



МИНОРБНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение высшего
образования*

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Отчёт по выполнению практического задания № 5

Тема:

«Однонаправленный динамический список»

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.....	6
2.3 Результаты тестирования программы.....	12
3 ВЫВОД.....	14
4 ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ.....	15

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Цель работы

Получить знания и практические навыки управления динамическим однонаправленным списком.

1.2 Задание

Реализуйте программу решения задачи варианта по использованию линейного однонаправленного списка.

Требования для всех вариантов

1. Информационная часть узла определена вариантом
2. Разработать функции вставки нового узла перед первым узлом и удаления узла по ключу.
3. Реализуйте возможность
 - а) создания нового списка вручную, а также
 - б) использования уже готового списка для тестирования заданий индивидуального варианта.
4. Разработать функцию вывода списка в консоль.
5. Разработать функции согласно индивидуальному варианту. При необходимости можно добавлять вспомогательные функции, декомпозируя задачу.
6. Реализуйте текстовое пользовательское меню.
7. В основной программе выполните тестирование каждой функции (пункты 2-5).
8. Составить отчет по выполненному заданию (формат отчета после вариантов).

1.3 Индивидуальный вариант

17	<степень, коэффициент > <int, real>	<p>Линейный многочлен n-ой степени представлен в программе как линейный однонаправленный список. Каждый i-ый узел списка хранит информацию по i-му члену многочлена. Поэтому информационная часть узла состоит из двух значений: степень члена и коэффициент при этой степени. Если i-ый член в многочлене отсутствует, то узел не создается.</p> <p>1. Разработать функцию, которая создает список по переданному в качестве параметра многочлену: он представлен массивом коэффициентов и их степеней.</p> <p>2. Разработать функцию, которая выводит многочлен и представляет его в форме выражения.</p> <p>3. Разработать функцию, которая вычисляет значение многочлена при заданном значении x. В вычислении использовать алгоритм Горнера.</p>
----	--	---

2 ВЫПОЛНЕНИЕ ЗАДАНИЯ

2.1 Определение списка операций над списком

Список представляет собой линейный многочлен n -ой степени, каждый узел представляет из себя одно слагаемое этого списка, состоящего из степени перед x и его коэффициентом.

В таком случае значение узла будет представлять собой пару чисел типа `int` — степень x , и `float` — коэффициент перед x .

В ходе выполнения задачи, понадобились дополнительные операции, такие как вычисление значения при заданном x и вывод списка в виде многочлена по схеме Горнера.

Для реализации функции вывода многочлена мы берём каждый узел и выводим для него его значение в виде коэффициента и икса в n -ой степени. Если степень равна 0, то x не выводится, если коэффициент равен единице, то он не выводится. Рисунок вывода списка в виде многочлена представлен на рис. 1.

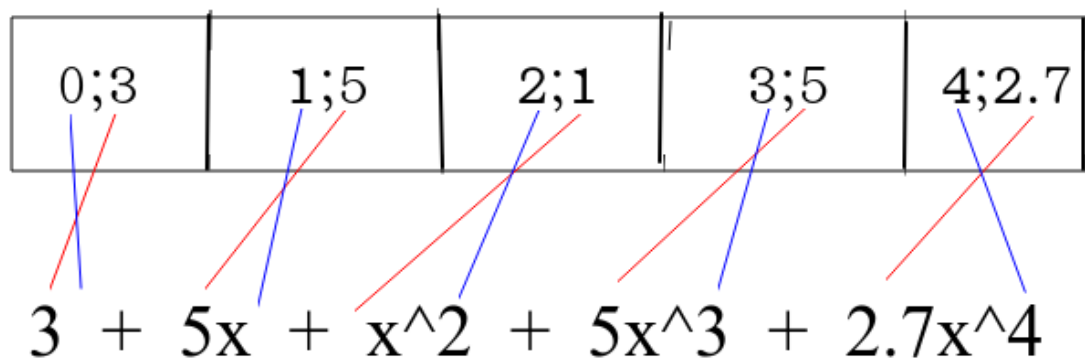


Рисунок 1 - Рисунок вывода списка

Алгоритм вычисления значения многочлена при заданном x представляет собой рекурсивный алгоритм Горнера. Рисунок алгоритма представлен на рис. 2.

