**МИНОБРНАУКИ РОССИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ижевский государственный  
технический университет имени М.Т. Калашникова»

(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Кафедра «Вычислительная техника»

Отчет по лабораторной работе №2

по дисциплине «Объектно-ориентированное программирование»

на тему «Библиотека STL, ассоциативные контейнеры и потоковая итераторы»

Вариант 14

Выполнил: студент гр. Б06-781-1

Пермитин И.А.

Проверил: к.т.н., доцент кафедры ВТ

Марков Е.М.

Ижевск 2020

**Задание**

1. Разработать программу по заданию, сама обработка выполняется через алгоритмы STL.

2. Использовать для обработки как множество, так и отображение, данные заполняются случайным образом, как в первой работе. Задание выполняется как для множества, так и для отображения. В качестве элемента контейнера допускается использовать стандартные типы данных (рекомендуется использовать string, самый простой вариант это int), для тех кто хочет лучше изучить рекомендуется использовать свои типы данных, например как в предыдущей работе bitset.

3. В зависимости от задания массив выводиться до и после обработки в удобном виде, как на экран, так и файл, через потоковые итераторы.

**Описание использованных технологий**

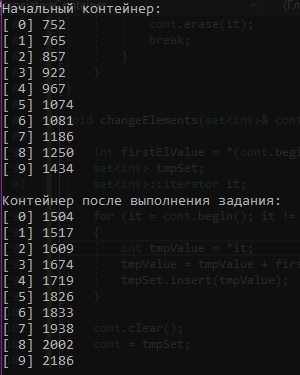
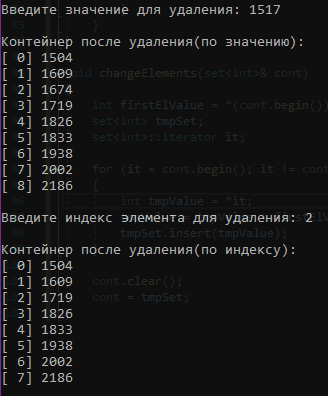
Лабораторная работа выполнялась в IDE «Visual Studio 2019».

Используемый язык программирования – C++.

Тип проекта – Консольное приложение.

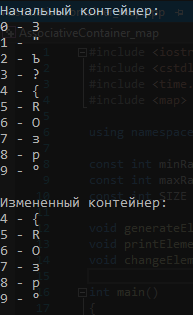
**Работа приложения**

Контейнер SET: Создается и заполняется сгенерированными значениями. Элементы контейнера выводятся в консоль. Происходит увеличение значения каждого элемента контейнера на значение первого элемента. Далее – удаление по значению и индексу. После каждого изменения контейнера выполняется вывод на экран.

Контейнер MAP: Создается и заполняется сгенерированными значениями.

Происходит удаление элемента с максимальными значением и вывод контейнера в консоль.



**Листинг программы**

SET:

#include <iostream>

#include <set>

#include <algorithm>

#include <cstdlib>

#include <time.h>

using namespace std;

const int minRand = 700;

const int maxRand = 900;

const int SIZE = 10;

void generateContainer(set<int>& cont);

void printElements(set<int> cont, string title);

void eraseElement\_value(set<int>& cont, int value);

void eraseElement\_index(set<int>& cont, int index);

void changeElements(set<int>& cont);

int main()

{

setlocale(LC\_ALL, "Russian");

int index;

int value;

set<int> cont;

generateContainer(cont);

printElements(cont, "Начальный контейнер");

changeElements(cont);

printElements(cont, "Контейнер после выполнения задания");

cout << "\nВведите значение для удаления: ";

cin >> value;

eraseElement\_value(cont, value);

printElements(cont, "Контейнер после удаления(по значению)");

cout << "\nВведите индекс элемента для удаления: ";

cin >> index;

eraseElement\_index(cont, index);

printElements(cont, "Контейнер после удаления(по индексу)");

return 0;

}

void generateContainer(set<int>& cont)

{

srand(time(NULL));

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

{

int val = minRand + rand() % maxRand;

cont.insert(val);

}

}

void printElements(set<int> cont, string title)

{

set<int>::iterator it;

cout << "\n" << title << ":" << endl;

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

{

cout << "[ " << distance(cont.begin(), it) << "] " << \*it << endl;

}

}

void eraseElement\_value(set<int>& cont, int value)

{

cont.erase(value);

}

void eraseElement\_index(set<int>& cont, int index)

{

set<int>::iterator it;

int tmp = 0;

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

{

tmp++;

if (tmp == index + 1)

{

cont.erase(it);

break;

}

}

}

void changeElements(set<int>& cont)

{

int firstElValue = \*(cont.begin());

set<int> tmpSet;

set<int>::iterator it;

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

{

int tmpValue = \*it;

tmpValue = tmpValue + firstElValue;

tmpSet.insert(tmpValue);

}

cont.clear();

cont = tmpSet;

}

MAP:

#include <iostream>

#include <cstdlib>

#include <time.h>

#include <map>

using namespace std;

const int minRand = 90;

const int maxRand = 150;

const int SIZE = 10;

void generateElements(map<int, char>& cont);

void printElements(map<int, char> cont, string title);

void changeElements(map<int, char>& cont);

int main()

{

setlocale(LC\_ALL, "Russian");

map<int, char> cont;

generateElements(cont);

printElements(cont, "Начальный контейнер");

changeElements(cont);

printElements(cont, "Измененный контейнер");

return 0;

}

void generateElements(map<int, char>& cont)

{

srand(time(NULL));

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

{

char value = minRand + rand() % maxRand;

cont.insert(pair<int, char>(i, value));

}

}

void printElements(map<int, char> cont, string title)

{

map<int, char>::iterator it;

cout << "\n" << title << ":" << endl;

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

{

cout << it->first << " - " << it->second << endl;

}

}

void changeElements(map<int, char>& cont)

{

int maxElement = 0;

map<int, char>::iterator it;

map<int, char>::iterator maxElementIt;

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

{

if (it->second > maxElement)

{

maxElement = it->second;

maxElementIt = it;

}

}

cont.erase(cont.begin(), maxElementIt);

}