



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

О Т Ч Е Т

по лабораторной работе № 6

Название: Динамические структуры данных. Списки

Дисциплина: Объектно-ориентированное программирование

Студент

ИУ6-22Б

(Группа)

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

(И.О. Фамилия)

Москва, 2020

Задание

Дана символьная строка, состоящая из слов, разделенных пробелом. Составить программу, которая вводит строку, организует из ее слов однонаправленный список. Используя сформированный список, создать новый список, в который включить слова исходного списка, начинающиеся на гласные буквы. Вывести на экран исходный список и сформированный или сообщение об отсутствии требуемых слов.

Исходный код

(файл Source.cpp)

```
#include <iostream>
```

```
#include <conio.h>
```

```
#include <string.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <set>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    set <char> gl = { 'a','e','y','u','i','o','j' };
```

```
    struct zap {
```

```
        char* content;
```

```
        zap* next;
```

```
    };
```

```
    char inp[50], buf[50], * cut, * buff;
```

```
    zap* q, * p, * st, * st1;
```

```
    bool exist = false;
```

```
    st = new zap;
```

```
    p = st;
```

```
    (*p).next = NULL;
```

```
    puts(">>Input string: ");
```

```
    gets_s(inp);
```

```
    int i = 0;
```

```
    while (inp[i] != '\0') {
```

```
        int j = 0;
```

```
        buff = new char[50];
```

```
        // ATTENTION - if you forgot it - all list's elements will be addressed to one
```

```
string
```

```
        while (inp[i] == ' ') {
```

```
            i++;
```

```
        }
```

```
        while ((inp[i] != '\0') && (inp[i] != ' ')) {
```

```

        *(buff + j) = inp[i];
        i++;
        j++;
    }
    *(buff + j) = '\0';
    q = new zap;
    (*q).content = buff;
    (*p).next = q;
    p = (*p).next;
    (*p).next = NULL;
}

```

```

if (((*st).next != NULL)) {
    puts(">>>list1:");
}
else {
    puts(">>>list1 is empty");
}

```

```

st = (*st).next;

```

```

p = st;

```

```

while (p != NULL) {
    puts((*p).content);
    p = (*p).next;
}

```

```

p = st;

```

```

//q = NULL;

```

```

//st1 = NULL;

```

```

st1 = new zap;

```

```

(*st1).next = NULL;

```

```

q = st1;

```

```

while (p != NULL) {

```

```

    if (gl.count((*p).content[0]) != 0) {

```

```

        q->next = new zap;

```

```

        q->next->content = p->content;

```

```

        q = q->next;

```

```

        q->next = NULL;

```

```

    }

```

```

    p = p->next;

