Министерство образования РФ

Пермский национальный исследовательский политехнический университет

Кафедра ИТАС

ЛАБОРАТОРНАЯ РАБОТА № 12

ПО ОСНОВАМ АЛГОРИТМИЗАЦИИ ЗА II СЕМЕСТР

Вариант 15

Выполнил студент:

Сташков Арсений Дмитриевич

Группа РИС-20-1бз

Шифр 20-ЭТФ-644

Проверила:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

ПЕРМЬ 2022

# Лабораторная работа №12 "Ассоциативные контейнеры библиотеки STL"

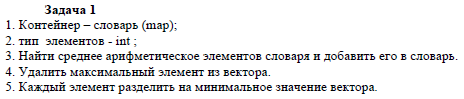
**Вариант №15**

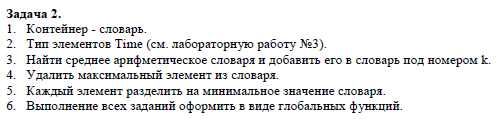
**Цель:**

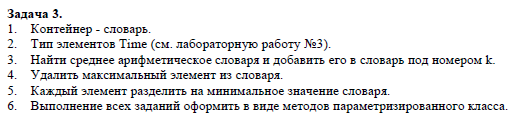
1) Создание консольного приложения, состоящего из нескольких файлов в системе программирования Visual Studio.

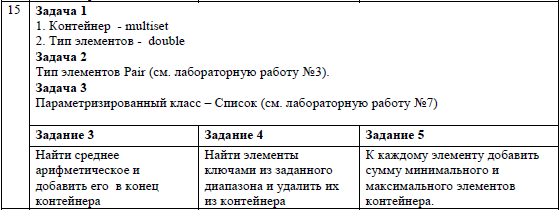
2) Использование ассоциативных контейнеров библиотеки STL в ОО программе.

## **Задача:**









**Решение:**

Файл Ассоциативные конт.STL.cpp

#include <iostream>

#include <set>

using namespace std;

// Определяем тип для работы контейнера

typedef set<double> TSet;

// Итератор

typedef TSet::iterator it;

// Формирование контейнера

TSet MakeSet(int n) {

// Пустой контейнер

TSet v;

double a;

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

cin >> a;

// Добавляем значение в контейнер

v.insert(a);

}

cout << "\n";

//cin.get();

return v;

}

// Вывод контейнера на экран

void PrintSet(TSet v) {

it p;

int i;

for (p = v.begin(), i = 0; p != v.end(); ++p, ++i) {

cout << i << " : " << \*p << " \n";

}

}

// Добавляем среднее арифметическое в контейнер

TSet Average(TSet v) {

it p;

double sum = 0;

int i;

for (p = v.begin(), i = 0; p != v.end(); ++p, ++i) {

sum += \*p;

}

int size = v.size();

double avg = sum / size;

cout << "\nВысчитываем среднее арифметическое и добавляем в контейнер: " << avg << "\n";

v.insert(avg);

return v;

}

// Удаление элементов в заданном диапазоне (10-50)

TSet DeleteRange(TSet v) {

for (auto i = v.begin(); i != v.end(); ) {

if (\*i >= 10 && \*i <= 50) {

i = v.erase(i);

}

else {

++i;

}

}

return v;

}

// Сумма каждого элемента с мин. и макс. значениями

TSet Sum(TSet v) {

TSet newSet;

double min = \*v.begin();

double max = NULL;

for (auto i = v.begin(); i != v.end(); i++) {

if (max == NULL || max < \*i) {

max = \*i;

}

}

double value;

for (auto i = v.begin(); i != v.end(); ) {

value = \*i;

i = v.erase(i);

newSet.insert(value + min + max);

}

return newSet;

}

int main()

{

setlocale(LC\_ALL, "Rus");

TSet s;

int n;

cout << "Введите кол-во значений контейнера: ";

cin >> n;

cout << "\n";

//cin.get();

cout << "Введите значения (double):\n";

s = MakeSet(n);

cout << "Наш контейнер:\n";

PrintSet(s);

//cin.get();

//s.insert();

s = Average(s);

PrintSet(s);

cout << "\nУдаляем элементы в диапазоне 10-50:\n";

s = DeleteRange(s);

PrintSet(s);

cout << "\nДобавляем к каждому элементу сумму макс. и мин. значений:\n";

s = Sum(s);

PrintSet(s);

return 0;

}

Файл Container.h

#pragma once

#include <iostream>

#include <set>

using namespace std;

