Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

**по лабораторной работе № 18.12**

Дисциплина: Информатика

# Тема: «Объектно-ориентированное программирование. Ассоциативные контейнеры библиотеки STL»

Вариант 1

Выполнил работу:

студент группы РИС-20-1Б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

Пермь

2021

**Цель работы**

* 1. Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.
  2. Использование ассоциативных контейнеров библиотеки STL в ОО программе.

**Постановка задачи**

### Задача 1.

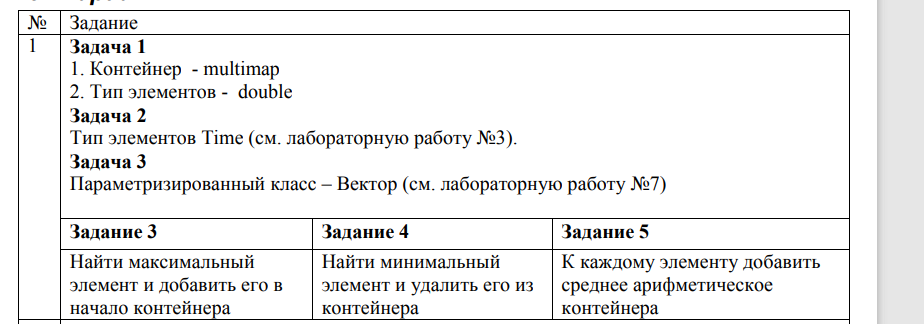
* + 1. Создать ассоциативный контейнер.
    2. Заполнить его элементами стандартного типа (тип указан в варианте).
    3. Добавить элементы в соответствии с заданием
    4. Удалить элементы в соответствии с заданием.
    5. Выполнить задание варианта для полученного контейнера.
    6. Выполнение всех 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 параметризированный 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 параметризированного класса



**Анализ задачи**

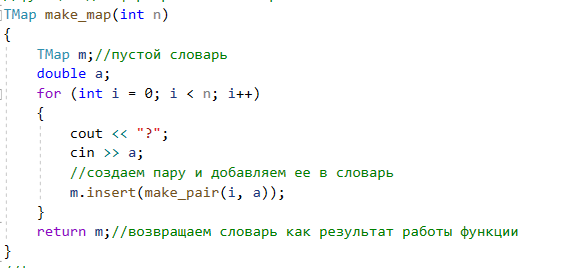
1. Необходимые действия

Задача 1

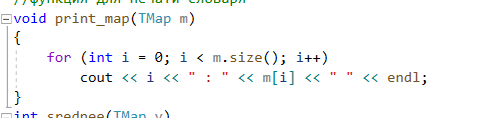
1. Ввести количество элементов в словаре



1. Написать функцию ввода элементов в словарь

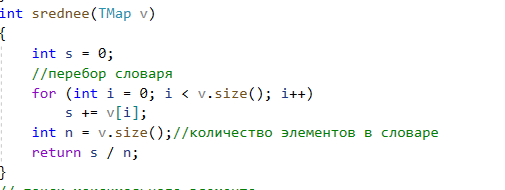


1. Вывести содержимое словря на консоль



1. Ввести значение и удалить найденный элемент из множества



1. Найти среднее арифметическое элементов словаря
2. Вывести среднее арифметическое и содержимое множества

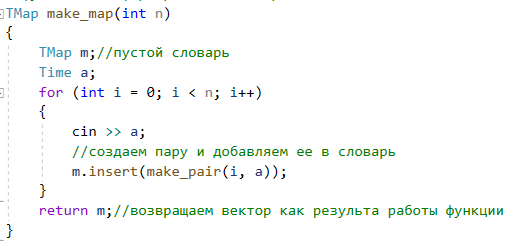


Задача 2

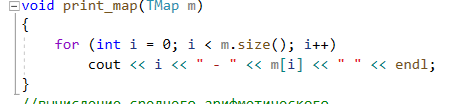
1. Ввести количество элементов во множестве



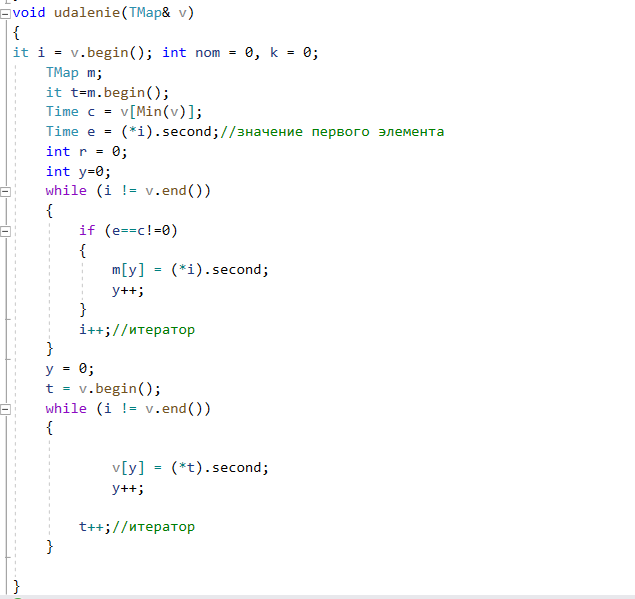
1. Написать функцию ввода элементов во множество

