

# “Java” do PL/0

Jan Čarnogurský, Vojtěch Danišík

# Základní informace

- vybráno 1. zadání
- snaha vytvořit kompilátor pro jazyk připomínající Javu
- (bude) verzováno v GITu <https://github.com/danisik/FJP>

# Funkcionalita

- minimální (celočíselné proměnné a konstanty, přiřazení, ..)
- zbylé cykly
- větev else
- datový typ boolean
- switch
- násobné přiřazení
- parametry předávané hodnotou
- návratová hodnota z podprogramu

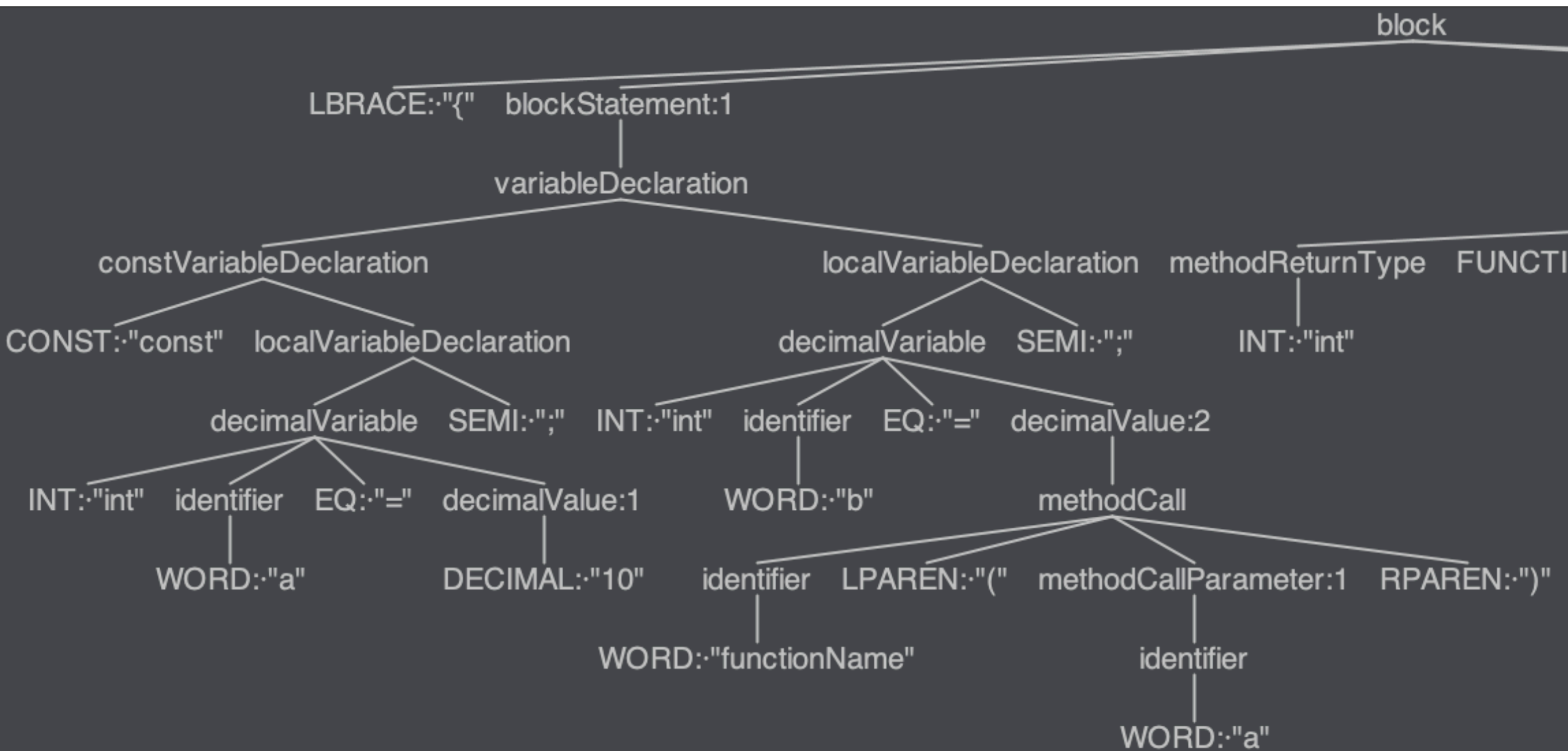
# Gramatika

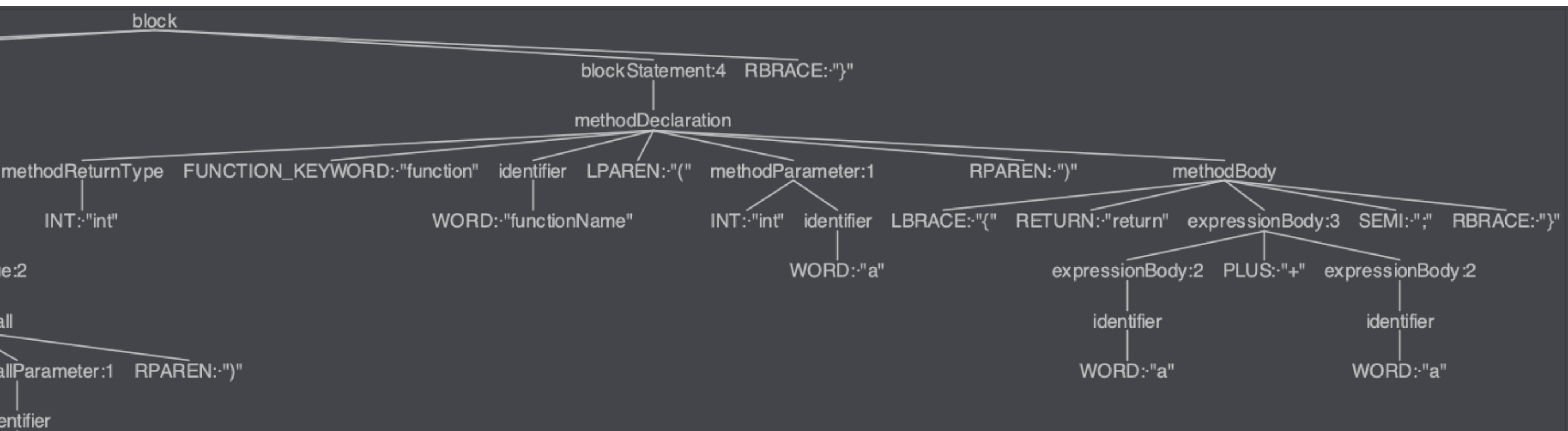
- zjednodušená Java
- použití nástroje ANTLRv4

```
const int a = 10;  
int b = functionName(a);  
  
int function functionName(int a) {  
    return a + a;  
}
```

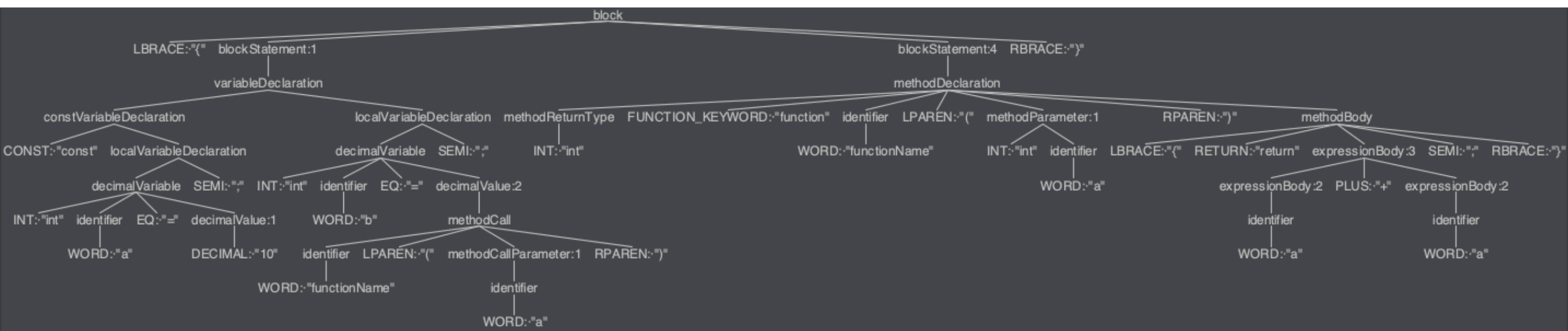
- **WORD** : [a-z] | [A-z] | '\_'
- **DECIMAL** : [0-9]<sup>+</sup>
- **identifier** : WORD (WORD | DECIMAL)<sup>\*</sup>
- **decimalVariable** : int identifier (parallelDeclaration)<sup>\*</sup> = (DECIMAL | methodCall)
- **boolVariable** : boolean identifier (parallelDeclaration)<sup>\*</sup> = (TRUE | FALSE | methodCall)
- **parallelDeclaration** : = identifier
- **localVariableDeclaration** (decimalVariable | boolVariable) ;
