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, очередь, дек

Студентка гр. 7383 _____ Ханова Ю.А.

Преподаватель _____ Размочаева Н.В

Санкт-Петербург
2018

Содержание

Цель работы

Познакомиться с понятием стека, дека и очереди, получить навыки реализации стека, дека и очереди на языке программирования C++.

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

Реализация задачи

В данной лабораторной работе был реализован класс Stack, содержащий структуру Stack и набор функций для работы со стеком:

base Stack::top (void); - возвращает элемент, который последним был помещен в стек;

base Stack::pop2(void); - удаляет последний добавленный в стек элемент и возвращает его значение;

void Stack::pop(void); - удаляет последний добавленный в стек элемент;

void Stack::push (const base &x); - добавляет элемент x в стек;

bool Stack::isNull(void); - проверяет стек на пустоту;

void Stack:: destroy(void); - удаляет все элементы в стеке;

Так же были разработаны функции для преобразования данных в соответствии с заданием:

void read(Stack &s,std::ifstream &in, std::ofstream &out); - считывает из входного файла символы и записывает их в стек;

void write(Stack &s, std::ofstream &out); - пока стек не опустеет вытаскивает из него элементы и записывает в выходной файл;

Тестирование

Программа собрана в операционной системе Ubuntu 17.04 с использованием компилятора g++. В других ОС и компиляторах тестирование не проводилось. Результаты тестирования показали, что поставленная цель выполнена. Результаты тестирования представлены в Приложении Б.

Выводы

В ходе выполнения данной лабораторной работы были освоены основные принципы работы со стеком на языке программирования C++. Также была написана программа для выполнения поставленной задачи.

ПРИЛОЖЕНИЕ А. Код программы

STACK.H

```
#ifndef STACK
#define STACK

typedef char base;
class Stack {
public:
    Stack (){
        topOfStack = NULL;
    };
    base top (void);
    base pop2(void);
    void pop(void);
    void push (const base &x);
    bool isNull(void);
    void destroy(void);
private:
    struct node;
    node *topOfStack;
};

#endif
```

STACK.CPP

```
#include <iostream>
#include <cstdlib>
#include "stack.h"
using namespace std ;

struct Stack::node {
    base hd;
    node *tl;
    node (){
     tl = NULL;
    }
};

base Stack::top (void){
    if (topOfStack == NULL) {
        cerr << "Error: top(null) \n";
        exit(1);
    }
        else return topOfStack->hd;
}

base Stack::pop2(void){
      if (topOfStack == NULL) {
```

```
            cerr << "Error: pop(null) \n";
            exit(1);
        }
        else {
            node *oldTop = topOfStack;
            base r = topOfStack->hd;
            topOfStack = topOfStack->tl;
            delete oldTop;
            return r;
        }
    }

void Stack::pop(void){
    if (topOfStack == NULL) {
        cerr << "Error: pop(null) \n";
        exit(1);
    }
    else {
        node *oldTop = topOfStack;
        topOfStack = topOfStack->tl;
        //delete oldTop->hd;
        delete oldTop;
    }
}

void Stack::push (const base &x){
  node *p;
  p = topOfStack;
  topOfStack = new node;
  if (topOfStack != NULL) {
       topOfStack->hd = x;
       topOfStack->tl = p;
  }
  else {
       cerr << "Memory not enough\n";
       exit(1);
  }
}

bool Stack::isNull(void){
    return (topOfStack == NULL) ;
}

void Stack:: destroy(void){
    while (topOfStack != NULL) {
         pop();
    }
}
```

MAIN.CPP

```
#include <cstdio>
#include <iostream>
#include <fstream>
#include "stack.h"
using namespace std;

void read(Stack &s,std::ifstream &in, std::ofstream &out){
    char c;
    while ((c = in.get()) != '\n' && c!=EOF)
        s.push(c);
}

void write(Stack &s, std::ofstream &out){
    while (!s.isNull())
        out << s.pop2();
    out<<endl;
}

int main(){
      string file_name;
  string out_file_name;
  ifstream fin;
  ofstream fout;
  int ch;
  bool ex = true;
  Stack  s;
     while(ex){
    cout << "0-exit from the program" << '\n';
    cout << "1-input a line from a file" << '\n';
    cin>>ch;
    cin.ignore();
           switch (ch)
           {
           case 1:{ cout << "Enter input file name:" << '\n';
            cin >> file_name;
        cin.ignore();
        cout << "Enter output file name:" << '\n';
        cin >> out_file_name;s
        cin.ignore();
        fin.open(file_name, ifstream::in);
        fout.open(out_file_name, ofstream::out);
        if (!fin.is_open() || !fout.is_open()) {
          cout << "Error opening file.\n";
        }
        else {
          while (!fin.eof()){
            read(s,fin,fout);
            write(s, fout);
          }
        }
```

```
        break;
      }
      case 0: {
        ex = false;
        break;
      }
      default:{
        ex = false;
        cout << "Error of input!";
        break;
      }
    }
  }
    fin.close();
    fout.close();
    return 0;
}
```

ПРИЛОЖЕНИЕ Б. Тестовые случаи

Таблица 1 - Результаты тестов.

input	output	True/false
ABC GH 56 DK	CBA KD 65 HG	True
NBK FJD NRN FB BN 6 MKS	DJF KBN 6 NB BF NRN SKM	True
VLH HLNJ GHJL;JGF JL0MJ 67GF GH 8	FGJ;LJHG JNLH HLV 8 HG FG76 JM0LJ	True