

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ФАКУЛЬТЕТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ НАУК  
ИМЕНИ ПРОФЕССОРА Н.И.ЧЕРВЯКОВА**

**ЛАБОРАТОРНАЯ РАБОТА №15**

Алгоритмизация и программирование

**Список**

**Выполнил студент:**

Сивко Иван Андреевич

студент 2 курса

группа ПМИ-б-о-23-2,

направление подготовки 01.03.02

**Проверил:**

Ассистент кафедры вычислительной  
математики и кибернетики, к.ф.-м.н.,

Черкашина Анастасия Андреевна

## Вариант 9

### Цель:

- Совершенствование навыков разработки программ в среде программирования
- Совершенствование навыков в программировании с использованием указателей
- Исследование процесса формирования элементов связанного списка
- Исследование операций с элементами связанных списков

## Задание 2

*В соответствии с вариантом написать и отладить программу, используя динамическую структуру данных: Односвязный список.*

### 1 Условие

Написать программу, позволяющую с использованием меню и описанных выше операций производить работу со связанным списком, выполняя ввод данных в список из файла, вывод всего списка и обработку данных, хранящихся в списке, при решении задачи согласно варианту и выгрузка списка в файл. Вариант решения задачи без файлов возможен, но максимальная оценка результата – «хорошо».

Записи содержат название издания, газета или журнал, цена экземпляра. Добавлять новые записи так, чтобы сначала располагались журналы, затем газеты.

### 2 Алгоритм/математическая модель:

программа позволят добавлять, показывать, загружать и сохранять в файлы публикации(газеты и журналы). алгоритм выполнения программы следующий:

1. **меню:** программа предлагает пользователю выбор действий через консоль:

- добавить публикацию.
- показать список публикаций.
- загрузить публикации из файла.
- сохранить публикации в файл.
- выйти из программы.

2. **добавить публикацию:**

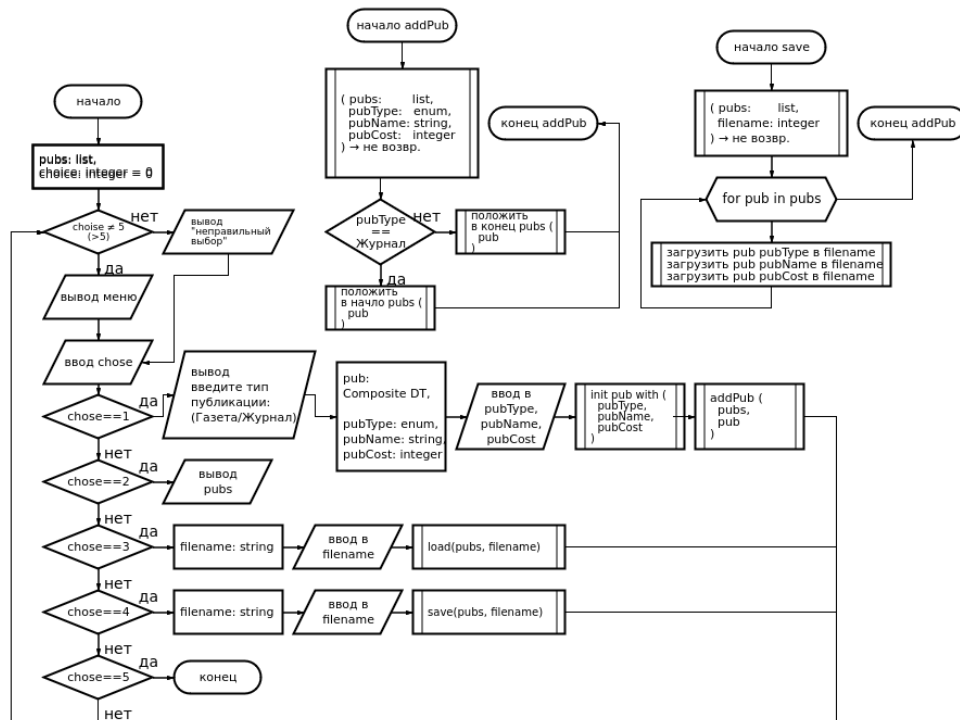
- запрашивает у пользователя тип публикации (газета или журнал), название и цену.
- добавляет публикацию в список в зависимости от типа (журнал — в начало списка, газета — в конец).

