

Specifikace jazyka

Formát programu

Program je tvořen posloupností příkazů. Příkazy jsou zapsány ve volném formátu, poznámky, mezery, tabulátory a konce řádků slouží pouze jako oddělovače a na význam programu nemají vliv. Poznámky jsou omezeny dvěma lomítky a koncem řádku. Klíčová slova jsou rezervovaná. V identifikátorech a klíčových slovech se rozlišují velká a malá písmena.

Proměnné

Názvy proměnných jsou tvořeny identifikátorem. Každá proměnná musí být před použitím deklarována, opakovaná deklarace proměnné téhož názvu je chybou. Po deklaraci má proměnná podle svého typu hodnotu 0, 0.0, "", resp. `False`.

Příkazy

Jsou definovány následující příkazy:

- | Prázdný příkaz ;
- | *typ proměnná, proměnná, ...;*
Deklarace proměnných uvedeného typu. Typem může být `int`, `float`, `boolean` nebo `String`.
- | *proměnná = výraz;*
Přiřazení hodnoty výrazu proměnné. Proměnná na levé straně musí být deklarovaná. Typ výrazu na pravé straně musí být kompatibilní vzhledem k přiřazení s typem proměnné, tj. typy musí být stejné nebo je možné přiřazovat celočíselné hodnoty do reálných proměnných s provedením automatické konverze.
- | *read proměnná, proměnná, ...;*
Přečtení hodnoty ze standardního vstupu a přiřazení do proměnných. Každá čtená hodnota je na samostatném vstupním řádku.
- | *print výraz, výraz, ...;*
Výpis hodnoty výrazů na standardní výstup. Za hodnotou posledního výrazu je vypsán znak konce řádku.
- | *if (podmínka) : příkazy else : příkazy end;*
Podmíněný příkaz, podmínka musí být booleovský výraz. Příkazy označují posloupnost příkazů oddělených středníkem. Část `else` je nepovinná.
- | *while (podmínka) : příkazy end;*
Příkaz cyklu, podmínka musí být booleovský výraz. Příkazy označují posloupnost příkazů oddělených středníkem.

Výrazy

V argumentech výrazů se jako operandy mohou vyskytovat celá desítková čísla, reálná čísla (včetně případného exponentu) a řetězce v uvozovkách (bez escape sekvencí). Operandy musí být typově kompatibilní, podle potřeby se provádí automatická konverze celočíselných operandů na reálné (ne naopak).

Jsou definovány následující operátory (písmena I, R, B, S vyjadřují datové typy int, float, boolean a String, typy T v relačních operátorech a ternárním operátoru ?: musí být v rámci jednoho operátoru kompatibilní).):

Popis	Operátor	Signatura
unární operátor pro změnu znaménka	-	I->I, R->R
binární aritmetické operátory	+ - * /	IxI->I, RxR->R
	%	IxI->I
operátor konkatenace	.	SxS->S
relační operátory	< <= > >= == !=	TxT->B (T = I,R,S,B)
logické operátory	&&,	BxB->B
	!	B->B
ternární operátor	<i>podmínka</i> ? <i>výraz</i> : <i>výraz</i>	BxTxT->T (T = I,R,S,B)

Výrazy mohou obsahovat závorky. Operátory jsou zleva asociativní, priorita operátorů je následující (v pořadí od nejnižší k nejvyšší):

1. ||
2. &&
3. relační operátory
4. + - .
5. * / %
6. !, unární -

Ve výrazech se dále mohou vyskytovat konstanty `True` a `False` typu boolean (pro relační operátory platí `False < True`).

Varianty zadání

Tým 3 studentů

Řešení musí odpovídat úplné specifikaci jazyka včetně příkazů `if` a `while`.

Tým 2 studentů

Řešení nemusí obsahovat příkazy `if` a `while`.

Jednotlivci

Řešení nemusí obsahovat příkazy `if` a `while`, relační a logické operátory a ternární operátor `?:`.

Konstrukce jazyka demonstruje následující ukázkové vstupy: [t1.txt](#), [t1e1.txt](#).

(c) Marek Běhálek, FEI VŠB- TU Ostrava Desing: Miroslav Beneš, FEI VŠB-TU Ostrava

19. 5. 2008
23:01:02