

Pascal – základné pojmy

program

– postupnosť príkazov popisujúcich nejakú činnosť. Každý program má autora, ktorý ho píše a procesor, ktorá ho zrealizuje. Program preto musí byť napísaný v jazyku, ktorému rozumie autor i procesor

- algoritmus napísaný v programovacom jazyku

proces – je to realizácia programu, ktorá sa skladá z realizácie príkazov

program je statický, proces, ktorý vzniká jeho realizáciou je dynamický

program - napr. kuchársky recept, podľa ktorého postupuje kuchár pri varení (kuchár=procesor, jednotlivé ingrediencie = objekty, napr. kurča na paprike=žiadaný efekt)

etapy tvorby programu:

1. definícia problému

problém – stav, v ktorom jestvuje rozdiel medzi tým, čo v danom momente poznáme (vieme, máme) a tým, čo potrebujeme

2. riešenie problému – je chápané vo význame splnenia cieľa

3. nájdenie postupu riešenia – tvorivá činnosť, je potrebné aktívne premyslieť podrobnosti a postupné kroky, iba človek, nejestvuje zariadenie, ktoré by bolo schopné tvorivo myslieť

4. vykonanie postupu – rutinná činnosť, ak je postup už daný alebo známy, je ho schopný zrealizovať človek i stroj (nemusí rozmýšľať, stačí ak daný postup presne zrealizuje)

algoritmus – je postup, ktorého realizáciou získavame zo zadaných vstupných údajov po konečnom počte činnosti v konečnom čase správne výsledky

vlastnosti algoritmu:

elementárnosť – postup je zložený z činností, ktoré sú pre realizátora elementárne, zrozumiteľné

determinovanosť – postup je zostavený tak, že je v každom momente jeho vykonávania jednoznačne určené, aká činnosť má nasledovať, alebo či sa už postup skončil

rezultatívnosť – postup dáva pre rovnaké vstupné údaje vždy rovnaké výsledky (ak skončí)

konečnosť – postup skončí vždy v konečnom čase a po vykonaní konečného postu činnosť

hromadnosť – postup je aplikovateľný na celú triedu prípustných vstupných údajov (postup pripúšťa premenlivé vstupné údaje a umožňuje riešiť úlohy podobného typu)

efektívnosť – postup sa uskutočňuje v čo najkratšom čase a s využitím čo najmenšieho počtu prostriedkov

algoritmizácia – schopnosť aktívne vytvárať algoritmy určené pre nemysliace zariadenie. Je nevyhnutnou súčasťou schopnosti programovať na počítačoch

algoritmus – je čiastočne správny, ak v prípade, že skončí, dáva vždy správne výsledky

algoritmus – je konečný, ak skončí v konečnom čase pre ľubovoľné vstupné údaje

algoritmus nazývame správny, ak je čiastočne správny a konečný

programovanie – je konštruktívna myšlienková činnosť (aj praktická činnosť) kedy vytvárame nové programové produkty realizovateľné na počítači (učíme sa myslieť, organizovať svoje myšlienky,...)

vytvorenie programu pozostáva z činností:

1. *algoritmizácia daného problému* – určenie vstupných a výstupných podmienok

vytvorenie programu

2. *zapísanie a odladenie programu priamo na počítači*

elementárne činnosti algoritmického jazyka:

príkaz vstupu – program potrebuje obvykle odniekiaľ brať informácie, ktoré spracúva (obvykle od užívateľa z klávesnice), a tieto hodnoty sa uložia do premenných, v každom momente vykonávania algoritmu vy mala byť v premennej nejaká konkrétna hodnota

To zabezpečí príkaz :

```
READ ( {zoznam premenných} ) ;
```

Zoznam premenných je zoznamom identifikátorov premenných, ktorých hodnoty chce počítač prečítať, sú oddelené čiarkami.

napr: read (a,b,c);

čítanie premenných a, b, c

readln

čítanie nového riadku

príkaz výstupu - umožňuje získať od procesora výsledky algoritmu alebo iné oznamy (napr. oznam o tom, že sme zadali nesprávne vstupné údaje)

To zabezpečí príkaz :

```
WRITE ( {zoznam výrazov} ) ;
```

write (výsledok) - write(p); výstupom môže byť iba premenná
write('Ahoj', meno); výstupom môže byť aj text, ktorý treba uzavrieť do apostrofov
writeln do výstupného súboru sa vypíše oddeľovač riadkov

priradovací príkaz - zmeniť hodnoty premenných počas vykonávania algoritmu môžeme pomocou dvoch spôsobov: príkazom vstupu alebo priradením novej hodnoty

- nariaďuje procesoru, aby vykonal ne jeho pravej strane naznačené operácie alebo funkcie a výsledok uložil do premennej, meno ktorej je na ľavej strane

má tvar

```
{premenná} := {výraz} ;
```

Pod pojmom výraz si možno predstaviť napr. aritmetický výraz: A :=4;

Postup spracovania priradovacieho príkazu:

- 1, určí sa cieľ priradenia (premenná na ľavej strane)
- 2, vyhodnotí sa výraz na pravej strane
- 3, získaná hodnota sa uloží do cieľovej premennej (tej v ľavo)

Poznámka:

vo výraze sa môže nachádzať aj cieľová hodnota (viď príklad), lebo pri vyhodnocovaní sa najprv výraz vyčíslí a až potom sa cieľová hodnota prepíše.

napr: riadok:= riadok+1; {zväčší hodnotu riadku o jedna}

zložený príkaz - jeho funkciou je zlúčiť viac príkazov do jedného.