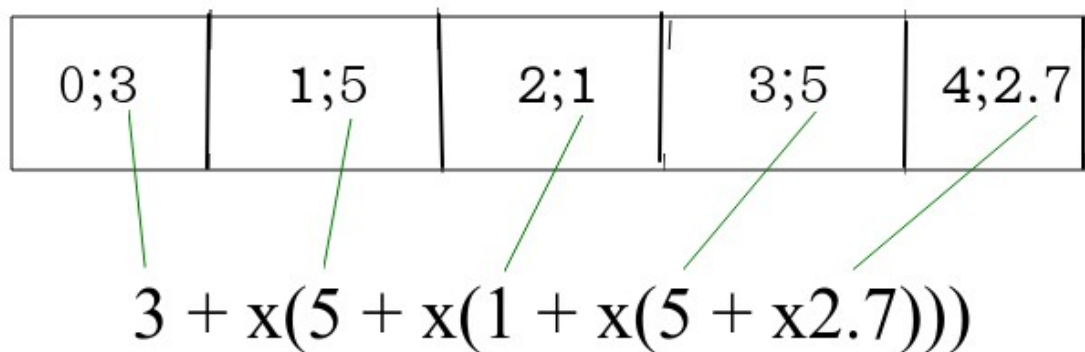


Рисунок 2 - Рисунок представления алгоритма Горнера

Запишем результаты тестов в таблицу. Таблица тестов для вывода и вычисления значения представлены в табл.1

Таблица 1 - Результаты тестирования функций

Исходный список	Вывод	x	Результат
[0;3, 1;5, 2;1, 3;5, 4;2.7]	$3 + 5x + x^2 + 5x^3 + 2.7x^4$	1	16.7
[0;0, 1;1, 2;4.35]	$x + 4.35x^2$	2	19.4
[0;0, 1;5, 2;8, 3;0, 4;3]	$5x + 8x^2 + 3x^4$	0	3

2.2 Код программы

Описание структуры узла представлен в листинге 1. Описание структуры списка представлен в листинге 2. Алгоритм пользовательского меню представлен в листинге 3. Вспомогательные функции в листинге 4. Код основной программы представлен в листинге 5.

Листинг 1 - Описание структуры узла

```
struct Node
{
    pair<int, float> val;
    Node *next;
    Node(pair<int, float> _val) : val(_val), next(nullptr) {}
    void print()
    {
        if(val.second == 0) return;
        if (val.first == 0) cout << val.second;
        else if (val.first == 1) cout << "x";
        else if (val.second == 1) cout << "x^" << val.first;
        else cout << val.second << "x^" << val.first;
    }
    void edit_val(pair<int, float> _val) { this->val = _val; }
};
```

Листинг 2 - Описание структуры списка

```
struct list
{
    Node *first;
    Node *last;
    list() : first(nullptr), last(nullptr) {}
    bool is_empty()
    {
        return first == nullptr;
    }
    void push_back(pair<int, float> _val)
    {
        Node *p = new Node(_val);
        if (is_empty())
        {
            first = p;
            last = p;
            return;
        }
        last->next = p;
        last = p;
    }
    void print()
    {
        if (is_empty())
            return;
        Node *p = first;
        while (p)
        {
            p->print();
            if (p->next != nullptr && p->val.second != 0)
                cout << " + ";
            p = p->next;
        }
        cout << endl;
    }
    Node *find(pair<int, float> _val)
    {
        Node *p = first;
        while (p && p->val != _val)
            p = p->next;
        return (p && p->val == _val) ? p : nullptr;
    }

    void insert_first(float coef)
    {
        // Создание нового узла
        Node *p = new Node(pair<int, float>{0, coef});
        // Задание ему следующего как первого текущего
        p->next = first;
        // Замена первого на новый узел
    }
}
```

```

        first = p;

        p = p->next;
        while (p->next != nullptr)
        {
            p->edit_val(pair<int, float>{p->val.first + 1, p->val.second});
            p = p->next;
        }
    }

    void remove_first()
    {
        first = first->next;
    }
    void remove_last()
    {
        Node *p = first;
        while (p->next != last)
        {
            p = p->next;
        }
        last = p;
        p->next = nullptr;
    }

    void remove(int power)
    {
        if (is_empty())
            return;
        if (first->val.first == power)
        {
            remove_first();
            return;
        }
        else if (last->val.first == power)
        {
            remove_last();
            return;
        }
        Node *slow = first;
        Node *fast = first->next;
        while (fast && fast->val.first != power)
        {
            fast = fast->next;
            slow = slow->next;
        }
        if (!fast)
        {
            cout << "This element does not exist" << endl;
            return;
        }
    }

