



МИНОРБНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего  
образования*

*«МИРЭА - Российский технологический университет»*

**РТУ МИРЭА**

---

Отчёт по выполнению практического задания № 6

**Тема:**

«Двунаправленные динамические списки»

Дисциплина: «Структуры и алгоритмы обработки данных»

Выполнил студент:

Фамилия И.О.

Фамилия И.О.

Группа:

AAAA-00-00

Номер группы

Москва 2023

## СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ.....	3
1.1 Цель работы.....	3
1.2 Задание.....	3
1.3 Индивидуальный вариант.....	4
2 ВЫПОЛНЕНИЕ ЗАДАНИЯ.....	5
2.1 Определение списка операций над списком.....	5
2.2 Код программы.....	8
2.3 Результаты тестирования.....	16
3 ВЫВОД.....	18
4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....	19

# 1 ПОСТАНОВКА ЗАДАЧИ

## 1.1 Цель работы

Получение знаний и практических навыков управления двунаправленным списком в программах на языке C++.

## 1.2 Задание

Разработать многомодульную программу, которая демонстрирует выполнение всех операций, определенных вариантом, над линейным двунаправленным динамическим списком.

Требования к разработке.

1. Разработать структуру узла списка, структура информационной части узла определена вариантом. Для определения структуры узла списка, используйте тип `struct` или `class`. Сохраните определение структуры узла и прототипы функций в заголовочном файле.

2. Разработайте функции для выполнения операции над линейным двунаправленным динамическим списком:

- создание списка;
- вставку узла;
- удаление узла;
- вывод списка в двух направлениях (слева направо и справа налево);
- поиск узла с заданным значением (операция должна возвращать указатель на узел с заданным значением).

3. Дополнительные операции над списком, указанные вариантом, оформите в виде функций и включите в отдельный файл с расширением `сpp`. Подключите к этому файлу заголовочный файл с определением структуры узла.

4. Разработайте программу, управляемую текстовым меню, и включите в меню демонстрацию выполнения всех операций задания и варианта.

5. Проведите тестирование операций.

- Оцените сложность алгоритма первой дополнительной операции.

### 1.3 Индивидуальный вариант

17	<p>Номер класса в школьном журнале, указатель на список тем уроков. О каждой теме должны храниться сведения: <u>код дисциплины</u>, название темы, дата изучения темы.</p> <p>Примечание. Для расшифровки названия дисциплины по ее коду создать массив для нескольких дисциплин, элемент массива должен хранить: код и название дисциплины. с кодом</p>	<p>Вставить сведения по теме урока по определенной дисциплине в список тем, в конец списка тем данной дисциплины.</p> <p>Сформировать массив (или список) указателей на начало списка тем по определенной дисциплине. Удалить темы по заданной дисциплине.</p>
----	--	--

## 2 ВЫПОЛНЕНИЕ ЗАДАНИЯ

### 2.1 Определение списка операций над списком

Список представляет собой запись тем и дат занятий по коду дисциплины.

В таком случае значение узла данного списка будет представлять собой три поля типа string — код, тема, дата.

Для представления дисциплины в текстовом формате был сделан словарь типа код-название.

В ходе выполнения задачи понадобились дополнительные операции, такие как вставка в список по коду после последней записи с таким кодом и удаление всех записей по коду дисциплины.

Приведём рисунки для каждой базовой операции над списком. Удаление из списка приведём на рис. 1, вставку в список на рис. 2. Дополнительные операции являются дополнением для базовых операций, поэтому приводить для них рисунки не будем. На рисунках красными стрелками показан список до, синими — после.

