

# Języki formalne i kompilatory

## Projekt

Program do upraszczania wyrażeń algebraicznych

## Autorzy

Ciałowicz Robert [robcial@student.agh.edu.pl](mailto:robcial@student.agh.edu.pl)

Szpila Magdalena [mszpila@student.agh.edu.pl](mailto:mszpila@student.agh.edu.pl)

## Opis

Program służy do upraszczania wyrażeń algebraicznych. Obsługuje operacje `+`, `-`, `*`, `/`, `(`, `)` oraz dowolną symboliczną nazwę zmiennej. Domyślnym zachowaniem programu jest pobieranie wyrażeń z pliku wsadowego `/example/example.txt` linia po linii oraz zwracanie wyników do `example/example_result.txt`. Lokalizację pliku wsadowego można zmienić poprzez zmianę marametru `inputFilePath` w klasie `main`.

## Uruchamianie

1. Uruchom projekt w IntelliJ
2. Zainstaluj plugin Antlr4
3. Uruchom `mvn clean package`
4. Uruchom metodę `main()` klasy `Main`

## Architektura i implementacja

### Wykorzystane technologie:

- Java 8
- Antrl4 i Antlr4 plugin

### Etapy przetwarzania wyrażeń algebraicznych:

Główna klasa programu do upraszczania wyrażeń algebraicznych składa się z następujących kroków:

#### Czytanie linii z pliku

```
val stream = CharStreams.fromString(polynomial);
```

Powyższa instrukcja odpowiada za podzielenie linii wczytanej z pliku jako łańcuch znaków do tablicy znaków.

#### Lexer

```
val lexer = new calculatorLexer(stream);
```

#### TOKENY

```
val tokens = new CommonTokenStream(lexer);
```

#### Parser

```
val parser = new calculatorParser(tokens);
```

#### Budowanie drzewa

```
val tree = parser.expression();
```

#### Ewaluacja wyrażeń

```
val result = new CalculatorVisitorImpl().visit(tree);
```

Visitor przechodzi po drzewie i zwraca wynik jako `PolynomialSum`.

#### Drukowanie wyniku

```
ResultParser.polynomialSumToString(result)
```

Powyższa metoda klasy `ResultParser` odpowiada za nadpisanie metody `toString()` na obiekcie typu `PolynomialSum`.

W tej metodzie zawarta jest również logika dodatkowa logika uwzględniająca poniższe przypadki:  $-x^1 \rightarrow x - x^0 \rightarrow 1 - 0 \cdot x \rightarrow 0 - x^{-1} \rightarrow x^{(-1)}$

## Gramatyka

```
grammar calculator;

expression
    : multiplyingExpression PLUS expression # Plus
    | multiplyingExpression MINUS expression # Minus
    | multiplyingExpression # toMultiplyingExpression
    ;

multiplyingExpression
    : powExpression TIMES multiplyingExpression # Times
    | powExpression DIV multiplyingExpression # Div
    | powExpression # toPowExpression
    ;

powExpression
    : signedAtom POW signedAtom # Pow
    | signedAtom # toSignedAtom
    ;

signedAtom
    : atom # PositiveAtom
    | MINUS atom # NegativeAtom
    ;

atom
    : FLOAT # Number
    | VARIABLE # Variable
    | LPAREN expression RPAREN # Parends
    ;

LPAREN : '(' ;
RPAREN : ')' ;
PLUS : '+' ;
MINUS : '-' ;
TIMES : '*' ;
DIV : '/' ;
COMMA : ',' ;
POINT : '.' ;
POW : '^' ;

VARIABLE
    : ('a' .. 'z') | ('A' .. 'Z')
    ;

FLOAT
    : ('0' .. '9') + ('.' ('0' .. '9') +)?
    ;

fragment SIGN
    : ('+' | '-')
    ;

WS
    : [ \r\n\t ] + -> skip
    ;
```

## Przykład działania