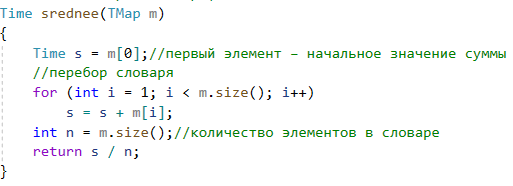


1. Вывести содержимое множества на консоль



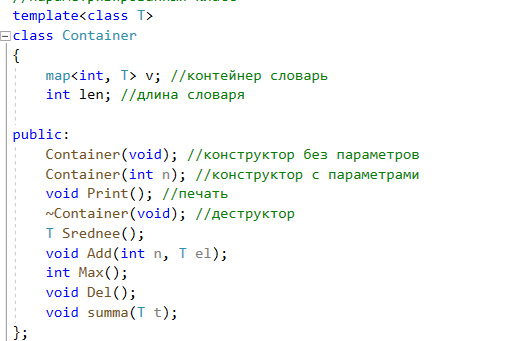
1. Ввести значение и удалить найденный элемент из множества



1. Найти среднее арифметическое элементов 

Задача 3

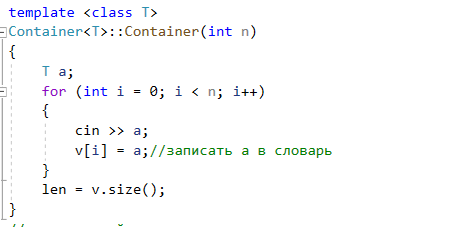
1. Создать класс-контейнер для работы со множеством, содержащим элементы любого типа (в данном случае int, double)



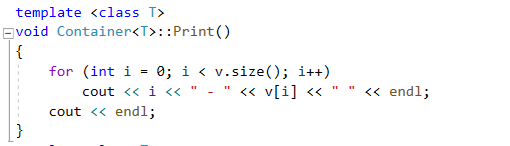
1. Ввести количество элементов в векторе



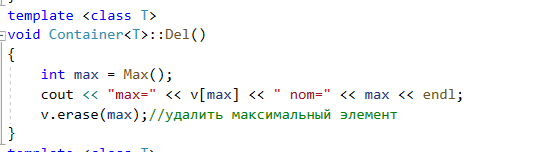
1. Написать функцию ввода элементов во множество



1. Вывести содержимое множества на консоль



1. Удалить максимальный элемент из контейнера



1. Программный код

Задача 1

#include <iostream>

#include <map>

using namespace std;

typedef map<int, double>TMap;//определяем тип для работы со словарем

typedef TMap::iterator it;//итератор

//функция для формирования словаря

TMap make\_map(int n)

{

TMap m;//пустой словарь

double a;

for (int i = 0; i < n; i++)

{

cout << "?";

cin >> a;

//создаем пару и добавляем ее в словарь

m.insert(make\_pair(i, a));

}

return m;//возвращаем словарь как результат работы функции

}

//функция для печати словаря

void print\_map(TMap m)

{

for (int i = 0; i < m.size(); i++)

cout << i << " : " << m[i] << " " << endl;

}

int srednee(TMap v)

{

int s = 0;

//перебор словаря

for (int i = 0; i < v.size(); i++)

s += v[i];

int n = v.size();//количество элементов в словаре

return s / n;

}

// поиск максимального элемента

int Max(TMap v)

{

it i = v.begin();

int nom = 0,//номер максимального

k = 0;//счетчик элементов

int m = (\*i).second;//значение первого элемента

while (i != v.end())

{

if (m < (\*i).second)

{

m = (\*i).second;

nom = k;

}

i++;//итератор

k++;//счетчик элементов

}

return nom;//номер max

}

int Min(TMap v)

{

it i = v.begin(); int nom = 0, k = 0;

int m = (\*i).second;//значение первого элемента

while (i != v.end())

{

if (m > (\*i).second)

{

m = (\*i).second;

nom = k;

