



**«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)

Отчет

по лабораторной работе № 6

Название лабораторной работы: Списки.

Дисциплина: Алгоритмизация и программирование

Студент гр. ИУ6-14Б

18.11.2023

А.С. Воеводин

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

18.11.2023

О.А. Веселовская

(Подпись, дата)

(И.О. Фамилия)

Москва, 2023

Цель работы – Решить поставленную задачу, используя списки.

Задание – С клавиатуры вводится последовательность, содержащая четное число символов. Составить из символов список. Определить, совпадают ли первая и вторая половины последовательности. Вывести на экран результат. Исключить из последовательности указанный символ. Вывести на экран исходный список и полученный список или сообщение об отсутствии введенного символа в списке. При завершении программы освободить динамическую память.

Ход работы:

- Написание программы.
- Тестирование программы при различных тестовых данных.
- Изображение схемы алгоритмов программы и диаграммы классов.
- Вывод.

Для начала создадим многофайловую структуру, а именно: файлы List.hpp, List.cpp, Node.hpp, Node.cpp, что позволит удобно работать с классами. Далее напомним код программы:

```
#pragma once

class Node {
public:
    friend class List;
    Node();
    Node(char symb_, Node* next_, Node* prev_);
    ~Node();

private:
    char symb;
    Node* next;
    Node* prev;
};
```

Рисунок 1 – Объявление класса Node и его полей, методов

```

#include "Node.h"

Node::Node() : symb('\0'), next(nullptr), prev(nullptr) {}

Node::Node(char symb_, Node* next_, Node* prev_) :
    symb(symb_), next(next_), prev(prev_) {}

Node::~~Node() {
    symb = '\0';
    prev = nullptr;
    delete next;
}

```

Рисунок 2 – Реализация класса Node

```

#pragma once

#include "Node.h"

class List
{
public:
    List();
    ~List();

    void SetList();
    bool RemoveElement();
    void EqualsHalf();
    void PrintList();

private:
    Node* head;
    Node* tail;
};

```

Рисунок 3 – Объявление класса List, его методов и полей

```

#include "List.hpp"
#include <string>
#include <iostream>

List::List() : head(nullptr), tail(nullptr) {}

List::~List() {
    delete head;
}

void List::SetList() {
    std::string tempStr;
    if (head != nullptr) {
        delete head;
    }
    head = new Node();
    std::cout << "Enter your string, length should be even: ";
    std::getline(std::cin, tempStr);
    while (tempStr.length() & 1 || tempStr.length() == 0) {
        std::getline(std::cin, tempStr);
        if (tempStr.length() & 1 || tempStr.length() == 0) {
            std::cout << "Wrong string, try again: ";
        }
    }
    Node* currElem = head;

    for (size_t i = 0; i < tempStr.length(); ++i) {
        currElem->symb = tempStr[i];
        currElem->next = new Node('\0', nullptr, currElem);
    }
}

```

Рисунок 4 – Реализация класса List часть 1

```

    for (size_t i = 0; i < tempStr.length(); ++i) {
        currElem->symb = tempStr[i];
        currElem->next = new Node('\0', nullptr, currElem);
        if (i == 0) {
            currElem->prev = nullptr;
        }
        if (i == tempStr.length() - 1) {
            delete currElem->next;
            currElem->next = nullptr;
        }
        tail = currElem;
        currElem = currElem->next;
    }
    std::cout << std::endl;
}

bool List::RemoveElement() {
    std::cout << "Enter your char, that will be removed"
    << "(if you enter a string the first symbol will be deleted) : ";
    char symbToRemove = '\0';
    std::cin >> symbToRemove;
    bool isRemoved = false;
    Node* currElem = head;
    Node* toDelete;
    while (currElem) {
        if (currElem->symb == symbToRemove) {
            isRemoved = true;
            toDelete = currElem;

```

Рисунок 5 – Реализация класса List часть 2

```

        if (currElem->prev != nullptr) {
            currElem->prev->next = currElem->next;
        }
        if (currElem->next != nullptr) {
            currElem->next->prev = currElem->prev;
        }
        if (currElem == head) {
            head = currElem->next;
        }
        if (currElem == tail) {
            tail = currElem->prev;
        }
        currElem = currElem->next;
        toDelete->next = nullptr;
        delete toDelete;
    }
    else {
        currElem = currElem->next;
    }
}

if (isRemoved) {
    std::cout << "Successfully removed " << symbToRemove << std::endl << std::endl;
}
else {
    std::cout << "Your symbol isn't in your string" << std::endl << std::endl;
}
return isRemoved;
}

