

Dokumentace k projektu IFJ/IAL

Implementace překladače imperativního jazyka IFJ19

Tým 048 varianta II

11. prosince 2019

Autoři:

Veverka Jiří xvever12

Halabica Michal xhalab00

Salih Adam xsalih01

Rusín Tomáš xrusin04

Abstrakt

Tento projekt se zabývá návrhem a implementací jazyka IFJ19, který je zjednodušenou podmnožinou jazyka Python 3. Implementace je realizována v jazyce C. Program je rozdělen na 3 části lexikální analýzu, syntaktickou analýzu zdola nahoru a syntaktickou analýzu shora dolů se sémantickou analýzou. Zvolili jsme si zadání II, kde bylo úkolem implementovat tabulku symbolů pomocí tabulky s rozptýlenými položkami.

Obsah

[1 Lexikální analýza 4](#_Toc26135124)

[2 Syntaktická analýza 4](#_Toc26135125)

[2.1 Syntaktická analýza shora dolů 4](#_Toc26135126)

[2.1.1 LL- gramatika 4](#_Toc26135127)

[2.2 Syntaktická analýza zdola nahoru 5](#_Toc26135128)

[2.2.1 Precedenční tabulka 5](#_Toc26135129)

[3 Datové typy 6](#_Toc26135130)

[3.1 Zásobník 6](#_Toc26135131)

[3.2 Tabulka s rozptýlenými položkami 6](#_Toc26135132)

[3.3 Řetězec 6](#_Toc26135133)

[3.4 Fronta 6](#_Toc26135134)

[4 Práce v týmu 6](#_Toc26135135)

[5 Rozdělení práce 6](#_Toc26135136)

[6 Závěr 6](#_Toc26135137)

[7 Přílohy 6](#_Toc26135138)

[7.1 Konečný automat lexikálního analyzátoru 6](#_Toc26135139)

# Lexikální analýza

Lexikální analyzátor, dále jen (LA) je implementován pomocí deterministického konečného automatu pomocí konstrukce //TODO(zásobník-switch) v jazyce C. Schéma je v příloze.

LA je vstupní částí překladače. Zpracovává vstupní zdrojový kód v jazyce IFJ19, který je přiveden na standartní vstup. Načítá vstup po jednom znaku a postupně generuje jednotlivé tokeny.

# **Syntaktická analýza**

## Syntaktická analýza shora dolů

### LL- gramatika

|  |
| --- |
| <body> → {statement}EOL{body} |
| <function> → def <id> ({ang-list}):EOL<ident>{body}<dedent> |
| <arg-list> → ε |
| <arg-list> → <id>,{arg-list} |
| <if> → if {expression}:EOL<indent>{body}<dedent> <if\_extra> |
| <if\_extra> → elif {expression}:EOL<indent>{body}<dedent><if\_extra> |
| <if\_extra> → else :EOL<indent>{body}<dedent> |
| <if\_extra> → ε |
| <while> → while {expression}:EOL<indent>{body}<dedent> |
| <statement> → <id> = TODO+-\*/ {expression}EOL |
| <statement> → {expression} |
| <statement> → {while} |
| <statement> → {if} |
| <statement> → {pass||break} |
| <statement> → return <expression> |
| <statement> → ε |
| <expression> → ε |
| <expresion> → <id> (<expresion-list>) |
| <expresion> → (<expression>) |
| <expresion> → <expression> <<operator>> <expression> |
| <expresion> → <<raw\_val>> |
| <expresion> → !<expression> |
| <expresion-list> →<expression> |
| <expresion-list> →<expression>, <expression\_list> |
| <expression\_list> → ε |
| <<value>> → <id> || <<const\_val>> |
| <<const\_val>> → <int || float || string || none> |
| TODO<<operator>> → +,-,\*,/ || <<logical\_op>> |
| TODO <<logical\_op>> → ==, !=,>,>=,<,<=,&&,|| |
|  |

TODO +-\*/ <assignment> →{+-\*/}

TODO psát operator ?

TODO boolop rozšíření bude ?

## Syntaktická analýza zdola nahoru

### Precedenční tabulka

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | +,- | \*, / | // | < | > | <= | >= | == | id | $ | ( | ) | != |
| +,- | > | < | < | > | > | > | > | > | < | > | < | > | > |
| \*, / | > | > | > | > | > | > | > | > | > | > | < | > | > |
| // | > | > | > | > | > | > | > | > | > | > | < | > | > |
| < | < | < | < | > | > | > | > | > | < | > | < | > | > |
| > | < | < | < | > | > | > | > | > | < | > | < | > | > |
| <= | < | < | < | > | > | > | > | > | < | > | < | > | > |
| >= | < | < | < | > | > | > | > | > | < | > | < | > | > |
| == | < | < | < | > | > | > | > | > | < | > | < | > | > |
| id | > | < | < | > | > | > | > | > |  | > |  | > | > |
| $ | < | < | < | < | < | < | < | < | < |  | < |  | < |
| ( | < | < | < | < | < | < | < | < | < |  | < | = | < |
| ) | > | > | > | > | > | > | > | > |  | > |  | > | > |
| != | < | < | < | > | > | > | > | > | < | > | < | > | > |

# Datové typy

## Zásobník

## Tabulka s rozptýlenými položkami

## Řetězec

## Fronta

# Práce v týmu

# Rozdělení práce

# Závěr

# Přílohy

## Konečný automat lexikálního analyzátoru