Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**СТЕКИ И ОЧЕРЕДИ**

**Отчет по лабораторной работе №2**

**По дисциплине**

**«Структуры и алгоритмы обработки данных в ЭВМ»**

Студент гр. 431-3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Гурулёв

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Проверил: профессор кафедры АСУ, д.т.н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.Н. Горитов

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Томск 2022

# Задание на лабораторную работу

Разработать программу, которая выполняет арифметические операции, операторы вводятся в постфиксной форме, для реализации использовать стек.

Для реализации АТД Стек использовать указатели.

# Алгоритм решения задачи

1. Открываем файл;
2. Считываем первое число, заносим его в стек;
3. Организуем цикл с условием: “Пока возможно считывание числа и символа”. Тело цикла:
4. Заносим считанное число в стек;
5. Находим операцию, которая была считана;
6. Выполняем данную операцию, результат заносим в стек;
7. Конец тела цикла;
8. Забираем из стека результат и выводим его.

# Листинг программы

**Для main.cpp:**

#include <iostream>

#include <locale.h>

#include <fstream>

#include "stackLogic.cpp"

using namespace std;

using namespace myTaskLogic;

struct Record

{

int num;

char operation;

};

int main()

{

setlocale(LC\_ALL, "rus");

MyStack stack;

Record rec = {-1, '?'};

ifstream f("File.txt");

if (f.is\_open())

{

f >> rec.num;

stack.Push(rec.num);

rec.num = -1;

while (f >> rec.num >> rec.operation)

{

if (rec.num != -1)

{

stack.Push(rec.num);

rec.num = -1;

}

switch (rec.operation)

{

case '+':

stack.Push(stack.Pop() + stack.Pop());

break;

case '-':

{

stack.Push((-stack.Pop()) + stack.Pop());

break;

}

case '\*':

stack.Push(stack.Pop() \* stack.Pop());

break;

case '/':

{

int a = stack.Pop();

int b = stack.Pop();

stack.Push(b / a);

break;

}

default:

break;

}

}

cout << (int)stack.Pop() << endl;

}

else

{

cout << "error" << endl;

}

system("pause");

return 0;

}

**Для stackLogic.cpp:**

#include <iostream>

using namespace std;

namespace myTaskLogic

{

class MyStack

{

public:

//Поместить элемент в стек

void Push(int num)

{

Node\* NewNode = new Node();

if (NewNode)

{

(\*NewNode).next = top;

(\*NewNode).data = num;

top = NewNode;

}

else

{

cout << "Ошибка внесения элемента";

exit(EXIT\_FAILURE);

}

}

//Изъять элемент из стека

int Pop()

{

if (top)

{

int out = (\*top).data;

Node\* newTop = (\*top).next;

delete[] top;

top = newTop;

return out;

}

else

{

cout << "Ошибка взятия элемента";

exit(EXIT\_FAILURE);

}

}

//Получение верхнего значения в стеке

int Top()const

{

return (\*top).data;

}

//Является ли стек пустым

bool IsEmpty()const

{

if (top)

{

return false;

}

else

{

return true;

}

}

private:

class Node

{

public:

int data;

Node\* next;

Node()

{

data = 0;

next = NULL;

}

};

Node\* top = NULL;

};

}

# Пример решения

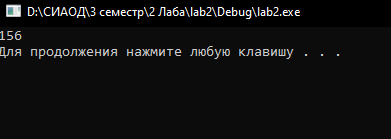


Рисунок 4.1 - Результат выполнения программы

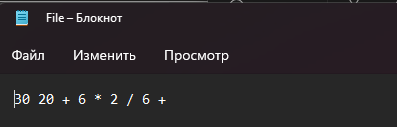


Рисунок 4.2 - Входные данные

# Вывод

Я изучил принципы работы со структурами данных “стек”.