Федеральное государственное автономное образовательное учреждение

высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КАФЕДРА СИСТЕМНОГО АНАЛИЗА И ТЕЛЕКОММУНИКАЦИИ

**ПРОВЕРИЛ ВЫПОЛНИЛ**

Старший преподаватель Студент группы КТбо1-2

Лапшин В.С. Макаричев СВ

«9» марта\_2023 г. «9» \_ марта \_\_2023 г.

**ЛАБОРАТОРНАЯ РАБОТА №2**

По дисциплине

«Основы алгоритмизации и программирования»

Вариант 4

На тему

«Стек»

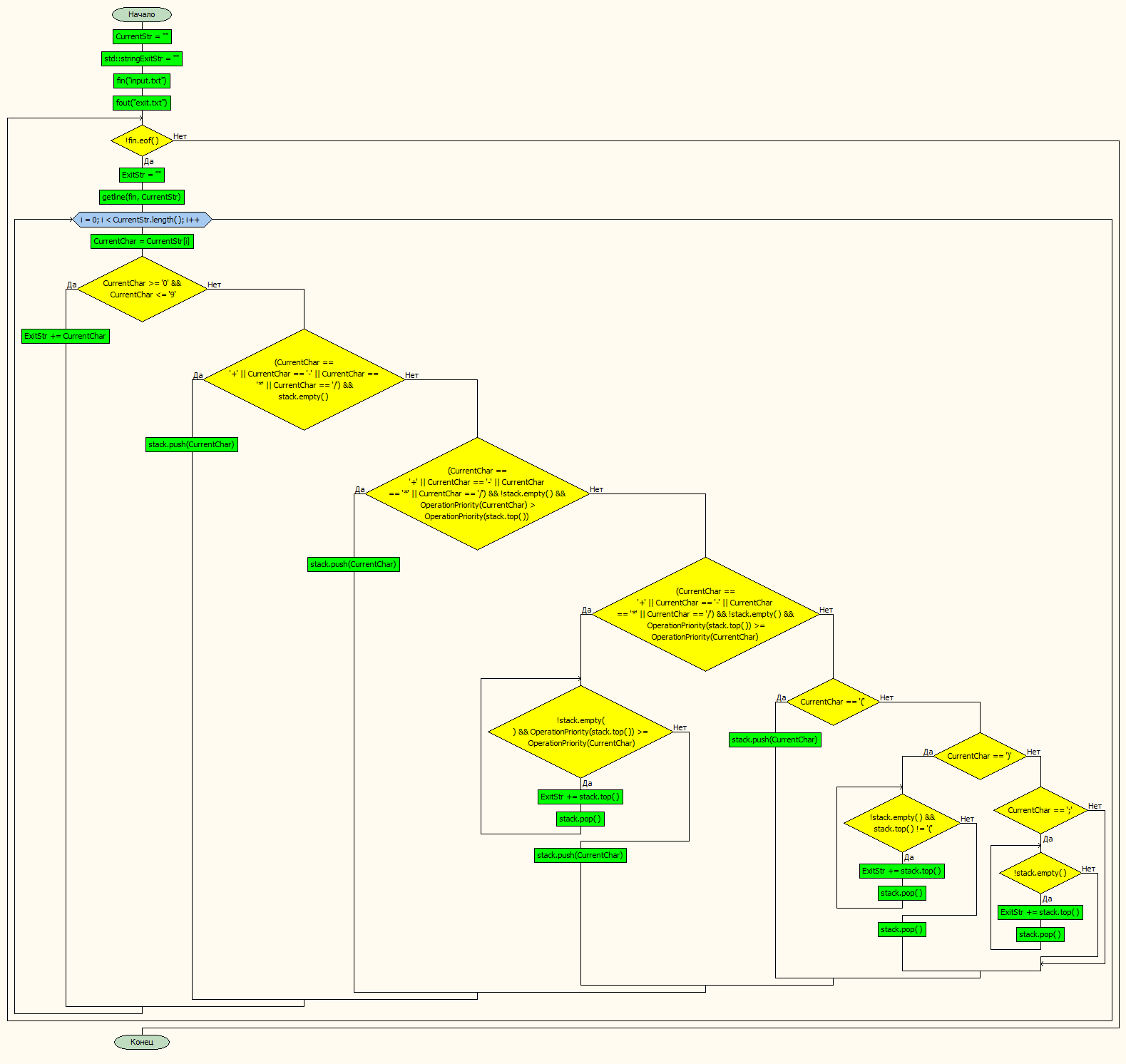
**Таганрог 2023**

**Цель**: Целью данной лабораторной работы является выработка у студента 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.

