

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
Пермский национальный исследовательский политехнический
университет
Электротехнический факультет
Кафедра информационных технологий и автоматизированных систем

Лабораторная работа №12

" Ассоциативные контейнеры библиотеки STL "

Вариант: 12

Выполнил студент ИВТ-24-26:
Шишкин Максим Григорьевич

(дата, подпись)

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

(дата, подпись)

Пермь 2025

Содержание

1 Постановка задачи	3
2 Код на C++	4-6
3 Результаты работы	7
4 UML-диаграмма классов	7
5 Ответы на контрольные вопросы	7-8
6 Ссылка на github.....	8

1 Постановка задачи

Задача 1.

1. Создать ассоциативный контейнер.
2. Заполнить его элементами стандартного типа (тип указан в варианте).
3. Добавить элементы в соответствии с заданием
4. Удалить элементы в соответствии с заданием.
5. Выполнить задание варианта для полученного контейнера.
6. Выполнение всех заданий оформить в виде глобальных функций.

Задача 2.

1. Создать ассоциативный контейнер.
2. Заполнить его элементами пользовательского типа (тип указан в варианте). Для пользовательского типа перегрузить необходимые операции.
3. Добавить элементы в соответствии с заданием
4. Удалить элементы в соответствии с заданием.
5. Выполнить задание варианта для полученного контейнера.
6. Выполнение всех заданий оформить в виде глобальных функций.

Задача 3

1. Создать параметризованный класс, используя в качестве контейнера ассоциативный контейнер.
2. Заполнить его элементами.
3. Добавить элементы в соответствии с заданием
4. Удалить элементы в соответствии с заданием.
5. Выполнить задание варианта для полученного контейнера.
6. Выполнение всех заданий оформить в виде методов параметризованного класса.

Вариант:

Задача 1

1. Контейнер - map
2. Тип элементов - int

Задача 2

Тип элементов Pair (см. лабораторную работу №3).

Задача 3

Параметризованный класс – Список (см. лабораторную работу №7)

Задание 3	Задание 4	Задание 5
Найти среднее арифметическое и добавить его на заданную позицию контейнера	Найти элементы ключами из заданного диапазона и удалить их из контейнера	Из каждого элемента вычесть среднее арифметическое контейнера.

2 Код на C++

```
Pair.h    Pair.cpp    MapFunctions.h    MapFunctions.cpp    MyMap.h    main.cpp
12лабаперделка (Глобальная область)

1  #include <iostream>
2  #include <map>
3  #include "Pair.h"
4  #include "MapFunctions.h"
5  #include "MyMap.h"
6
7  int main() {
8      setlocale(LC_ALL, "Russian");
9
10     std::cout << "\n--- Задача 1 (map<int, int>) ---\n";
11     std::map<int, int> m1 = { {1, 10}, {2, 20}, {3, 30} };
12     insert_average(m1, 1);
13     delete_in_range(m1, 1, 2);
14     subtract_average(m1);
15     for (auto& kv : m1)
16         std::cout << "Ключ: " << kv.first << ", Значение: " << kv.second << std::endl;
17
18     std::cout << "\n--- Задача 2 (map<int, Pair>) ---\n";
19     std::map<int, Pair> m2 = { {1, {1, 1.1}}, {2, {2, 2.2}}, {3, {3, 3.3}} };
20     insert_average(m2, 1);
21     delete_in_range(m2, 1, 2);
22     subtract_average(m2);
23     for (auto& kv : m2)
24         std::cout << "Ключ: " << kv.first << ", Значение: " << kv.second << std::endl;
25
26     std::cout << "\n--- Задача 3 (MyMap<Pair>) ---\n";
27     MyMap<Pair> m3;
28     m3.insert(1, { 1, 1.1 });
29     m3.insert(2, { 2, 2.2 });
30     m3.insert(3, { 3, 3.3 });
31     m3.insert_average(1);
32     m3.delete_in_range(1, 2);
33     m3.subtract_average();
34     m3.print();
35
36     return 0;
37 }
```

```
Pair.h    Pair.cpp    MapFunctions.h    MapFunctions.cpp    MyMap.h    main.cpp
12лабаперделка (Глобальная область)

1  #pragma once
2  #include <iostream>
3
4  class Pair {
5      int first;
6      double second;
7  public:
8      Pair();
9      Pair(int f, double s);
10     Pair operator+(const Pair& other) const;
11     Pair operator/(int val) const;
12     Pair operator-(const Pair& other) const;
13     friend std::ostream& operator<<(std::ostream& os, const Pair& p);
14 }
```

```
Pair.h    Pair.cpp    MapFunctions.h    MapFunctions.cpp    MyMap.h    main.cpp
12лабаперделка (Глобальная область)

1  #pragma once
2  #include <map>
3  #include "Pair.h"
4
5  void insert_average(std::map<int, int>& m, int pos);
6  void delete_in_range(std::map<int, int>& m, int low, int high);
7  void subtract_average(std::map<int, int>& m);
8
9  void insert_average(std::map<int, Pair>& m, int pos);
10 void delete_in_range(std::map<int, Pair>& m, int low, int high);
11 void subtract_average(std::map<int, Pair>& m);
12
```

