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 №7.**

**«Шаблоны классов»**

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 шаблон класса-контейнера (см. лабораторную работу №6).
2. Реализовать конструкторы, деструктор, операции ввода-вывода, операцию присваивания.
3. Перегрузить операции, указанные в варианте.
4. Инстанцировать шаблон для стандартных типов данных (int, float, double).
5. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы стандартных типов данных.
6. Реализовать пользовательский класс (см. лабораторную работу №3).
7. Перегрузить для пользовательского класса операции ввода-вывода.
8. Перегрузить операции необходимые для выполнения операций контейнерного класса.
9. Инстанцировать шаблон для пользовательского класса.
10. Написать тестирующую программу, иллюстрирующую выполнение операций для контейнера, содержащего элементы пользовательского класса.

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];

Пользовательский класс Pair (пара чисел). Пара должна быть представлено двумя полями: типа int для первого числа и типа double для второго. Первое число при выводе на экран должно быть отделено от второго числа двоеточием.

**Анализ задачи**

**1. Архитектура решения**  
Программа реализует шаблонный двусвязный список с использованием:

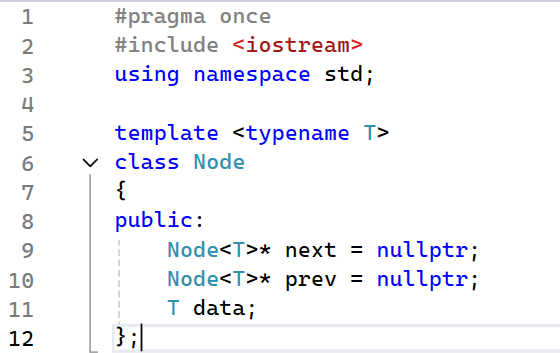
* Node<T> - узел списка (хранит данные и указатели)
* List<T> - основной контейнер
* Iterator<T> - механизм обхода элементов
* Pair<T1,T2> - пользовательский тип для демонстрации

**2. Ключевые особенности**

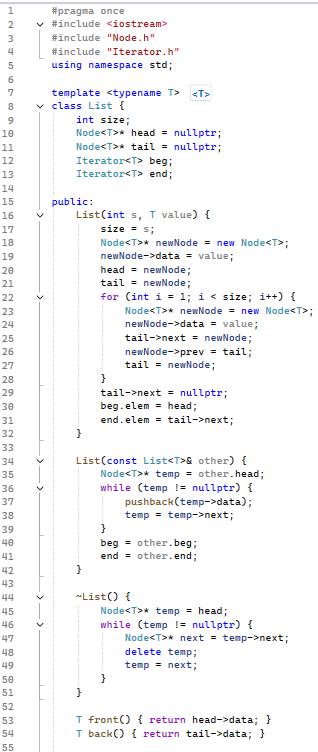
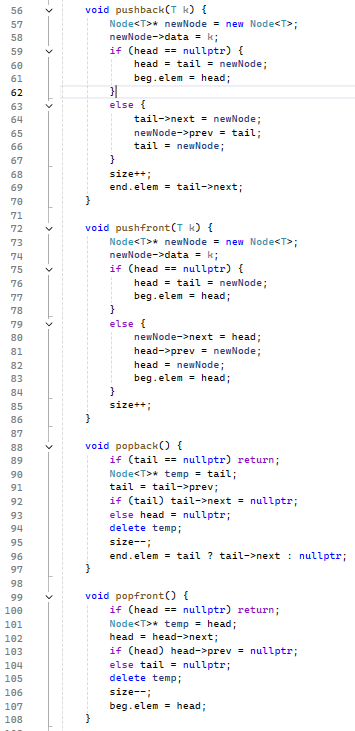
* Полная шаблонизация всех классов
* Поддержка стандартных операций:
  + Добавление/удаление элементов
  + Доступ по индексу
  + Поэлементные операции
* Реализация итератора с перегруженными операторами
* Полиморфная работа с разными типами данных