```

```

    }

    if (((*st1).next != NULL)) {
        puts(">>>list2:");
    }
    else {
        puts(">>>list2 is empty");
    }
    st1 = (*st1).next;

    p = st1;

    while (p != NULL) {
        puts((*p).content);
        p = (*p).next;
    }

    p = st;

    while (p != NULL) {
        q = p;
        p = (*p).next;
        delete[](*q).content;
        delete q;
    }

    p = st1;

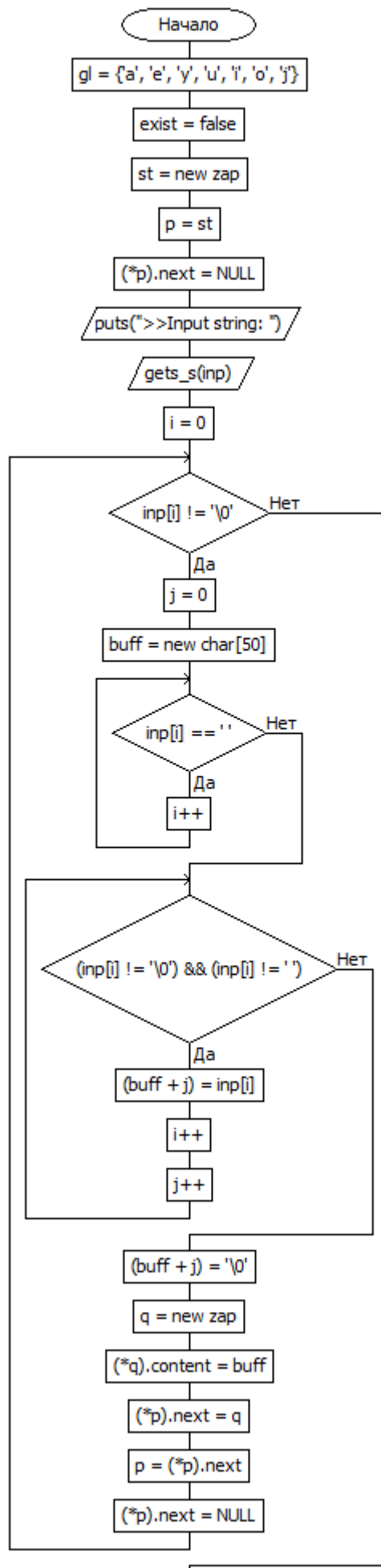
    while (p != NULL) {
        q = p;
        p = (*p).next;
        delete q;
    }
}

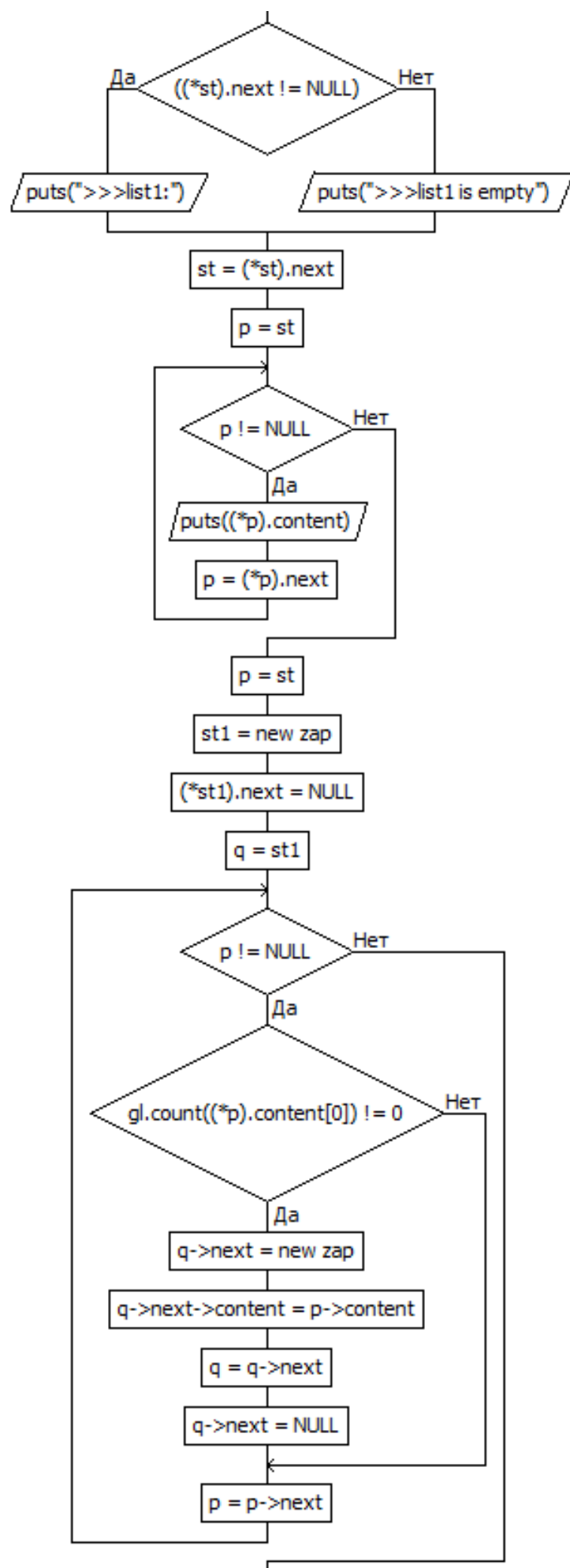
```

