Федеральное государственное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ИНТЕГРАЦИИ IT-РЕШЕНИЙ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Доцент кафедры САИТ Студент группы КТбо1-2

Лапшин Вячеслав Сергеев Хиль Дмитрий Сергеевич

«9» марта 2023 г. «08» марта 2023 г.

**ЛАБОРАТОРНАЯ РАБОТА №2**

По дисциплине

«Основы алгоритмизации и

программирования»

Вариант 6

На тему

«Стек»

**Цель**: Целью данной лабораторной работы является выработка практических навыков работы со стеком на примере построения обратной польской записи арифметических и логических выражений.

**Задачи**: написать программу, реализующую перевод в ОПЗ простых арифметических и логических выражений. Ввод и вывод данных организовать во внешние файлы. Запрограммировать работу стека на основе двух функций push () и pop (), программно реализовать алгоритм Дейкстры. Программу протестировать на тестовом наборе арифметических выражений. Кроме того, нужно построить десять тестовых логических выражений и продемонстрировать на них работу программы.

**Составление математической модели:**

Вход: 9–5/ (5 + 9)

Читаем «9»

Добавим «9» к выходной строке

Выход: 9

Читаем «-»

Кладём «-» в стек

Выход: 9

Стек: -

Читаем «5»

Добавим «5» к выходной строке

Выход: 9 5

Стек: -

Читаем «/»

Кладём «/» в стек

Выход: 9 5

Стек: - /

Читаем «(»

Кладём «(» в стек

Выход: 9 5

Стек: - /

Читаем «5»

Добавим «5» к выходной строке

Выход: 9 5 5

Стек: - /

Читаем «+»

Кладём «+» в стек

Выход: 9 5 5

Стек: - / +

Читаем «9»

Добавим «9» к выходной строке

Выход: 9 5 5 9

Стек: - / +

Читаем «)»

Выталкиваем «+» из стека в выходную строку, выталкиваем «(»

