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 №12.**

**«Ассоциативные контейнеры библиотеки STL»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

**Задача 1: Работа с**multiset<double>

1. Создать ассоциативный контейнер multiset для хранения элементов типа double.
2. Заполнить контейнер данными (например, случайными числами или вводом с клавиатуры).
3. Реализовать глобальные функции для выполнения операций:
   * **Задание 1:** Найти среднее арифметическое всех элементов и добавить его в конец контейнера.
   * **Задание 2:** Удалить элементы, значения которых попадают в заданный диапазон [a, b].
   * **Задание 3:** К каждому элементу добавить сумму минимального и максимального элементов контейнера.
4. Вывести результаты после каждой операции.

**Задача 2: Работа с пользовательским** типом Pair

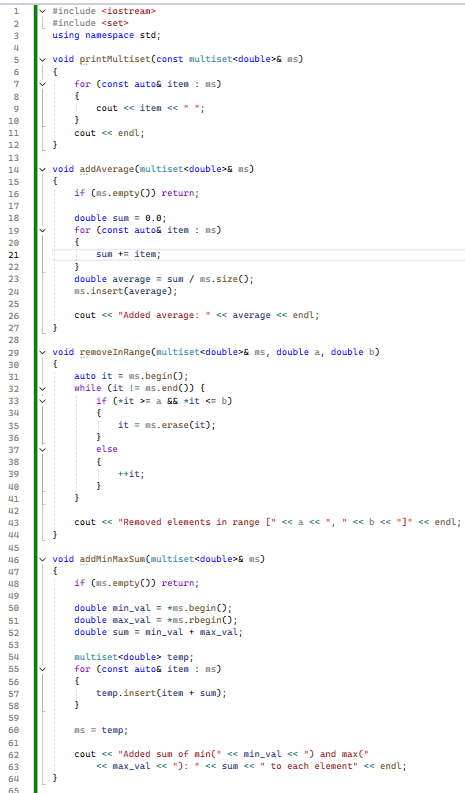
1. Создать класс Pair (из лабораторной работы №3) с полями:
   * first — тип int;
   * second — тип double.
2. Перегрузить для класса Pair необходимые операции (<, ==, +, << и др.).
3. Создать multiset<Pair> и заполнить его элементами.
4. Реализовать глобальные функции для тех же заданий, что и в Задаче 1, но с учётом особенностей типа Pair:
   * **Задание 1:** Среднее арифметическое вычисляется отдельно для first и second.
   * **Задание 2:** Удаление происходит по полю first в заданном диапазоне.
   * **Задание 3:** Добавление суммы min + max применяется только к полю second.
5. Вывести результаты.

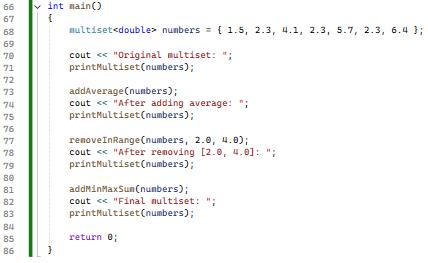
**Задача 3: Параметризированный класс**MyList