3. **показать публикации:** выводит все публикации в списке с указанием их типа, названия и цены.

4. **загрузить публикации из файла:** программа читает данные из файла, ожидая строки с типом публикации, названием и ценой. Эти данные добавляются в список публикаций.

5. **сохранить публикации в файл:** программа сохраняет список публикаций в файл в формате: тип, название, цена.

6. **выход.**



### 3 Диаграмма:

### 4 Код:

```
#include <cstdlib>
#include <cstdint>
#include <limits>
#include <list>
#include <fstream>
#include <sstream>
#include <iostream>
```

```
enum class PublicationType { Newspaper, Magazine };
```

```
enum : size_t { Type, Name, Cost };
```

```
using Publication = std::tuple<PublicationType, std::string, double>;
```

```
std::istream& operator>>(std::istream& is, PublicationType& type) {
    std::string str;
    is >> str;
    if (str == "Газета")
        type = PublicationType::Newspaper;
    else if (str == "Журнал")
        type = PublicationType::Magazine;
    else
        is.setstate(std::ios::failbit);
    return is;
}
```

```
std::ostream& operator<<(std::ostream& out, PublicationType type) {
    return out << (type == PublicationType::Newspaper ? "Газета" : "Журнал");
}
```

```

}

std::ostream& operator<<(std::ostream& out, const std::list<Publication>& pubs) {
    if (pubs.empty())
        return out << "Список пуст.\n";
    for (const auto& pub : pubs)
        out << std::get<Type>(pub) << ": " << std::get<Name>(pub)
            << ", Цена: " << std::get<Cost>(pub) << " монеты.\n";
    return out;
}

void addPublication(std::list<Publication>& pubs, const Publication& pub) {
    if (std::get<Type>(pub) == PublicationType::Magazine)
        pubs.push_front(pub);
    else
        pubs.push_back(pub);
}

bool loadPublications(std::list<Publication>& pubs, const std::string& filename) {
    std::ifstream file{filename};
    if (!file.is_open()) {
        std::cerr << "Ошибка: Не удалось открыть файл " << filename << '\n';
        return false;
    }

    std::string line, pubType, name;
    double price;
    while (std::getline(file >> std::ws, line)) {
        std::istringstream iss{line};
        std::getline(iss >> std::ws, pubType, ',');
        std::getline(iss >> std::ws, name, ',');
        if (!(iss >> price)) {
            std::cerr << "Ошибка: Некорректный формат строки: " << line << '\n';
            return false;
        }
        PublicationType type = pubType == "Журнал" ? PublicationType::Magazine : PublicationType::Newspaper;
        addPublication(pubs, {type, name, price});
    }
    return true;
}

bool savePublications(const std::list<Publication>& pubs, const std::string& filename) {
    std::ofstream file{filename};
    if (!file.is_open()) {
        std::cerr << "Ошибка: Не удалось открыть файл для записи: " << filename << '\n';
        return false;
    }

    for (const auto& pub : pubs) {
        file << (std::get<Type>(pub) == PublicationType::Newspaper ? "Газета" : "Журнал")
            << ", " << std::get<Name>(pub) << ", " << std::get<Cost>(pub) << '\n';
    }
}

```

```

    return true;
}

int main() {
    // std::setlocale(LC_ALL, "ru_RU.UTF-8");
    std::list<Publication> pubs;
    uint32_t choice = 0;

    while (choice != 5) {
        std::cout << "\nМеню:\n"
            << "1. Добавить публикацию\n"
            << "2. Показать публикации\n"
            << "3. Загрузить из файла\n"
            << "4. Сохранить в файл\n"
            << "5. Выход\n"
            << "Ваш выбор: ";
        if (!(std::cin >> choice)) {
            std::cerr << "Ошибка: Некорректный ввод. Ожидается число.\n";
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            continue;
        }

        switch (choice) {
            case 1: {
                PublicationType type;
                std::string name;
                double price;

                std::cout << "Введите тип (Газета или Журнал): ";
                if (!(std::cin >> type)) {
                    std::cerr << "Ошибка: Неверный тип публикации.\n";
                    std::cin.clear();
                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
                    break;
                }

                std::cout << "Введите название: ";
                std::getline(std::cin >> std::ws, name);

                std::cout << "Введите цену: ";
                if (!(std::cin >> price)) {
                    std::cerr << "Ошибка: Неверный ввод цены.\n";
                    std::cin.clear();
                    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
                    break;
                }

                addPublication(pubs, {type, name, price});
                std::cout << "Публикация добавлена.\n";
                break;
            }
        }
    }
}