Pri spracovaní budú príkazy vykonávané v poradí akom sú zapísané. Špeciálny prípad zloženého cyklu je telo programu. Zložené príkazy môžeme do seba vnárať, zložený príkaz považujeme ako jeden príkaz, ktorý keď je zadané, že ho treba preskočiť, tak preskočí všetky príkazy v ňom.

Begin

```
{príkaz 1};
```

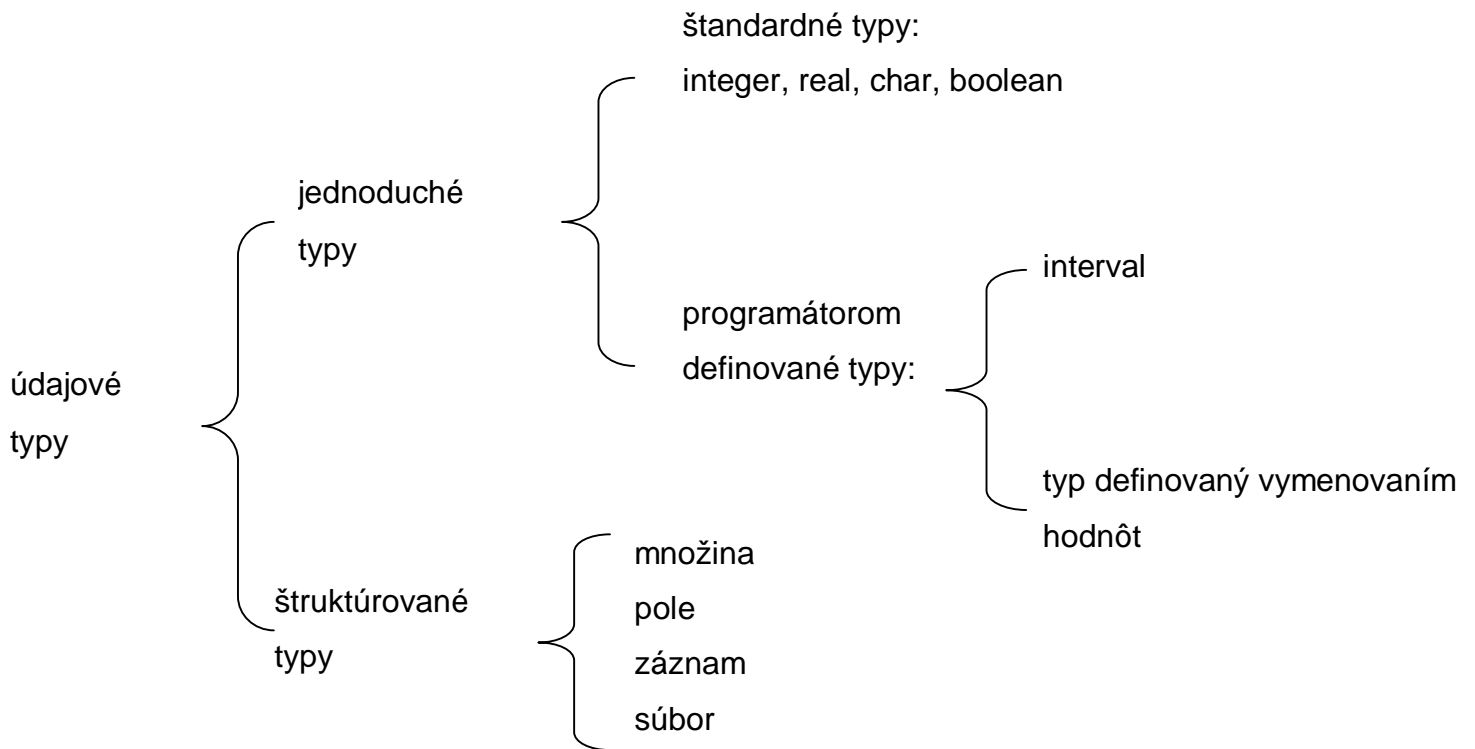
```
{príkaz 2};
```

```
:
```

```
{príkaz N};
```

