**Formální jazyky a překladače**

2016/2017

**Dokumentácia interpretu imperativního jazyka IFJ16**

**Tým 013, varianta b/3/I**

Rozšírenia: BOOLOP

**Zoznam autorov:**

Šuba Adam xsubaa00 25%

Tóth Adrián xtotha01 25%

Šuhaj Peter xsuhaj02 25%

Paliesek Jakub xpalie00 25%

**Obsah**

[1. Úvod 2](#_Toc468124121)

[2. Interpret 3](#_Toc468124122)

[2.1. Návrh 3](#_Toc468124123)

[2.2. Implementácia 3](#_Toc468124124)

[2.3. Vývojový cyklus 3](#_Toc468124125)

[2.4. Spôsob práce 3](#_Toc468124126)

[2.5. Špeciálne použité techniky 3](#_Toc468124127)

[3. Algoritmy 4](#_Toc468124128)

[3.1. (b) vyhľadávanie s použitím Boyer-Moorovho algoritmu 4](#_Toc468124129)

[3.2. (3) radenie pomocou Shell sort algoritmu 4](#_Toc468124130)

[3.3. (I) implementácia tabuľky symbolov pomocou binárneho vyhľadávacieho stromu 4](#_Toc468124131)

[4. Záver 5](#_Toc468124132)

[5. Príloha 6](#_Toc468124133)

[5.1. Diagram konečného automatu lexikálnej analýzy 6](#_Toc468124134)

[5.2. LL-gramatika syntaktického analyzátora 7](#_Toc468124135)

[5.3. Precedenčná tabuľka syntaktického analyzátora 8](#_Toc468124136)

# Úvod

Dokumentácia popisuje vývoj interpretu imperatívneho jazyka IFJ16. Jazyk IFJ16 je podmnožina jazyka Java SE 8, čo je staticky typovaný objektovo orientovaný jazyk.

Úlohou nášho interpretu je kontrola zdrojového kódu napísaného v jazyku IFJ16 a následne jeho interpretácia.

Nami zvolený Tím 013 mal priradenú variantu b/3/I. Varianta obsahovala: (b) vyhľadávanie s použitím Boyer-Moorovho algoritmu, (3) radenie pomocou Shell sort algoritmu, (I) implementácia tabuľky symbolov pomocou binárneho vyhľadávacieho stromu.

Dokumentácia je rozdelená na kapitoly, ktoré bližšie popisujú jednotlivé časti interpretu a spôsob ich riešenia. Taktiež obsahuje aj popis algoritmov, popis rozdelenia úloh a spôsobu práce v tímu. V závere dokumentácie sa nachádza zhrnutie našej práce.

V prílohe sa nachádzajú: diagram konečného automatu lexikálnej analýzy, LL-gramatika a precedenčná tabuľka syntaktického analyzátora.

# Interpret

## Návrh

## Implementácia

## Vývojový cyklus

## Spôsob práce

## Špeciálne použité techniky

# Algoritmy

## (b) vyhľadávanie s použitím Boyer-Moorovho algoritmu

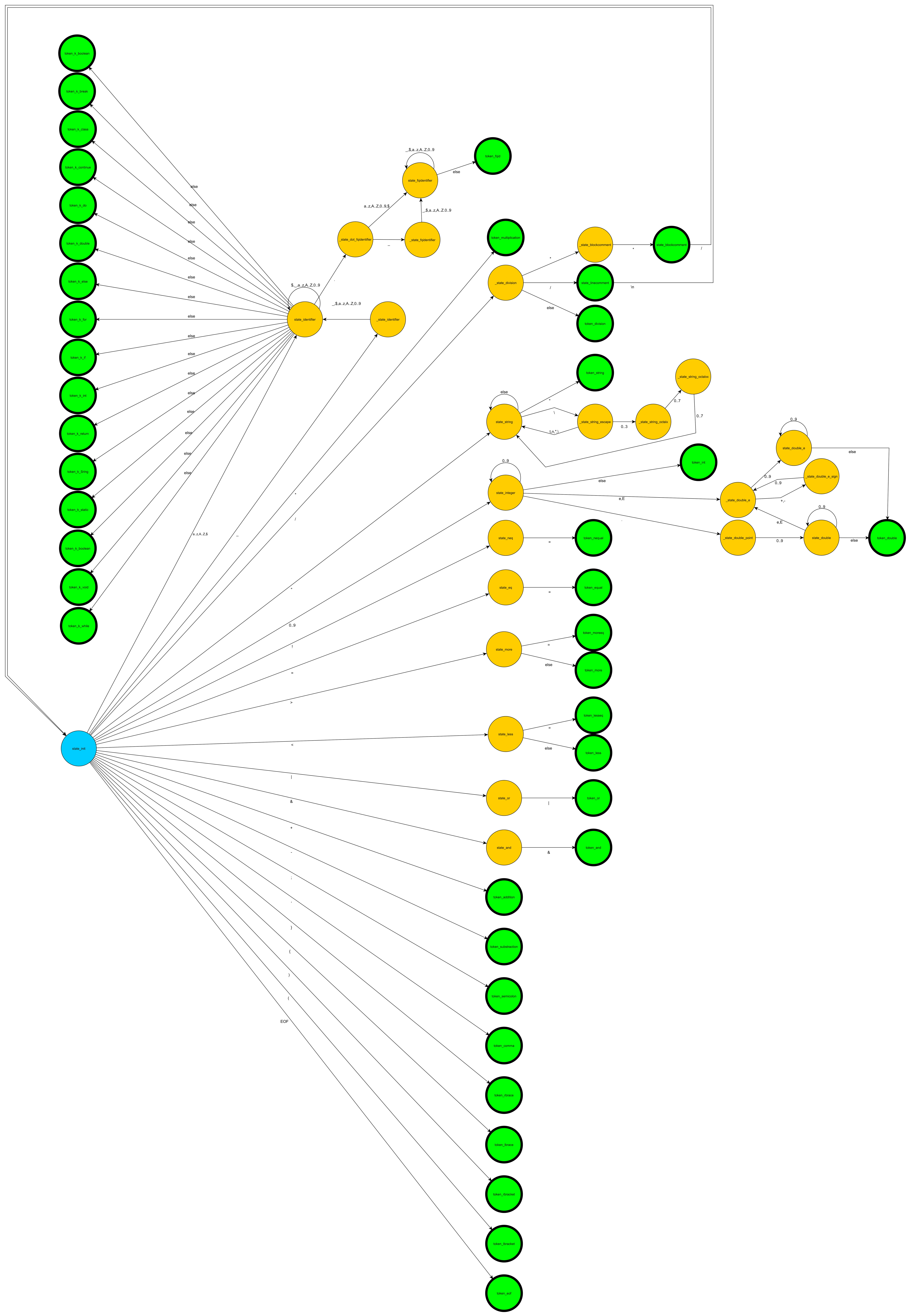
## (3) radenie pomocou Shell sort algoritmu

