

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

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Алгоритмы и структуры данных»
Тема: стек и очередь

Студент гр. 7383

_____ Власов Р.А.

Преподаватель

_____ Размочаева Н.В.

Санкт-Петербург

2018

Содержание

1. Цель работы	3
2. Реализация задачи	4
3. Тестирование	5
3.1 Процесс тестирования	5
3.2 Результаты тестирования	5
4. Вывод	6
Приложение А: Исходный код	7

Цель работы

Цель работы: познакомиться со стеком и очередью, создать реализацию стека на базе массива на языке программирования C++.

Формулировка задачи: Вариант 4-в. Содержимое заданного текстового файла F , разделенного на строки, переписать в текстовый файл G , выписывая литеры каждой строки в обратном порядке.

Реализация задачи

Для реализации стека на базе массива было принято создать класс с базовыми функциями стека. Структура класса приведена ниже.

```
class Stack {  
private:  
    int st_p;  
    char *st;  
public:  
    Stack(string str);  
    void push(char ch);  
    char pop();  
    bool isEmpty();  
    ~Stack();  
};
```

Метод `void push(char ch)` добавляет в стек один элемент, а метод `char pop()` возвращает последний элемент и убирает его из стека, метод `bool isEmpty()` возвращает `true`, если стек пуст. Конструктор класса инициализирует стек строкой.

Программа получает на вход названия файлов для ввода и для вывода, после чего поочередно добавляет строки из первого файла в стек, а после записывает их из стека во второй файл.

Исходный код программы представлен в приложении А.

Тестирование

1. Процесс тестирования

Программа собрана в операционной системе Ubuntu 18.04.1 LTS bionic компилятором g++ (Ubuntu 7.3.0-16ubuntu3) 7.3.0. В других ОС и компиляторах тестирование не проводилось.

2. Результаты тестирования

По результатам тестирования было выявлено, что программа записывает не все строки, если файл ввода и вывода совпадает. В программу была добавлена проверка на совпадение названий файлов.

Вывод

В ходе выполнения данной работы были изучены стек и очередь. Был создан класс стека и написана программа, записывающая в обратном порядке литеры строк из одного файла в другой.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cctype>
#include <string>

using namespace std;

class Stack {
private:
    int st_p;
    char *st;
public:
    Stack(string str);
    void push(char ch);
    char pop();
    bool isEmpty();
    ~Stack();
};

Stack::Stack(string str)
{
    if (str.size() == 0)
        return;
    st = new char[str.size() + 1];
    st_p = 0;
    st[st_p] = '\\n';
    st_p++;
    for (auto ch : str)
    {
        st[st_p] = ch;
        st_p++;
    }
}

void Stack::push(char ch)
{
    st[st_p] = ch;
    st_p++;
}

char Stack::pop()
```

```

{
    st_p--;
    return st[st_p];
}

bool Stack::isEmpty()
{
    return (st_p > 0 ? false : true);
}

bool write(string str_o, Stack &st)
{
    fstream f(str_o, ios_base::out | ios_base::app);
    if (!f)
    {
        cout << "Unable to open the output file!" << endl;
        return false;
    }
    while (!st.isEmpty())
        f.put(st.pop());
    f.close();
    return true;
}

Stack::~~Stack()
{
    if (st)
        delete[] st;
}

int main()
{
    int n, c;
    int *el;
    string str_i, str_o;
    while(true)
    {
        cout << "Press 1 to reverse letters in strings in a file\n" <<
            "Press 2 to exit." << endl;
        cin >> str_i;
        if (!isdigit(str_i[0]))
            continue;
        c = stoi(str_i);
    }
}

```



```

str_i.clear();
switch (c)
{
    case 1:
        break;
    case 2:
        return 0;
    default:
        cout << "Something went wrong. try again!" << endl;
        continue;
}
if(c == 1)
{
    cout << "Enter input file name: ";
    cin >> str_i;
    cout << "Enter output file name: ";
    cin >> str_o;
    if (str_i == str_o)
    {
        cout << "Input and output files can't be the same." <<
endl;
        continue;
    }
    ifstream f;
    f.open(str_i);
    if (!f)
    {
        cout << "Unable to open the input file!" << endl;
        continue;
    }
    while(!f.eof())
    {
        getline(f, str_i);
        Stack el(str_i);
        if (write(str_o, el) == false);
            break;
    }
    f.close();
}
}
}

```