```

Рисунок 6 – Реализация класса List часть 3

```

void List::PrintList() {
    Node* currElem = head;
    if (currElem == nullptr) {
        std::cout << "Your string is empty" << std::endl << std::endl;
    }
    else {
        std::cout << "Your string: ";
        while (currElem) {
            std::cout << currElem->symb;
            currElem = currElem->next;
        }
        std::cout << std::endl << std::endl;
    }
}

void List::EqualsHalf() {
    Node* left = head;
    Node* right = tail;
    bool isEqual = true;
    while (left->next != right && isEqual) {
        if (left->symb != right->symb) {
            isEqual = false;
        }
        left = left->next;
        right = right->prev;
    }
    if (isEqual) {
        std::cout << "Your string is a palindrome" << std::endl << std::endl;
    }
}

```

Рисунок 7 – Реализация класса List часть 4

```

    }
    if (isEqual) {
        std::cout << "Your string is a palindrome" << std::endl << std::endl;
    }
    else {
        std::cout << "Your string is not a palindrome" << std::endl << std::endl;
    }
}

```

Рисунок 8 – Реализация класса List часть 5

```

#include <iostream>
#include "List.hpp"

int main()
{
    List str;
    bool isWantContinue = true;
    bool isSetted = false;
    bool isRemoved = false;
    while (isWantContinue) {
        char ans = '\0';
        std::cout << "What do you want to do?" << std::endl
        << "1) Enter string" << std::endl
        << "2) Remove symbol" << std::endl
        << "3) Print string" << std::endl
        << "4) Check for palindrome" << std::endl
        << "5) End" << std::endl << std::endl;
        std::cin >> ans;
        switch (ans)
        {
            case '1': {
                str.SetList();
                isSetted = true;
                isRemoved = false;
                break;
            }
            case '2': {
                if (!isSetted) {
                    std::cout << "Your string is not setted. Set your string first" << std::endl << std::endl;
                }
            }
        }
    }
}

```

Рисунок 9 – Основной файл часть 1


```

        if (!isSetted) {
            std::cout << "Your string is not setted. Set your string first" << std::endl << std::endl;
        }
        else {
            isRemoved = str.RemoveElement();
        }
        break;
    }
    case '3': {
        str.PrintList();
        break;
    }
    case '4': {
        if (!isSetted) {
            std::cout << "Your string is not setted. Set your string first" << std::endl << std::endl;
        }
        else if (isRemoved) {
            std::cout << "Can not check for palindrome, "
                << "because can check only not removed strings. (check your task)"
                << std::endl << std::endl;
        }
        else {
            str.EqualsHalf();
        }
        break;
    }
    case '5': {
        isWantContinue = false;
        break;
    }
    default:
        std::cout << "Wrong command. Try again" << std::endl << std::endl;
        break;
    }
}

return 0;
}

```

Рисунок 10 – Основной файл часть 2

Далее введём несколько тестирующих команд, проверяющих функционал написанной программы:

```
What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

2
Your string is not setted. Set your string first

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

3
Your string is empty

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

4
Your string is not setted. Set your string first

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

5
C:\Users\artem\source\repos\laba_6\x64\Debug\laba_6.exe (процесс 16516) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

Рисунок 11 – Тестовые команды часть 1

```
What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

1
Enter your string, length should be even: aaa
Wrong string, try again: aaaa

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

3
Your string: aaaa

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

4
Your string is a palindrome

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

2
Enter your char, that will be removed(if you enter a string the first symbol will be deleted) : a
Successfully removed a
```

Рисунок 12 – Тестовые команды часть 2

```
What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

3
Your string is empty

What do you want to do?
1) Enter string
2) Remove symbol
3) Print string
4) Check for palindrome
5) End