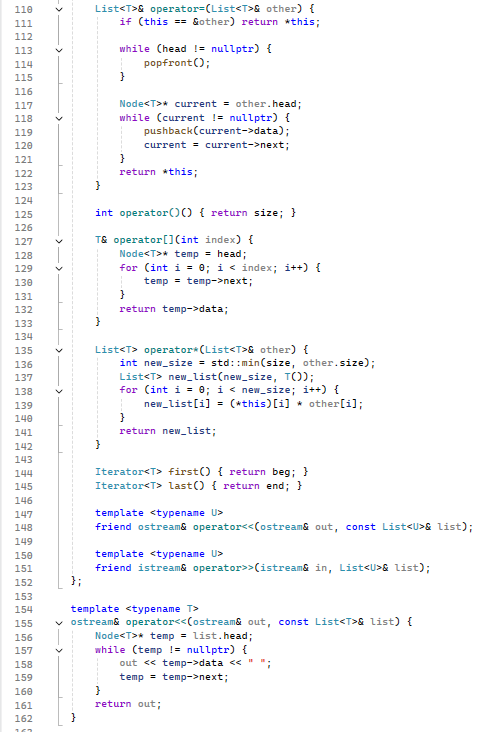
**Код программы**

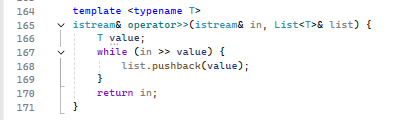
**Node.h**



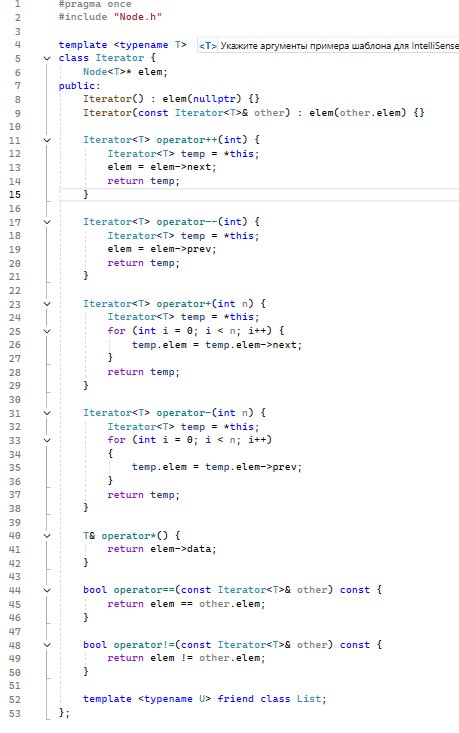
**List.h**



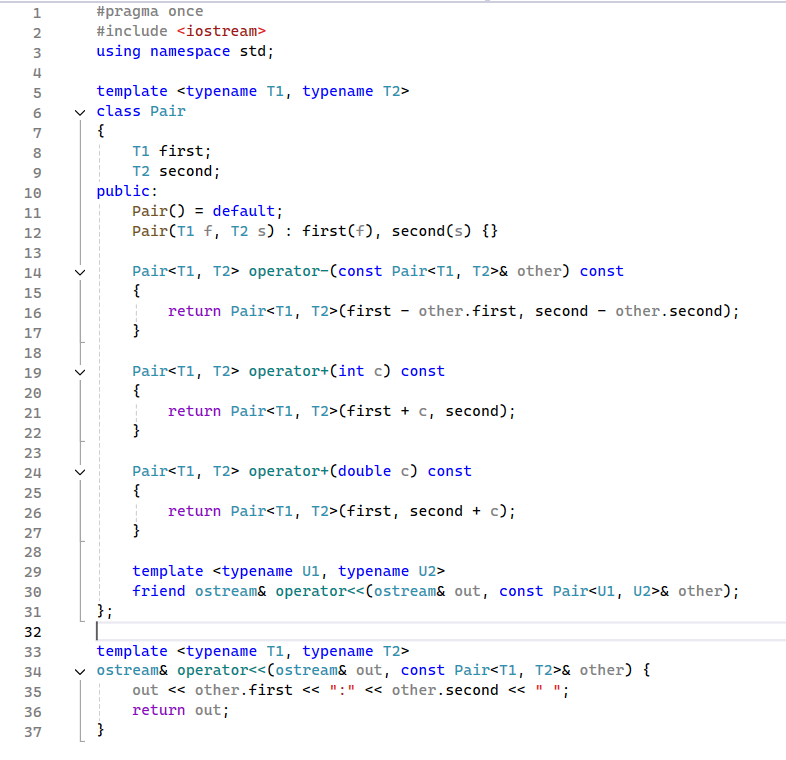




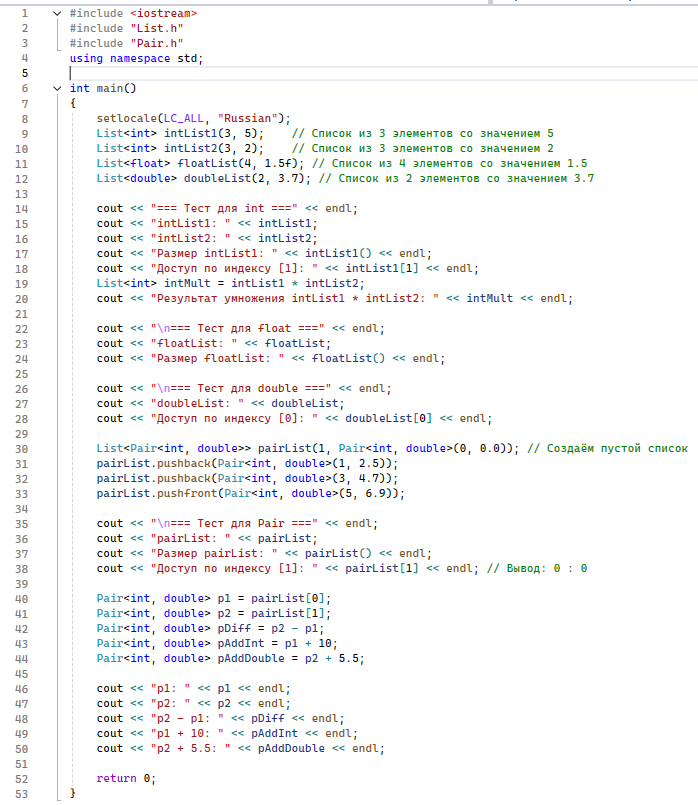
**Iterator.h**



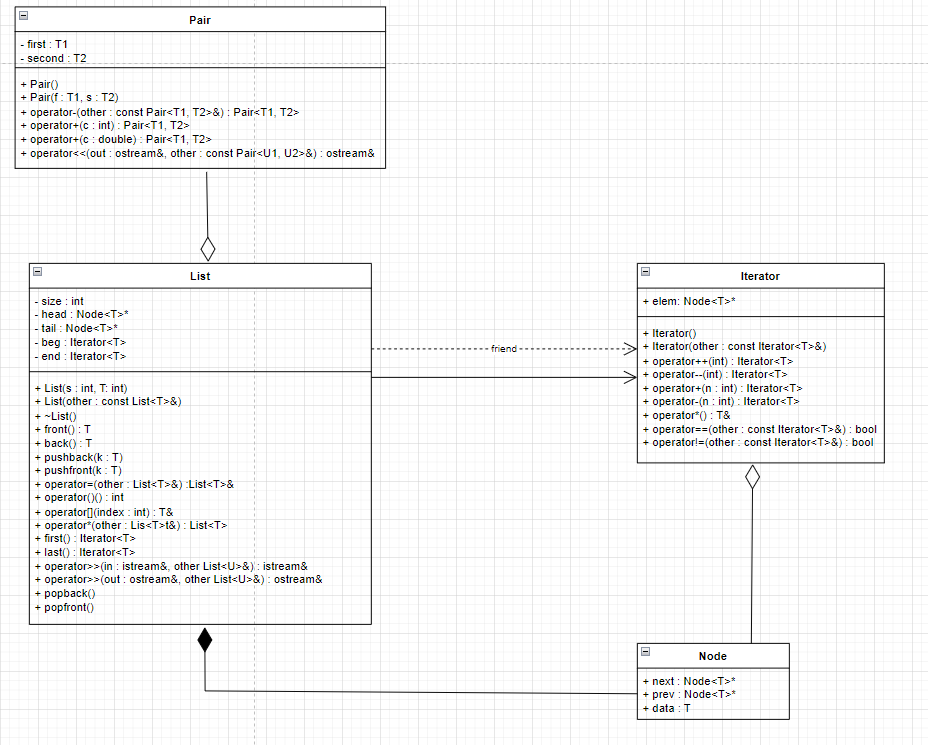
**Pair.h**



**main.cpp**



**Диаграмма классов**



**Контрольные вопросы**

**1. В чем смысл использования шаблонов?**  
Ответ: Шаблоны позволяют создавать обобщённый код, работающий с разными типами данных без их явного указания на этапе написания кода.

**2. Каковы синтаксис/семантика шаблонов функций?**  
Ответ:  
template <typename T>

T функция(T параметр) { ... }

**3. Каковы синтаксис/семантика шаблонов классов?**  
Ответ:  
template <typename T>

class ИмяКласса { ... };

**4. Что такое параметры шаблона функции?**  
Ответ: Это типы или значения, указываемые в угловых скобках, которые определяют поведение шаблона.

**5. Основные свойства параметров шаблона функции:**

* Могут быть типами (typename/class) или конкретными значениями
* Поддерживают несколько параметров
* Можно указывать параметры по умолчанию

**6. Как записывать параметр шаблона?**  
Ответ:

template <typename T> // для типов

template <int N> // для значений

**7. Можно ли перегружать параметризованные функции?**  
Ответ: Да, шаблонные функции можно перегружать как другими шаблонными, так и обычными функциями.

**8. Основные свойства параметризованных классов:**

* Могут содержать методы-шаблоны
* Позволяют создавать обобщённые контейнеры

**9. Все ли компонентные функции параметризованного класса являются параметризованными?**  
Ответ: Нет, класс может содержать как шаблонные, так и обычные методы.

**10. Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?**  
Ответ: Да, если они объявлены внутри шаблонного класса, они автоматически становятся шаблонными.

**11. Могут ли шаблоны классов содержать виртуальные компонентные функции?**  
Ответ: Да, но сама функция не может быть шаблонной - только обычной виртуальной функцией внутри шаблонного класса.

**12. Как определяются компонентные функции параметризованных классов вне определения шаблона класса?**  
Ответ:

template <typename T>

тип Класс<T>::метод(параметры) { ... }

**13. Что такое инстанцирование шаблона?**  
Ответ: Процесс создания конкретной реализации шаблона для определённого типа или значения параметра.

**14. На каком этапе происходит генерирование определения класса по шаблону?**  
Ответ: На этапе компиляции, когда компилятор встречает использование шаблона с конкретными параметрами.