

Dokumentácia k projektu z predmetu IFJ a IAL

Implementácia interpretu imperatívneho jazyka IFJ16

Tým 016, varianta b/4/II

5. decembra 2016

Riešitelia:

Sámel Šimon, xsamel02 , 0%

Patrik Sztefek, xsztef02 , 25%

Tomáš Szúcs, xszucs01 , 25%

Marek Šipoš, xsipos03 , 25%

Jakub Štol, xstolj00 , 25%

Obsah

<i>Dokumentácia k projektu z predmetu IFJ a IAL</i>	<i>2</i>
<i>Implementácia interpreteru imperatívneho jazyka IFJ16</i>	<i>2</i>
<i>Tým 016, varianta b/4/II</i>	<i>2</i>
<i>5. decembra 2016.....</i>	<i>2</i>

1. Úvod

V tejto dokumentácii je popísaný vývoj a implementácia interpreta imperatívneho jazyka IFJ16, ktorý je zjednodušenou podmnožinou jazyka Java SE 8. Celá dokumentácia je rozdelená do kapitol, v ktorých sú popísané jednotlivé časti interpreta. Od návrhu, implementácie až po samotnú prácu v tíme.

2. Návrh a implementácia

V tejto kapitole sa popisuje návrh a implementácia interpretu.

2.1 Návrh interpretu

Lexikálna Analýza

Syntaktická analýza

- prediktívna
- precedentná

Sémantická analýza

Generátor trojadresného kódu

Interpret

4. Vývoj interpretu

V tejto kapitole je popísane rozdelenie práce na interprete, až po použité nástroje pri vývoji.

3.1 Rozdelenie práce

Prácu pred začatím vývoja interpretu rozdeľoval náš vedúci Patrik Sztefek. Rozdeľoval ju spôsobom, kto bol ako zručný v programovaní. Ale zároveň nám dal nám možnosť si vybrať, ktorú časť chceme riešiť. Aj keď sa môže zdať, že rozdelenie bodov neprináleží k odvedenej práci jednotlivých členov, každý sa snažil ako len mohol. Nižšie uvedené popisy slúžia pre predstavu, kto akú časť problematiky riešil.

Patrik Sztefek - Generátor, algoritmy, interpret, testovanie

Tomáš Szűcs – Lexikálny analyzátor, výpomoc, dokumentácia,

Marek Šipoš – Syntaktický analyzátor, Sémantický analyzátor, testovanie, Generátor

Jakub Štol - Lexikálny analyzátor, vstavané funkcie, výpomoc

3.2 Komunikácia a schôdze

Už pred začatím vývoja sme vedeli, že nebude čas na pravidelne schôdze z dôvodu odlišnosti rozvrhu každého člena, tak sme sa stretávali len vo vážnych prípadoch. Namiesto schôdze sme zvolili variantu Facebook-ovej skupiny a Facebook instant-messaging. Výhoda spočíva v tom, že vieme komunikovať z ktoréhokoľvek miesta. Zároveň podávať správu o pokroku na svojej práci.

