*Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет***  ***имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ: Информатики и систем управления

КАФЕДРА: Компьютерные системы и сети

**Отчет**

**по усложнённой лабораторной работе №** 5

**Название лабораторной работы:** *Динамические структуры данных. Списки.*

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент гр. ИУ6-22 |  |  | | Боярских Никита |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |
| Преподаватель |  | |  | Черноусова Т. Г. |
|  | (Подпись, дата) | |  | (И. О. Фамилия) |

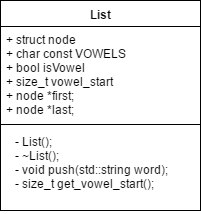
**Вариант 1**

Москва, 2017

**Задание:**

Дана символьная строка, состоящая из слов, разделенных пробелом. Составить программу, которая вводит строку, организует из ее слов два сортированных списка, (в первый входят все нечетные слова, а во второй - четные), считает в каждом списке количество слов, начинающихся на гласные буквы, и определяет, в каком из списков таких слов больше.

Для поставленной задачи разработан класс List (диаграмма классов представлена на рис. 1)



*Рисунок 1 - диаграмма классов программы, работающей со списком*

**Код программы, работающей со списком**

#include <iostream>

#include <string>

#include <cstring>

// Односвязный список

class List {

private:

// Структура одного элемента списка

struct node {

std::string data;

struct node \*next;

};

// Гласные буквы

char const VOWELS[11] = "AEUIOaeiou";

// Гласная ли буква

bool isVowel(char c);

// Количество слов, начинающихся с гласной буквы

size\_t vowel\_start;

(Продолжение кода на следующей странице)

(Продолжение кода)

// Указатели на начало и на конец списка

node \*first;

node \*last;

public:

List();

~List();

void push(std::string word);

size\_t get\_vowel\_start();

};

int main(int argc, char \*\*argv) {

using std::cin;

using std::cout;

using std::endl;

List odd, even;

std::string word;

// Счётчик слов

size\_t word\_counter = 0;

cout <<"Введите слова:"<< endl;

// Читвем слова, пока они есть в STDIN

while(cin>>word) {

// Чётные пишем в один список ...

if(++word\_counter % 2 == 0)

even.push(word);

// ... нечётные в другой

else

odd.push(word);

}

// Выводим результаты

cout<<"Слов, начинающихся на гласную букву в первом списке: "<<odd.get\_vowel\_start()<<endl;

cout<<"Слов, начинающихся на гласную букву во втором списке: "<<even.get\_vowel\_start()<<endl;

if(odd.get\_vowel\_start() < even.get\_vowel\_start()) {

cout<<"Слов, начинающихся на гласную букву во втором списке больше, чем в первом"<<endl;

}

else if(odd.get\_vowel\_start() > even.get\_vowel\_start()) {

cout<<"Слов, начинающихся на гласную букву в первом списке больше, чем во втором"<<endl;

} else {

cout<<"Слов, начинающихся на гласную букву в обоих списках одинаковое количество"<<endl;

}

return 0;

}

(Продолжение кода на следующей странице)

(Продолжение кода)

// Конструктор

List::List() {

vowel\_start = 0;

first = NULL;

last = NULL;

}

// Вставка элемента

void List::push(std::string str) {

if(first == NULL) {

first = new node;

last = first;

}

else {

last->next = new node;

last = last->next;

}

last->data = str;

last->next = NULL;

if( isVowel(str[0]) ) vowel\_start++;

}

// Геттер счётчика слов, начинающихся на гласную букву

size\_t List::get\_vowel\_start() {

return vowel\_start;

}

// Гласная ли буква

bool List::isVowel(char c) {

return (strchr(VOWELS, c));

}

// Деструктор

List::~List() {

while(first != NULL) {

node \*temp = first;

first = first->next;

delete temp;

}

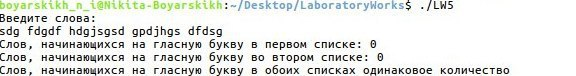
}

Проведено тестирование программы (см. табл. 1)

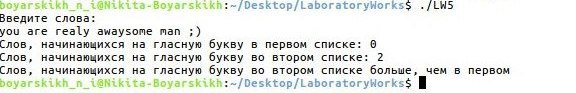
Таблица 1 – результаты тестирования программы, работающей со списком

|  |  |  |  |
| --- | --- | --- | --- |
| № | Исходные данные | Ожидаемый результат | Полученный результат |
| 1 | sag fdgdf hdgjsgsd gpdjhgs | 0 0 = | 0 0 = |
| 2 | you are really awesome man ;) | 0 2 < | 0 2 < |
| 3 | lol kek cheburek Tramp and Putin is already presidents | 2 1 > | 2 1 > |

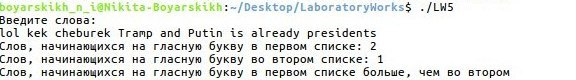
**Скриншоты, демонстрирующие работу программы**

**

*Рисунок 2 – тестирование программы при исходных данных №1 (см. табл. 1)*

**

*Рисунок 3 – тестирование программы при исходных данных №2 (см. табл. 1)*



*Рисунок 4 – Тестирование программы при исходных данных №3 (см. табл. 1)*

**Вывод:**

1) Разработаны классы для решения поставленной задачи. Составлена диаграмма в среде [draw.io](http://draw.io)(рис. 1)

2) Создан код программы по алгоритму в среде Vim

3) Проведено тестирование (см. табл. 1)

4) Тестирование показало корректность работы при заданных исходных данных (см. табл. 1 и рис. 2 - 4)