Pair.h
Pair.cpp
MapFunctions.h
MapFunctions.cpp
MyMap.h
main.cpp

12лабаперделка
(Глобальная обла

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

#include "Pair.h"

Pair::Pair() : first(0), second(0.0) {}
Pair::Pair(int f, double s) : first(f), second(s) {}

Pair Pair::operator+(const Pair& other) const {
 return Pair(first + other.first, second + other.second);
}

Pair Pair::operator/(int val) const {
 return Pair(first / val, second / val);
}

Pair Pair::operator-(const Pair& other) const {
 return Pair(first - other.first, second - other.second);
}

std::ostream& operator<<(std::ostream& os, const Pair& p) {
 os << p.first << " " << p.second;
 return os;
}

Pair.h
Pair.cpp
MapFunctions.h
MapFunctions.cpp
MyMap.h
main.cpp

12лабаперделка
(Глобальная область)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

#pragma once
#include <map>
#include <iostream>

template<typename T>
class MyMap {
 std::map<int, T> data;
public:
 void insert(int key, const T& value) {
 data[key] = value;
 }

 void insert_average(int pos) {
 if (data.empty()) return;
 T sum;
 for (auto& kv : data) sum = sum + kv.second;
 T avg = sum / (int)data.size();

 std::map<int, T> result;
 int i = 0;
 for (auto& kv : data) {
 if (i == pos) result[-1] = avg;
 result[kv.first] = kv.second;
 ++i;
 }
 data = result;
 }

 void delete_in_range(int low, int high) {
 for (auto it = data.begin(); it != data.end();) {
 if (it->first >= low && it->first <= high)
 it = data.erase(it);
 else
 ++it;
 }
 }

 void subtract_average() {
 if (data.empty()) return;
 T sum;
 for (auto& kv : data) sum = sum + kv.second;
 T avg = sum / (int)data.size();
 for (auto& kv : data) kv.second = kv.second - avg;
 }

 void print() const {
 for (auto& kv : data)
 std::cout << "Ключ: " << kv.first << ", Значение: " << kv.second << std::endl;
 }
};