}

i++;//итератор

k++;//счетчик элементов

}

return nom;//номер max

}

void delenie(TMap& v)

{

double m = v[Min(v)];//значение минимального элемента

for (int i = 0; i < v.size(); i++)

v[i] = v[i] / m;

}

//основная функция

void main()

{

int n;

cout << "N?"; cin >> n;//количество элементов

TMap m = make\_map(n);//создать словарь

print\_map(m);//напечатать словарь

int max = Max(m);

cout << "max=" << m[max] << " nom=" << max << endl;

m.erase(max);//удаление элемента

print\_map(m);

int min = Min(m);

cout << "min=" << m[min] << " nom=" << min << endl;

delenie(m);

print\_map(m);

}

Задача 2

#include <iostream>

# #include <iostream>

#include <map>

#include "Time.h"

using namespace std;

typedef map<int, Time>TMap;//определяем тип для работы со словаре

typedef TMap::iterator it;

//функция для формирования словаря

TMap make\_map(int n)

{

TMap m;//пустой словарь

Time a;

for (int i = 0; i < n; i++)

{

cin >> a;

//создаем пару и добавляем ее в словарь

m.insert(make\_pair(i, a));

}

return m;//возвращаем вектор как результа работы функции

}

//функция для печати словаря

void print\_map(TMap m)

{

for (int i = 0; i < m.size(); i++)

cout << i << " - " << m[i] << " " << endl;

}

//вычисление среднего арифметического

Time srednee(TMap m)

{

Time s = m[0];//первый элемент – начальное значение суммы

//перебор словаря

for (int i = 1; i < m.size(); i++)

s = s + m[i];

int n = m.size();//количество элементов в словаре

return s / n;

}

//поиск максимального элемента

int Max(TMap v)

{

it i = v.begin(); int nom = 0, k = 0;

Time m = (\*i).second;//значение первого элемента

while (i != v.end())

{

if (m < (\*i).second)

{

m = (\*i).second;

nom = k;

}

i++;//итератор

k++;//счетчик элементов

}

return nom;//номер max

}

//поиск минимального элемента

int Min(TMap v)

{

it i = v.begin(); int nom = 0, k = 0;

Time m = (\*i).second;//значение первого элемента

while (i != v.end())

{

if (m > (\*i).second)

{

m = (\*i).second;

nom = k;

}

i++;//итератор

k++;//счетчик элементов

}

return nom;//номер max

}

void delenie(TMap& v)

{

Time m = v[Min(v)];

for (int i = 0; i < v.size(); i++)

v[i] = v[i] / m;

}

void udalenie(TMap& v)

{

it i = v.begin(); int nom = 0, k = 0;

TMap m;

it t=m.begin();

Time c = v[Min(v)];

Time e = (\*i).second;//значение первого элемента

int r = 0;

int y=0;

while (i != v.end())

{

if (e==c!=0)

{

m[y] = (\*i).second;

y++;

}

i++;//итератор

}

y = 0;

t = v.begin();

while (i != v.end())

{

v[y] = (\*t).second;

y++;

t++;//итератор

}

}

void main()

{

int n;

cout << "N?"; cin >> n;

map<int, Time> m = make\_map(n);

print\_map(m);

//вычисление среднего

Time el = srednee(m);

cout << "srednee=" << el << endl;

//добавление в конец

m.insert(make\_pair(n, el));

print\_map(m);

int min = Min(m);

//cout << "max=" << m[max] << " nom=" << max << endl;

//m.erase(max);

m.erase(min);

//cout << "udalenie min: ";

//udalenie(m);

print\_map(m);

//cout << "min=" << m[min] << " nom=" << min << endl;

//delenie(m);

//print\_map(m);

}

Задача 3

Сontainer.h

#include <iostream>

#include <map> //словарь

using namespace std;

//параметризированный класс

template<class T>

class Container

{

map<int, T> v; //контейнер словарь

int len; //длина словаря

public:

Container(void); //конструктор без параметров

Container(int n); //конструктор с параметрами

void Print(); //печать

~Container(void); //деструктор

T Srednee();

void Add(int n, T el);

int Max();

void Del();

void summa(T t);

};

//реализация методов

//конструктор без параметров

template <class T>

Container<T>::Container() {

len = 0;

}

//деструктор

template <class T>

Container<T>::~Container(void)

{

}

//конструктор c параметрами

template <class T>

Container<T>::Container(int n)

{

T a;

for (int i = 0; i < n; i++)

{

cin >> a;

v[i] = a;//записать а в словарь

}

len = v.size();

}

//вывод контейнера

template <class T>

void Container<T>::Print()

