Gramatika bez výrazů

			Predict
1. <prog>	→	<declarations> <body>	{var, begin}
2. <declarations>	→	**var** <declaration> <decl-list>	{var}
3. <declarations>	→	ϵ	{begin}
4. <declaration>	→	**id :** <type> **;**	{id}
5. <decl-list>	→	<declaration> <decl-list>	{id}
6. <decl-list>	→	ϵ	{begin}
7. <body>	→	**begin** <statements> **end .**	{begin}
8. <statements>	→	<statement> <st-list>	{begin, readln, write, id, if, while}
9. <statements>	→	ϵ	{end}
10. <st-list>	→	**;** <statement> <st-list>	{;}
11. <st-list>	→	ϵ	{end}
12. <statement>	→	**begin** <statements> **end**	{begin}
13. <statement>	→	**readln (id** <id-list> **)**	{readln}
14. <statement>	→	**write (expr** <expr-list> **)**	{write}
15. <statement>	→	**id := expr**	{id}
16. <statement>	→	**if expr then** <statement> **else** <statement>	{if}
17. <statement>	→	**while expr do** <statement>	{while}
18. <expr-list>	→	**, expr** <expr-list>	{,}
19. <expr-list>	→	ϵ	{)}
20. <id-list>	→	**, id** <id-list>	{,}
21. <id-list>	→	ϵ	{)}
22. <type>	→	**integer**	{integer}
23. <type>	→	**double**	{double}
24. <type>	→	**string**	{string}

	Empty	First	Follow
<prog>	$\emptyset$	{var, begin}	{$}
<declarations>	{ ϵ }	{var}	{begin}
<declaration>	$\emptyset$	{id}	{id, begin}
<decl-list>	{ ϵ }	{id}	{begin}
<body>	$\emptyset$	{begin}	{$}
<statements>	{ ϵ }	{begin, readln, write, id, if, while}	{end}
<st-list>	{ ϵ }	{;}	{end}
<statement>	$\emptyset$	{begin, readln, write, id, if, while}	{;, else, end}
<expr-list>	{ ϵ }	{,}	{)}
<id-list>	{ ϵ }	{,}	{)}
<type>	$\emptyset$	{integer, double, string}	{;}

LL tabulka	begin	end	var	id	if	while	readln	write	;	,	)	integer	double	string
<prog>	1		1											
<declarations>	3		2											
<declaration>				4										
<decl-list>	6			5										
<body>	7													
<statements>	8	9		8	8	8	8	8						
<st-list>		11							10					
<statement>	12			15	16	17	13	14						
<expr-list>										18	19			
<id-list>										20	21			
<type>												22	23	24

* u ostatních terminálů v LL tabulce je automaticky prázdné políčko

Zpracování výrazů

Priorita operátorů (všechny mají asociativitu zleva doprava):

1. * div
2. + -
3. < <= > >=
4. <> =

Gramatika:

1. E → id
2. E → E * E
3. E → E div E
4. E → E + E
5. E → E – E
6. E → E < E
7. E → E <= E
8. E → E > E
9. E → E >= E
10. E → E = E
11. E → E <> E
12. E → (E)

	*	div	+	-	<	<=	>	>=	=	<>	(	)	id	$
*	>	>	>	>	>	>	>	>	>	>	<	>	<	>
div	>	>	>	>	>	>	>	>	>	>	<	>	<	>
+	<	<	>	>	>	>	>	>	>	>	<	>	<	>
-	<	<	>	>	>	>	>	>	>	>	<	>	<	>
<	<	<	<	<	>	>	>	>	>	>	<	>	<	>
<=	<	<	<	<	>	>	>	>	>	>	<	>	<	>
>	<	<	<	<	>	>	>	>	>	>	<	>	<	>
>=	<	<	<	<	>	>	>	>	>	>	<	>	<	>
=	<	<	<	<	<	<	<	<	>	>	<	>	<	>
<>	<	<	<	<	<	<	<	<	>	>	<	>	<	>
(	<	<	<	<	<	<	<	<	<	<	<	=	<	
)	>	>	>	>	>	>	>	>	>	>		>		>
id	>	>	>	>	>	>	>	>	>	>		>		>
$	<	<	<	<	<	<	<	<	<	<	<	OK	<	OK

* [$,)] je OK protože závorka už nemusí patřit do výrazu a místo toho uzavírá seznam parametrů funkce