Pair.h	Pair.cpp	MapFunctions.h	MapFunctions.cpp	MyMap.h	main.cpp
--------	----------	----------------	------------------	---------	----------

```

1      #include "MapFunctions.h"
2
3      void insert_average(std::map<int, int>& m, int pos) {
4          if (m.empty()) return;
5          int sum = 0;
6          for (auto& kv : m) sum += kv.second;
7          int avg = sum / m.size();
8
9          std::map<int, int> result;
10         int i = 0;
11         for (auto& kv : m) {
12             if (i == pos) result[-1] = avg;
13             result[kv.first] = kv.second;
14             ++i;
15         }
16         m = result;
17     }
18
19     void delete_in_range(std::map<int, int>& m, int low, int high) {
20         for (auto it = m.begin(); it != m.end(); it++) {
21             if (it->first >= low && it->first <= high)
22                 it = m.erase(it);
23             else
24                 ++it;
25         }
26     }
27
28     void subtract_average(std::map<int, int>& m) {
29         if (m.empty()) return;
30         int sum = 0;
31         for (auto& kv : m) sum += kv.second;
32         int avg = sum / m.size();
33         for (auto& kv : m) kv.second -= avg;
34     }
35
36     void insert_average(std::map<int, Pair>& m, int pos) {
37         if (m.empty()) return;
38         Pair sum;
39         for (auto& kv : m) sum = sum + kv.second;
40         Pair avg = sum / (int)m.size();
41
42         std::map<int, Pair> result;
43         int i = 0;
44         for (auto& kv : m) {
45             if (i == pos) result[-1] = avg;
46             result[kv.first] = kv.second;
47             ++i;
48         }
49         m = result;
50     }

```

```

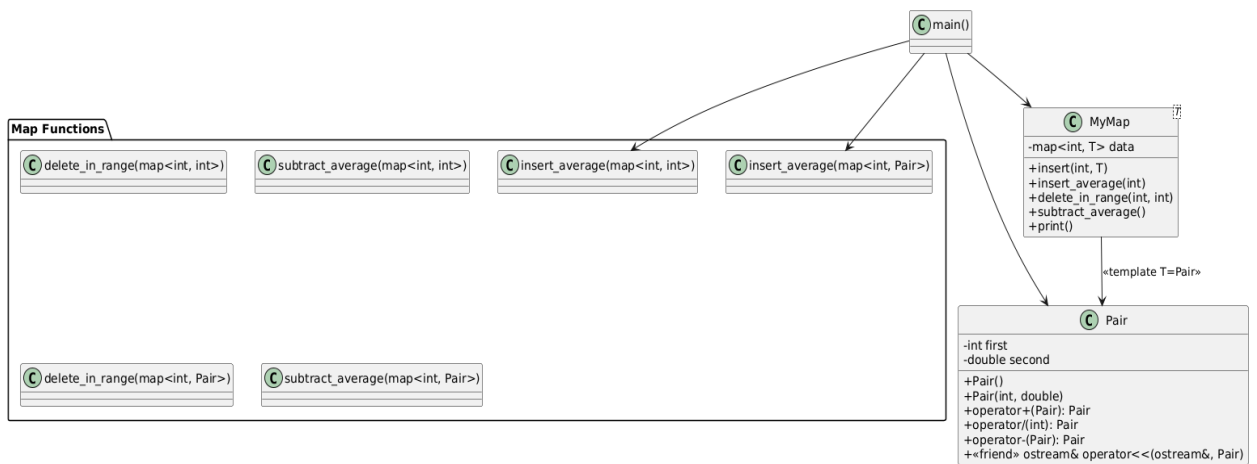
51
52     void delete_in_range(std::map<int, Pair>& m, int low, int high) {
53         for (auto it = m.begin(); it != m.end(); it++) {
54             if (it->first >= low && it->first <= high)
55                 it = m.erase(it);
56             else
57                 ++it;
58         }
59     }
60
61     void subtract_average(std::map<int, Pair>& m) {
62         if (m.empty()) return;
63         Pair sum;
64         for (auto& kv : m) sum = sum + kv.second;
65         Pair avg = sum / (int)m.size();
66         for (auto& kv : m) kv.second = kv.second - avg;
67     }

```

3 Результаты работы

```
--- Задача 1 (map<int, int>) ---  
Ключ: -1, Значение: -5  
Ключ: 3, Значение: 5  
  
--- Задача 2 (map<int, Pair>) ---  
Ключ: -1, Значение: 0:-0.55  
Ключ: 3, Значение: 1:0.55  
  
--- Задача 3 (MyMap<Pair>) ---  
Ключ: -1, Значение: 0:-0.55  
Ключ: 3, Значение: 1:0.55
```

