

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: Стек. Очередь. Дек

Студент гр. 7383

Левкович Д.В.

Преподаватель

Размочева Н.В.

Санкт-Петербург

2018

ОГЛАВЛЕНИЕ

Цель работы.....	3
Реализация задачи.....	3
Тестирование.....	4
Вывод.....	5
Приложение А. Тестовые случаи.....	6
Приложение Б. Исходный код программы.....	6

1. ЦЕЛЬ РАБОТЫ

Цель работы: познакомиться со структурой и реализацией стека и использованием их в практических задачах на языке программирования C++.

Формулировка задачи: вычислить как целое число значение выражения (без переменных), записанного в постфиксной форме. Вариант 11-а-в.

2. РЕАЛИЗАЦИЯ ЗАДАЧИ

В функции `main` было реализовано меню для пользователя, где можно выбрать способ ввода входных данных. Данные можно ввести либо из файла, либо из терминала.

Функция `computation` принимает на вход введенную строку. Пока строка не закончится, из нее считываются символы. Если мы встретили число, то мы добавляем его в стек. Далее, если мы встречаем `fhbavtnbxtcrbq` оператор, то мы извлекаем последние два числа из стека, производим необходимые вычисления и добавляем в стек результат данного вычисления.

Шаблонный класс `Stack` с шаблонным параметром `T` (тип хранимых элементов) представляет из себя стек на базе динамического массива. Класс содержит в себе следующие поля:

- `T * arr.`
- `int size` – базовый размер массива.
- `int len` – текущая длина массива.

Методы класса `Stack`:

- Конструктор, устанавливающий первоначальные значения полей.
- `Push` – принимает объект типа `T`, который добавляется в стек, при этом увеличивается индекс массива на единицу и проверяется, достигли ли мы конца массива, если да, то вызывается функция `new_size`, которая создает массив нового размера и перезаписывает данные.
- `Pop` – возвращает нам последний элемент типа `T`.

- Clear – очищает весь стек.
- IsEmpty – проверяет стек на наличие в нем элементов.
-

3. ТЕСТИРОВАНИЕ

Программа была собрана в компиляторе g++ с ключом -std=c++14 в OS Linux Ubuntu 16.04 LTS.

В ходе тестирования ошибки не были найдены.

Тестовые случаи представлены в приложении А.

4. ВЫВОД

В ходе работы были получены навыки работы со стеком. Был реализован шаблонный класс, представляющий из себя стек на базе массива. Стек является удобной структурой данных для вычисления выражения, записанного в постфиксной форме.

ПРИЛОЖЕНИЕ А. ТЕСТОВЫЕ СЛУЧАИ

Таблица 2 — Корректные тестовые случаи

Входные данные	Результат
$162 \cdot 2 \cdot 1 + 4^{\wedge} /$	2
$6 \cdot 3 \cdot 2^{\wedge} - 11 -$	-14
2	2
$6 \cdot 3 - 2^{\wedge} 11 -$	-2
-2	Error

ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД ПРОГРАММЫ

```
//вариант 11-а-в
#include <iostream>
#include <sstream>
#include <cstdlib>
#include <fstream>
#include <cstring>
#include <cmath>
#include <exception>

using namespace std;
using base = int;

template <class T>
class Stack
{
private:
    T * arr;
    int size;
```

```

int len;
void new_size(int s)
{
    if (s<=0) return;
    T * new_arr = new T[s];
    int min_s = s<len?s:len;
    for (int i = 0; i<min_s; i++) new_arr[i] = arr[i];
    delete[] arr;
    size = s;
    arr = new_arr;
}

public:
    Stack();
    Stack(Stack &st);
    ~Stack();
    int Size();
    void Push(T x);
    T Pop();
    T Top();
    void Clear();
    bool IsEmpty();
};

template <class T>
Stack<T>::Stack()
{
    len = 0;
    arr = new T[size = 5];
}

template <class T>
Stack<T>::Stack(Stack & st)
{
    arr = new T[size = st.size];
    len = st.len;
    for (int i = 0; i < len; i++) arr[i] = st.arr[i];
}

template <class T>
Stack<T>::~~Stack()
{
    delete[] arr;
}

template <class T>
int Stack<T>::Size()
{
    return len;
}

template <class T>
void Stack<T>:: Push(T x)
{
    if (len == size) new_size(size<<1);
    arr[len++] = x;
    cout<<"Push -> "<<x<<endl;
}

template <class T>
T Stack<T>::Pop()
{

