Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №6.**

**«АТД. Контейнеры»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

1. Определить класс-контейнер.
2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.
3. Перегрузить операции, указанные в варианте.
4. Реализовать класс-итератор. Реализовать с его помощью операции последовательного доступа.
5. Написать тестирующую программу, иллюстрирующую выполнение операций.

Класс-контейнер СПИСОК с ключевыми значениями типа int.  
Реализовать операции:  
[] - доступа по индексу;  
int() - определение размера списка;

* вектор - умножение элементов списков a[i]\*b[i];  
  +n - переход вправо к элементу с номером n (с помощью класса-итератора).

**Анализ задачи**

**1. Архитектура решения**  
Программа реализует двусвязный список через три основных класса:

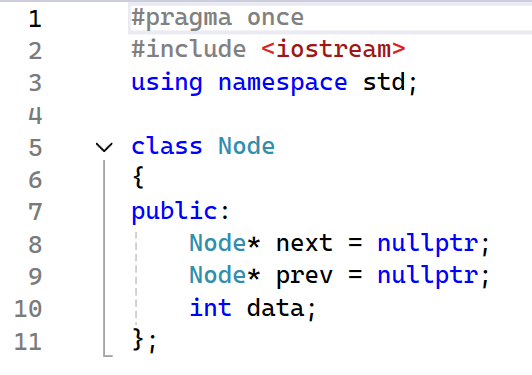
* Node - базовый элемент списка
* List - класс-контейнер
* Iterator - класс для последовательного доступа

**2. Реализованные операции**

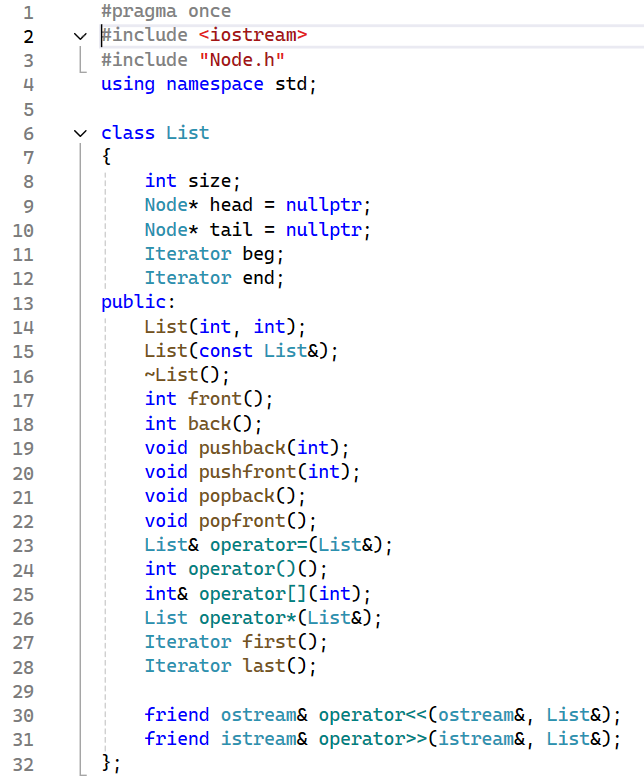
1. **Базовые операции контейнера**:
   * Конструкторы (с параметрами, копирования)
   * Деструктор (корректное освобождение памяти)
   * Добавление/удаление элементов (pushback, pushfront, popback, popfront)
2. **Перегруженные операторы**:
   * [] - доступ по индексу
   * () - получение размера списка
   * \* - поэлементное умножение списков
   * << и >> - ввод/вывод
3. **Итераторные операции**:
   * ++/-- - перемещение между элементами
   * +n/-n - переход на n позиций
   * \* - доступ к данным текущего элемента