```



```

    }
    slow->next = fast->next;
    delete fast;
}

// Рекурсивный подсчёт значения при заданном x по схеме
горнера
float calc_x(float x, Node *p)
{
    if (p->next == nullptr)
    {
        return p->val.second;
    }

    return p->val.second + x * calc_x(x, p->next);
}
};

```

Листинг 3 - Алгоритм пользовательского меню

```

void menu()
{
    cout << "Односвязный список: линейный многочлен n-ой
степени" << endl;
    cout << "Выберите команду из предложенных" << endl;
    cout << "1 - задать список вручную" << endl;
    cout << "2 - сгенерировать список вручную" << endl;
    string comm;
    cin >> comm;
    list lst;
    if (comm == "1")
    {
        cout << "Введите количество элементов: ";
        int size = 0;
        cin >> size;
        vector<float> coefs;
        cout << "Введите коэффициенты многочлена: ";
        for (int i = 0; i < size; i++)
        {
            float el;
            cin >> el;
            coefs.push_back(el);
        }
        lst = create_list(coefs);
        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" << endl;
        cout << "2 - вставить значение перед первым элементом"
        << endl;
        cout << "3 - вычислить значение многочлена при заданном
        x" << endl;
        cout << "4 - удалить значение по ключу (степени)" <<
        endl;
        cout << "5 - выход" << endl;

        cin >> comm;
        if (comm == "1")
        {
            cout << endl;
            lst.print();
            cout << endl;
        }
        else if (comm == "2")
        {
            cout << "Введите коэффициент: ";
            int cskn;
            cin >> cskn;
            lst.insert_first(cskn);
            cout << "Вставка успешна" << endl;
        }
        else if (comm == "3")
        {
            cout << "Введите x: ";
            int x;
            cin >> x;
            cout << "Вычисленная сумма: " << lst.calc_x(x,
            lst.first) << endl;
        }
        else if (comm == "4")
        {
            cout << "Введите степень перед x: ";
            int x;
            cin >> x;
            lst.remove(x);
            cout << "Удаление успешно" << endl;
        }
        else if (comm == "5")

```

```

        {
            cout << "Выход..." << endl;
            return;
        }
    }
}

```

Листинг 4 - Вспомогательные функции

```

int rnd()
{
    random_device dev;
    mt19937 rng(dev());
    uniform_int_distribution<mt19937::result_type> d(0, 9);

    return d(rng);
}

list get_list()
{
    list lst;
    for (int i = 0; i < 11; i++)
    {
        pair<int, float> a = {i, rnd()};
        lst.push_back(a);
    }
    return lst;
}

list create_list(vector<float> coefs)
{
    list lst;
    for (int i = 0; i < coefs.size(); i++)
    {
        pair<int, float> a = {i, coefs[i]};
        lst.push_back(a);
    }
    return lst;
}

```

Листинг 5 - Код основной программы

```

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

```

2.3 Результаты тестирования программы

Результаты запуска представлены на рис. 3, 4.

```
Односвязный список: линейный многочлен n-ой степени
Выберите команду из предложенных
1 - задать список вручную
2 - сгенерировать список вручную
1
Введите количество элементов: 4
Введите коэффициенты многочлена: 1 0 2 5

Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
1

1 + 2x^2 + 5x^3

Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
2
Введите коэффициент: 5
Вставка успешна
Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
1

5 + x + 2x^3 + 5x^3

Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
5
Выход...
```

Рисунок 3 - Результаты тестирования

```

Односвязный список: линейный многочлен n-ой степени
Выберите команду из предложенных
1 - задать список вручную
2 - сгенерировать список
2
Список сгенерирован
Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
1

1 + x + 2x^3 + 9x^4 + 2x^5 + 6x^6 + x^7 + 3x^8 + 2x^9 + 4x^10

Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
4
Введите степень перед x: 7
Удаление успешно
Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
3
Введите x: 1
Вычисленная сумма: 36
Выберите команду из предложенных
1 - вывести список
2 - вставить значение перед первым элементом
3 - вычислить значение многочлена при заданном x
4 - удалить значение по ключу (степени)
5 - выход
5
Выход...

```

Рисунок 4 - Результаты тестирования

3 ВЫВОД

Была проведена работа с односвязным списком и его оптимизации 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.

2. Схема Горнера — Материал из Википедии — свободной энциклопедии [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Схема_Горнера (дата обращения 31.03.2024)