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 №12**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Ассоциативные контейнеры библиотеки STL

Вариант 6.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**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.

Условия варианта:

1. Контейнер - multiset;

2. Тип элементов - double.

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.

Условие варианта:

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

Задача 3:

1. Создать параметризированный класс, используя в качестве контейнера ассоциативный контейнер.

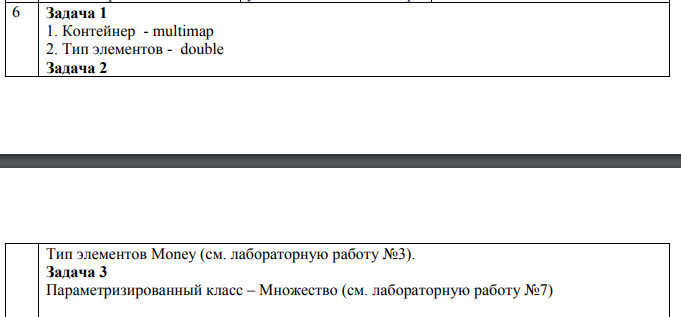
2. Заполнить его элементами.

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.



**Код программы**

**Задача 1**

#include <iostream>

#include <set>

using namespace std;

typedef set<double> TSet;

set<double>::iterator it;

TSet make\_set(int n)

{

TSet m;

double a;

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

{

cout << "Number: ";

cin >> a;

m.insert(a);

}

return m;

}

void print\_set(TSet m)

{

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

cout << \*it << " ";

}

TSet erase\_set(TSet m, double el)

{

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

{

double n = \*it;

if (el == n)

{

cout << \*it << endl;

m.erase(it++);

}

else

++it;

}

return m;

}

double Min\_set(TSet m)

{

double min, temp;

it = m.begin();

min = \*it;

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

{

temp = \*it;

if (min > temp)

min = temp;

}

return min;

}

double Max\_set(TSet m)

{

double max, temp;

it = m.begin();

max = \*it;

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

{

temp = \*it;

if (max < temp)

max = temp;

}

return max;

}

TSet zadanie(TSet m, double min, double max)

{

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

{

double h = \*it + min - max;

cout << h << " ";

}

return m;

}

void main()

{

int n;

cout << "Number: ";

cin >> n;

TSet m = make\_set(n);

print\_set(m);

double d;

cout << "Whats to remove: ";

cin >> d;

TSet j = erase\_set(m, d);

print\_set(j);

double h = Min\_set(j);

double u;

cout << "Add: " << endl;

cin >> u;

j.insert(u);

print\_set(j);

cout << endl;

double g = Max\_set(j);

cout << "Add max and min: " << endl;

zadanie(j, h, g);

}

**Задача 2**

**#include** <iostream>

#include <map>

#include "Money.h"

using namespace std;

typedef map<int, Money> TMap;

typedef TMap::iterator it;

TMap make\_map(int n)

{

TMap m;

Money a;

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

{

cin >> a;

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

}

return m;

}

void print\_map(TMap m)

{

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

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

}

Money middle(TMap m)

{

Money s = m[0];

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

s = s + m[i];

return s / m.size();

}

int Max(TMap v)

{

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

Money m = (\*i).second;

while (i != v.end())

{

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

{

m = (\*i).second;

nom = k;

}

i++;

k++;

}

return nom;

}

int Min(TMap v)

{

it i = v.begin();

int nom = 0, k = 0;

Money m = (\*i).second;

while (i != v.end())

{

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

{

m = (\*i).second;

nom = k;

}

i++;

k++;

}

return nom;

}

void delenie(TMap& v)

{

Money m = v[Min(v)];

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

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

}

void main()

{

int n;

cout << "Number: ";

cin >> n;

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

print\_map(m);

Money el = middle(m);

cout << "Middle: " << el << endl;

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

print\_map(m);

int max = Max(m);

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

m.erase(max);

print\_map(m);

int min = Min(m);

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

delenie(m);

print\_map(m);

}

**Задача 3**

**#include** <iostream>

#include "Many.h"

#include "Money.h"

using namespace std;

void main()

{

int n;

cout << "Number: ";

cin >> n;

Many <Money> v(n);

v.Print();

Money t = v.Srednee();

cout << "Middle: " << t << endl;

cout << "Adding Middle" << endl;

cout << "Pos: ";

int pos;

cin >> pos;

v.Add(pos, t);

v.Print();

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

v.Del();

v.Print();

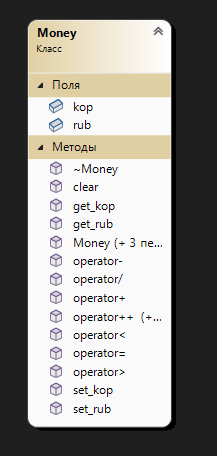
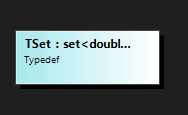
cout << "Division on min: " << endl;

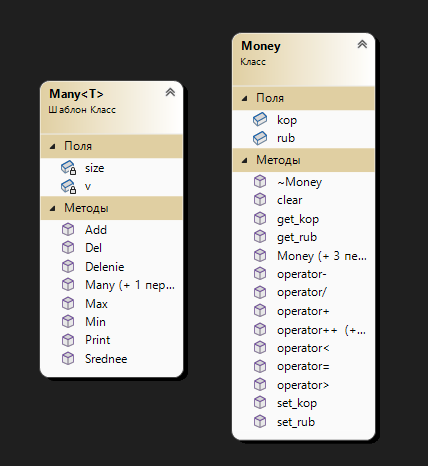
v.Delenie();

v.Print();

}

**UML диаграммы**

****



**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.