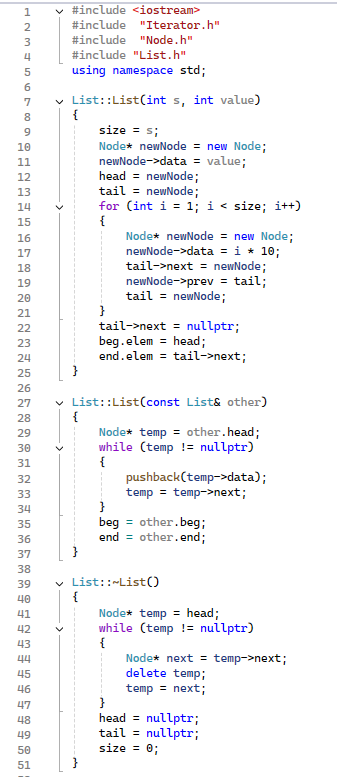
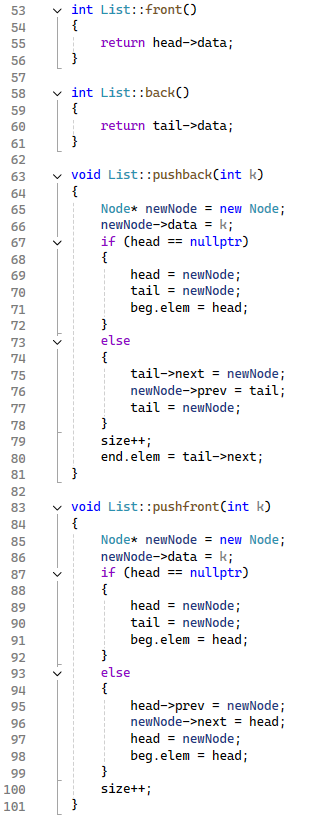
**Код программы**

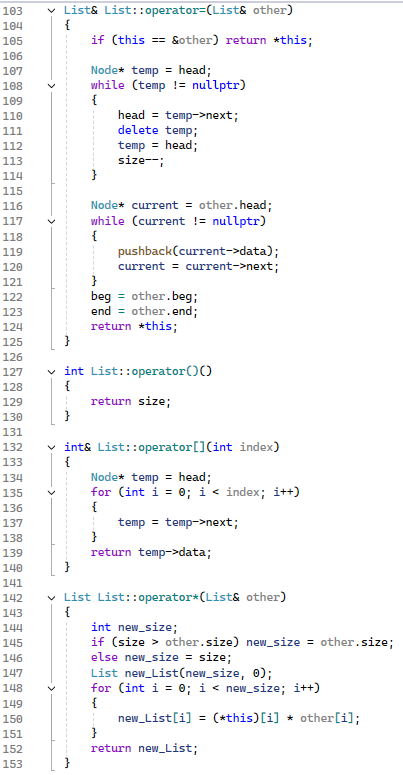
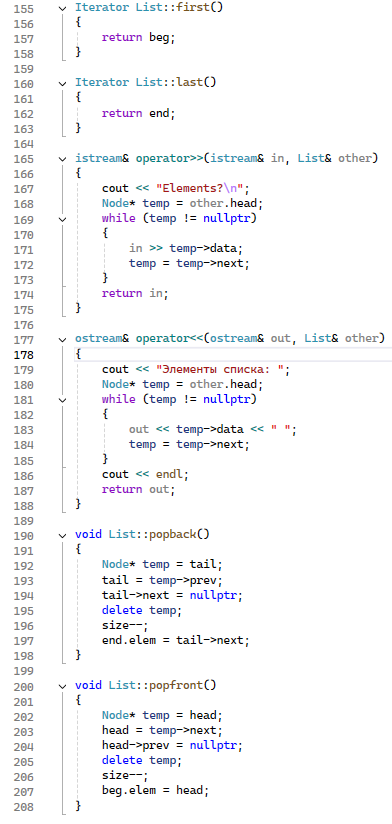
**Node.h**



