ОТЧЕТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №8  
ПО ДИСЦИПЛИНЕ «ПРОГРАММИРОВАНИЕ» Вариант №4

Факультет: АВТФ Преподаватель: Новицкая Ю.В.

Группа: АВТ-208

Студенты: Нестеренко М.К.

Паршин Н.С.

Цель работы:

Ознакомиться с шаблонными классами библиотеки STL. Изучить применение этих классов и их методов на практике.

Задание:

Для встроенного типа (например, int или char) провести временной анализ заданных шаблонных классов на основных операциях: добавление, удаление, поиск, сортировка. Использовать итераторы для работы с контейнерами. Для получения времени выполнения операции засекать системное время начала и окончания операции и автоматически генерировать большое количество данных

**Вариант 4**

Стек; множество с дубликатами.

Исходные коды модулей проекта:

/**/set\_test.h**

#pragma once

#include <set>

#include <ctime>

#include <random>

using namespace std;

unsigned set\_add\_test();

unsigned set\_delete\_test();

unsigned set\_search\_test();

//unsigned set\_sort\_test(); set изначально отсортирован

**//stack\_test.h**

#pragma once

#include <stack>

#include <ctime>

#include <random>

using namespace std;

unsigned stack\_add\_test();

unsigned stack\_delete\_test();

//unsigned stack\_search\_test(); доступ только к первому элементу, поиск невозможен

//unsigned stack\_sort\_test(); сортировка для стека невозможна

**//main.cpp**

#include <iostream>

#include "../Header\_files/set\_test.h"

#include "../Header\_files/stack\_test.h"

using namespace std;

int main() {

cout << "Set testing\n";

cout << "Time for 1b floats set add test: " << set\_add\_test() / CLOCKS\_PER\_SEC << " seconds" << endl;

cout << "Time for 1b floats set delete test: " << set\_delete\_test() / CLOCKS\_PER\_SEC << " seconds" << endl;

cout << "Time for 1b floats set search test: " << set\_search\_test() / CLOCKS\_PER\_SEC << " seconds" << endl;

cout << "Set always sorted\n";

cout << "Stack testing\n";

cout << "Time for 1b floats stack add test: " << stack\_add\_test() / CLOCKS\_PER\_SEC << " seconds" << endl;

cout << "Time for 1b floats stack delete test: " << stack\_delete\_test() / CLOCKS\_PER\_SEC << " seconds" << endl;

cout << "Access to first element only, sorting is not possible\n";

cout << "Access to first element only, search is not possible\n";

return 0;

}

**//set\_test.cpp**

#include "../Header\_files/set\_test.h"

unsigned set\_add\_test() {

set<float> s;

unsigned time\_begin = clock();

for (int i = 0; i < 1000000000; i++) {

s.insert(rand() / RAND\_MAX);

}

unsigned time\_end = clock() - time\_begin;

return time\_end;

}

unsigned set\_delete\_test() {

set<float> s;

for (int i = 0; i < 1000000000; i++) {

s.insert(i / 127);

}

unsigned time\_begin = clock();

for (int i = 0; i < 1000000000; i++) {

s.erase(i / 127);

}

unsigned time\_end = clock() - time\_begin;

return time\_end;

}

unsigned set\_search\_test() {

set<float> s;

for (int i = 0; i < 1000000000; i++) {

if (i == 1324235) {

s.insert(1.0);

continue;

}

s.insert(rand() / RAND\_MAX);

}

unsigned time\_begin = clock();

s.count(1.0);

unsigned time\_end = clock() - time\_begin;

return time\_end;

}

**//stack\_test.cpp**

#include "../Header\_files/stack\_test.h"

unsigned stack\_add\_test() {

stack<float> s;

unsigned time\_begin = clock();

for (int i = 0; i < 1000000000; i++) {

s.push(rand() / RAND\_MAX);

}

unsigned time\_end = clock() - time\_begin;

return time\_end;

}

unsigned stack\_delete\_test() {

stack<float> s;

for (int i = 0; i < 1000000000; i++) {

s.push(rand() / RAND\_MAX);

}

unsigned time\_begin = clock();

for (int i = 0; i < 1000000000; i++) {

s.pop();

}

unsigned time\_end = clock() - time\_begin;

return time\_end;

}

Выводы:

В данной лабораторной работе были изучены основные принципы работы с библиотекой шаблонных классов STL (Standard Template Library).

В первой части работы был рассмотрен класс "Стек" из библиотеки STL. Стек представляет собой контейнер, основанный на принципе "последним пришел - первым вышел" (LIFO - Last In, First Out). Было изучено использование стека для различных задач.

Во второй части работы был рассмотрено множество с дубликатами. Множество представляет собой контейнер, который содержит только уникальные элементы. Однако, в отличие от классического множества, множество с дубликатами позволяет хранить повторяющиеся элементы.

Шаблонные классы позволяют создавать обобщенные классы, которые могут работать с различными типами данных. Благодаря этому, можно обобщить решение задачи и сделать его более универсальным.

В ходе лабораторной работы была изучена структура и основные методы работы с классами "Стек" и "Множество с дубликатами". Были выполнены задачи, связанные с использованием этих классов, что позволило закрепить полученные знания. Таким образом, данная лабораторная работа позволила изучить базовые принципы работы с библиотекой шаблонных классов STL на примере классов "Стек" и "Множество с дубликатами". Полученные знания и навыки могут быть использованы при разработке различных программных решений на языке C++.

**Опирались на данные:**

**1)**



**2)**

