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 № 18.13**

Дисциплина: Информатика

# Тема: «Объектно-ориентированное программирование Функциональные объекты и предикаты»

Вариант 1

Выполнил работу:

студент группы РИС-20-2Б

Азмагулов Артём Вадимович

Проверила:

Доцент кафедры ИТАС

Полякова О.А.

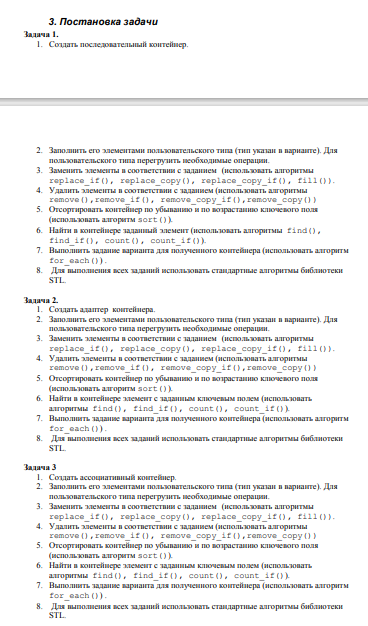
Пермь

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**



**Анализ задачи**

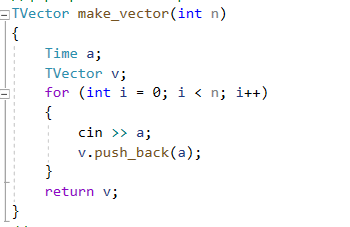
1. Необходимые действия

Задача 1

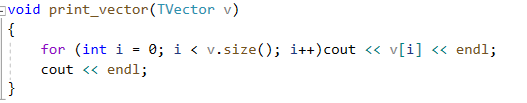
1. Ввести количество элементов во множестве



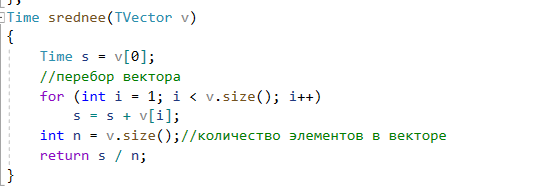
1. Написать функцию ввода элементов во множество



1. Вывести содержимое множества на консоль



1. Найти среднее арифметическое элементов множества

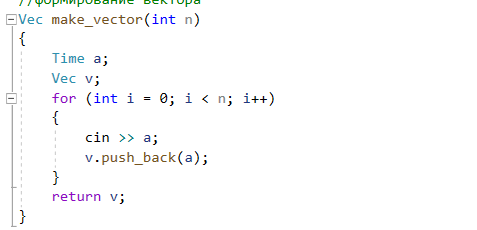


Задача 2

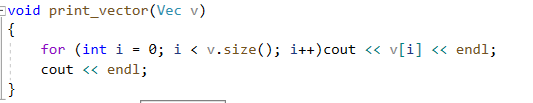
1. Ввести количество элементов во множестве

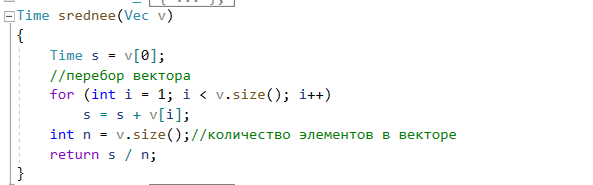


1. Написать функцию ввода элементов во множество



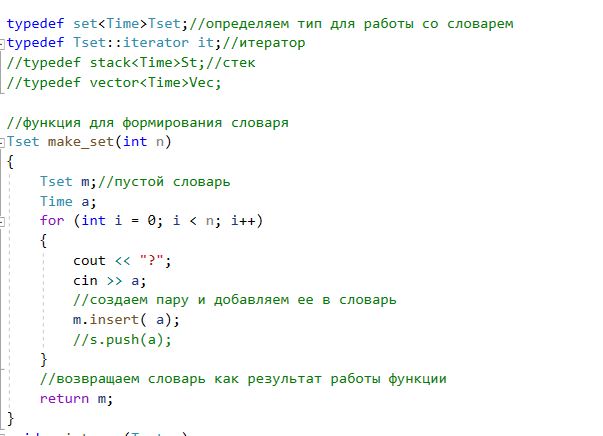
1. Вывести содержимое множества на консоль



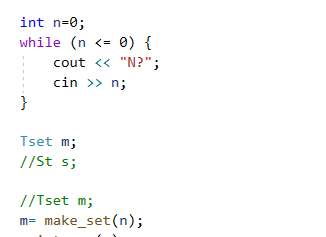
1. Найти среднее арифметическое элементов множества 