```

```

        if (size > 10 && len < size>>2) new_size(size>>1);
        if (len == 0) return arr[len];
        return arr[--len];
    }
    template <class T>
    T Stack<T>::Top()
    {
        return arr[len];
    }
    template <class T>
    void Stack<T>::Clear()
    {
        delete arr;
        len = 0;
        arr = new T[size = 5];
    }

    template <class T>
    bool Stack<T>::IsEmpty()
    {
        return len == 0;
    }
    void computation(string a){
        int x = 0;
        Stack<int> s;
        int n=a.length();
        for (int i = 0; i < n; i++){
            cout << "шаг: " << i+1 << " СИМВОЛ = " << a[i] << endl; //
            if (a[i] == '+')
                s.Push(s.Pop() + s.Pop());
            if (a[i] == '-') {
                x=s.Pop();
                s.Push(s.Pop()-x);
            }
            if (a[i] == '*')
                s.Push(s.Pop() * s.Pop());
            if (a[i] == '/') {
                x= s.Pop();
                s.Push(s.Pop()/x);
            }
            if (a[i] == '^')
                s.Push(pow(s.Pop(), s.Pop()));
            if ((a[i] >= '0') && (a[i] <= '9'))
                s.Push(0);
            while ((a[i] >= '0') && (a[i] <= '9')){
                s.Push(10*s.Pop() + (a[i++] - '0'));
                cout << "шаг_: " << i+1 << " СИМВОЛ = " << a[i] << endl;
            }
        }
        int res = 0;
        res = s.Pop();
        if(!s.IsEmpty()){
            throw invalid_argument("Error\n");
        }
        else
            cout << "Результат = " << res << endl;
        s.Clear();
    }
    string Space(string str0){
        int i = 0, k = 0;

```

dem

```

        char str[100];
        string str1;
        while(str0[i]!='\0'){
            if((str0[i]=='^' || str0[i]=='/' || str0[i]=='*' || str0[i]=='-' ||
str0[i+1]=='+')){
                str[k]=' ';
                str[k+1]=str0[i];
                str[k+2]=' ';
                k+=3;
            }
            else{
                if(str0[i]=='\n') break;
                str[k]=str0[i];
                k++;
            }
            i++;
        }
        str[k]='\0';
        str1=str;
        return str1;
    }
    int main ()
    {
        string a;
        int run = 1;
        string k;
        int m;
        while(run){
            cout<<"Введите 1, если хотите ввести выражение из консоли,
введите 2, если хотите ввести выражение из файла, 3 - выход из
программы."<<endl;
            getline(cin, k);
            try{
                m=stoi(k);
            }
            catch(exception& e){
                cout<<"неверный символ"<<endl;
            }
            try{
                switch(m){
                    case 1:{
                        cout << "введите выражение:" << endl;
                        getline(cin, a);
                        a=Space(a);
                        computation(a);
                        break;
                    }
                    case 2:{
                        ifstream infile("postfix.txt");
                        if(!infile){
                            cout<<"File not open for reading"<<endl;
                            break;
                        }
                        getline(infile, a);
                        a=Space(a);
                        computation(a);
                        break;
                    }
                    case 3:
                        cout<<"End!"<<endl;
                        return 0;
                }
            }
        }
    }

```



```
        default:
            break;
    }
}
catch(exception& e){
    cout << e.what() << endl;
}
}
return 0;
}
```