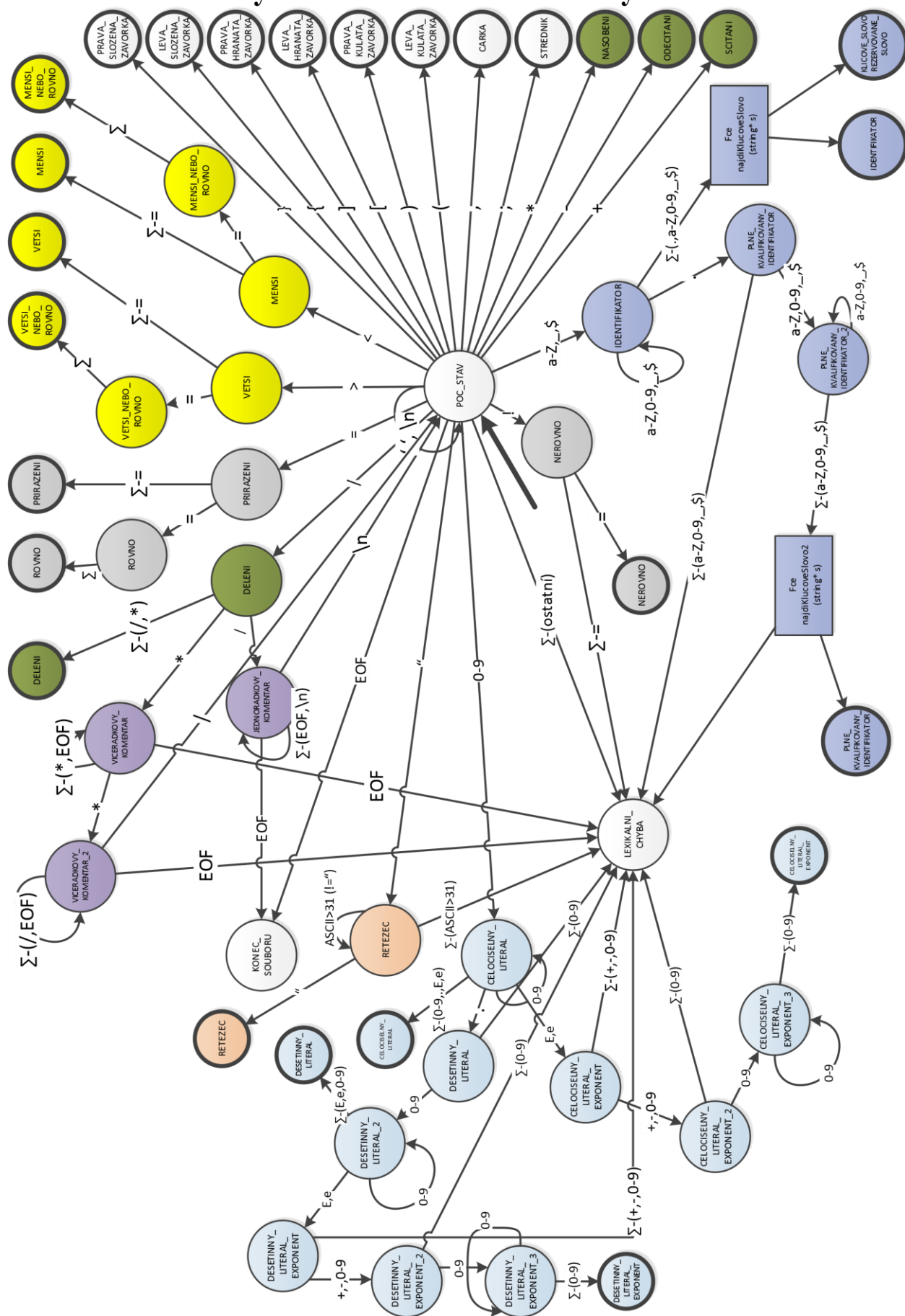
3.3 Použité nástroje

Na vývoj interpretu sa použil verzovací systém **Git**. Ako úložný priestor pre náš projekt poslúžil **GitHub**. Z dôvodu dostupnosti na internete a zároveň má zabudované funkcie. Ktoré sprahľadňujú zmeny, a to v konkrétnom súbore na konkrétnom riadku. Aj z dôvodu poskytnutia voľnej licencie pre študentov, aj kvôli chuti vyskúšať moderné trendy. Ako ďalší nástroj sme použili **Google docs**. Najviac užitočný bol vo fázach vývoja, keď sme mali obrovské množstvo hlavičkových a zdrojových súborov. Zároveň slúžil ako poznámkový blok, či už pre brainstorming alebo popis jednotlivej časti interpretu.

3.4 Metodika vývoja softwaru

Ranné fázy vývoja by sa dali prirovnať k V-modelu, či už z pohľadu testovania alebo samotnej implementácie. Z dôvodu, že každá časť interpretu sa navrhla, implementovala následne testovala.