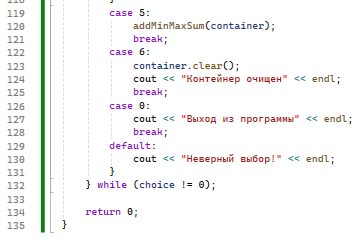
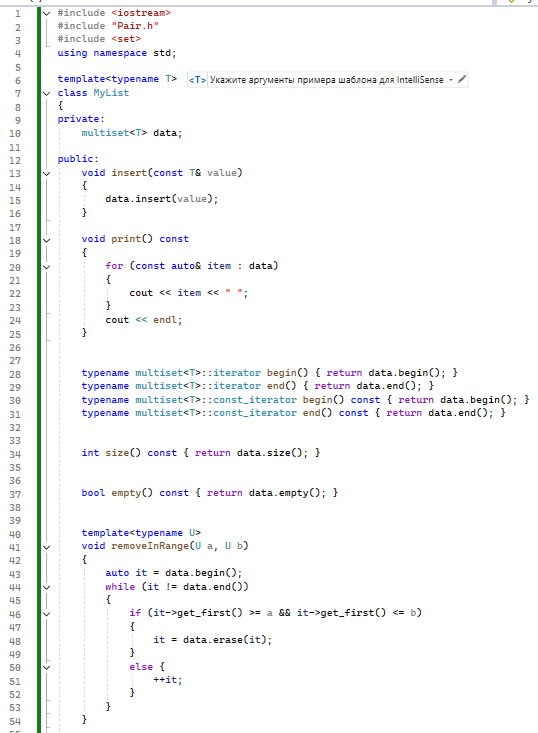
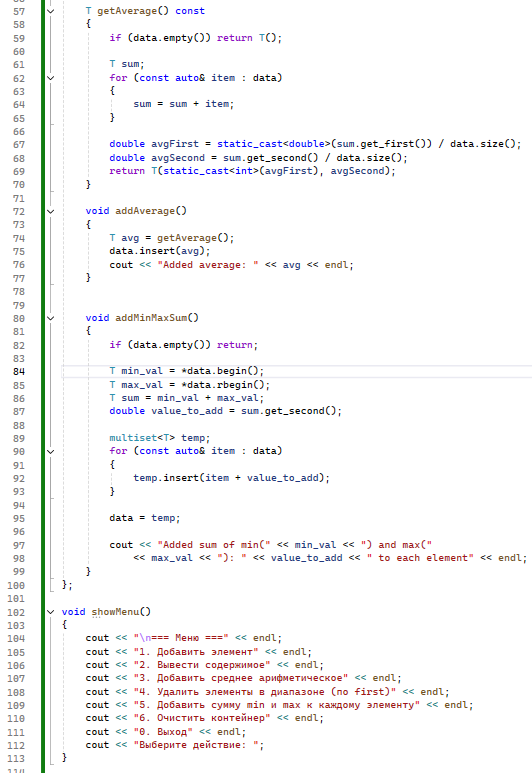
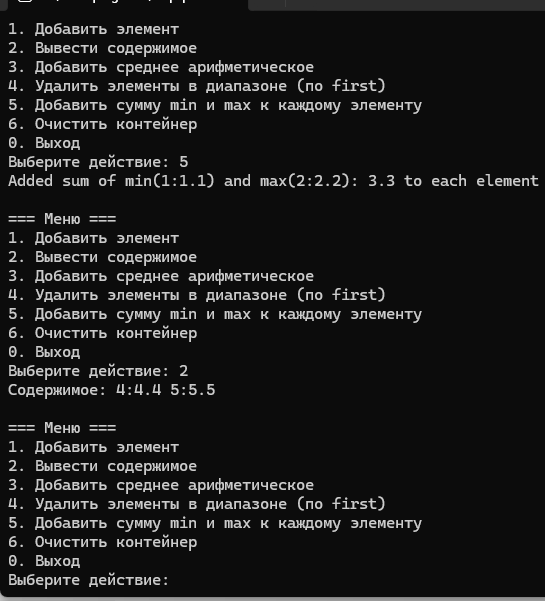
1. Реализовать шаблонный класс MyList<T> на основе контейнера multiset (или list, если в лабораторной №7 использовался список).
2. Класс должен содержать:
   * Поле data типа multiset<T>.
   * Методы для вставки, удаления и доступа к элементам.
3. Добавить методы для выполнения заданий:
   * **Задание 1:** void addAverage().
   * **Задание 2:** void removeInRange(T a, T b).
   * **Задание 3:** void addMinMaxSum().