Выход: 9 5 5 9 +

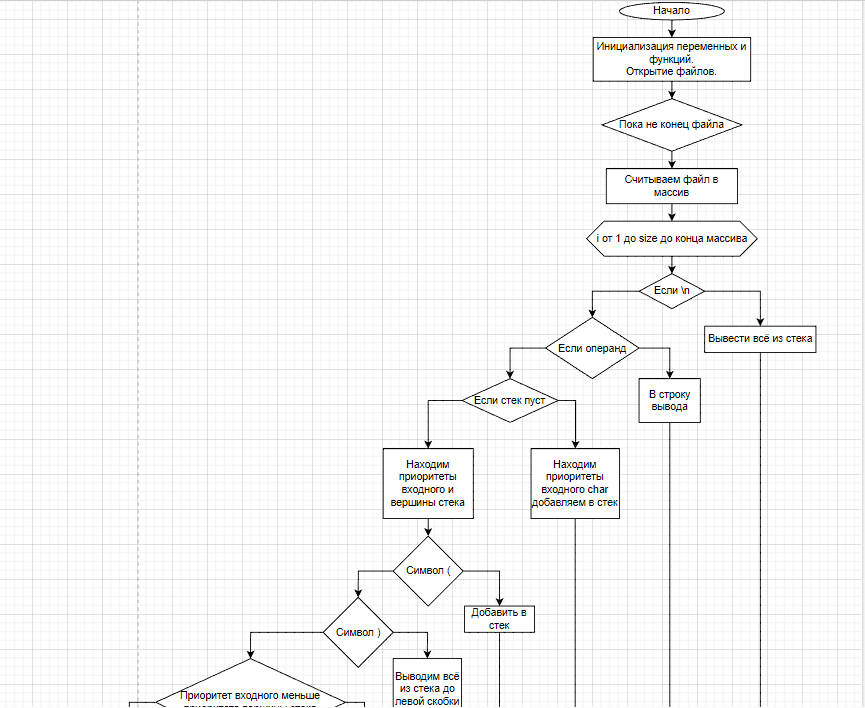
Стек: - /

Конец выражения

Выталкиваем все элементы из стека в строку

Выход: 9 5 5 9+/-

**Ход работы**:



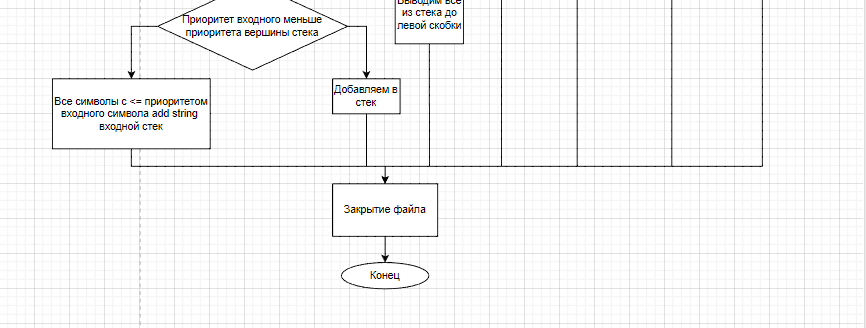


Рисунок 1. – Главная процедура алгоритма ОПС.

**Реализация программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

char stack[1000];

char str[1000];

int top = 0;

void push(char i) {

if (top >= 1000)

printf("Вне стека\n");

else

stack[top++] = i;

}

char pop() {

top -= 1;

if (top < 0) {

printf("Вне стека\n");

}

return stack[top];

}

int Prior(char b) {

int ans = 0;

switch (b)

{

case '(': {ans = 0; break; }

case ')': {ans = 1; break; }

case '+':

case '-':

case '|':

{ans = 2; break; }

case '\*':

case'/':

case'&': {ans = 3; break; }

case'^': {ans = 4; break; }

default:

break;

}

return ans;

}

void OPZ(FILE\* in, FILE\* out, int& temp1, int& temp2, char& temp3)

{

while (!feof(in))

{

fgets(str, 999, in);

int size = strlen(str);

for (int i = 0; i < size; ++i)

{

if (str[i] == '\n')

{

while (top > 0)

{

fprintf(out, "%c", pop());

}

fprintf(out, "\n");

}

else if (str[i] != '(' && str[i] != ')' && str[i] != '+'

&& str[i] != '-' && str[i] != '\*' && str[i] != '/'

&& str[i] != '|' && str[i] != '&' && str[i] != '^')

fprintf(out, "%c", str[i]);

else

{

if (top == 0)

{

push(str[i]);

}

else

{

temp1 = Prior(str[top - 1]);

temp2 = Prior(str[i]);

if (str[i] == '(')

{

push(str[i]);

}

else if (str[i] == ')')

{

while ((temp3 = pop()) != '(')

fprintf(out, "%c", temp3);

}

else {

if (temp1 > temp2)

push(str[i]);

else {

while (temp2 <= Prior(stack[top - 1]))

fprintf(out, "%c", pop());

push(str[i]);

}

}

}

}

}

}

}

int main()

{

setlocale(LC\_ALL, "Russian");

FILE\* in = fopen("input.txt", "r");

FILE\* out = fopen("output.txt", "w");

int temp1, temp2;

char temp3;

if (in != NULL)

{

printf("Файл открыт успешно\n");

}

else printf("Файл не удалось открыть!\n");

if (out != NULL)

{

printf("Файл открыт успешно\n");

}

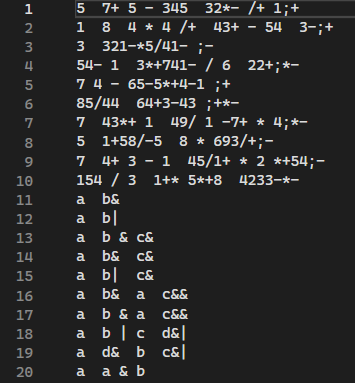
else printf("Файл не удалось открыть!\n");

OPZ(in, out, temp1, temp2, temp3);

}

**Результат работы программы.**

**Изображение выглядит как текст

Автоматически созданное описание**

**Вывод:** выработал практических навыков работы со стеком. Построил обратную польскую запись арифметических и логических выражений.