Príloha A: Konečný automat Lexikálneho analyzátoru



Priloha B: LL gramatika a precedencna tabulka

01 PROGRAM \rightarrow TRIDA PROGRAM

02 PROGRAM \rightarrow eof

03 TRIDA \rightarrow class identifikator { SEZNAM-DEFINIC-STATIC }

04 SEZNAM-DEFINIC-STATIC \rightarrow static DATOVY-TYP DEFINICE-STATIC SEZNAM-DEFINIC-STATIC

05 SEZNAM-DEFINIC-STATIC $\rightarrow \epsilon$

06 DEFINICE-STATIC \rightarrow DEFINICE-FUNKCE

07 DEFINICE-STATIC \rightarrow DEFINICE-PROMENNA ;

08 DEFINICE-PROMENNA \rightarrow identifikator DEF-PROM-KONEC

09 DEF-PROM-KONEC \rightarrow PRIRAZENI

10 DEF-PROM-KONEC $\rightarrow \epsilon$

11 DEFINICE-FUNKCE \rightarrow f_identifikator (SEZNAM-PARAMETRU) SLOZENY-PRIKAZ

12 SEZNAM-PARAMETRU \rightarrow PARAMETR-PRVNI PARAMETR-DALSI

13 SEZNAM-PARAMETRU $\rightarrow \epsilon$

14 PARAMETR-PRVNI \rightarrow primitivni_typ identifikator

15 PARAMETR-DALSI \rightarrow , primitivni_typ identifikator PARAMETR-DALSI

16 PARAMETR-DALSI $\rightarrow \epsilon$

17 SEZNAM-VSTUPU \rightarrow vyraz VSTUP-DALSI

18 SEZNAM-VSTUPU \rightarrow identifikator VSTUP-DALSI

19 SEZNAM-VSTUPU $\rightarrow \epsilon$

20 VSTUP-DALSI \rightarrow , VSTUP-KONEC

21 VSTUP-DALSI $\rightarrow \epsilon$

22 VSTUP-KONEC \rightarrow vyraz VSTUP-DALSI

23 VSTUP-KONEC \rightarrow identifikator VSTUP-DALSI

24 SLOZENY-PRIKAZ \rightarrow { BLOK-PRIKAZU }

25 BLOK-PRIKAZU \rightarrow PRIKAZ BLOK-PRIKAZU

26 BLOK-PRIKAZU $\rightarrow \epsilon$

27 PRIKAZ \rightarrow primitivni_typ DEFINICE-PROMENNA ;

28 PRIKAZ \rightarrow vyraz ;

29 PRIKAZ \rightarrow identifikator POUZITI ;

30 PRIKAZ \rightarrow f_identifikator VOLANI-FUNKCE ;

31 PRIKAZ \rightarrow return NAVRAT-KONEC ;

32 PRIKAZ \rightarrow if (vyraz) SLOZENY-PRIKAZ else SLOZENY-PRIKAZ

33 PRIKAZ \rightarrow while (vyraz) SLOZENY-PRIKAZ

34 POUZITI \rightarrow PRIRAZENI

35 POUZITI $\rightarrow \epsilon$

36 VOLANI-FUNKCE \rightarrow (SEZNAM-VSTUPU)

37 NAVRAT-KONEC \rightarrow vyraz

38 NAVRAT-KONEC \rightarrow identifikator

39 NAVRAT-KONEC $\rightarrow \epsilon$

40 PRIRAZENI \rightarrow = PRAVA-STRANA

41 PRAVA-STRANA \rightarrow vyraz

42 PRAVA-STRANA \rightarrow identifikator

43 PRAVA-STRANA \rightarrow f_identifikator VOLANI-FUNKCE

44 DATOVY-TYP \rightarrow void

45 DATOVY-TYP \rightarrow primitivni_typ

vyraz \rightarrow Předá se precedenční synt. analýze - symboly: identifikator(proměnné) číslo řetězec () aritmetické + relační operátory

Naše gramatika je jednoznačná LL(1) gramatika. Protože však jazyk IFJ16 nelze být popsán čistou LL(1) gramatikou, bylo pro popsání jazyka použito drobných heuristik, které mírně “ohýbají” princip syntaktické analýzy.

Precedenčná tabuľka

[illegible]

