

Intervallum, ill. halmaz típusú adat:

[] -ben adjuál meg az értéket
fel kell sorolni, hogy milyen értéket vehet fel a halmaz

pl.: [1..20]

['a'..'j']

pl.: [2, 3, 5, 7, 21]

['x', 'a', 'y', 'c']

['kétő', 'edd', 'erde', 'üti', 'pente', 'ko', 'vas']

összetett adattípusok:

a, tömb b, rekord c, fájl

a, homogén adatszerkezet

azonos típusú elemi adatokból épül fel, és index(es) elöl-
bőltetés) meg egymástól

Attribútuma: meg kell adni 1 nevet \rightarrow nev: array [indexhatárok] of típusú

pl.: adni: array [1..50] of integer \Rightarrow lehet [-5; ... 10], ['a'..'j']

nev[1]; nev[2]; ... nev[50]

nev[-5]; ... nev[10]

nev['a']; ... nev['j']

vektor: 1 dimenziós tömb

A tömb lehet többdimenziós is

- nev: array [1..10, 1..50] of integer \rightarrow 500 elemű a
tömb

a valóságban: tömb [1; 1], tömb [1; 2] ... tömb [1..50], tömb [2..1] ...
tömb [10, 50]

matrixba foglalható:

	1	2	3	...	50
1					
2					
...					
10					

- 1 tömb lehet 3 dimenziós is, vagy akár több.

sorszám: egész szám (előjeles vagy nem) v. karakter típusú

Ha olyan sorszámot adunk meg, amit nem definiáltak, írja:

out of range ⇒ értéken aluli

szöveg: array [1..50] of char → 50-byte-os karakter típusú

azt mondja rá, hogy string [50]

↳ karakterlánc

max hossza: 255 karakter,
mert a 0. -ban újat tárol.

A Pascal programozás utasítás és izlete:

adatbemenet és adatkiírás ⇒ ez a system unit eljárásai

readln (a) olvasó utasítás (l₁..l_n)

ált. 1 read-del 1 változóval adunk értéket

az adatbemenet a bill-ről történik

↳ eljárás során a program ellenőri, hogy megfelelő adatot kaptunk-e be, megfelel-e a deklarációnak.

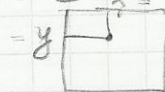
WRITE(LN) (o₁; o₂... o_n)

mind a kettő a képernyőre ír

write: a cursor ott marad a sor végén, még a writeln
sorozat.

clrscr; törlő a képernyőt (clearscreen)

ha a képernyő adott pontjára akarunk jöni: GOTO xy (oszl., sor)



write ('a='); readln (a);

writeln ('A száma:', a, '!');

a: kéréség: b-edes száma

- ha kisebb a szám:

$$2,73507 : 2 \rightarrow 2,735,07$$

- ha kicsi a nagyság hely, akkor növekvő alapértelmezésbe.

Értékesítő utasítás:

vált új := Pascal formula;

a változó vegye fel a Pascal formula által előírt értékét.

a := b + c; Teljesen mindegy, hogy mi volt az "a" értéke, megváltozik

1, Nincs típuskonverzió

alsóhatár: a baloldali változó magasabbrendű lehet, mint a jobboldali formula által szolgáltatott érték

aritmetikai

a változó numerikus változó \Rightarrow nincs automatikus típuskonv.

TYPE MISMATCH ERROR

a program futása alatt

a és b aritmetikai változó \Rightarrow logvált := a < b

itt érték lehet: 1 v. 0.

stringvált3 := stringvált2 + stringvált1.

'BUDAPEST' 'BUDA' 'PEST'

nem kommutatív.

[konkaténáció = összekapcsolás]

aritmetikai kifejezés, értékelés:

$v :=$ aritmetikai kifejezés \rightarrow egész típusú változóra pl. $20 \div 3 = 6$
alkalmazható

szempelket: + - * / div (egész osztás) = mod, sin, cos, 11, $\sqrt{\quad}$
levegőp. maradékos $\rightarrow x \in \mathbb{Z}$

azt adjó meg, hogy mennyi lesz a maradék
 $20 \bmod 3 = 2$

$v := a + b * c - d / 5 * e;$

érvényes rd a műveleti sorrend

1: $b * c$

2: $- d / 5 * e$

3: $1 - 2$

4: $a + 3$

egyenrangú műveletek a
BARÓL-FOBBRA szabály
érvényes

- ha a műveleti sorrendtől el akarunk térni, akkor ()-t kell használni (bármennyit)

- ha 1 aritmetikai kifejezésben előforduló típusok a változók: a végeredmény igazodik a legmagasabb operációs szinthez

egész / levegőpontos \Rightarrow mindig real

Mindig enged megfelelő választ meg a változót!!!

ans := Pascal kifejezés;
bereni↓ a változóba az érték

b1: b2 + b3 2 byte típusú összege byte

a d b fixpontos \Rightarrow a/b lebegőpontos lesz

OUT OF RANGE \leftarrow

(ha tilosordulás van, akkor lép fel!)

$$a := b/c * d \Rightarrow \frac{b \cdot d}{c}$$

$$a := b/(c * d) \Rightarrow \frac{b}{c * d}$$

Logikai műveletek

logvalt := logikai kifejezés

- logvalt := $a < b$

ahol a reláció értelmezhető: 1, ha nem "H"

- logvalt := $a \geq b$

helyettesíthető not ($a < b$)