END;

podmienka: logický výraz t.j. zistenie vzťahov medzi výrazmi, prip. zviazané logickými operáciami
(and = a, or = alebo, no t= neplatí, že)



typ boolean:

množina hodnôt: true (pravda) – false (nepravda)

operácie:

and	logický súčin
or	logický súčet
not	logický negácia
<, >, =	menší, väčší, rovná sa
<=, >=, < >	menší alebo rovná sa, väčší alebo rovná sa, nerovná sa

typ integer:

množina hodnôt: podmnožina množiny celých čísel

operácie:

+, -	sčítanie, odčítanie
*	nasobenie
div	celočíselný podiel (celá časť podielu)
mod	zvyšok celočíselného podielu

<, >, = menší, väčší, rovná sa
 <=, >=, < > menší alebo rovná sa, väčší alebo rovná sa, nerovná sa

štandardné funkcie :

abs absolútna hodnota
 sqr umocnenie na druhú
 trunc (x) výsledkom je celočíselná časť reálnej hodnoty x napr, trunc(2,5)=2
 round(x) výsledkom je zaokrúhlená hodnota reálnej hodnoty x
 odd (x) dáva hodnotu true, ak x je nepárna, inak false

TYP	ROZSAH	VEĽKOSŤ
INTEGER	-32 768...32 767	2B
SHORTINT	-128...127	1B
LONGINT	-2 147 483 648...2 147 836 647	4B
BYTE	0...255	1B
WORD	0...65 535	2B

typ real:

množina hodnôt typu: podmnožina množiny reálnych čísel

operácie :

+, - *, / sčítanie, odčítanie, násobenie, delenie
 <, >, = menší, väčší, rovná sa
 <=, >=, < > menší alebo rovná sa, väčší alebo rovná sa, nerovná sa

štandardné funkcie :

abs absolútna hodnota
 sqr umocnenie na druhú
 sqrt druhá odmocnina
 sin, cos sinus cosínus
 arctg arkustangens
 ln prirodzený logaritmus
 exp(x) e^x
 trunc (x) výsledkom je celočíselná časť reálnej hodnoty x napr, trunc(2,5)=2
 round(x) výsledkom je zaokrúhlená hodnota reálnej hodnoty x

TYP	ROZSAH	VEĽKOSŤ
REAL	$2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$	6B
SINGLE	$1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$	4B
DOUBLE	$5 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$	8B
EXTENDED	$3,4 \cdot 10^{-4932} \dots 1,1 \cdot 10^{4932}$	10B
COMP	$-2^{63} + 1 \dots 2^{63} - 1$	8B

typ char:

množina hodnôt typu: množina znakov

operácie:

<, >, = menší, väčší, rovná sa

<=, >=, < > menší alebo rovná sa, väčší alebo rovná sa, nerovná sa

štandardné funkcie :

ord(x) hodnota ord(x) je poradové číslo znaku x v danej množine znakov, poradové čísla sa nezačínajú nulou

chr(i) hodnotou je znak, ktorého hodnota ordinárneho čísla je i

vymenovaný typ:

definujeme nasledovne: type menotypu = (h₁, h₂, ..., h_n)

napr. type den = (pondelok, utorok, streda,.....)

type farba = (cervena, zelena, modra,.....)

operácie nad celými číslami:

OPERÁCIA	VÝZNAM	VÝSTUP
+	sčítovanie	integer/real
-	odčítovanie	integer/real
*	násobenie	integer/real
/	delenie	real
DIV	celočíselné delenie	integer
MOD	zvyšok po celočíselnom delení	integer
SIN (X)	sinus (x je v radianoch)	real
ARCTAN (X)	inverzná k tangensu	real
SQR (X)	mocnina	real
SQRT (X)	odmocnina	real
EXP (X)	exponent - e^x	real
LN (X)	logaritmus prirodzený	real
INT (X)	vynuluje desatinnú časť reálneho čísla	real
FRAC (X)	ponechá si iba desatinnú časť reálneho čísla	real
ABS (X)	absolutná hodnota	integer/real
ROUND (X)	zaokruhlí reálne číslo na najbližšie celé číslo	integer
RANDOM (X)	generuje náhodné čísla v rozpätí x	real
TRUNC (X)	odtrhne desatinú časť reálneho čísla	integer

organizácia programu v jazyku pascal :

hlavička programu (obsahuje meno programu a parameter, pomocou, ktorých identifikujeme vstupné a výstupné údaje programu)

deklaračná časť (deklarujeme a definujeme objekty, ktoré budeme používať v príkazovej časti – návestia, konštanty, typy, premenné, procedúry, funkcie)

príkazová časť (postupnosť príkazov, ktoré sa vykonávajú v takom poradí v akom sú napísané)

program meno programu(parameter);

deklaracia návesti, konštant, typov, premenných, procedúr, funkcií

begin – začiatok príkazovej časti

príkaz 1;

príkaz 2;

end. – koniec príkazovej časti

