WOJSKOWA AKADEMIA TECHNICZNA IM. JAROSŁAWA DĄBROWSKIEGO W WARSZAWIE

***Sprawozdanie z przedmiotu „Języki formalne i kompilatory”***

**Temat zajęć:** *Analiza*

**Autorzy:** *Jakub Kapusta, Adam Kochański, Mikołaj Bednarek*

**Grupa:** *I6B2S1*

**Prowadzący*:*** *mgr inż. Krzysztof Mierzejewski*

**Opis stworzonej gramatyki bez kontekstowej (funkcje, operatory):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Funkcja** | **Wyjaśnienie** | **Operator** | **Przykład wywołania** |
| Add | Operacja dodawania (dwuargumentowa) | + | F1 + F2 |
| Sub | Operacja odejmowania (dwuargumentowa) | - | F1 - F2 |
| Mul | Operacja mnożenia (dwuargumentowa) | \* | F1 \* F2 |
| Div | Operacja dzielenia (dwuargumentowa) | : | F1 : F2 |
| Pow | Operacja potęgowania (dwuargumentowa, drugi z argumentów jest liczbą całkowitą) | ^ | F1 ^ I1 |
| Mod | Operacja modulo (dwuargumentowa) | % | F1 % F2 |
| Cong | Operacja kongruencji (dwuargumentowa) | cong | F1 cong F2 |
| Sqrt | Pierwiastek drugiego stopnia (operacja jednoargumentowa) | sqrt | sqrt(F1) |
| Neg | Operacja negacji (operacja jednoargumentowa zwracająca liczbę przeciwną do liczby wejściowej) | neg | neg(F1) |
| Abs | Wartość bezwzględna z liczby (operacja jednoargumentowa) | abs | abs(F1) |
| Floor | Zaokrąglenie w dół (operacja jednoargumentowa) | floor | floor(F1) |
| Ceil | Zaokrąglenie w górę (operacja jednoargumentowa) | ceil | ceil(F1) |
| Round | Zaokrąglenie (operacja jednoargumentowa) | round | round(F1) |
| Min | Operacja wyznaczenia najmniejszej liczby (operacja wieloargumentowa) | min | min(F1, F2, … Fn) |
| Max | Operacja wyznaczenia największej liczby (operacja wieloargumentowa) | max | max(F1, F2, … Fn) |

* F1, F2, … Fn - ułamki reprezentowane za pomocą klasy Fraction w postaci np. 3/4
* I1 – liczba całkowita

**Zapis gramatyki w formacie EBNF:**

Add= '+';

Sub= '-';

Mul= '\*';

Div= ':';

Pow= '^';

Mod= '%';

Cong= 'cong';

Sqrt= 'sqrt';

Neg= 'neg';

Abs= 'abs';

Floor= 'floor';

Ceil= 'ceil';

Round= 'round';

Min= 'min';

Max= 'max';

Digit\_excluding\_zero= "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";

Digit= "0" | Digit\_excluding\_zero

IntNum= '0' | (Digit\_excluding\_zero , {Digit});

FractionBar= '/';

Fraction= IntNum , FractionBar , ['+' | '-'] , IntNum;

Point= '.';

LeftBracket= '(';

RightBracket= ')';

Comma= ',';

fragment Space= ' ';

fragment Tab= '\t';

fragment NextLine= '\n';

fragment Return= '\r';

Blank= {Space | Tab | NextLine | Return}; (\*-> skip\*)

number= ['+' | '-'] , (IntNum | Fraction);

function= ['+' | '-'] , (Min | Max);

pow\_operator= Pow;

additive\_operator= (Add | Sub);

multiplicative\_operator= (Mul | Div | Mod | Cong);

function\_single\_value= ['+' | '-'] , (Sqrt | Neg | Abs | Floor | Ceil | Round );

function\_expression=

function ,

LeftBracket , expression , {Comma , expression} , RightBracket;

function\_single\_value\_expression=

function\_single\_value ,

LeftBracket , expression , RightBracket;

pow\_expression=

safe\_expression ,

pow\_operator ,

IntNum;

safe\_expression=

function\_expression |

function\_single\_value\_expression |

number |

expression\_in\_brackets;

secondary\_expression=

safe\_expression |

pow\_expression;

multiplicative\_expression=

secondary\_expression

{

multiplicative\_operator ,

secondary\_expression

};

expression=

{multiplicative\_expression}

{

additive\_operator,

multiplicative\_expression

};

expression\_in\_brackets=

(LeftBracket , expression , RightBracket);

**Opis implementacji rozwiązania w wybranej technologii:**

Projekt został wykonany w języku Python 3 w oparciu o generator i wtyczkę ANTLR4.

***frac.py***- skrypt zawierający funkcje obliczeniowe oraz klasę *Fraction* zawierającą m.in. metody operacji na ułamkach.

***eval.py*** - skrypt zawierający logikę ewaluacji wpisanych działań. Uruchomiony skrypt pozwala na ewaluację wyrażeń wpisanych do konsoli złożonych z działań podanych w opisie. Ułamki są skracane do postaci prostych.

***tree.py*** - Po uruchomieniu skrypt wypisuje drzewo parsowania dla wyrażeń złożonych z działań podanych w opisie.

Podczas wykonywania się skryptów mogą zostać wyrzucone błędy:

* “Blad skladni” - towarzyszący źle wprowadzonym wyrażeniom
  + “0 w mianowniku” - gdy wprowadzony ułamek będzie posiadał 0 w mianowniku
  + “Pierwiastek niewymierny” - gdy pierwiastek kwadratowy dla danego ułamka nie jest możliwy do zapisania w postaci ułamka zwykłego
  + Inne błędy związane z bazowymi błędami obliczania(np. dzielenie przez zero lub pierwiastek z liczby ujemnej)

**Przykładowe wywołania programu wraz z wypisanym w konsoli drzewem(obok forma graficzna z wtyczki ANTLR dla lepszej wizualizacji):**