{

for (int i = 0; i < v.size(); i++)

cout << i << " - " << v[i] << " " << endl;

cout << endl;

}

template<class T>

T Container<T>::Srednee()

{

T s = v[0];//начальное значение суммы – первый элемент словаря

//перебор словаря

for (int i = 1; i < v.size(); i++)s = s + v[i];

int n = v.size();//количество элементов в словаре

return s / n;

}

//добавление

template<class T>

void Container<T>::Add(int n, T el)

{

v.insert(make\_pair(n, el));//формируем пару и добавляем ее в словарь

}

template <class T>

int Container<T>::Max()

{

typedef map<int, T>TMap;//определяем тип для работы со словаре

typedef TMap::iterator it;

it i = v.begin();//итератор поставили на первый элемент

int nom = 0, k = 0;

T m = (\*i).second;//значение первого элемента

while (i != v.end()) //пока нет конца контейнера

{

if (m < (\*i).second)

{

m = (\*i).second;

nom = k;

}

i++;//итератор

k++;//счетчик элементов

}

return nom;//номер max

}

template <class T>

void Container<T>::Del()

{

int max = Max();

cout << "max=" << v[max] << " nom=" << max << endl;

v.erase(max);//удалить максимальный элемент

}

template <class T>

void Container<T>::summa(T t) {

//typedef map<int, T>TMap;//определяем тип для работы со словаре

//typedef TMap::iterator it;

//it i = v.begin();//итератор поставили на первый элемент

for (int i = 0; i < v.size(); i++)

v[i] = v[i] + t;

}

18.12(3).cpp

// 18.12(3).cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include "Time.h"

#include "Container.h"

#include <iostream>

using namespace std;

void main()

{

int n;

cout << "N?"; cin >> n;

Container<Time> v(n);

v.Print();

Time t = v.Srednee();//найти среднее арифметическое

v.summa(t);

v.Print();

cout << "Srednee=" << t << endl;

cout << "Add srednee" << endl;

/\*cout << "pos?";

int pos;

cin >> pos;//позиция для добавления\*/

//v.insert(make\_pair(n, t));

v.Add(n, t);//добавление

v.Print();//печать

cout << "Delete max: " << endl;

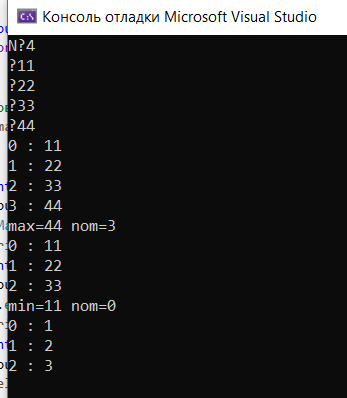
v.Del();

v.Print();

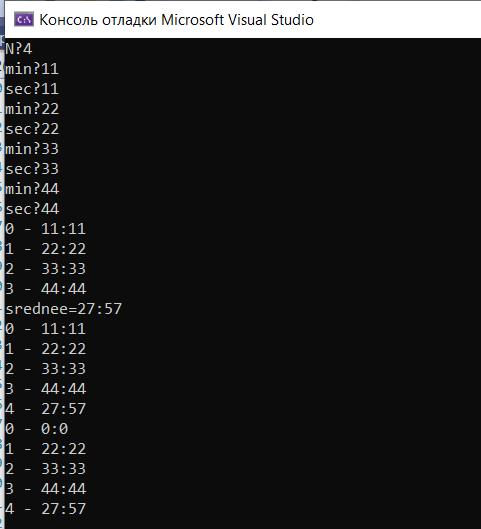
}

1. Демонстрация работы программы

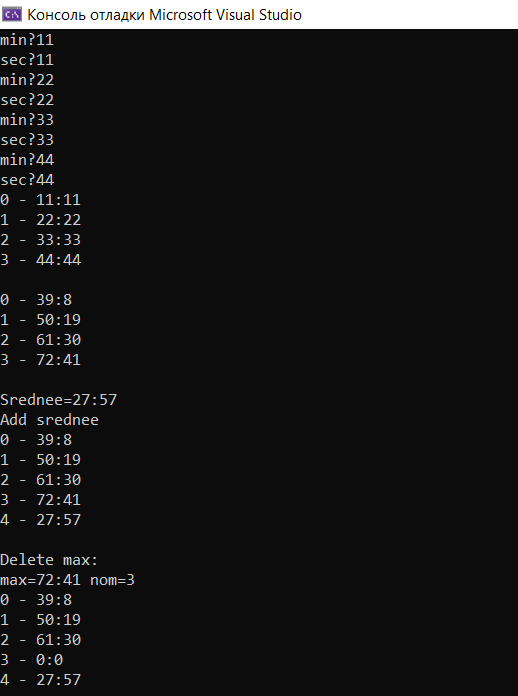
Задача 1



Задача 2



Задача 3



1. Типы переменных и представление данных

Задача 1

Set<int, less<int>> - множество для хранения элементов целого типа в возрастающем порядке, int – элемент множества, размер множества, элемент для удаления из множества, сумма элементов во множестве, среднее арифметическое, iterator – итератор для прохождения по элементам множества, vector<int> - вектор для временного хранения элементов множества

