

Programski jezik PROTEUS

1 Leksikalna zgradba

Vrsta besede	Opis znakovne predstavitve
Ime	[A-Za-z_] [A-Za-z0-9_]*
Celo število	[+-]? [0-9]+
Realno število	[+-]? [0-9]+\.[0-9]+([Ee] [+-]? [0-9]+)?
Logični konstanti	true false
Nizi	\"([^\\""] \\\"\\\\\")*\"
Imena tipov	int real bool string
Ključne besede	arr else for fun if rec then typ var where while
Simboli	+ - * / % ! & == <> <= >= < > () [] { } = . , : ;

Komentarji. Komentar se začne z znakom # in se konča na koncu vrstice. V komentarju se lahko pojavijo vsi znaki ASCII abecede s kodami od 32₁₀ do 126₁₀.

Belo besedilo. Presledki, tabulatorji in znaki za skok v novo vrstico predstavljajo belo besedilo. Tabulator ustreza 4 presledkom.

2 Sintaksna zgradba

source → *expressions*

expressions → *expression*

expressions → *expressions* , *expression*

expression → *or_expression*

or_expression → *and_expression*

or_expression → *or_expression* | *and_expression*

and_expression → *relational_expression*

and_expression → *and_expression* & *relational_expression*

relational_expression → *additive_expression*

relational_expression → *additive_expression* == *additive_expression*

relational_expression → *additive_expression* <> *additive_expression*

relational_expression → *additive_expression* <= *additive_expression*

relational_expression → *additive_expression* >= *additive_expression*

relational_expression → *additive_expression* < *additive_expression*

relational_expression → *additive_expression* > *additive_expression*

additive_expression → *multiplicative_expression*

additive_expression → *additive_expression* + *multiplicative_expression*

additive_expression → *additive_expression* - *multiplicative_expression*

multiplicative_expression → *prefix_expression*

multiplicative_expression → *multiplicative_expression* * *prefix_expression*

multiplicative_expression → *multiplicative_expression* / *prefix_expression*

multiplicative_expression → *multiplicative_expression* % *prefix_expression*

prefix_expression → *postfix_expression*

prefix_expression → + *prefix_expression*

prefix_expression → - *prefix_expression*

prefix_expression → * *prefix_expression*

prefix_expression → & *prefix_expression*

prefix_expression → ! *prefix_expression*

$postfix_expression \rightarrow INTCONST$
 $postfix_expression \rightarrow REALCONST$
 $postfix_expression \rightarrow BOOLCONST$
 $postfix_expression \rightarrow STRINGCONST$
 $postfix_expression \rightarrow \{ \}$
 $postfix_expression \rightarrow IDENTIFIER$
 $postfix_expression \rightarrow IDENTIFIER (expressions)$
 $postfix_expression \rightarrow (expressions)$
 $postfix_expression \rightarrow postfix_expression . IDENTIFIER$
 $postfix_expression \rightarrow postfix_expression [expression]$
 $postfix_expression \rightarrow postfix_expression \text{ WHERE } declarations$
 $postfix_expression \rightarrow \{ IDENTIFIER = expression \}$
 $postfix_expression \rightarrow \{ IF expression \text{ THEN } expressions \}$
 $postfix_expression \rightarrow \{ IF expression \text{ THEN } expressions \text{ ELSE } expressions \}$
 $postfix_expression \rightarrow \{ FOR IDENTIFIER = expression , expression : expressions \}$
 $postfix_expression \rightarrow \{ WHILE expression : expressions \}$
 $declarations \rightarrow declaration$
 $declarations \rightarrow declarations declaration$
 $declaration \rightarrow type_declaration$
 $declaration \rightarrow function_declaration$
 $declaration \rightarrow variable_declaration$
 $type_declaration \rightarrow TYP IDENTIFIER : type ;$
 $type \rightarrow INT$
 $type \rightarrow REAL$
 $type \rightarrow BOOL$
 $type \rightarrow STRING$
 $type \rightarrow \{ \}$
 $type \rightarrow IDENTIFIER$
 $type \rightarrow * type$
 $type \rightarrow ARR [expression] type$
 $type \rightarrow REC (record_components)$
 $type \rightarrow (type)$
 $record_components \rightarrow record_component$
 $record_components \rightarrow record_components , record_component$
 $record_component \rightarrow IDENTIFIER : type$
 $function_declaration \rightarrow FUN IDENTIFIER (function_parameters) : type = expressions ;$
 $function_parameters \rightarrow function_parameter$
 $function_parameters \rightarrow function_parameters , function_parameter$
 $function_parameter \rightarrow IDENTIFIER : type$
 $variable_declaration \rightarrow VAR IDENTIFIER : type ;$