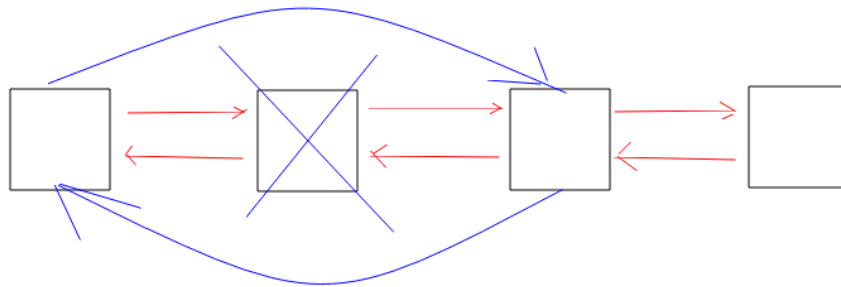


Рисунок 1 - Удаление из списка

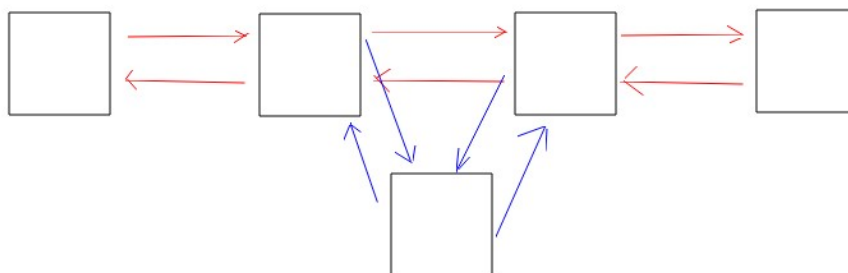


Рисунок 2 - Вставка в список

Рассмотрим дополнительные операции. В случае вставки в список по данному коду используется вставка после последнего найденного элемента. В случае удаления из списка всех записей с данным кодом, используется удаление из списка до тех пор, пока в списке содержится дисциплина с таким кодом.

Временная сложность первой дополнительной операции будет являться линейной, так как вставка будет представлять собой одиночный проход по списку с дальнейшей вставкой, а значит  $O(n) = n$ .

Приведём тесты для дополнительных операций. Тесты на удаление представлены в табл. 1, на вставку в табл. 2.

Таблица 1 - Результаты тестирования дополнительного удаления

Исходный список	Ключ	Результат
История России Смута 13.10.2020 Физика Электричество 17.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021	1	История России Смута 13.10.2020 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020
	2	История России Смута 13.10.2020 Физика Электричество 17.05.2022 Физика Механика 12.01.2021
	4	Физика Электричество 17.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021

Таблица 2 - Результаты тестирования дополнительной вставки

Исходный список	Значение	Результат
История России Смута 13.10.2020 Физика Электричество 17.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021	1 Теорема гаусса 18.05.2022	История России Смута 13.10.2020 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 История России Смута 13.10.2020 Физика Электричество 17.05.2022 Физика Теорема гаусса 18.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021
	4 Раскол 18.10.2020	История России Смута 13.10.2020 Физика Электричество 17.05.2022 Физика Механика 12.01.2021 История России Смута 13.10.2020 История России Раскол 18.10.2020 Физика Электричество 17.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021
	2 Векторы 01.01.2021	История России Смута 13.10.2020 Физика Электричество 17.05.2022 Линейная алгебра и аналитическая геометрия Операторы 24.11.2020 Линейная алгебра и аналитическая геометрия Векторы 01.01.2021 Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020 Физика Механика 12.01.2021

## 2.2 Код программы

Заголовочные файлы представлены в листингах 1-3. Коды реализации представлены в листингах 4-6. Алгоритм основной программы представлен в листинге 7.

Листинг 1 - Файл Node.h

```
#include "const.h"

struct Node
{
    Node *next;
    Node *prev;
    string code, name, date;
    Node(string code, string name, string date);
    void print();
};
```

Листинг 2 - Файл List.h

```
#include "Node.h"

struct List
{
    Node *first;
    Node *last;

    List();
    bool is_empty();
    Node *find(string code);

    void push_back(string code, string name, string date);
    void insert(string code, string name, string date);

    void remove_first();
    void remove_last();
    bool remove(string code);
    void remove_all(string code);

    void print_l2r();
    void print_r2l();
};
```



### Листинг 3 - Файл const.h

```
#include <map>
#include <iostream>
#include <vector>
#include <string>
#include <random>

using namespace std;

extern map<string, string> disciplines;

extern map<string, vector<string>> themes;

extern vector<string> dates;
```

### Листинг 4 - Файл Node.cpp

```
#include "Node.h"

Node::Node(string code, string name, string date)
{
    this->code = code;
    this->name = name;
    this->date = date;
}

void Node::print()
{
    cout << disciplines[code] << " " << name << " " << date;
}
```