**Составление математической модели**:

1. Читать очередной символ входной строки, пока не будет достигнут конец выражения: 1.1. Если символ операнд, то он перемещается в выходную строку. 1.2. Если символ операция и стек пуст, то он проталкивается в стек. 1.3. Если символ операция и стек не пуст, а приоритет символа операции меньше, чем у символа операции вершины стека, то прочитанный символ проталкивается и стек. 1.4. Если символ операция и стек не пуст, а приоритет символа вершины стека, меньше приоритета входного символа, то из стека в выходную строку выталкиваются все символы операций с меньшим или равным приоритету входного символа, после чего входной символ операции заносится в стек. 1.5. Если входной символ левая открывающая скобка, то она проталкивается в стек. 1.6. Если входной символ правая закрывающая скобка, то она выталкивает из стека в выходную строку все символы до левой открывающей скобки. сами скобки уничтожаются и в выходную строку не попадают. 15

 *(Блок-схема программы.)*

**Реализация .**

#include <iostream>

#include <fstream>

#include <stack>

#include <string>

int OperationPriority(char a)

{

if (a=='(')

{

return 0;

}

if (a == ')')

{

return 1;

}

if (a == '+' || a == '-')

{

return 2;

}

if (a == '\*' || a == '/')

{

return 3;

}

if (a == '^')

{

return 4;

}

return 0;

}

int main()

{

char CurrentChar;

std::string CurrentStr = "";

std::string ExitStr="";

std::stack <char> stack;

std::ifstream fin("input.txt");

std::ofstream fout("exit.txt");

while (!fin.eof())

{

ExitStr = "";

getline(fin, CurrentStr);

for (int i = 0; i < CurrentStr.length(); i++)

{

CurrentChar = CurrentStr[i];

if (CurrentChar >= '0' && CurrentChar <= '9')

{

ExitStr += CurrentChar;

}

else if ((CurrentChar == '+' || CurrentChar == '-' || CurrentChar == '\*' || CurrentChar == '/') && stack.empty())

{

stack.push(CurrentChar);

}

else if ((CurrentChar == '+' || CurrentChar == '-' || CurrentChar == '\*' || CurrentChar == '/') && !stack.empty() && OperationPriority(CurrentChar) > OperationPriority(stack.top()))

{

stack.push(CurrentChar);

}

else if ((CurrentChar == '+' || CurrentChar == '-' || CurrentChar == '\*' || CurrentChar == '/') && !stack.empty() && OperationPriority(stack.top()) >= OperationPriority(CurrentChar))

{

while (!stack.empty() && OperationPriority(stack.top()) >= OperationPriority(CurrentChar))

{

ExitStr += stack.top();

stack.pop();

}

stack.push(CurrentChar);

}

else if (CurrentChar == '(')

{

stack.push(CurrentChar);

}

else if (CurrentChar == ')')

{

while (!stack.empty() && stack.top() != '(')

{

ExitStr += stack.top();

stack.pop();

}

stack.pop();

}

else if (CurrentChar == ';')

{

while (!stack.empty())

{

ExitStr += stack.top();

stack.pop();

}

}

}

fout << ExitStr <<std::endl;

}

}

**Результат работы программы.**

**При таких вводных данных**

**Изображение выглядит как текст

Автоматически созданное описание**

**Имеем такой ответ**

**Изображение выглядит как текст

Автоматически созданное описание**

**Вывод:** Рабочий код программы, реализующий перевод в ОПЗ простых арифметических и логических выражений.