

História, pôvod Pascalu

Jazyk bol navrhnutý začiatkom 70 rokov. Autorom je profesor Niklaus Wirth z vysokej školy technickej v Zurichu. Názov jazyka bol zvolený na počesť francúzskeho filozofa Blaise Pascala (1623 – 1662). Cieľom bolo:

- navrhnuť jazyk tak, aby bol vhodný na výučbu programovania a bol založený na obmedzenom počte zrozumiteľných konštrukcií,
- navrhnuť štruktúru jazyka tak, aby bolo ľahké implementovať Pascal na väčšine súčasných počítačov.

Prvé verzie boli publikované v 1971, neskoršie (1974) bola uverejnená oprávnená definícia jazyka. [Hruška, T.: Pascal pro začátečníky. Praha: STNL, 1989]

Pascal – základné pojmy

Identifikátor

- sa používa na označenie jednotlivých objektov v programe - premenných, konštánt, typov, procedúr, funkcií...
- mal by byť zvolený tak, aby vyjadroval obsah (napr. procedure vypis, const = pocet, ...)
- nesmie začínať číslicou
- nesmie obsahovať medzery a znaky s diakritikou
- nesmie byť zhodný s kľúčovým slovom

Kľúčové slová

- sú slová, ktoré sa nemôžu používať v žiadnom inom význame než v tom, na ktorý sú určené (napr. and, array, begin, case, const, div, do, downto, else, end, for, function, goto, if, label, mod, not, or, program, repeat, string, then, to, type, unit, until, uses, var, while, xor, ...)

Konštanta

- predstavuje hodnotu, ktorá je nemenná. Môže byť zadefinovaná v deklaračnej časti programu (napr. const n=5).

Komentár (poznámky) obsahuje text, ktorý vysvetľuje význam objektov a konštrukcií programu, popisuje časti programu tak, aby sa v programe dalo ľahko zorientovať aj po dlhšom čase.

- uzatvára sa do zložených zátvoriek {komentar} alebo okrúhlych zátvoriek s hviezdíčkou (*komentar*)
- pri vykonávaní programu sa neberie do úvahy - kompilátor ho nepovažuje za súčasť programu
- hlavne pri rozsiahlejších programoch ho dávame všade tam, kde by mohol neskôr pomôcť pri identifikácii príkazu;

Premenná

- predstavuje logické pamäťové miesto, ktoré má svoje pomenovanie - názov, pomocou ktorého k nemu pristupujeme. Premennej musí byť priradený typ (napr. var a: integer)

Organizácia programu

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)

Štruktúra programu

PROGRAM *názov_programu*;

USES {zoznam použitých knižníc (jednotiek, unitov) - system, dos, overlay, printer, crt, graph}

LABEL {deklarácia návěstí - definuje miesto, na ktoré skočíme príkazom goto}

CONST {deklarácia konštánt - číselná, znaková, reťazcová}

TYPE {definícia dátových typov}

VAR {deklarácia premenných {celočíselné, reálne, znaky, reťazce, polia, ...}}

PROCEDURE *názov_procedury*; {deklarácia procedúr}

FUNCTION *názov_funkcie*; {deklarácia funkcií}

BEGIN {začiatok hlavného programu}

{Telo hlavného programu}

END. {koniec programu}

ŠTANDARDNÉ PRÍKAZY

Príkaz vstupu

READ, READLN (readln - prechod na nový riadok). Príkaz vstupu zabezpečuje vstup údajov z klávesnice alebo zo súboru do programu.

Napríklad:	príkaz	vstup z kláv.	efekt
	READ(a)	23	hodnota a je 23
	READ(a,b,c)	1 aa 3.14	hodnota a je 1, b je 'aa' a c je 3.14

Príkaz výstupu

WRITE, WRITELN (writel - prechod na nový riadok). Príkaz výstupu zabezpečuje výstup údajov z programu na monitor alebo do súboru.

Napríklad: (a=2, b=3)	príkaz	výstup na monitor
	WRITE(1);	1
	WRITE(1,2);	12
	WRITE(a,b);	23
	WRITE(a,' ',b);	2 3
	WRITE('a','b');	ab
	WRITE('a'+b');	ab
	WRITE(a+b);	5

Prirad'ovací príkaz

:=

Realizáciou prirad'ovacieho príkazu sa priradí premennej na ľavej strane príkazu hodnota výrazu na pravej strane príkazu.

Napríklad:	príkaz	efekt
	a := 1;	hodnota a je 1
	b := 21 - a;	hodnota b je 20
	c := 'ret' + 'azec';	hodnota c je 'reťazec'

Zložený príkaz

- zlúčenie viac príkazov do jedného. Pri spracovaní budú príkazy vykonávané v poradí akom sú zapísané. Špeciálny prípad zloženého cyklu je telo programu.