Задача 2

Money – элемент множества, элемент для удаления из множества, set<Money, less<Money>> - множество для хранения элементов в порядке возрастания, int – размер множества, сумма элементов во множестве, среднее арифметическое, iterator – итератор для прохождения по элементам множества, vector<Money> - вектор для временного хранения элементов класса Money

Задача 3

Vector – класс для взаимодействия со множеством, set<T, less<T>> - хранение элементов множества в порядке возрастания, T – элемент вектора, элемент для удаления, сумма элементов в векторе, среднее арифметическое, vector<T>, int – размер вектора, iterator - итератор для прохождения по элементам множества

1. Контрольные вопросы
   1. Что представляет собой ассоциативный контейнер?

Это ассоциативный массив, содержащий пару значений (ключ и значение)

* 1. Перечислить ассоциативные контейнеры библиотеки STL.

Set, multiset, map, multimap

* 1. Каким образом можно получить доступ к элементам ассоциативного контейнера?

Получить доступ можно с помощью итераторов

* 1. Привести примеры методов, используемых в ассоциативных контейнерах.

Insert(), erase(), count(), size()

* 1. Каким образом можно создать контейнер map? Привести примеры.

map<int, float> mymap;

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

По умолчанию – возрастание элементов.

map<int, greater<int>> mymap; - убывание элементов

* 1. Какие операции определены для контейнера map?

Size(), empty(), find(), [], insert(pair(k, v)), erase(), begin(), end()

* 1. Написать функцию для добавления элементов в контейнер map с помощью функции make\_pair().

map<int, int> mymap;

void add\_element(map<int, int>& some\_map, int size) {

int number;

for (int i = 0; i < size; i++) {

cin >> number;

some\_map.insert(make\_pair(i, number));

}

}

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

map<int, int> mymap;

void add\_element(map<int, int>& some\_map, int size) {

int number;

for (int i = 0; i < size; i++) {

cin >> number;

some\_map[i] = number;

}

}

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

void print\_map(map<int, int> some\_map) {

map<int, int>::iterator iter;

for (iter = some\_map.begin(); iter != some\_map.end(); iter++)

cout << (iter->second) << endl;

}

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

void print\_map(map<int, int> some\_map) {

for (int i = 0; i < some\_map.size(); i++)

cout << some\_map[i] << endl;

}

* 1. Чем отличаются контейнеры map и multimap?

Multimap позволяет хранить элементы с одинаковыми ключами и для multimap не определена операция доступа по индексу

* 1. Что представляет собой контейнер set?

Множества set можно рассматривать как ассоциативные массивы, в которых значения не играют роли, так что мы отслеживаем только ключи.

* 1. Чем отличаются контейнеры map и set?

Map хранит в себе пару значений, а set – одно значение

* 1. Каким образом можно создать контейнер set? Привести примеры.

set<int> set1; // создается пустое множество int а[5] = { 1. 2. 3. 4, 5};

set<int> set2(a, а + 5);// инициализация копированием set<int> set3(set2); // инициализация другим множеством

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

По умолчанию элементы упорядочены по возрастанию.

Можно изменить порядок вывода с помощью реверсивного итератора

set<int> myset = { 10,50,30,40,20 };

set<int>::reverse\_iterator rit;

for (rit = myset.rbegin(); rit != myset.rend(); ++rit)

cout << \*rit << '\n';

* 1. Какие операции определены для контейнера set?

Begin(), end(), find(), insert(), copy(), erase(), count(), lower\_bound(), upper\_bound(), equal\_range()

* 1. Написать функцию для добавления элементов в контейнер set.

void add\_element(set<int>& myset, int size) {

int number;

for (int i = 0; i < size; i++) {

cin >> number;

myset.insert(number);

}

}

* 1. Написать функцию для печати контейнера set

void print\_set(set<int> myset) {

set<int>::iterator it;

for (it = myset.begin(); it != myset.end(); ++it)

cout << \*it << '\n';

}

* 1. Чем отличаются контейнеры set и multiset?

Multiset может хранить дубликаты