

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

**Пермский национальный исследовательский
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №13 по классам

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Задача:

15	Задача 1 1. Контейнер - список 2. Тип элементов Pair (см. лабораторную работу №3). Задача 2 Адаптер контейнера – очередь с приоритетами. Задача 3 Ассоциативный контейнер - словарь		
	Задание 3	Задание 4	Задание 5
	Найти среднее арифметическое и добавить его в конец контейнера	Найти элементы ключами из заданного диапазона и удалить их из контейнера	К каждому элементу добавить сумму минимального и максимального элементов контейнера.

Текст программы
файл Pair.h

```
1  #pragma once
2  #include <iostream>
3
4  using namespace std;
5  class Pair{
6      int first;
7      double second;
8  public:
9      Pair();
10     Pair(int, double);
11     Pair(const Pair&);
12     ~Pair();
13     void set_first(int);
14     void set_second(double);
15     int get_first() const;
16     double get_second() const;
17     Pair& operator+(const Pair&);
18     Pair& operator/(int);
19     friend ostream& operator<<(ostream& out, const Pair& p);
20     friend istream& operator>>(istream& in, Pair& p);
21     friend bool operator <=(const Pair&, double s);
22     friend bool operator <= (double, const Pair&);
23     friend bool operator<(const Pair&, const Pair&);
24 };

```

Файл Pair.cpp

```
1  #include "Pair.h"
2
3  Pair::Pair() {
4      first = 0;
5      second = 0.0;
6  }
7  Pair::Pair(int f, double s) {
8      first = f;
9      second = s;
10 }
11 Pair::Pair(const Pair& p) {
12     first = p.first;
13     second = p.second;
14 }
15 Pair::~Pair() {}
16 void Pair::set_first(int f) {
17     first = f;
18 }
19 void Pair::set_second(double s) {
20     second = s;
21 }
22 int Pair::get_first() const{
23     return first;
24 }
25 double Pair::get_second() const{
26     return second;
27 }
28 Pair& Pair::operator+(const Pair& p) {
29     first += p.first;
30     second += p.second;
31     return *this;
32 }
33 Pair& Pair::operator/(int d) {
34     first /= d;
35     second /= d;
36     return *this;
37 }
38 ostream& operator<<(ostream& out, const Pair& p) {
39     return (out << p.first << " : " << p.second);
40 }
41 istream& operator>>(istream& in, Pair& p) {
42     istream& operator>>(istream& in, Pair& p) {
43         in >> p.first;
44         in >> p.second;
45         return in;
46     }
47
48     bool operator <= (const Pair& pair1, double s) {
49         return (pair1.get_first() <= s and pair1.get_second() <= s);
50     }
51     bool operator <= (double s, const Pair& pair1) {
52         return (pair1.get_first() >= s and pair1.get_second() >= s);
53     }
54     bool operator < (const Pair& pair1, const Pair& pair) {
55         return (pair1.get_first() + pair1.get_second() < pair.get_first() + pair.get_second());
56     }
57 }
```

Задание 1

Файл List_funktion.h

```
1  #pragma once
2  #include<iostream>
3  #include <list>
4  #include <numeric>
5  #include <algorithm>
6  #include "Pair.h"
7
8  using namespace std;
9
10 void Show_list(const list<Pair>& m_list) {
11     if (m_list.size() > 0) {
12         for_each(m_list.begin(), m_list.end(), [](Pair p) {
13             cout << p << endl;
14         });
15     }
16     else {
17         cout << "Список пуст!!";
18     }
19     cout << '\n';
20 }
21
22 void Push_average_value(list<Pair>& m_list) {
23     if (m_list.size() > 0) {
24         Pair Average = accumulate(m_list.begin(), m_list.end(), Pair(0, 0));
25         Average.set_first(Average.get_first() / m_list.size());
26         Average.set_second(Average.get_second() / m_list.size());
27         m_list.push_back(Average);
28     }
29     else {
30         cout << "Список пуст!!\n";
31     }
32 }
33
34 void Delete_range_keys(list<Pair>& m_list, const double Begin_value, const double End_value) {
35     auto i = remove_if(m_list.begin(), m_list.end(), [Begin_value, End_value](Pair p) {
36         return ((Begin_value <= p.get_first() && p.get_first() <= End_value) && (Begin_value <= p.get_second() && p.get_second() <= End_value));
37     });
38     m_list.erase(i, m_list.end());
39 }
40
41 void Plus_sum_Min_Max(list<Pair>& m_list) {
42     if (m_list.size() > 0) {
43         Pair max = *max_element(m_list.begin(), m_list.end());
44         Pair min = *min_element(m_list.begin(), m_list.end());
45         for (Pair& c : m_list) {
46             c = max + min;
47         }
48     }
49 }
```

