FJP překladač do PL/0

Martin Kantořík, Michal Tušl

Jak se změnilo naše zadání

- Základní funkcionalita
- Přidání *else* větve
- Cykly (*for*, *do-while*, *repeat-until*)
- Datový typ boolean
- Násobné přiřazení
- Pole
- Návratová hodnota
- String (1)
- Operátor pro porovnání řetězců (2)
- Switch (1)
- Podmíněné přiřazení (1)
- parametry předávané hodnotou (2)

Stav gramatiky

- V ANTLRv4
- Generována do Javy
- 95% procent hotovo
- Funkční všechny vlastnosti

```
grammar SLLanguage;

compilationUnit
    :   translationUnit? EOF
    ;

translationUnit
    :   allCode
    |   allCode translationUnit
    ;

allCode
    :   functionDefinition
    |   declaration
    |   cycle
    |   functionCall
    |   constDeclaration
    |   expressionStatement
    ;
```

```
cycle
    : 'for' '(' forCondition ')' compoundStatement
    | 'while' '(' expression ')' compoundStatement
    | 'do' compoundStatement 'while' '(' expression ')' ';'
    | 'until' compoundStatement 'until' '(' expression ')' ';'
    | 'if' '(' logicalOrExpression ')' compoundStatement ('else' compoundStatement)?
    | 'switch' '(' expression ')' '{' (labeledStatement)* '}'
    ;

statement
    :   labeledStatement
    |   compoundStatement
    |   expressionStatement
    |   cycle
    |   jumpStatement
    |   functionCall
    ;

labeledStatement
    :   'case' DigitSequence ':' (compoundStatement | expressionStatement | jumpStatement | functionCall)*
    |   'default' ':' (compoundStatement | expressionStatement | jumpStatement | functionCall)
    ;

compoundStatement
    :   '{' blockItemList? '}'
    ;
```

Testovací soubor

- Zatím testuje především gramatiku
- Testuje téměř všechny důležité věci

```
int ahoj = 5;
int b = 3;
boolean c = true;
neee(3,3);

b = 5;

b = ahoj = c;


int function neee(int first, int second){
    int ahoj = 8;
    int i;
    for (i = 0; i < 5; i++) {
        ahoj = 5 * i + ahoj;
    }
    return ahoj;
}

int p = (((5 + 5) * 6) + 5 * (1 + 3));

int [] pole = new int [5];

int b = pole [2];

const int CON = 5;

string test = "ahoj";

if (ahoj > 5) {
    c = (10 > ahoj) ? b = 5 : b = 1;
}
```

Stav překladače

- Struktury pro:
  - Jednotlivé proměnné
  - Generované řádky kódu
- Struktura pro proměnné
  - Zajišťuje vhodné hodnoty
  - Jedinečné názvy v rámci zanoření
  - Udržuje nezměnitelné konstanty
  - a další
- Struktura pro kód
  - Pro snadnější zapsání do souboru
  - Snadno zjistitelné, kde je chyba
  - Uchovávání kódu, hodnoty, indexu a zanoření

Vygenerovaná gramatika

- Vygenerovaný strom z testovacího souboru
  - Nastavení vlastních listenerů
  - Volání funkcí z jiných tříd
  - Listování v Contextu

Závěr

- Chybí nám toho ještě dost
  - Ale už bude více času a méně rozptýlení

Děkujeme za pozornost