5
```

Рисунок 13 – Тестовые команды часть 3

Как видно из рисунков, программа работает корректно. Далее изобразим программу в виде схемы алгоритма:

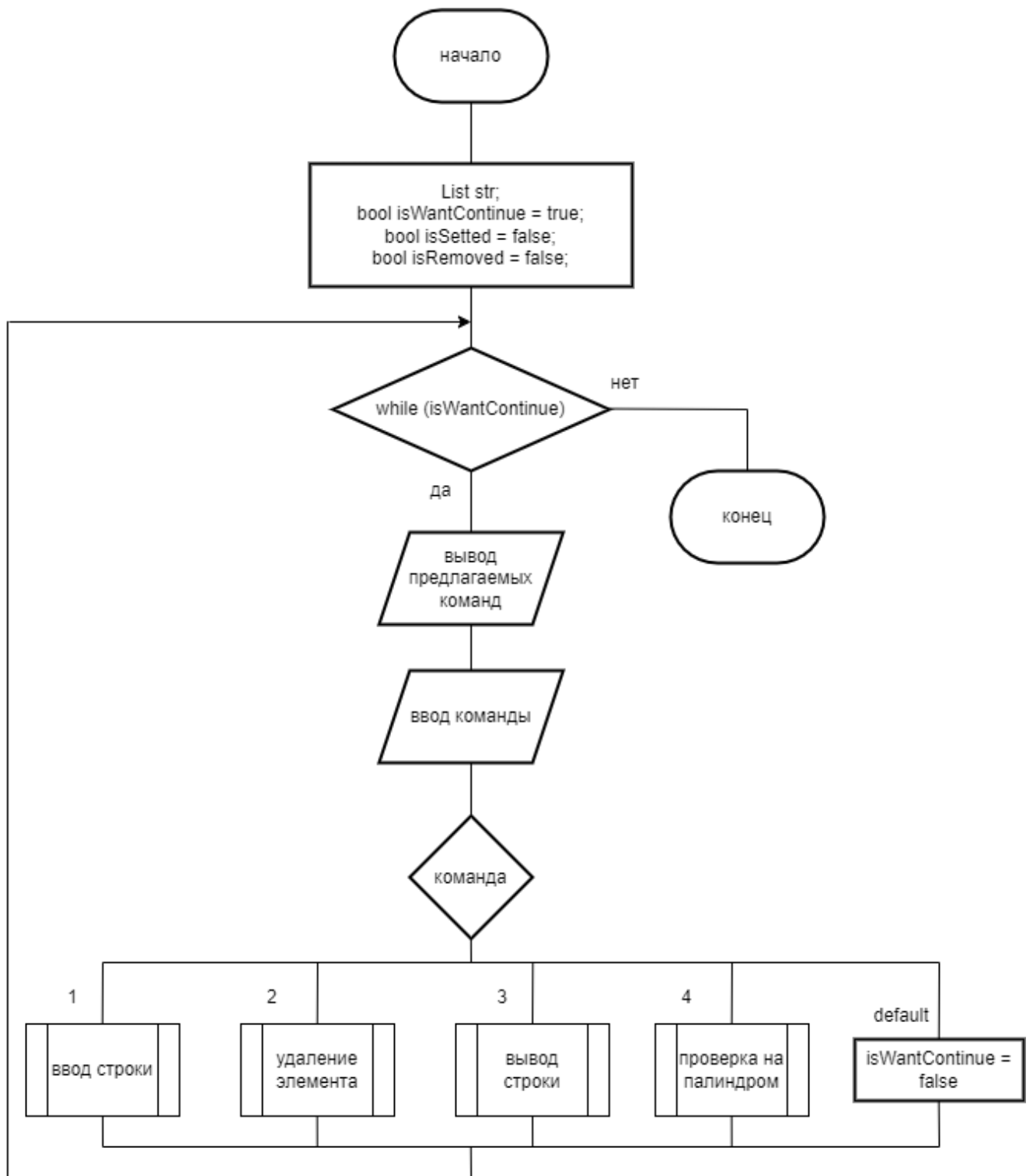


Рисунок 14 – Схема алгоритма основного файла

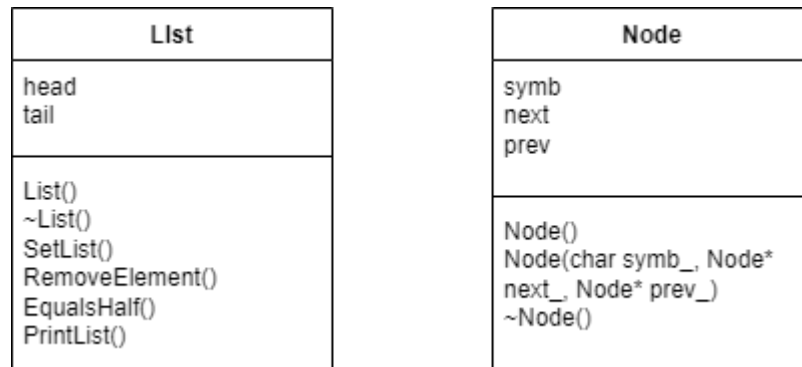


Рисунок 15 – Диаграммы классов, используемых в программе

Вывод: в ходе лабораторной работы были получены навыки работы с динамическими структурами, а именно списком, классами, его полями и методами, конструкторами и деструкторами.