Файл infa.cpp

```
1  #include "Pair.h"
2  #include "List_function.h"
3
4
5  ✓ int main() {
6      system("chcp 1251 > null");
7
8
9      cout << "Исходный список" << endl;
10     list<Pair> a = { Pair(6, 2.9), Pair(7, 9), Pair(2, 3.5) };
11     Show_list(a);
12
13     cout << "Список после добавления среднего значения в конец" << endl;
14     Push_average_value(a);
15     Show_list(a);
16
17     double begin, end;
18
19
20     cout << "Введите диапазон, элементы содержащие ключи внутри которого будут удалены\n";
21     cout << "От: ";
22     cin >> begin;
23     cout << "До: ";
24     cin >> end;
25     cout << "Список после удаления элементов содержащих ключи в заданном диапазоне\n";
26     Delete_range_keys(a, begin, end);
27     Show_list(a);
28
29     cout << "Список, после добавления к каждому элементу суммы максимального и минимального элементов\n";
30     Plus_sum_Min_Max(a);
31     Show_list(a);
32
33     return 0;
34 }
```

Задание 2

Файл Que_priority_function.h

```
1  #pragma once
2  #include<iostream>
3  #include <list>
4  #include <numeric>
5  #include <algorithm>
6  #include<Queue>
7  #include"Pair.h"
8
9  using namespace std;
10
11  list<Pair> priority_queueTolist(priority_queue<Pair>& m_priority_queue) {
12      priority_queue <Pair> tmp_priority_queue = m_priority_queue;
13      list <Pair> m_list;
14      while (tmp_priority_queue.size() > 0) {
15          m_list.push_back(tmp_priority_queue.top());
16          tmp_priority_queue.pop();
17      }
18      return m_list;
19  }
20
21  void Push_average_value(priority_queue<Pair>& m_priority_queue) {
22      if (m_priority_queue.size() > 0) {
23          list <Pair> m_list = priority_queueTolist(m_priority_queue);
24          Pair Average = accumulate(m_list.begin(), m_list.end(), Pair(0, 0));
25          Average.set_first(Average.get_first() / m_list.size());
26          Average.set_second(Average.get_second() / m_list.size());
27          m_priority_queue.push(Average);
28      }
29      else {
30          cout << "Очередь пуста!\n";
31      }
32  }
33
34  void Delete_range_keys(priority_queue<Pair>& m_priority_queue, const double BeginValue, const double EndValue) {
35      if (m_priority_queue.size() > 0) {
36          list <Pair> m_list = priority_queueTolist(m_priority_queue);
37          auto i = remove_if(m_list.begin(), m_list.end(), [BeginValue, EndValue](Pair p) {
38              return (BeginValue <= p.get_first() && p.get_first() <= EndValue) && (BeginValue <= p.get_second() && p.get_second() <= EndValue);
39          });
40          m_priority_queue = {};
41          if (i != m_list.begin()) {
42              for_each(m_list.begin(), i--, [&m_priority_queue](Pair p) {
43                  m_priority_queue.push(p);
44              });
45          }
46      }
47  }
```

Файл infa.cpp

```
1  #include "Pair.h"
2  #include "Que_priority_function.h"
3
4
5  int main() {
6      system("chcp 1251 > null");
7
8      priority_queue <Pair> a;
9      for (Pair i : { Pair(4, 2.52), Pair(5, 2.5), Pair(10, 23.5) }) {
10         a.push(i);
11     }
12     cout << "Исходная очередь с приоритетом" << endl;
13     Show_priority_queue(a);
14
15     cout << "Очередь с приоритетом после добавления среднего значения в конец" << endl;
16     Push_average_value(a);
17     Show_priority_queue(a);
18
19     double begin, end;
20
21     cout << "Введите диапазон, элементы содержащие ключи внутри которого будут удалены\n";
22     cout << "От: ";
23     cin >> begin;
24     cout << "До: ";
25     cin >> end;
26     cout << "Очередь с приоритетом, после удаления элементов содержащих ключи в заданном диапазоне\n";
27     Delete_range_keys(a, begin, end);
28     Show_priority_queue(a);
29
30     cout << "Очередь с приоритетом, после добавления к каждому элементу суммы максимального и минимального элементов\n";
31     Plus_sum_Min_Max(a);
32     Show_priority_queue(a);
33
34     return 0;
35 }
```