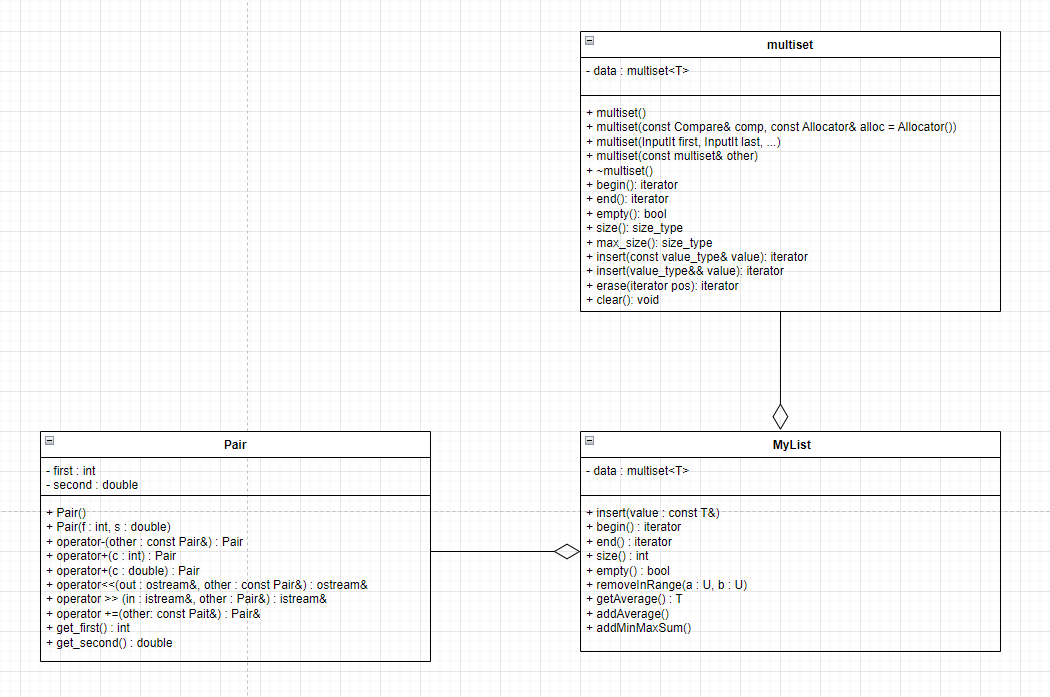
**Код программы**

**Main.cpp**





**Контрольные вопросы**

1. **Что представляет собой ассоциативный контейнер?**  
   Ответ: Контейнер, хранящий элементы в виде пар "ключ-значение" или только ключей, с быстрым поиском по ключу.
2. **Перечислить ассоциативные контейнеры библиотеки STL.**  
   Ответ: map, multimap, set, multiset, unordered\_map, unordered\_set, unordered\_multimap, unordered\_multiset.
3. **Каким образом можно получить доступ к элементам ассоциативного контейнера?**  
   Ответ: Через ключи (для map), итераторы, методы find(), at(), или оператор [] (только для map).
4. **Привести примеры методов, используемых в ассоциативных контейнерах.**  
   Ответ: insert(), erase(), find(), size(), empty(), clear(), begin(), end().
5. **Каким образом можно создать контейнер map? Привести примеры.**

map<int, string> m1; // Пустой map

map<string, double> m2 = {{"pi", 3.14}, {"e", 2.71}}; // С инициализацией

1. **Каким образом упорядочены элементы в контейнере map по умолчанию? Как изменить порядок на обратный?**

Ответ: По возрастанию ключей. Для обратного порядка:

map<int, string, greater<int>> m;

1. **Какие операции определены для контейнера map?**  
   Ответ: Вставка (insert), удаление (erase), поиск (find, count), доступ (at, []), обход (итераторы).
2. **Написать функцию для добавления элементов в контейнер map с помощью функции make\_pair().**

void addPair(map<int, string>& m, int key, string value)

{

m.insert(make\_pair(key, value));

}

1. **Написать функцию для добавления элементов в контейнер map с помощью операции прямого доступа [].**

void add (map<int, string>& m, int key, string value)

{

m[key] = value;

}

1. **Написать функцию для печати контейнера map с помощью итератора.**

void printMap(const map<int, string>& m)

{

for (auto it = m.begin(); it != m.end(); ++it)

cout << it->first << ": " << it->second << endl;

}

1. **Написать функцию для печати контейнера map с помощью операции прямого доступа [].**  
   Ответ:

template<typename KeyType, typename ValueType>

void printMap (const map<KeyType, ValueType>& m)

{

map<KeyType, ValueType> temp = m;

for (const auto& pair : temp) {

// Используем оператор [] для доступа к значению

cout << "Key: " << pair.first << ", Value: " << temp[pair.first] << endl;

}

}

1. **Чем отличаются контейнеры map и multimap?**  
   Ответ: map хранит уникальные ключи, multimap разрешает дубликаты ключей.
2. **Что представляет собой контейнер set?**  
   Ответ: Коллекция уникальных упорядоченных ключей без значений.
3. **Чем отличаются контейнеры map и set?**  
   Ответ: map хранит пары "ключ-значение", set — только ключи.
4. **Каким образом можно создать контейнер set? Привести примеры.**

set<int> s1; // Пустой set

set<string> s2 = {"apple", "banana"}; // С инициализацией

1. **Каким образом упорядочены элементы в контейнере set по умолчанию? Как изменить порядок на обратный?**

Ответ: По возрастанию. Для обратного порядка:

set<int, greater<int>> s;

1. **Какие операции определены для контейнера set?**  
   Ответ: insert(), erase(), find(), count(), size(), empty().
2. **Написать функцию для добавления элементов в контейнер set.**

void addToSet(set<int>& s, int value) {

s.insert(value);

}

1. **Написать функцию для печати контейнера set.**

void printSet(const set<int>& s) {

for (auto it = s.begin(); it != s.end(); ++it)

cout << \*it << " ";

}

1. **Чем отличаются контейнеры set и multiset?**  
   Ответ: set хранит уникальные элементы, multiset разрешает дубликаты.