



Dokumentace

Implementace překladače imperativního jazyka IFJ23

Varianta - vv-BVS

Bez implementace rozšíření

Tým xeffen00

vedoucí Marek Effenberger (xeffen00) 25%

Samuel Hejníček (xhejniXX) 25%

Dominik Horut (xhorut01) 25%

Adam Valík (xvalik05) 25%

Brno, 3. prosince 2023

Obsah

1	Práce v týmu	2
1.1	Rozdělení práce mezi jednotlivé členy týmu	2
2	Implementace překladače	3
2.1	Lexikální analýza	3
2.2	Syntaktická a sémantická analýza	3
2.2.1	Parser	3
2.2.2	LL gramatika	3
2.2.3	Expression parser	5
2.2.4	Precedenční tabulka	5
2.2.5	Gramatika výrazů	5
2.3	Generování kódu	5

1 Práce v týmu

1.1 Rozdělení práce mezi jednotlivé členy týmu

Marek Effenberger

- Návrh automatu pro lexikální analýzu
-
-
-

Samuel Hejníček

- Návrh automatu pro lexikální analýzu
- Implementace lexikálního analyzátoru
-
-

Dominik Horut

- Implementace lexikálního analyzátoru
-
-
-

Adam Valík

- Návrh automatu pro lexikální analýzu
-
-
-

2 Implementace překladače

2.1 Lexikální analýza

2.2 Syntaktická a sémantická analýza

2.2.1 Parser

2.2.2 LL gramatika

1. $\langle prog \rangle \rightarrow EOF$
2. $\langle prog \rangle \rightarrow \langle func_def \rangle \langle prog \rangle$
3. $\langle prog \rangle \rightarrow \langle body \rangle \langle prog \rangle$
4. $\langle func_def \rangle \rightarrow func\ id(\langle params \rangle) \langle ret_type \rangle \{ \langle local_body \rangle \}$
5. $\langle params \rangle \rightarrow \langle par_name \rangle \langle par_id \rangle : \langle type \rangle \langle params_n \rangle$
6. $\langle params \rangle \rightarrow \varepsilon$
7. $\langle par_name \rangle \rightarrow _$
8. $\langle par_name \rangle \rightarrow id$
9. $\langle par_id \rangle \rightarrow _$
10. $\langle par_id \rangle \rightarrow id$
11. $\langle type \rangle \rightarrow Int$
12. $\langle type \rangle \rightarrow Int?$
13. $\langle type \rangle \rightarrow Double$
14. $\langle type \rangle \rightarrow Double?$
15. $\langle type \rangle \rightarrow String$
16. $\langle type \rangle \rightarrow String?$
17. $\langle params_n \rangle \rightarrow , \langle params \rangle$
18. $\langle params_n \rangle \rightarrow \varepsilon$
19. $\langle ret_type \rangle \rightarrow \rightarrow \langle type \rangle$
20. $\langle ret_type \rangle \rightarrow \varepsilon$
21. $\langle local_body \rangle \rightarrow \langle body \rangle \langle local_body \rangle$
22. $\langle local_body \rangle \rightarrow \varepsilon$
23. $\langle body \rangle \rightarrow \langle var_def \rangle$

24. $\langle body \rangle \rightarrow \langle condition \rangle$
25. $\langle body \rangle \rightarrow \langle cycle \rangle$
26. $\langle body \rangle \rightarrow \langle assign \rangle$
27. $\langle body \rangle \rightarrow \langle func_call \rangle$
28. $\langle body \rangle \rightarrow \langle ret \rangle$
29. $\langle body \rangle \rightarrow \varepsilon$
30. $\langle ret \rangle \rightarrow return \langle exp \rangle$
31. $\langle ret \rangle \rightarrow return$
32. $\langle var_def \rangle \rightarrow let\ id \langle opt_var_def \rangle$
33. $\langle var_def \rangle \rightarrow var\ id \langle opt_var_def \rangle$
34. $\langle opt_var_def \rangle \rightarrow : \langle type \rangle$
35. $\langle opt_var_def \rangle \rightarrow \langle assign \rangle$
36. $\langle opt_var_def \rangle \rightarrow : \langle type \rangle \langle assign \rangle$
37. $\langle assign \rangle \rightarrow = \langle exp \rangle$
38. $\langle assign \rangle \rightarrow = \langle func_call \rangle$
39. $\langle func_call \rangle \rightarrow id (\langle args \rangle)$
40. $\langle args \rangle \rightarrow \varepsilon$
41. $\langle arg \rangle \rightarrow \langle exp \rangle$
42. $\langle arg \rangle \rightarrow id : \langle exp \rangle$
43. $\langle args_n \rangle \rightarrow , \langle arg \rangle \langle args_n \rangle$
44. $\langle args_n \rangle \rightarrow \varepsilon$
45. $\langle condition \rangle \rightarrow if \langle exp \rangle \{ \langle local_body \rangle \} else \{ \langle local_body \rangle \}$
46. $\langle condition \rangle \rightarrow if let\ id \{ \langle local_body \rangle \} else \{ \langle local_body \rangle \}$
47. $\langle cycle \rangle \rightarrow while \langle exp \rangle \{ \langle local_body \rangle \}$

2.2.3 Expression parser

2.2.4 Precedenční tabulka

	+	-	*	/	<	>	<=	>=	!=	==	!	??	()	i	\$
+	>	>	<	<	>	>	>	>	>	>	<	>	<	>	<	>
-	>	>	<	<	>	>	>	>	>	>	<	>	<	>	<	>
*	>	>	>	>	>	>	>	>	>	>	<	>	<	>	<	>
/	>	>	>	>	>	>	>	>	>	>	<	>	<	>	<	>
<	<	<	<	<							<	>	<	>	<	>
>	<	<	<	<							<	>	<	>	<	>
<=	<	<	<	<							<	>	<	>	<	>
>=	<	<	<	<							<	>	<	>	<	>
!=	<	<	<	<							<	>	<	>	<	>
==	<	<	<	<							<	>	<	>	<	>
!	>	>	>	>	>	>	>	>	>	>		>		>	>	>
??	<	<	<	<	<	<	<	<	<		<	<	<	>	<	>
(<	<	<	<	<	<	<	<	<	<	<	<	<	=	<	
)	>	>	>	>	>	>	>	>	>	>	>	>		>		>
i	>	>	>	>	>	>	>	>	>	>	>	>		>		>
\$	<	<	<	<	<	<	<	<	<	<	<	<	<		<	

2.2.5 Gramatika výrazů

1. $E \rightarrow i$
2. $E \rightarrow (E)$
3. $E \rightarrow E + E$
4. $E \rightarrow E - E$
5. $E \rightarrow E * E$
6. $E \rightarrow E / E$
7. $E \rightarrow E < E$
8. $E \rightarrow E > E$
9. $E \rightarrow E <= E$
10. $E \rightarrow E >= E$
11. $E \rightarrow E == E$
12. $E \rightarrow E != E$
13. $E \rightarrow E!$
14. $E \rightarrow E ?? E$

2.3 Generování kódu