```

```

    }
    case 2:
        std::cout << pubs;
        break;

    case 3: {
        std::string filename;
        std::cout << "Введите имя файла: ";
        std::getline(std::cin >> std::ws, filename);

        if (loadPublications(pubs, filename))
            std::cout << "Публикации успешно загружены.\n";
        break;
    }

    case 4: {
        std::string filename;
        std::cout << "Введите имя файла: ";
        std::getline(std::cin >> std::ws, filename);

        if (savePublications(pubs, filename))
            std::cout << "Публикации успешно сохранены.\n";
        break;
    }

    case 5:
        std::cout << "Выход из программы.\n";
        break;

    default:
        std::cerr << "Ошибка: Неверный выбор.\n";
        break;
}

return EXIT_SUCCESS;
}

```

source code

## 5 Результат работы программы

```
Меню:
1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход
Ваш выбор: 1
Введите тип (Газета или Журнал): Газета
Введите название: Новая
Введите цену: 100
Публикация добавлена.
```

```
Меню:
1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход
Ваш выбор: 1
Введите тип (Газета или Журнал): Журнал
Введите название: Зеленый
Введите цену: 102
Публикация добавлена.
```

```
Меню:
1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход
Ваш выбор: 1
Введите тип (Газета или Журнал): Газета
Введите название: Фиолетовая
Введите цену: 20
Публикация добавлена.
```

```
Меню:
1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход
Ваш выбор: 2
Журнал: Зеленый, Цена: 102 монеты.
Газета: Новая, Цена: 100 монеты.
Газета: Фиолетовая, Цена: 20 монеты.
```

```
Меню:
1. Добавить публикацию
2. Показать публикации
3. Загрузить из файла
4. Сохранить в файл
5. Выход
Ваш выбор: 5
Выход из программы.
```

## Задание 3

### 1 Условие:

Дана запись многочлена (от переменной  $x$ ) произвольной степени с целыми коэффициентами, причем они могут быть и неупорядочены по степеням  $x$ , а одночлены одной и той же степени могут повторяться (например,  $-8x^4+3x^5+2x^4-1$ ). Требуется написать программу, которая выполняет следующие функции:

- процедуру **ввод(P)**, которая считывает с клавиатуры набор чисел и формирует список-многочлен P.
- удаляет указанный элемент из списка.
- вставляет элемент в произвольную позицию списка
- процедуру, которая приведет подобные и сформирует новый многочлен.
- процедуру **вывод(P,x)**, которая печатает многочлен p как многочлен от переменной x.

### 2 Алгоритм / Математическая модель

Программа предназначена для работы с многочленами (полиномами) и позволяет добавлять, удалять и отображать члены полинома. Алгоритм выполнения программы следующий:

1. **Меню:** программа предлагает пользователю выбор действий через консоль:

- Ввести новый член полинома.
- Удалить член полинома по указанной степени.
- Вывести полином на экран.
- Выйти из программы.

2. **Ввести новый член полинома:**

- Запрашивает у пользователя многочлен или одночлен в строковом формате.
- Строка анализируется на наличие коэффициента и степени переменной.
- Если степень не указана, то по умолчанию считается, что степень равна 1.
- Добавляет новый член полинома в список. Если степень уже существует, коэффициенты складываются.

3. **Удалить член полинома по указанной степени:**

- Программа запрашивает у пользователя степень члена, который нужно удалить.
- Если в полиноме существует член с указанной степенью, он удаляется из списка.

4. **Вывести полином:**

- Программа выводит все члены полинома в формате: коэффициент, переменная, степень.
- Если полином пуст, выводится сообщение "Полином пуст".

5. **Выход:** программа завершает свою работу.

## 2.1 Математическая модель

1. **Член полинома** (терм) представляется как пара:

$$T(x) = (a, n)$$

где  $a$  — коэффициент, а  $n$  — степень переменной  $x$ .

2. **Полином** состоит из списка членов, каждый из которых представляет собой пару  $(a, n)$ , где  $a$  — коэффициент, а  $n$  — степень.