Begin

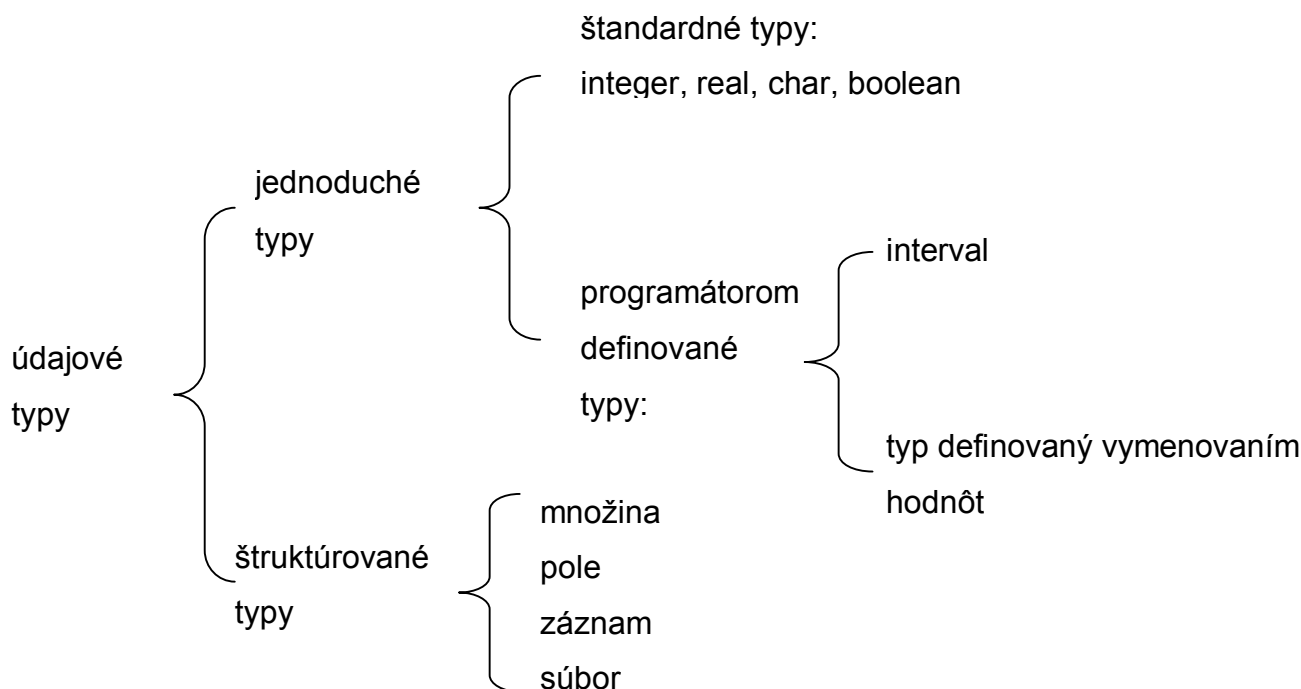
```
{príkaz 1};
{príkaz 2};
:
{príkaz N};
```

END;

Údajový typ

- definuje množinu prípustných hodnôt

Napríklad: Ak si zadefinujem premennú *a* typu integer, množinu prípustných hodnôt tvorí interval celých čísel <-32768,32767>. Žiaden typ nemôže definovať nekonečnú množinu hodnôt. Je to obmedzené hranicami intervalu, počtom platných číslic alebo jednoducho veľkosťou pamäte počítača



ŠTANDARDNÉ TYPY DÁT

Celočíselné typy:	typ	rozsah
	byte	0..255
	shortint	-128..127
	word	0..65535
	integer	-32768..32767
	longint	-2147483648..2147483647

Množina operácií: aritmetické: +, -, *, **DIV**, **MOD**
 relačné: >, >=, <, <=, <>, =

(operácia / sem nepatrí, dôvodom je fakt, že napr.: výraz 4/2 má v Pascale hodnotu 2.000000000 čo je číslo reálne a to aj napriek tomu, že jeho desatinná časť je nulová)

Reálne typy:	typ	rozsah
	real	$2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$
	single	$1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$
	double	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$
	extended	$3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$
	comp	$-263+1 \dots 263-1$

Množina operácií: aritmetické: +, -, *, /
 relačné: >, >=, <, <=, <>, =

Štandardné funkcie pre typ real:

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 char: premenná typu **char** je určená pre prácu so znakmi a obsahuje práve jeden znak v ASCII (American Standard Code for Information Interchange, zjednodušene sú to všetky znaky, ktoré môžeme napísať pomocou klávesnice) kóde. Konštanty typu char sú uzavreté medzi apostrofmami, napr. 'd', 'D'.
 Pozor: 'd' <>'D'.

Typ string: premenná typu **string** je určená pre prácu s reťazcami (postupnosťami) znakov dĺžky 1 až 255. Ak nie je uvedená dĺžka, predpokladá sa dĺžka 255 znakov.

Typ boolean: premennej typu **boolean** môže byť priradená práve jedna z dvoch hodnôt: true (pravda) alebo false (nepravda). Je to logická premenná a používa sa pri práci s logickými premennými.

Množina operácií: logické: **AND, OR, NOT, XOR**
relačné: **>, >=, <, <=, <>, =**