4 UML-диаграмма классов



5 Ответы на контрольные вопросы

```
// 1. Что представляет собой ассоциативный контейнер?  
// Ответ: Контейнер, который хранит элементы в виде пар "ключ-значение" с автоматической сортировкой по ключу.  
  
// 2. Перечислить ассоциативные контейнеры библиотеки STL.  
// Ответ: map, multimap, set, multiset.  
  
// 3. Каким образом можно получить доступ к элементам ассоциативного контейнера?  
// Ответ: С помощью итераторов и оператора [] (только для map).  
  
// 4. Привести примеры методов, используемых в ассоциативных контейнерах.  
// Ответ: insert(), find(), erase(), begin(), end(), size(), clear().
```

```

// 5. Каким образом можно создать контейнер map? Примеры:
map<int, string> m1; // Пустой map
map<int, string> m2 = { {1, "one"}, {2, "two"} }; // Инициализация

// 6. Каким образом упорядочены элементы в контейнере map по умолчанию? Как изменить порядок на обратный?
// Ответ: По возрастанию ключей. Чтобы изменить – использовать std::greater.
map<int, string, greater<int>> m_desc; // по убыванию

// 7. Какие операции определены для контейнера map?
// Ответ: insert(), operator[], at(), erase(), find(), count(), size(), empty(), clear(), begin(), end().

// 8. Функция добавления в map через make_pair:
void addWithMakePair(map<int, string>& m, int key, const string& value) {
    m.insert(make_pair(key, value));
}

// 9. Функция добавления в map через []:
void addWithBracket(map<int, string>& m, int key, const string& value) {
    m[key] = value;
}

// 10. Печать map с использованием итератора:
void printMapIterator(const map<int, string>& m) {
    for (auto it = m.begin(); it != m.end(); ++it)
        cout << it->first << ": " << it->second << endl;
}

// 11. Печать map с использованием []:
void printMapBracket(const map<int, string>& m) {
    for (const auto& kv : m)
        cout << kv.first << ": " << m.at(kv.first) << endl;
}

// 12. Чем отличаются map и multimap?
// Ответ: map хранит уникальные ключи, multimap допускает дубликаты ключей.

// 13. Что представляет собой контейнер set?
// Ответ: Контейнер, хранящий только уникальные элементы, упорядоченные по возрастанию.

// 14. Чем отличаются map и set?
// Ответ: map хранит пары "ключ-значение", set хранит только ключи.

// 15. Пример создания set:
set<int> s1;
set<string> s2 = { "apple", "banana", "cherry" };

// 16. Как упорядочены элементы в set? Как сделать обратный порядок?
// Ответ: По возрастанию. Для убывания:
set<int, greater<int>> s_desc;

// 17. Какие операции определены для set?
// Ответ: insert(), erase(), find(), count(), size(), empty(), clear(), begin(), end().

// 18. Функция добавления в set:
void addToSet(set<int>& s, int value) {
    s.insert(value);
}

// 19. Функция печати set:
void printSet(const set<int>& s) {
    for (int val : s) cout << val << " ";
    cout << endl;
}

// 20. Чем отличаются set и multiset?
// Ответ: set хранит уникальные значения, multiset – допускает дубликаты.

int main() {
    // Пример использования
    map<int, string> m;
    addWithMakePair(m, 1, "один");
    addWithBracket(m, 2, "два");

    cout << "Печать map (итератор):\n";
    printMapIterator(m);

    set<int> s;
    addToSet(s, 10);
    addToSet(s, 20);
    cout << "Печать set:\n";
    printSet(s);

    return 0;
}

```

6 Ссылка на github

ссылка на github - <https://github.com/MAKSPOWERO/mas1>