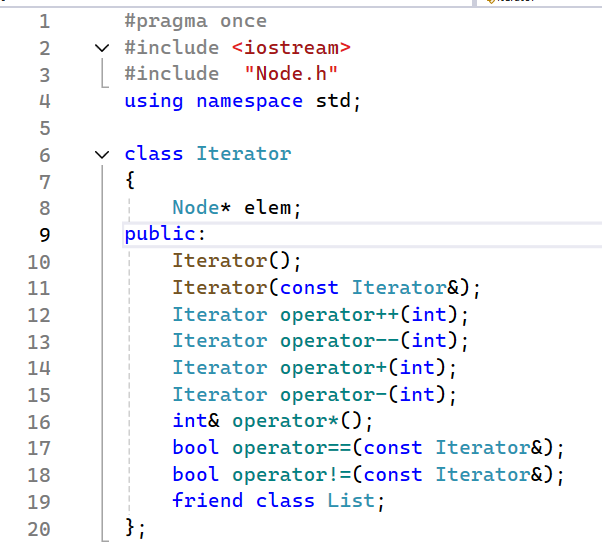
**List.h**



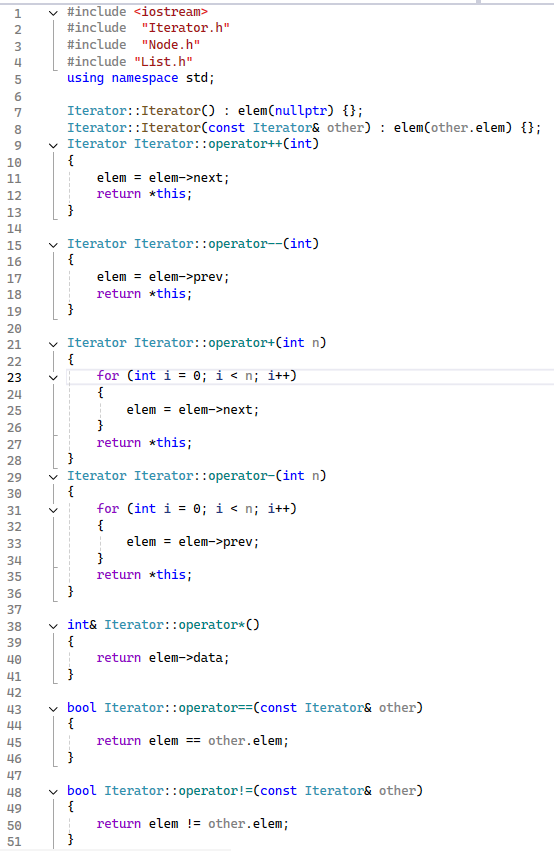
**List.cpp**



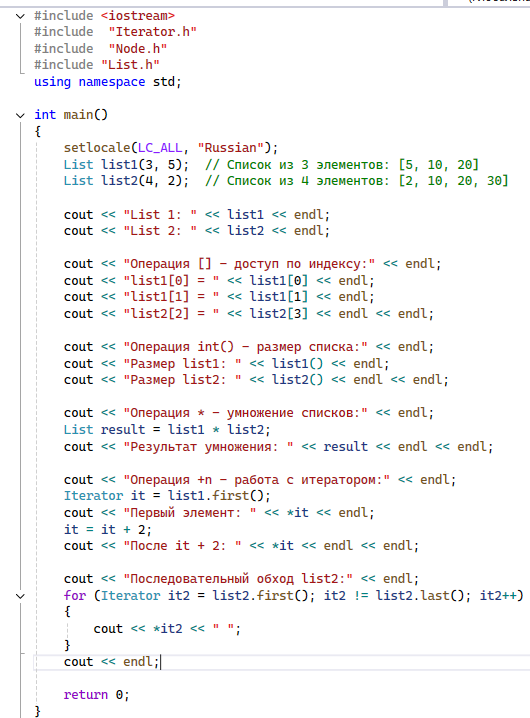
**Iterator.h**

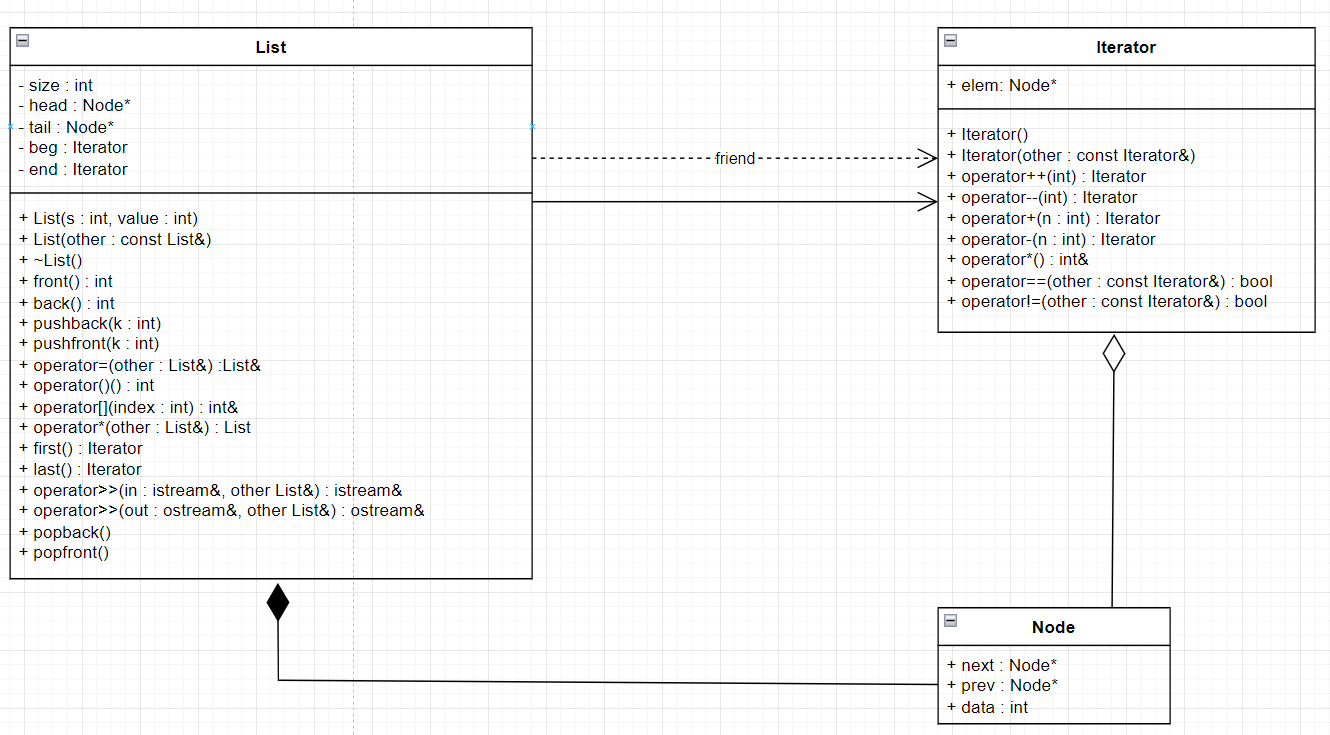


**Iterator.cpp**



**main.cpp**



**Диаграмма классов**

**Контрольные вопросы**

**1. Что такое абстрактный тип данных? Привести примеры АТД.**  
Ответ: АТД - это модель данных, определяющая поведение независимо от реализации. Примеры: стек, очередь, список, множество.

**2. Привести примеры абстракции через параметризацию.**  
Ответ: Шаблоны классов в C++ (например, vector<T>) где T - параметр типа.

**3. Привести примеры абстракции через спецификацию.**  
Ответ: Интерфейсы (чисто виртуальные классы), где задается только поведение без реализации.

**4. Что такое контейнер? Привести примеры.**  
Ответ: Объект для хранения других объектов. Примеры: массив, список, словарь, множество.

**5. Какие группы операций выделяют в контейнерах?**  
Ответ: Вставка, удаление, поиск, доступ, обход, проверка размера.

**6. Какие виды доступа к элементам контейнера существуют? Привести примеры.**  
Ответ:

* Последовательный (списки)
* Прямой/индексный (массивы)
* По ключу (словари)

**7. Что такое итератор?**  
Ответ: Объект для обхода элементов контейнера без раскрытия его внутренней структуры.

**8. Каким образом может быть реализован итератор?**  
Ответ: Как класс с операциями ++, --, \*, ==, != для доступа к элементам.

**9. Каким образом можно организовать объединение контейнеров?**  
Ответ: Через методы или создание нового контейнера с элементами обоих.

**10. Какой доступ к элементам предоставляет контейнер, состоящий из элементов «ключ-значение»?**  
Ответ: Доступ по уникальному ключу.

**11. Как называется контейнер, в котором вставка и удаление элементов выполняется на одном конце контейнера?**  
Ответ: Стек

**12. Какой из объектов (a,b,c,d) является контейнером?**  
Ответ: d) int mas[100] (массив).

**13. Какой из объектов (a,b,c,d) не является контейнером?**  
Ответ: d) int mas (одиночная переменная).

**14. Контейнер реализован как динамический массив, в нем определена операция доступ по индексу. Каким будет доступ к элементам контейнера?**  
Ответ: Прямой (индексный) доступ за O(1).

**15. Контейнер реализован как линейный список. Каким будет доступ к элементам контейнера?**  
Ответ: Последовательный доступ за O(n).