Московский авиационный институт

(национальный исследовательский институт)

Институт «Компьютерные науки и прикладная математика»

**Курсовая работа**

**«Разработка и реализация алгоритма представления арифметического выражения в ПОЛИЗ»**

**по курсу**

**«Системы программирования»**

**IV семестр**

*Студент:* Попов Н.А

*Группа:* М8О-308Б-21

*Руководитель:* Семенов А.С

*Оценка:*

*Дата:*

**Москва. 2023**

**Содержание**

[Требования к алгоритму (поставленная задача) 3](#_heading=h.5tsl531kls00)

[Теоретические сведения 3](#_heading=h.90bmcwcle4ji)

[Реализация алгоритма 5](#_heading=h.dqj2o5xqz2qh)

[Код программы 7](#_heading=h.n7zstc3w0gd7)

# 

# 

# 

# 

# 

# 

# 

# Требования к алгоритму (поставленная задача):

Разработка и реализация алгоритма представления арифметического выражения, содержащего тригонометрические функции, функцию модуль и операции плюс минус в ПОЛИЗ, для выражения | cos ( x ) + sin ( y ) - z |.

# Теоретические сведения

Постфиксная форма и указатель функции

Деревья могут использоваться для представления произвольных арифметических выражений. В таком дереве каждому листу соответствует операнд, а каждому родительскому узлу - операция. Если число операндов любой операции бу дет меньше или равно двум, то дерево будет бинарным. Если все операции будут иметь два операнда, то дерево окажется строго бинарным.

В компиляторах чаще всего используют постфиксную форму, в котором операции записываются справа от операндов. Постфиксная форма (польская инверсная запись – ПОЛИЗ) определяется следующими правилами:

Всякая переменная или константа есть выражение.

1. Если *θ*1 – знак унарной (одноместной) операции, а *α* – выражение, то *αθ*1 является выражением.
2. Если *θ*2 – знак бинарной (двухместной) операции, а *α*1 и *α*2 – выражения, то *α1 α2θ*2 является выражением.
3. Если *θn*– знак *n*-местной операции, а *α*1*,α*2, … , *αn*  являются выражениями, то *α*1*,α*2, … , *αn θn* – выражение.
4. Других выражений не существует

С точки зрения представления в польской инверсной записи указатель функции (оператор обращения к функции) мало чем отличается от переменной с индексом. Для представления указателя функции введем в ПОЛИЗ дополнительную операцию CALL, имеющую переменное число операндов, висящее от числа аргументов в вызове функции. Количество операндов операции CALL определяется выражением:

*k = n +* 1,

где *n* –число аргументов функции.

Т.е. наше выражение в ПОЛИЗ имеет вид: | cos x 1CALL sin y 1CALL + z -3CALL |

**Графическое представление выражений**

Деревья могут использоваться для представления произвольных арифметических выражений. В таком дереве каждому листу соответствует операнд, а каждому родительскому узлу - операция. Если число операндов любой операции будет меньше или равно двум, то дерево будет бинарным. Если все операции будут иметь два операнда, то дерево окажется строго бинарным.