### Листинг 5 - Файл List.cpp

```
#include "List.h"

List::List() : first(nullptr), last(nullptr){};

bool List::is_empty()
{
    return first == nullptr;
}

Node *List::find(string code)
{
    Node *p = first;
    while (p && p->code != code)
        p = p->next;
    return (p && p->code == code) ? p : nullptr;
}
```

```

void List::push_back(string code, string name, string date)
{
    Node *p = new Node(code, name, date);
    if (is_empty())
    {
        p->prev = nullptr;
        p->next = nullptr;
        first = p;
        last = p;
        return;
    }
    p->prev = last;
    last->next = p;
    last = p;
}

void List::insert(string code, string name, string date)
{
    Node *f = first;

    while (f->next && f->code != code)
    {
        f = f->next;
    }

    Node *p = new Node(code, name, date);

    Node *pr = f;
    Node *nx = f->next;

    pr->next = p;
    nx->prev = p;
    p->prev = pr;
    p->next = nx;
}

void List::remove_first()
{
    first = first->next;
    first->prev = nullptr;
}

void List::remove_last()
{
    last = last->prev;
    last->next = nullptr;
}

bool List::remove(string code)
{

```

```

    if (is_empty())
        return false;
    if (first->code == code)
    {
        remove_first();
        return true;
    }
    else if (last->code == code)
    {
        remove_last();
        return true;
    }

    Node *f = first->next;
    while (f && f->code != code)
    {
        f = f->next;
    }

    if (!f)
    {
        return false;
    }

    Node *pr = f->prev;
    Node *nx = f->next;

    nx->prev = pr;
    pr->next = nx;
    return true;
}

void List::remove_all(string code)
{
    while (remove(code))
    {
    }
}

void List::print_l2r()
{
    Node *p = first;

    while (p)
    {
        p->print();
        cout << endl;
        p = p->next;
    }
}

```

```

void List::print_r2l()
{
    Node *p = last;

    while (p)
    {
        p->print();
        cout << endl;
        p = p->prev;
    }
}

```

#### Листинг 6 - Файл const.cpp

```

#include "const.h"

using namespace std;

map<string, string> disciplines = {
    {"1", "Физика"},
    {"2", "Линейная алгебра и аналитическая геометрия"},
    {"3", "Математический анализ"},
    {"4", "История России"},
    {"5", "Иностранный язык"}
};

map<string, vector<string>> themes = {
    {"1", {"Электричество", "Мкт", "Кинематика", "Механика", "Магнетизм"}},
    {"2", {"Матрицы", "Определители", "СЛАУ", "Пространства", "Операторы"}},
    {"3", {"Пределы", "Производные", "Дифференциал", "Интегралы", "Тейлор"}},
    {"4", {"Древняя Русь ", "Рюриковичи", "Смута", "Романовы", "Великая Отечественная Война"}},
    {"5", {"Грамматика", "Лексика", "Синтаксис", "Аудирование", "Чтение"}}};

vector<string> dates = {
    "01.09.2020", "08.09.2020", "15.09.2020", "22.09.2020",
    "29.09.2020",
    "06.10.2020", "13.10.2020", "20.10.2020", "27.10.2020",
    "03.11.2020", "10.11.2020", "17.11.2020", "24.11.2020",
    "01.12.2020", "08.12.2020", "15.12.2020", "22.12.2020",
    "29.12.2020",
    "05.01.2021", "12.01.2021", "19.01.2021", "26.01.2021",
    "02.02.2021", "09.02.2021", "16.02.2021", "23.02.2021",
    "02.03.2021", "09.03.2021", "16.03.2021", "23.03.2021",
    "30.03.2021",
    "03.05.2022", "10.05.2022", "17.05.2022", "24.05.2022",
    "31.05.2022"};