3 Semantična pravila

3.1 Deklaracije

Območje deklariranosti imen: Vsa imena tipov, funkcij in spremenljivk pripadajo istemu območju deklariranosti imen. Vsaka deklaracija tipa zapis ustvari lastno območje deklariranosti imen; vsako ime komponente zapisa pripada območju deklariranosti imen zapisa, ki mu komponenta pripada.

Območje vidnosti imen:

1. Vsak izraz oblike

expression where declarations

ustvari novo območje vidnosti imen, ki obsega celotni izraz te oblike.

2. Vsaka deklaracija funkcije oblike

FUN IDENTIFIER (*function_parameters*) : *type* = *expressions* ;

ustvari novo območje vidnosti imen, ki se začne pri (in konča pri ; .

Območja vidnosti imen so gnezdena:

1. Vsaka deklaracija imena tipa, funkcije ali spremenljivke je vidna v območju vidnosti, kjer je ime deklarirano.
2. Vsako ime komponente tipa zapis je vidno v območju vidnosti imen, v katerem je viden tip zapis, ki mu komponenta pripada.
3. Deklaracije imen tipov, funkcij in spremenljivk na vgnezdenem območju vidnosti imen prekrijejo deklaracije, ki so vidne v obsegajočem območju vidnosti imen.

3.2 Sistem tipov

3.2.1 Deklaracije

$$\frac{\text{typ id : type ; } \quad \text{type :: } \tau}{\text{id :: } \tau} \quad \frac{\text{var id : type ; } \quad \text{type :: } \tau}{\text{id :: } \tau}$$
$$\frac{\text{fun id (id}_1 : \tau_1 , \text{id}_2 : \tau_2 , \dots , \text{id}_m : \tau_m) : \text{type} = \text{exp ; } \quad \text{type :: } \tau \quad \text{exp :: } \tau}{\text{id :: (}\tau_1, \tau_2, \dots, \tau_m\text{) } \longrightarrow \tau \quad \text{id}_1 :: \tau_1 \quad \text{id}_2 :: \tau_2 \quad \dots \quad \text{id}_m :: \tau_m}$$

3.2.2 Podatkovni tipi

Osnovni podatkovni tipi:

$$\frac{}{\text{INT :: } \textit{int}} \quad \frac{}{\text{REAL :: } \textit{real}}$$
$$\frac{}{\text{BOOL :: } \textit{bool}} \quad \frac{}{\text{STRING :: } \textit{string}} \quad \frac{}{\text{\{\}} :: \textit{void}}$$

Sestavljeni podatkovni tipi:

$$\frac{\text{type}_1 :: \tau_1 \quad \text{type}_2 :: \tau_2 \quad \dots \quad \text{type}_m :: \tau_m}{\text{rec(id}_1 : \text{type}_1, \text{id}_2 : \text{type}_2, \dots, \text{id}_m : \text{type}_m) :: \text{rec}_{\langle\langle \text{id}_1, \text{id}_2, \dots, \text{id}_m \rangle\rangle}(\tau_1, \tau_2, \dots, \tau_m)}$$
$$\frac{\text{type :: } \tau}{*\text{type} :: \textit{ptr}(\tau)} \quad \frac{\text{type :: } \tau \quad \text{exp :: } \textit{int}}{\text{type}[\text{exp}] :: \textit{arr}_{\langle\langle \text{exp} \rangle\rangle}(\tau)} \quad \frac{\text{type :: } \tau}{(\text{type}) :: \tau}$$

3.2.3 Izrazi

Enostavni izrazi:

$$\frac{}{\text{INTCONST :: } \textit{int}} \quad \frac{}{\text{REALCONST :: } \textit{real}}$$
$$\frac{}{\text{BOOLCONST :: } \textit{bool}} \quad \frac{}{\text{STRINGCONST :: } \textit{string}} \quad \frac{}{\text{\{\}} :: \textit{void}}$$

