Лабораторная работа № 5

Вариант 13

Задание №1

Постановка задачи:

В строке записано арифметическое выражение. Построить для него бинарное дерево выражения. Вывести полученное дерево на экран.

(Инфиксная форма записи арифметического выражения)

Код:

class Tree: # Класс дерево

pass

def Struk(k): # Структура

root = Tree()

root.data = k

root.left = None

root.right = None

return root

def Pr(op): # Приоритет операции

P = 4

if op in '\*/' : P = 2

if op in '+-' : P = 1

return P

def PostPr(s): # Последняя позиция

min\_P = 3

z = -1

for i in range(len(s)):

P = Pr(s[i])

if P <= min\_P:

min\_P = P

z = i

return z

def del\_Tree(s): # Делаем дерево

Z = PostPr(s)

if Z < 0:

tree = Struk(s)

else:

tree = Struk(s[Z])

if s[Z] == '-' and s[:Z] == '':

tree.left = del\_Tree('0')

else:

tree.left = del\_Tree(s[:Z])

tree.right = del\_Tree(s[Z + 1:])

return tree

def PoZ(Tree, a = 0): # Поворот на 90 градусов и вывод на экран

if Tree:

PoZ(Tree.right, a + 1)

print(' '\*a, end='')

print(Tree.data)

PoZ(Tree.left, a + 1)

Fin = open('input.txt', "r")

Fout = open('output.txt', "a+")

s = Fin.readline()

if not s:

print('Файл пустой')

else:

print(s)

Tree = del\_Tree(s) # Создание дерева

PoZ(Tree) # Вывод дерева

Fin.close()

Fout.close()

Тесты:

| № | Исходные данные | Ожидаемый результат | Вывод программы |
| --- | --- | --- | --- |
| 1 | 5+15+20/10+1/2 | 2  /  1  +  10  /  20  +  15  +  5 | 2  /  1  +  10  /  20  +  15  +  5 |
| 2 |  |  |  |
| 3 |  |  |  |

Задание №2

Постановка задачи:

Используя результаты п.1, получить запись арифметического выражения в префиксной форме. Вывести его на экран и вычислить, используя функцию из лабораторной работы 4. Результат вывести на экран.

Код:

# Префиксная форма(КЛП)

def PPREF(Tree):

global S

if Tree:

S += Tree.data + ' '

PPREF(Tree.left)

PPREF(Tree.right)

def VV(stack, a, b): # Вычисление

v = stack.pop()

if v == '+':

Z = a + b

elif v == '-':

Z = a - b

elif v == '\*':

Z = a \* b

else:

Z = a // b

stack.append(Z)

return stack

def Pred(stack): # Проверка на значение предыдущего знака. Рекурсия.

i = str(stack.pop())

a = str(stack.pop())

if a not in '+-\*/': # если предыдущий не знак

stack = VV(stack, int(a), int(i))

if len(stack) >= 3:

Pred(stack)

if a in '+-\*/':

stack.append(a)

stack.append(i)

return stack

def PREF(s): # Ориентирование по стеку

stack = []

for i in s:

if i in '+-\*/':

stack.append(i)

elif i not in '+-\*/':

a = str(stack.pop())

stack.append(a)

stack.append(i)

stack = Pred(stack)

return stack

...

S = ''

PPREF(Tree)

print('Префиксная форма: ', S)

print('Выражение равно: ', PREF(S.split()))

...

Тесты:

| № | Исходные данные | Ожидаемый результат | Вывод программы |
| --- | --- | --- | --- |
| 1 | 7 / 1 + 4 - 1 + 5 | Префиксная форма: + - + / 7 1 4 1 5  Выражение равно: [15] | Префиксная форма: + - + / 7 1 4 1 5  Выражение равно: [15] |
| 2 |  |  |  |
| 3 |  |  |  |

Задание №3

Постановка задачи:

Используя результаты п.1, получить запись арифметического выражения в постфиксной форме. Вывести его на экран и вычислить, используя функцию из лабораторной работы 4. Результат вывести на экран.

Код:

# Постфиксная форма(ЛПК)

def PPOST(Tree):

global SS

if Tree:

PPOST(Tree.left)

PPOST(Tree.right)

SS += Tree.data + ' '

def POST(s):

stack = []

for i in s:

if i in '+-\*/':

b = int(stack.pop())

a = int(stack.pop())

if i == '+':

Z = a + b

elif i == '-':

Z = a - b

elif i == '\*':

Z = a \* b

else:

Z = a // b

stack.append(Z)

else:

stack.append(i)

return stack

...

SS = ''

PPOST(Tree)

print('Постфиксная форма: ', SS)

print('Выражение равно: ', POST(SS.split()))

...

Тесты:

| № | Исходные данные | Ожидаемый результат | Вывод программы |
| --- | --- | --- | --- |
| 1 | 7 / 1 + 4 - 1 + 5 | Постфиксная форма: 7 1 / 4 + 1 - 5 +  Выражение равно: [15] | Постфиксная форма: 7 1 / 4 + 1 - 5 +  Выражение равно: [15] |
| 2 |  |  |  |
| 3 |  |  |  |

Задание №4

Постановка задачи:

Используя результаты п.1, вычислить значение арифметического выражения при помощи бинарного дерева. Результат вывести на экран.

Код:

# Вычисление значения по бинарному дереву

def VBin(Tree):

if not Tree.left:

return int(Tree.data)

else:

a = VBin(Tree.left)

b = VBin(Tree.right)

return int(Vich(a, b, Tree.data))

...

print('Выражение вычисленного по бинарному дереву равно: ', VBin(Tree))

...

Тесты:

| № | Исходные данные | Ожидаемый результат | Вывод программы |
| --- | --- | --- | --- |
| 1 | 7 / 1 + 4 - 1 + 5 | Выражение вычисленного по бинарному дереву равно: 15 | Выражение вычисленного по бинарному дереву равно: 15 |
| 2 |  |  |  |
| 3 |  |  |  |