// Параметризированный класс

template <class T>

class Listing {

set<double, T> v;

int len;

public:

Listing(void);

Listing(int n);

void Print();

T Average();

void Add(double, T);

double Min();

double Max();

void Sum();

~Listing(void);

};

// Конструктор без параметров

template <class T>

Listing<T>::Listing() { len = 0; }

// Деструктор

template <class T>

Listing<T>::~Listing(void) {}

// Конструктор с параметрами

template <class T>

Listing<T>::Listing(int n) {

T a;

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

cin >> a; v[i] = a;

}

len = v.size();

}

// Вывод контейнера на экран

template <class T>

void Listing<T>::Print() {

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

cout << i << " - " << v[i] << " \n";

}

cout << endl;

}

template <class T>

T Listing<T>::Average() {

Pair a = v[0];

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

a = a + v[i];

}

int n = v.size();

return a/n;

}

template <class T>

void Listing<T>::Add(double n, T el) {

v.insert(maker\_pair(n, el));

}

template <class T>

double Listing<T>::Min() {

set<double, T>::iterator i = v.begin();

int num, k;

num = 0, k = 0;

Pair mi = (\*i).second;

while (i != v.end()) {

if ((\*i).second.get\_min() != 0 && (\*i).second.get\_sec() != 0) {

if (mi > (\*i).second) { mi = (\*i).second; num = k; }

i++;

k++;

}

retrun num;

}

}

template <class T>

double Listing<T>::Max() {

set<double, T>::iterator i = v.begin();

int num, k;

num = 0, k = 0;

Pair ma = (\*i).second;

while (i != v.end()) {

if ((\*i).second.get\_min() != 0 && (\*i).second.get\_sec() != 0) {

if (ma < (\*i).second) {

ma = (\*i).second;

num = k;

}

i++;

k++;

}

retrun num;

}

}

template <class T>

void Listing<T>::Sum() {

T min = v[Min()];

T max = v[Max()];

cout << "Минимальное значение: " << min << endl;

cout << "Максимальное значение: " << max << endl;

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

{ v[i] = v[i] + mi + ma; }

}

Файл Task3.cpp

#include "C:\Users\arsenii.stashkov\Documents\Политех\Лабораторные работы C++\2 семестр\Основы алгоритмизации\Лабораторная работа №11\Task2\Pair.h"

#include "Container.h"

#include <iostream>

int main()

{

setlocale(LC\_ALL, "Rus");

int n;

cout << "N?";

cin >> n;

Listing <Pair> v(n);

v.Print();

Pair p = v.Average();

cout << "Среднее значение: " << p << endl;

cout << "Добавляем значение 2.5" << endl;

v.Add(2.5, p);

v.Print();

cout << "Сумма: \n";

v.Sum();

v.Print();

}

Файл Task2.cpp

#include <iostream>

#include <set>

#include "C:\Users\arsenii.stashkov\Documents\Политех\Лабораторные работы C++\2 семестр\Основы алгоритмизации\Лабораторная работа №11\Task2\Pair.h"

using namespace std;

typedef set<double, Pair>TSet;

typedef TSet::iterator it;

TSet MakeSet(int n) {

TSet s;

double a;

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

cin >> a; s.insert(a);

}

return s;

}

void PrintSet(TSet s) {

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

cout << i << " - " << i << " \n";

}

}

Pair Average(TSet s) {

Pair a = s[0];

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

a += i;

}

int n = s.size();

return a/n;

}

int Min(TSet v) {

it i = v.begin();

int num, k;

num = 0;

k = 0;

Pair s = (\*i).second;

while (i != v.end()) {

if (s > (\*i).second) {

s = (\*i).second; num = k;

}

i++;

k++;

}

return num;

}

int Max(TSet v) {

it i = v.begin();

int num, k;

num = 0;

k = 0;

Pair s = (\*i).second;

while (i != v.end()) {

if (s < (\*i).second) { s = (\*i).second; num = k; }

i++;

k++;

}

return num;

}

void Sum(TSet& v) {

Pair mi = v[Min(v)];

Pair ma = v[Max(v)];

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

v[i] = v[i] + v[mi] + v[ma];

}

}

int main()

{

int n;

cout << "Введите значение n (int)" << endl;

cin >> n;

set<double, Pair> s = MakeSet(n);

PrintSet(s);

Pair el = Average(s);

cout << "Среднее значение: " << el << endl;

s.insert(n);

PrintSet(s);

Sum(s);

PrintSet(s);

}

