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

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Программирование»

ТЕМА: Динамические структуры данных

Студент гр. 1304 _____ Дешура Д.В.

Преподаватель _____ Чайка К.В.

Санкт-Петербург
2022

Цель работы.

Создать программу, проверяющую html страницу на валидность, реализовать для неё класс со стеком на базе массива и интерфейсом к нему. Изучить работу с классами.

Задание.

Расстановка тегов.
Требуется написать программу, получающую на вход строку, (без кириллических символов и не более 3000 символов) представляющую собой код "простой" html-страницы и проверяющую ее на валидность. Программа должна вывести correct если страница валидна или wrong.

html-страница, состоит из тегов и их содержимого, заключенного в эти теги. Теги представляют собой некоторые ключевые слова, заданные в треугольных скобках. Например, <tag> (где tag - имя тега). Область действия данного тега распространяется до соответствующего закрывающего тега </tag> который отличается символом /. Теги могут иметь вложенный характер, но не могут пересекаться.
<tag1><tag2></tag2></tag1> - верно
<tag1><tag2></tag1></tag2> - не верно

Существуют теги, не требующие закрывающего тега.

Валидной является html-страница, в коде которой всякому открывающему тегу соответствует закрывающий (за исключением тегов, которым закрывающий тег не требуется).

Во входной строке могут встречаться любые парные теги, но гарантируется, что в тексте, кроме обозначения тегов, символы < и > не встречаются. аттрибутов у тегов также нет.

Теги, которые не требуют закрывающего тега:
, <hr>.

Стек (который потребуется для алгоритма проверки парности тегов) требуется реализовать самостоятельно на базе массива. Для этого необходимо:

Реализовать класс CustomStack, который будет содержать перечисленные ниже методы. Стек должен иметь возможность хранить и работать с типом данных char*

Объявление класса стека:

```
class CustomStack {

public:

// методы push, pop, size, empty, top + конструкторы, деструктор

private:

// поля класса, к которым не должно быть доступа извне

protected:  // в этом блоке должен быть указатель на массив данных

   char** mData;
};
```

Перечень методов класса стека, которые должны быть реализованы:

void push(const char* val) - добавляет новый элемент в стек

void pop() - удаляет из стека последний элемент

char* top() - доступ к верхнему элементу

size_t size() - возвращает количество элементов в стеке

bool empty() - проверяет отсутствие элементов в стеке

extend(int n) - расширяет исходный массив на n ячеек

Примечания:

Указатель на массив должен быть protected.

Подключать какие-то заголовочные файлы не требуется, всё необходимое подключено(<cstring> и <iostream>).

Предполагается, что пространство имен std уже доступно.

Использование ключевого слова using также не требуется.

Выполнение работы.

Класс состоит из protected стека (представляет собой динамически расширяемый массив указателей на char*), private переменных типа size_t, отвечающих за количество элементов в стеке (numEl) и за его размер (sizeOfStack), а также public методы интерфейса (void push(), void pop(), char* top(), size_t size(), bool empty(), void extend()) и методы конструктора (CustomStack()) и деструктора (~CustomStack()).

Мы инициализируем переменную mData при помощи конструктора класса и попарно ища первые вхождения символов ‘<’ и ‘>’ вытягиваем из входящей строки теги, далее мы складываем открывающие парные теги в стек, одиночные теги игнорируем, а встретив закрывающий парный тег мы сравниваем его с последним записанным открывающим тегом, если они равны, мы стираем открывающий тег из памяти, иначе выдаём ошибку.

Разработанный программный код см. в приложении А.

Тестирование.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные
1.	<html><a> adsdfsdabr> dsdfg><a><1><2><3><4><hr> rrffrr </4></3></2></1></html>	correct
2.	<html><head><title>HTML Document</title></head><body><p>This	correct

	text is bold, this is bold and italics</p></body></html>	
3.	<html>	wrong

Выводы.

Выполнив лабораторную работу мы создали программу, проверяющую html страницу на валидность, реализовали для неё класс со стеком на базе массива и интерфейсом к нему. Изучили работу с классами.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 1304_PR_Дешура_ДВ_ЛР4.c

```
#include <iostream>
#include <cstring>

using namespace std;

class CustomStack {
public:
    CustomStack(){
        mData = new char*[4];
        sizeOfStack = 4;
    }

    void push(const char* val){
        if(numEl + 1 > sizeOfStack){
            extend(4);
        }
        mData[numEl] = (char *)val;
        numEl ++;
    }

    void pop(){
        if(!empty()){
            mData[--numEl] = NULL;
        }
    }

    char* top(){
        if(!empty()){
            return mData[numEl - 1];
        }
        return NULL;
    }

    size_t size(){
        return numEl;
    }

    bool empty(){
        return !numEl;
    }

    void extend(int n){
        char** buf;
        buf = new char*[sizeOfStack + n];
        for(int i = 0; i < sizeOfStack; i++){
            buf[i] = mData[i];
        }
        sizeOfStack += (size_t)n;
        delete[] mData;
        mData = buf;
    }
```

```
    ~CustomStack(){
        delete[] mData;
    }

private:
    size_t sizeOfStack, numEl = 0;

protected:
    char** mData;
};

int main() {

    CustomStack mData = CustomStack();
    char str[3001], *s, *f;
    int errorFlag = 0;
    cin.getline(str, 3001);

    s = strchr(str, '<');
    while(s){

        f = strchr(s, '>');
        if(!f){
            errorFlag = 1;
            break;
        }

        *f = '\0';
        if(*(s + 1) == '/'){
            if(!mData.top()){errorFlag = 1; break;}
            if(!strcmp(s + 2, mData.top()))
                mData.pop();
            else {errorFlag = 1; break;}
        } else if(strcmp(s + 1, "br") && strcmp(s + 1, "hr"))
            mData.push(s+1);

        s = f + 1;
        s = strchr(s, '<');

    }

    if(errorFlag || !mData.empty())
        cout << "wrong";
    else
        cout << "correct";

    return 0;
}
```