Задание 3

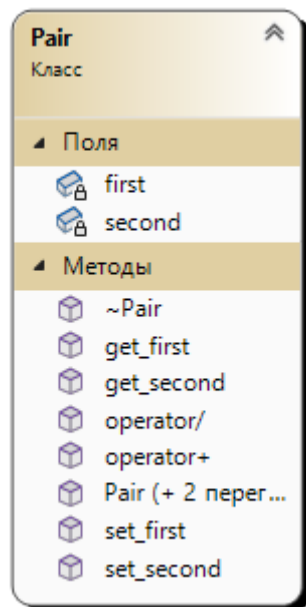
Файл Map_function.h

```
1  #pragma once
2  #include <map>
3  #include <iostream>
4  #include <list>
5  #include <numeric>
6  #include <algorithm>
7
8
9  using namespace std;
10
11 void Push_average_value(map <int, Pair>& m_map) {
12     if (m_map.size() > 0) {
13         int MaxKey = max_element(m_map.begin(), m_map.end(), [](const auto& p1, const auto& p2) {
14             return p1.first < p2.first;
15         })->first + 1;
16         Pair Middle_second = accumulate(m_map.begin(), m_map.end(), Pair(0,0), [](Pair value, const auto& p) {
17             return value + p.second;
18         }) / m_map.size();
19         m_map[MaxKey] = Middle_second;
20         m_map.insert({ MaxKey, Middle_second });
21     }
22     else {
23         cout << "Словарь пуст!\n";
24     }
25 }
26
27 void Delete_range_keys(map <int, Pair>& m_map, const double BeginValue, const double EndValue) {
28     for (auto p = m_map.begin(); p != m_map.end(); ) {
29         if (BeginValue <= (*p).second && (*p).second <= EndValue) {
30             p = m_map.erase(p);
31         }
32         else {
33             p++;
34         }
35     }
36 }
37
38 void Plus_sum_Min_Max(map <int, Pair>& m_map) {
39     if (m_map.size() > 0) {
40         Pair MaxValue = max_element(m_map.begin(), m_map.end(), [](const auto& p1, const auto& p2) {
41             return p1.second < p2.second;
42         })->second;
43         Pair MinValue = min_element(m_map.begin(), m_map.end(), [](const auto& p1, const auto& p2) {
44             return p1.second < p2.second;
45         })->second;
46         for (auto& p : m_map) {
47             p.second = p.second + MaxValue + MinValue;
48         }
49     }
50 }
51
52 void Show_map(const map <int, Pair>& m_map) {
53     if (m_map.size() > 0) {
54         for (auto& p : m_map) {
55             cout << p.first << " " << p.second << '\n';
56         }
57     }
58     else {
59         cout << "Словарь пуст!";
60     }
61     cout << '\n';
62 }
```

Файл infa.cpp

```
1  #include "Pair.h"
2  #include "Map_function.h"
3
4
5  int main() {
6      system("chcp 1251 > null");
7
8      map <int, Pair> a = { {1, {2, 3.14}}, {2, {4, 5.44}}, {3, {12, 2.15}}, {4, {2, 6.28}} };
9      cout << "Исходный словарь" << endl;
10     Show_map(a);
11
12     cout << "Словарь после добавления среднего значения в конец" << endl;
13     Push_average_value(a);
14     Show_map(a);
15
16     double begin, end;
17     cout << "Введите диапазон, элементы содержащие ключи, внутри которого будут удалены\n";
18     cin >> begin;
19     cin >> end;
20     Delete_range_keys(a, begin, end);
21     cout << "Слова, после удаления элементов содержащих ключи в заданном диапазоне\n";
22     Show_map(a);
23
24     cout << "Словарь, после добавления к каждому элементу суммы максимального и минимального элементов\n";
25     Plus_sum_Min_Max(a);
26     Show_map(a);
27
28     return 0;
29 }
```

UML-диаграмма



Тесты

Задание 1

```
Исходный список
6 : 2.9
7 : 9
2 : 3.5

Список после добавления среднего значения в конец
6 : 2.9
7 : 9
2 : 3.5
5 : 5.13333

Введите диапазон, элементы содержащие ключи внутри которого будут удалены
От: 2
До: 4
Список после удаления элементов содержащих ключи в заданном диапазоне
6 : 2.9
7 : 9
5 : 5.13333

Список, после добавления к каждому элементу суммы максимального и минимального элементов
19 : 14.8
20 : 20.9
18 : 17.0333

C:\Users\MOKAS\source\repos\infax64\Debug\infax64.exe (процесс 35616) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Задание 2

```
Исходная очередь с приоритетом
10 : 23.5
5 : 2.5
4 : 2.52

Очередь с приоритетом после добавления среднего значения в конец
10 : 23.5
6 : 9.50667
5 : 2.5
4 : 2.52

Введите диапазон, элементы содержащие ключи внутри которого будут удалены
От: 2
До: 4
Очередь с приоритетом, после удаления элементов содержащих ключи в заданном диапазоне
10 : 23.5
6 : 9.50667
5 : 2.5

Очередь с приоритетом, после добавления к каждому элементу суммы максимального и минимального элементов
25 : 49.5
21 : 35.5067
20 : 28.5

C:\Users\MOKAS\source\repos\infax64\Debug\infax64.exe (процесс 14440) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Задание 3

Исходный словарь

- 1) 2 : 3.14
- 2) 4 : 5.44
- 3) 12 : 2.15
- 4) 2 : 6.28

Словарь после добавления среднего значения в конец

- 1) 2 : 3.14
- 2) 4 : 5.44
- 3) 12 : 2.15
- 4) 2 : 6.28
- 5) 5 : 4.2525

Введите диапазон, элементы содержащие ключи, внутри которого будут удалены

2

4

Словарь, после удаления элементов содержащих ключи в заданном диапазоне

- 2) 4 : 5.44
- 3) 12 : 2.15
- 4) 2 : 6.28
- 5) 5 : 4.2525

Словарь, после добавления к каждому элементу суммы максимального и минимального элементов

- 2) 18 : 13.87
- 3) 26 : 10.58
- 4) 16 : 14.71
- 5) 19 : 12.6825

C:\Users\MOkAS\source\repos\infa\x64\Debug\infa.exe (процесс 35552) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...