Logični operatorji:

$$\frac{\text{exp} :: \text{bool}}{! \text{exp} :: \text{bool}} \quad \frac{\text{exp}_1 :: \text{bool} \quad \text{exp}_2 :: \text{bool}}{\text{exp}_1 \& \text{exp}_2 :: \text{bool}} \quad \frac{\text{exp}_1 :: \text{bool} \quad \text{exp}_2 :: \text{bool}}{\text{exp}_1 | \text{exp}_2 :: \text{bool}}$$

Primerjalni operatorji:

$$\frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 == \text{exp}_2 :: \text{bool}} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 <> \text{exp}_2 :: \text{bool}}$$

$$\frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 < \text{exp}_2 :: \text{bool}} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 <= \text{exp}_2 :: \text{bool}}$$

$$\frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 > \text{exp}_2 :: \text{bool}} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 >= \text{exp}_2 :: \text{bool}}$$

pri $\tau \in \{\text{int}, \text{real}, \text{bool}, \text{string}\} \cup \{\text{ptr}(\tau'); \forall \tau' \in T\}$

Aritmetični operatorji:

$$\frac{\text{exp} :: \tau}{+ \text{exp} :: \tau} \quad \frac{\text{exp} :: \tau}{- \text{exp} :: \tau}$$

$$\frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 + \text{exp}_2 :: \tau} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 - \text{exp}_2 :: \tau}$$

$$\frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 * \text{exp}_2 :: \tau} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 / \text{exp}_2 :: \tau} \quad \frac{\text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\text{exp}_1 \% \text{exp}_2 :: \tau}$$

pri $\tau \in \{\text{int}, \text{real}\}$

Kazalčni operatorji:

$$\frac{\text{exp} :: \tau}{\& \text{exp} :: \text{ptr}(\tau)} \quad \frac{\text{exp} :: \text{ptr}(\tau)}{* \text{exp} :: \tau}$$

Sestavljeni izrazi:

$$\frac{\text{exp} :: \tau}{(\text{exp}) :: \tau} \quad \frac{\text{exp}_1 :: \tau_1 \quad \text{exp}_2 :: \tau_2 \quad \dots \quad \text{exp}_m :: \tau_m}{\text{exp}_1, \text{exp}_2, \dots, \text{exp}_m :: \tau_m}$$

$$\frac{\text{exp} :: \text{rec}_{\langle\langle \text{id}_1, \text{id}_2, \dots, \text{id}_m \rangle\rangle}(\tau_1, \tau_2, \dots, \tau_m)}{\text{exp}.\text{id}_j :: \tau_j} \quad \frac{\text{exp}_1 :: \text{arr}_{\langle\langle \cdot \rangle\rangle}(\tau) \quad \text{exp}_2 :: \text{int}}{\text{exp}_1[\text{exp}_2] :: \tau}$$

$$\frac{\text{id} :: (\tau_1, \tau_2, \dots, \tau_m) \longrightarrow \tau \quad \text{exp}_1 :: \tau_1 \quad \text{exp}_2 :: \tau_2 \quad \dots \quad \text{exp}_m :: \tau_m}{\text{id}(\text{exp}_1, \text{exp}_2, \dots, \text{exp}_m) :: \tau}$$

$$\frac{\text{exp} :: \tau}{\text{exp where decls} :: \tau}$$

$$\frac{\text{id} :: \tau \quad \text{exp} :: \tau}{\{ \text{id} = \text{exp} \} :: \tau}$$

$$\frac{\text{exp} :: \text{bool} \quad \text{exp}_1 :: \tau}{\{ \text{if exp then exp}_1 \} :: \text{void}} \quad \frac{\text{exp} :: \text{bool} \quad \text{exp}_1 :: \tau \quad \text{exp}_2 :: \tau}{\{ \text{if exp then exp}_1 \text{ else exp}_2 \} :: \tau}$$

$$\frac{\text{id} :: \text{int} \quad \text{exp}_1 :: \text{int} \quad \text{exp}_2 :: \text{int} \quad \text{exp}_3 :: \tau}{\{ \text{for id = exp}_1, \text{exp}_2 : \text{exp}_3 \} :: \text{void}} \quad \frac{\text{exp}_1 :: \text{bool} \quad \text{exp}_2 :: \tau}{\{ \text{while exp}_1 : \text{exp}_2 \} :: \text{void}}$$