3. **Процесс парсинга строки:** - Если строка содержит символ 'x', то извлекаются коэффициент и степень. - Если 'x' отсутствует, то это константа (степень 0, коэффициент равен числу). - Если степень не указана явно, то она считается равной 1.

4. **Алгоритм добавления члена в полином:** - Если степень уже присутствует в полиноме, то коэффициенты складываются. - Если степень отсутствует, новый член вставляется в нужное место, поддерживая упорядоченность полинома по убыванию степени.

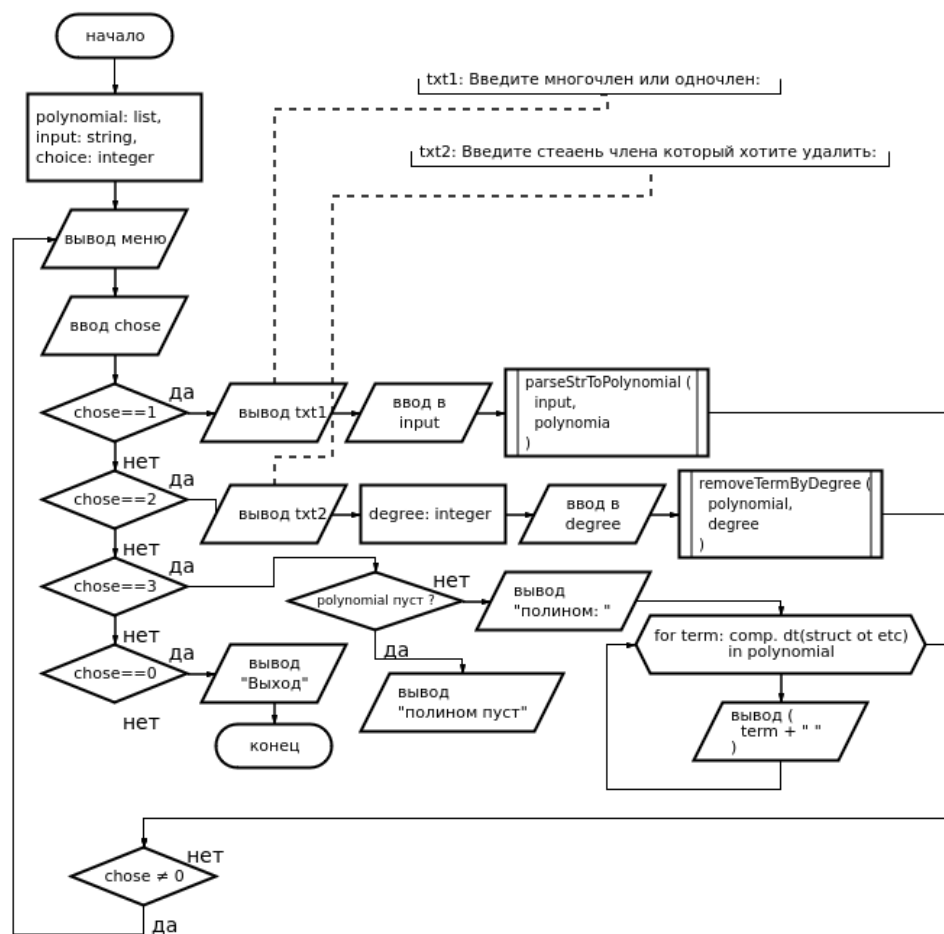
5. **Удаление члена по степени:** - Процесс удаления члена полинома с указанной степенью из списка.

6. **Вывод полинома:** - Все члены полинома выводятся в читаемом виде, с учетом знака коэффициента, степени и форматирования.

Таким образом, программа позволяет эффективно манипулировать многочленами, обеспечивая функции для их редактирования и отображения.



### 3 Диаграмма:



### 4 Код:

```

#include <cstdlib>      // for EXIT_SUCCESS, abs
#include <cstdint>      // for int32_t
#include <limits>
#include <list>         // std::list
#include <string>        // std::string
#include <string_view>   // std::string_view
#include <utility>       // for std::pair
#include <iostream>     // std::cout, std::cin

namespace {
    constexpr auto DEBUG_PRINT{ false };
}

using Term = std::pair<int32_t, int32_t>;

Term parseStrToTerm(std::string_view term) {
    int32_t cfcnt { 1 },
        degree{ 1 };
    size_t xPos{ term.find('x') };
    if (xPos == std::string_view::npos) { // if x not exist