## (I) implementácia tabuľky symbolov pomocou binárneho vyhľadávacieho stromu

# Záver

# Príloha

## Diagram konečného automatu lexikálnej analýzy



## LL-gramatika syntaktického analyzátora

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rule** | **Empty** | **First** | **Follow** | **Predict** |
| C-LIST → **class id {** MEMB-LIST **}** C-LIST | 0 | class, $ | $ | class |
| C-LIST → **$** | $ |
| MEMB-LIST → C-MEMB MEMB-LIST | 1 | static | } | static |
| MEMB-LIST → ε | } |
| C-MEMB → **static** C-MEMB1 | 0 | static | static, } | static |
| C-MEMB1 → TYPE **id** C-MEMB2 | 0 | void, int, double, String, boolean | static, } | int, double, String, boolean |
| C-MEMB1 → **void id** C-MEMBFUNC | void |
| TYPE → **int** | 0 | int, double, String, boolean | id | int |
| TYPE → **double** | double |
| TYPE → **String** | String |
| TYPE **→ boolean** | boolean |
| C-MEMBFUNC → **(** FN-DEF-PLIST **) {** FN-BODY **}** | 0 | ( | static, } | ( |
| C-MEMB2 → **=** E**;** | 0 | ;, =, ( | static, } | = |
| C-MEMB2 → **;** | ; |
| C-MEMB2 **→** C-MEMBFUNC | ( |
| FN-DEF-PLIST → ε | 1 | int, double, String, boolean | ) | ) |
| FN-DEF-PLIST → PAR-DEF FN-DEF-PLIST1 | int, double, String, boolean |
| FN-DEF-PLIST1 → **,** PAR-DEF FN-DEF-PLIST1 | 1 | "," | ) | , |
| FN-DEF-PLIST1 → ε | ) |
| PAR-DEF → TYPE **id** | 0 | int, double, String, boolean | ",", ) | int, double, String, boolean |
| FN-BODY → STAT FN-BODY | 1 | id, fqid, {, if, while, return, int, double, String, boolean | } | id, fqid, {, if, while, return |
| FN-BODY → TYPE **id** OPT-ASSIGN **;** FN-BODY | int, double, String, boolean |
| FN-BODY → ε | } |
| STAT-COM → **{** STAT-LIST **}** | 0 | { | else, id, fqid, {, if, while, return,  int, double, String, boolean, } | { |
| STAT-LIST → STAT STAT-LIST | 1 | id, fqid, {, if, while, return | } | id, fqid, {, if, while, return |
| STAT-LIST → ε | } |
| ID → **id** | 0 | id, fqid | (, =, ), “,” | id |
| ID → **fqid** | fqid |
| STAT → IDAS-CA **;** | 0 | id, fqid, {, if, while, return | fqid, id, {, if, while, return, int,  double, String, boolean, } | id, fqid |
| STAT **→** STAT-COM | { |
| STAT → **if** **(** E **)** STAT-COM **else** STAT-COM | if |
| STAT → **while** **(** E **)** STAT-COM | while |
| STAT **→ return** RET-VAL **;** | return |
| AS-CA → **(** FN-PLIST **)** | 0 | (, = | ; | ( |
| AS-CA → **=** ASSIGN | = |
| ASSIGN → E | 0 | E, id, fqid | ; | E |
| ASSIGN → ID **(** FN-PLIST **)** | id, fqid |
| OPT-ASSIGN → ε | 1 | = | ; | ; |
| OPT-ASSIGN → **=** ASSIGN | = |
| FN-PLIST → ε | 1 | id, fqid, string-literal, int-literal,  double-literal,  boolean-literal | ) | ) |
| FN-PLIST → VAL-ID FN-PLIST1 | id, fqid, string-literal,  int-literal, double-literal, boolean-literal |
| FN-PLIST1 → ε | 1 | "," | ) | ) |
| FN-PLIST1 → **,** VAL-ID FN-PLIST1 | , |
| VAL-ID → ID | 0 | id, fqid, string-literal, int-literal,  double-literal,  boolean-literal | ",", ) | id, fqid |
| VAL-ID → **string-literal** | string-literal |
| VAL-ID **→ int-literal** | int-literal |
| VAL-ID **→ double-literal** | double-litera |
| VAL-ID **→ boolean-literal** | boolean-literal |
| RET-VAL → E | 1 | E | ; | E |
| RET-VAL → ε | ; |

## Precedenčná tabuľka syntaktického analyzátora

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