```

## Листинг 7 - Файл main.cpp

```
#include "List.h"

List get_list()
{
    List lst;
    vector<string> keys;

    for (auto const &imap : disciplines)
        keys.push_back(imap.first);
    for (int i = 0; i < 5; i++)
    {
        string theme = keys[rand() % 5];
        lst.push_back(theme, themes[theme][rand() % 5],
dates[rand() % 40]);
    }
    return lst;
}

List create_list(vector<string> codes, vector<string> themes,
vector<string> dates)
{
    List lst;
    for (int i = 0; i < codes.size(); i++)
    {
        lst.push_back(codes[i], themes[i], dates[i]);
    }
    return lst;
}

void menu()
{
    cout << "Двусвязный список: список тем уроков" << endl;
    cout << "Выберите команду из предложенных" << endl;
    cout << "1 - задать список вручную" << endl;
    cout << "2 - сгенерировать список" << endl;
    string comm;
    cin >> comm;
    List lst;
    if (comm == "1")
    {
        cout << "Введите количество элементов: ";
        int size = 0;
        cin >> size;
        vector<string> codes, themes, dates;
        cout << "Доступные темы: " << endl;
        for (auto const &imap : disciplines)
            cout << imap.first << " - " << imap.second << endl;
        cout << "Введите элементы списка через строку в
формате: { код тема дата }: " << endl;
```

```

        for (int i = 0; i < size; i++)
        {
            string code, theme, date;
            cin >> code >> theme >> date;
            codes.push_back(code);
            themes.push_back(theme);
            dates.push_back(date);
        }
        lst = create_list(codes, themes, dates);
        cout << endl;
    }
    else if (comm == "2")
    {
        lst = get_list();
        cout << "Список сгенерирован" << endl;
    }
    else
    {
        cout << "Неизвестная команда" << endl;
        return;
    }

    while (true)
    {
        cout << "Выберите команду из предложенных" << endl;
        cout << "1 - вывести список слева направо" << endl;
        cout << "2 - вывести справа слева налево" << endl;
        cout << "3 - найти узел по ключу (коду дисциплины) и вывести " << endl;
        cout << "4 - вставить сведения по теме урока после последней темы с такой дисциплиной" << endl;
        cout << "5 - удалить все темы по ключу (коду дисциплины)" << endl;
        cout << "6 - выход" << endl;

        cin >> comm;
        if (comm == "1")
        {
            cout << endl;
            lst.print_l2r();
            cout << endl;
        }
        else if (comm == "2")
        {
            cout << endl;
            lst.print_r2l();
            cout << endl;
        }
        else if (comm == "3")
        {
            cout << "Введите код дисциплины: ";

```

```

        string code;
        cin >> code;
        lst.find(code)->print();
        cout << endl;
    }
    else if (comm == "4")
    {
        cout << "Введите новую запись в формате: { код тема
дата }: ";
        string code, theme, date;
        cin >> code >> theme >> date;
        lst.insert(code, theme, date);

        cout << "Вставка успешна" << endl;
    }
    else if (comm == "5")
    {
        cout << "Введите код дисциплины: ";
        string code;
        cin >> code;
        lst.remove(code);
        cout << "Удаление успешно" << endl;
    }
    else if (comm == "6")
    {
        cout << "Выход..." << endl;
        return;
    }
}

int main()
{
    menu();
    return 0;
}

```

## 2.3 Результаты тестирования

Результаты тестирования представлены на рис. 3, 4.

```
Двусвязный список: список тем уроков
Выберите команду из предложенных
1 - задать список вручную
2 - сгенерировать список
2
Список сгенерирован
Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
3
Введите код дисциплины: 2
Линейная алгебра и аналитическая геометрия Операторы 24.11.2020
Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
1

История России Смута 13.10.2020
Физика Электричество 17.05.2022
Линейная алгебра и аналитическая геометрия Операторы 24.11.2020
Линейная алгебра и аналитическая геометрия СЛАУ 15.09.2020
Физика Механика 12.01.2021

Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
6
Выход...
```

Рисунок 3 - Результаты тестирования



```
Двусвязный список: список тем уроков
Выберите команду из предложенных
1 - задать список вручную
2 - сгенерировать список
1
Введите количество элементов: 3
Доступные темы:
1 - Физика
2 - Линейная алгебра и аналитическая геометрия
3 - Математический анализ
4 - История России
5 - Иностранный язык
Введите элементы списка через строку в формате: { код тема дата }:
1 Электромагнетизм 01.09.2024
2 Векторы 01.09.2024
3 Пределы 02.09.2024

Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
4
Введите новую запись в формате: { код тема дата }: 2 Матрицы 08.09.2024
Вставка успешна
Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
5
Введите код дисциплины: 2
Удаление успешно
Выберите команду из предложенных
1 - вывести список слева направо
2 - вывести справа слева налево
3 - найти узел по ключу (коду дисциплины) и вывести
4 - вставить сведения по теме урока после последней темы с такой дисциплиной
5 - удалить все темы по ключу (коду дисциплины)
6 - выход
6
```

Рисунок 4 - Результаты тестирования

### **3 ВЫВОД**

Была проведена работа с двусвязным списком и с его оптимизацией для определённой задачи. Разработаны программный интерфейс для работы с данным списком и протестирована его работа.

#### **4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ**

1. Структуры и алгоритмы обработки данных : Лекционные материалы / Рысин М. Л. МИРЭА — Российский технологический университет, 2022/23. – 77 с.