```

```

        cfcnt = std::stoi(std::string(term));
        degree = 0;
    } else {
        if (xPos > 0)
            switch (term[xPos-1]) {
                case '-':
                    cfcnt = -1;
                    break;
                case '+':
                    cfcnt = +1;
                    break;
                default:
                    cfcnt = std::stoi(std::string(term.substr(0, xPos)));
            }
        size_t powPos{ term.find('^', xPos) };
        if (powPos == std::string_view::npos || cfcnt == 0)
            degree = 1;
        else
            degree = std::stoi(std::string(term.substr(powPos + 1)));
    }
    return { cfcnt, degree };
}

void insertTerm(std::list<Term>& poly, const Term& term) {
    for (auto it{ poly.begin() }; it != poly.end(); ++it) {
        if (term.second == it->second) {
            it->first += term.first;
            if (it->first == 0)
                poly.erase(it);
            return;
        } else if (term.second > it->second) {
            poly.insert(it, term);
            return;
        }
    }
    poly.push_back(term); // if term has the lowest degree put to the end
}

void removeTermByDegree(std::list<Term>& poly, int32_t degree) {
    for (auto it{ poly.begin() }; it != poly.end(); ++it)
        if (it->second == degree) {
            poly.erase(it);
            return;
        }
}

void parseStrToPolynomial(std::string_view str, std::list<Term>& poly) {
    size_t start {0},
           len    { str.size() };
    while (start < len) {
        size_t next{ str.find_first_of("+-", start + 1) };
        if (next == std::string_view::npos)

```

```

        next = len;
        insertTerm(poly, parseStrToTerm(str.substr(start, next - start)));
        start = next;
    }
}

std::ostream& operator<<(std::ostream& out, const Term& term) {
if constexpr (! DEBUG_PRINT) {
    out << (term.first < 0? '-' : '+');
    if (term.first == 0)
        return out << 0;
    if (std::abs(term.first) != 1 || term.second == 0)
        out << std::abs(term.first);
    if (term.second > 0) {
        out << 'x';
        if (term.second != 1)
            out << '^' << term.second;
    }
}
else {
    out << "{ " << term.first << ", " << term.second << " }";
}
    return out;
}

int main() {
    std::list<Term> polynomial;
    std::string input;
    uint32_t choice;
    do {
        std::cout << "\nМеню:\n"
            "1. Ввести новый член полинома\n"
            "2. Удалить член по указанной степени\n"
            "3. Вывести полином\n"
            "0. Выйти\n"
            "Выберите: ";
        if (!(std::cin >> choice)) {
            std::cerr << "Ошибка: Некорректный ввод. Ожидается число.\n";
            std::cin.clear();
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
            continue;
        }
        switch(choice) {
            case 1:
                std::cout << "Введите многочлен или одночлен: ";
                std::getline(std::cin >> std::ws, input);
                parseStrToPolynomial(input, polynomial);
                break;
            case 2:
                std::cout << "Введите степень члена который хотите удалить: ";
                int32_t degree;
                std::cin >> degree;

```

```

        removeTermByDegree(polynomial, degree);
        break;
    case 3:
        if (polynomial.empty())
            std::cout << "Полином пуст\n";
        else {
            std::cout << "Полином:\n";
            for (const auto& term : polynomial)
                std::cout << term << " ";
            std::cout << '\n';
        }
        break;
    case 0:
        std::cout << "Выход\n";
        break;
    default:
        std::cout << "Неизвестный ввод, Выберите из доступных: 1, 2, 3, 0\n";
        break;
    }
} while(choice);
return EXIT_SUCCESS;
}

```

source code

## 5 Результат работы программы

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 1

Введите многочлен или одночлен:  $2+3+4x^5+2x^2+x-x^4$

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 3

Полином:

$+4x^5 -x^4 +2x^2 +x +5$

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 1

Введите многочлен или одночлен:  $3x^6-x^7+1-x^2$

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 3

Полином:

$-x^7 +3x^6 +4x^5 -x^4 +x^2 +x +6$

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 2

Введите степень члена который хотите удалить: 5

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 3

Полином:

$-x^7 +3x^6 -x^4 +x^2 +x +6$

Меню:

1. Ввести новый член полинома
2. Удалить член по указанной степени
3. Вывести полином
0. Выйти

Выберите: 0

Выход