↓
nem művelet

↳ negáció jele \neg

a	$\neg a$
i	k
k	i

művelet : $a > b$

$a < b$

$a = b$

$a \geq b$

$a \leq b$

$a > b$

vagy or diszjunktív jele: \vee

a	b	$a \vee b$
i	i	i
i	h	i
h	i	i
h	h	h

xor : ez a b vagy ekskluszív diszjunktív jele: \oplus

a	b	$a \oplus b$
i	i	h
i	h	i
h	i	i
h	h	h

pl.: Győzelem v. maghalalás.

and és konjunktív jele \wedge

a	b	$a \wedge b$
i	i	i
i	h	h
h	i	h
h	h	h

A logikai műveletek síkja lehet kifejezve:

$$p: (a < b) \wedge (c = 5)$$

reláció: logikai operandusok (Mindig zárójellel kell lenni)

legmagasabb rendű: $\neg \rightarrow \wedge \rightarrow \vee, \times$

Kiértékelés sorrendje: először zárójel, majd \neg

$$\text{pl.: } (a=b) \wedge (b=c)$$

jelentése: $a=b=c$

$$\text{DE: } \underbrace{(a \leftrightarrow b) \wedge (b \leftrightarrow c)}_{\text{és nem zárja ki, hogy } a \leftrightarrow c}$$

es nem zárja ki, hogy $a \leftrightarrow c$

De Morgan azonosság

$$\neg(a \wedge b) = \neg a \vee \neg b$$

$\neg a = \text{logikai } a$

$$\neg(a \vee b) = \neg a \wedge \neg b$$

$$\text{not}((a=b) \text{ and } (b=c)) = (a \neq b) \text{ or } (b \neq c)$$

$$\neg \neg a = a$$

$\text{not}(a=b) \Rightarrow a \neq b$ (nem kell zárjellel, de így áttekinthetőbb).

szöveges változó := string kifejezés

kar := stringvall [3]

↳ 3. karakter

Type mismatch error

String 1 := string 2 + stringváltozó 3;

nem kommutatív: balról jobbra felől össze

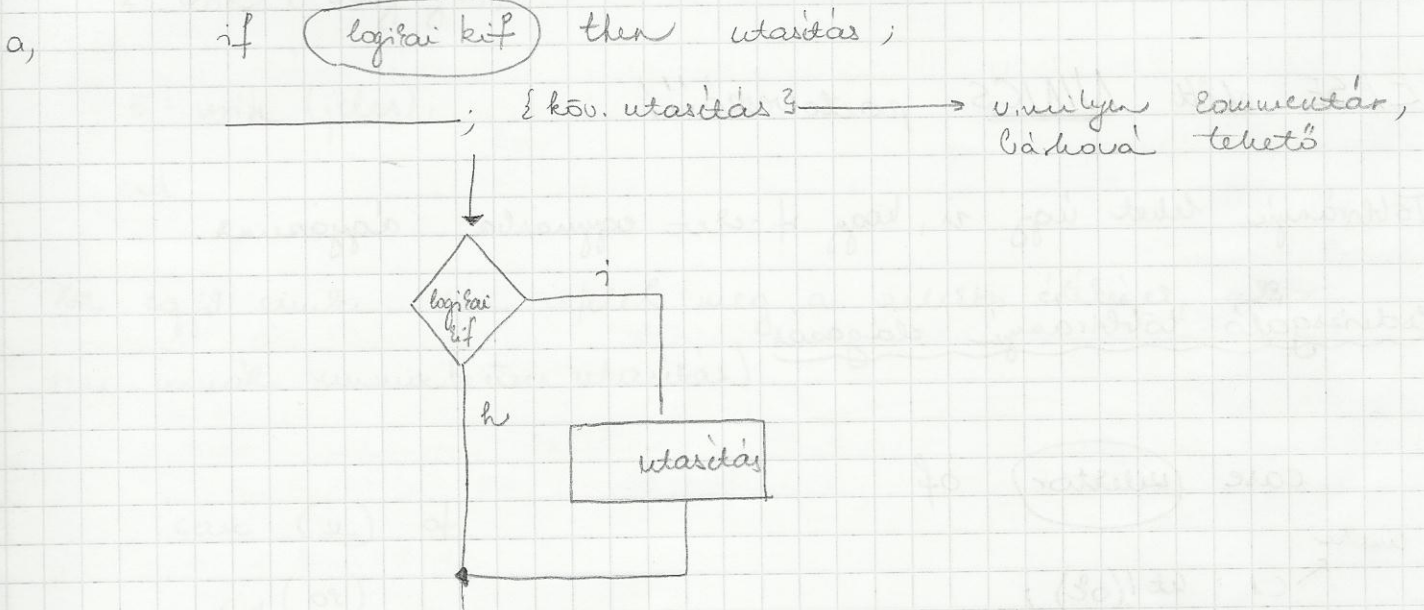
Bzélkaió meqvalósítása

Pascal-ban

2 típusa van \rightarrow a, alternatív selekció (2 lehetőség)

\rightarrow b, többrányú

(nem logikai feltételhez kötött)



több utasítás \Rightarrow utasításblokk \Rightarrow

begin
utasítás 1;
ut 2;
ut 3;
end;

} utasítás blokk
utolsó: az end
előtt nem kell, de
nem baj \rightarrow adasható.

b./1;

if (log-i kif) then ut1 (v. utasítások)
else ut2 (" ");