- **constVariableDeclaration** : const localVariableDeclaration
- **variableDeclaration** : (localVariableDeclaration | constDeclaration)<sup>+</sup>
- **block** : { blockStatement<sup>\*</sup> }
- **blockStatement** : variableDeclaration | statement | methodCall ; |  
methodDeclaration | setVariable
- **methodDeclaration** : (int | boolean | void) function identifier ( methodParametr? )  
{ blockStatement<sup>\*</sup> return (identifier | DECIMAL | TRUE | FALSE | methodCall )? ; }
- **methodCall** : identifier ( methodCallParametr? )
- **methodCallParameter** : identifier | identifier, methodCallParameter
- **methodParametr** : (boolean | int) identifier | (boolean | int) identifier,  
methodParametr

- **statement :**
  - IF expression block (ELSE block)?
  - FOR forControl block
  - WHILE expression block
  - DO block WHILE expression
  - SWITCH expression { switchBlockStatement\* }
- **expression :** LPAREN expressionBody RPAREN
- **expressionBody :**
  - (DECIMAL | TRUE | FALSE)
  - identifier
  - expressionBody op=(PLUS | MINUS | .... ) expressionBody
- **forControl :** ( DECIMAL '...' DECIMAL )
- **switchBlockStatement :** case DECIMAL : block | default : block









**Děkujeme za pozornost**