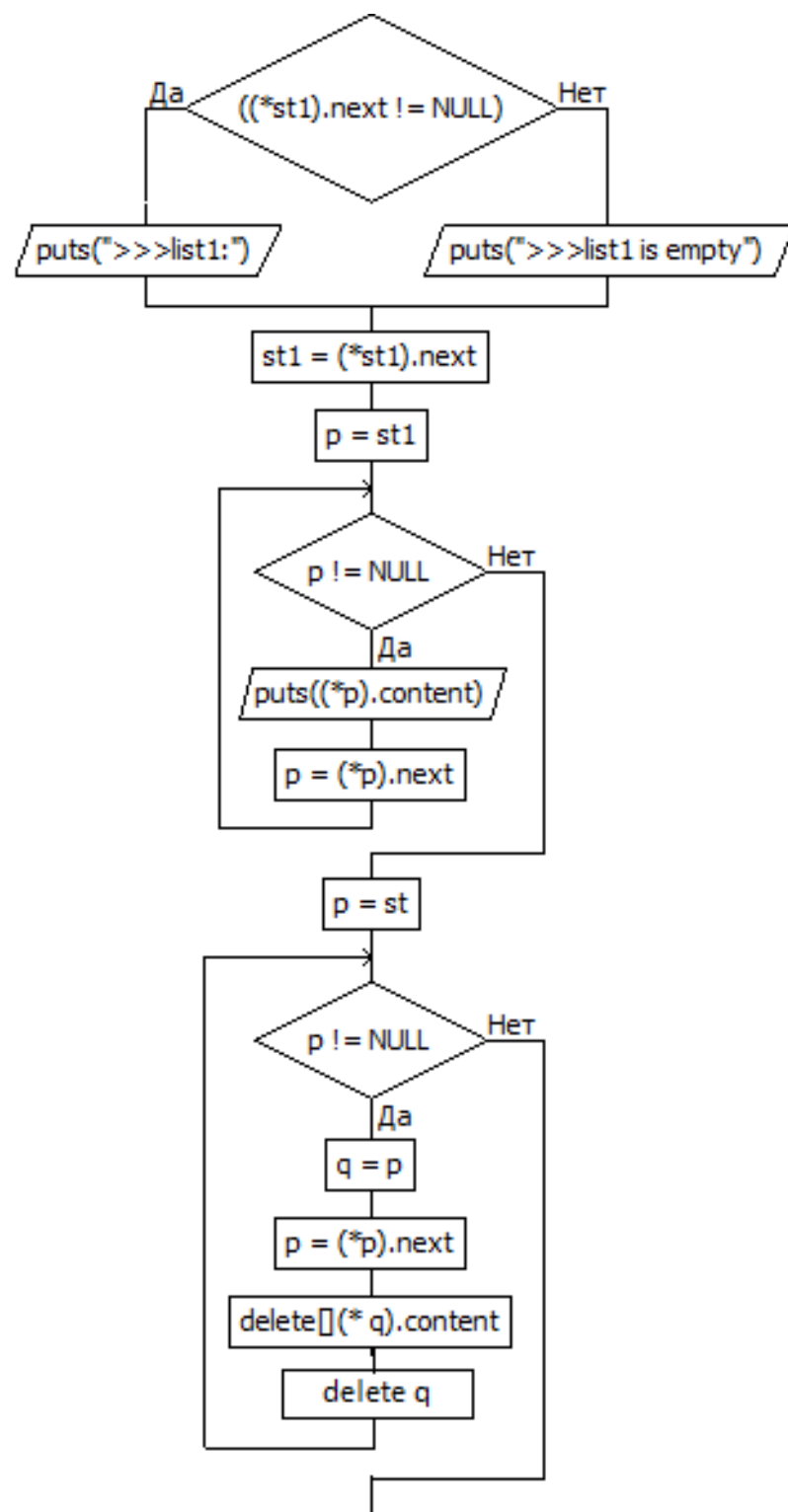
Тесты

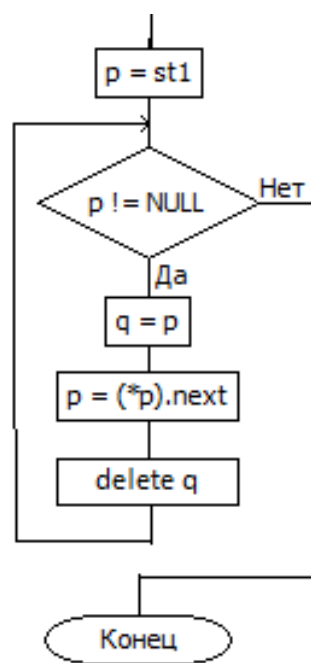
Входные данные	Ожидаемые выходные данные	Выходные данные
	>>>list1 is empty >>>list2 is empty	>>>list1 is empty >>>list2 is empty
abc vn obv	>>>list1: abc vn obv >>>list2: abc obv	>>>list1: abc vn obv >>>list2: abc obv
bn mn vc	>>>list1: bn mn vc >>>list2 is empty	>>>list1: bn mn vc >>>list2 is empty

Схема алгоритма









Вывод

- В общем случае списки в C++ не имеют значительных отличий от списком в Delphi
- При создании и очистке списка строк необходимо помнить, что операции со строками в C++ производятся по указателям, а значит для строк необходимо выделять память и очищать ее