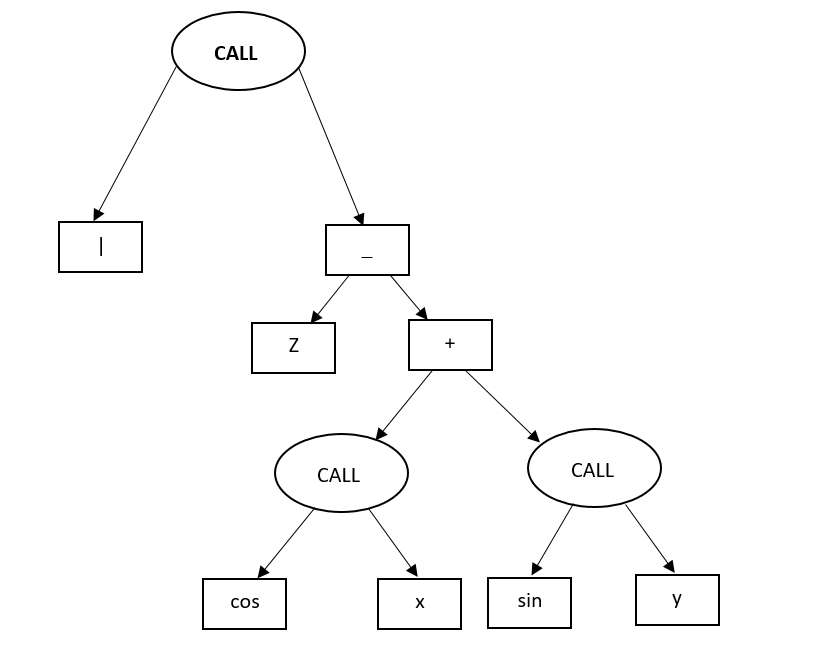


Рисунок 1 – Графическое представление выражений

# 

# 

# Реализация алгоритма

Таблица 1 – СУ-схема перевода

| **Правило грамматики G0** | **Элемент перевода G1** |
| --- | --- |
| E → |E| | E → |E N CALL| |
| E → G | E → T |
| G → C | S | T | P | Z | X | Y | G → C | S | T | P | Z | X | Y |
| P → E – G | P → E G – |
| T → E + G | T → E G + |
| C → cos(G) | C → cos G N CALL |
| S → sin(G) | S → sin G N CALL |
| Z → z | Z → z |
| N → +{1,2,3,4,5,6,7,8,9}\*{0,1,2,3,4,5,6,7,8,9} | N → +{1,2,3,4,5,6,7,8,9}\*{0,1,2,3,4,5,6,7,8,9} |
| X → x | X → x |
| Y → y | Y → y |
| G → ε | G → ε |

**Пример цепочки:**

Породим цепочку: |x- sin(y)| (в ПОЛИЗ: | x sin y 1 CALL - 2 CALL|)

Правило грамматики G0:

E →1|E|→2|G| →3|P| →4|E-G| →3|E-S| →2|G-S| →3|X-S| →10|x-S| →7|x-sin(G)| →3|x-sin(Y)| →11|x-sin(y)|

Правило грамматики G1:

E →1|E 2 CALL| →2|G 2 CALL| →3|P 2 CALL| →4|E G - 2 CALL| →3| E S - 2 CALL| →2|G S - 2 CALL| →3| X S- 2 CALL| →10| x S - 2 CALL|→7| x sin G 1 CALL - 2 CALL| →3| x sin Y 1 CALL - 2 CALL| →11| x sin y 1 CALL - 2 CALL|

**Комментарии к программе:**

На вход программы в качестве примера подается строка в качестве примера:

string expression = "| ( cos ( x ) + sin ( y ) - z ) |";

Скобки после модуля ставятся для обозначения, что модуль является функций от нескольких переменных.

На выводе программы должно в консоль выводится выражение представленное в виде ПОЛИЗ: | cos x 1CALL sin y 1CALL + z -3CALL |

Вывод в консоль приведен ниже после кода программы.

# 

# 

# 

# 

# 

# Код программы

using System;

using System.Collections.Generic;

class Program

{

static void Main(string[] args)

{

string expression = "| ( cos ( x ) + sin ( y ) - z ) |"; //| cos x 1CALL sin y 1CALL + z -3CALL |

string[] substring = expression.Split(' ');

string[] binaryOperatorsArray = { "+", "-" };

string[] multidimensionalOperationArray = { "cos", "sin","|"};

Stack<string> stack = new Stack<string>();

string result = "";

int counter = 0;

for (int i = 0; i < substring.Length; i++)

{

string elem = substring[i];

if (elem == "(")

{

stack.Push(elem);

}

else if (elem == ")")

{

while (stack.Peek() != "(")

{

counter += 1;

result += stack.Pop();

}

counter += 1;

result += counter.ToString();

result += "CALL";

result += " ";

stack.Pop();

}

else if (Array.IndexOf(multidimensionalOperationArray,elem) != -1)

{

result += elem;

result += " ";

counter = 0;

}

else if (Array.IndexOf(binaryOperatorsArray, elem) != -1)

{

while (stack.Count > 0 && Array.IndexOf(binaryOperatorsArray, stack.Peek()) != -1)

{

result += stack.Pop();

result += " ";

}

stack.Push(elem);

}

else

{

result += elem;

result += " ";

}

}

while (stack.Count > 0)

{

result += stack.Pop();

result += " ";

}

Console.WriteLine(result); }}

Рисунок 2 – Вывод программы