10;
  i : szamok;
begin
  paratlan := [1,3,5,7,9];
  paros    := [2,4,6,8,10];
  ketto_oszthato := paros;
  harom_oszthato := [3,6,9];
  negy_oszthato  := [4,8];

  teljes_halmaz := paratlan + paros;
  ketto_harom_oszt := ketto_oszthato * harom_oszthato;
  ketto_negy_oszt  := ketto_oszthato * negy_oszthato;
  write('A teljes halmaz: ');
  for i:=1 to 10 do
    if i in teljes_halmaz then write(i:3);
  writeln;
  write('Kettovel es harommal oszthato: ');
  for i:= 1 to 10 do
    if i in ketto_harom_oszt then write(i:3);
  writeln;
  write('Kettovel es neggyel oszthato: ');
  for i:= 1 to 10 do
    if i in ketto_negy_oszt then write(i:3);
  writeln;
  readln;
end.
```