Задача 3

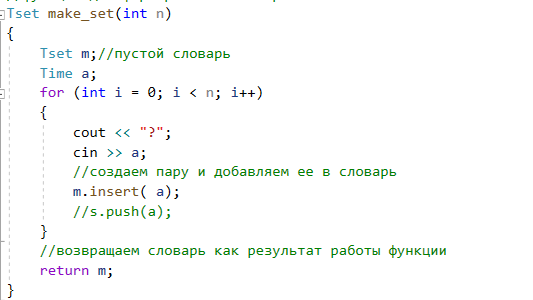
1. Создать класс-вектор для работы со множеством, содержащим элементы любого типа (в данном случае int, double)



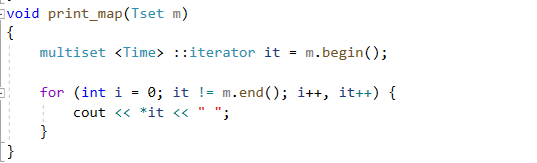
1. Ввести количество элементов в векторе



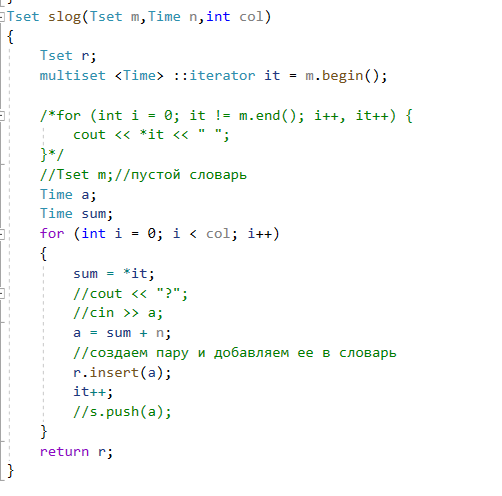
1. Написать функцию ввода элементов во множество



1. Вывести содержимое множества на консоль



1. Ввести значение и сложить его со всеми остальными элементами



1. Программный код

Задача 1

#include "Time.h"

#include "vector"

#include <iostream>

#include <algorithm>

using namespace std;

typedef vector<Time> TVector;

Time s;

//формирование вектора

TVector make\_vector(int n)

{

Time a;

TVector v;

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

{

cin >> a;

v.push\_back(a);

}

return v;

}

//печать вектора

void print\_vector(TVector v)

{

for (int i = 0; i < v.size(); i++)cout << v[i] << endl;

cout << endl;

}

struct Greater\_s //больше, чем s

{

bool operator()(Time t)

{

if (t > s) return true; else return false;

}

};

Time srednee(TVector v)

{

Time s = v[0];

//перебор вектора

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

s = s + v[i];

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

return s / n;

}

struct Comp\_less // для сортировки по убыванию

{

bool operator()(Time t1, Time t2)

{

if (t1 > t2)return true;

else return false;

}

};

struct Equal\_s

{

bool operator()(Time t)

{

if (t == s == 0) {

return true;

}

else {

return false;

}

}

};

void main()

{

setlocale(LC\_ALL, "rus");

int n;

cout << "N?";

cin >> n;

TVector v;

v = make\_vector(n);

print\_vector(v);

TVector::iterator i;

//поставили итератор i на максимальный элемент

i = max\_element(v.begin(), v.end());

cout << "max=" << \*(i) << endl;

Time m = \*(i);

s = m;//нашли среднее арифметическое вектора

/\*cout << "sred=" << s << endl;

//замена с использованием предиката

replace\_if(v.begin(), v.end(), Greater\_s(), m);

cout << "ZAMENA" << endl;

print\_vector(v);

\*/

cout << "ZAMENA" << endl;

Time r;

cin >> r;

replace\_if(v.begin(), v.end(), Equal\_s(), r);

print\_vector(v);

//по убыванию

cout << "Sortirovka po ubivaniu:" << endl;

sort(v.begin(), v.end(), Comp\_less());

print\_vector(v);

//по возрастанию

cout << "Sortirovka po vozrasaniu:" << endl;

sort(v.begin(), v.end());

print\_vector(v);

cout << "POISK" << endl;

cin >> s;

//поиск элементов, удовлетворяющих условию предиката

i = find\_if(v.begin(), v.end(), Equal\_s());

if (i != v.end())//если нет конца вектора

cout << \*(i) << endl;

else

cout << "Not such element!" << endl;

cout << "UDALENIE" << endl;

i = min\_element(v.begin(), v.end());

s = \*i;

//переместили элементы совпадающие с min в конец вектора

i = remove\_if(v.begin(), v.end(), Equal\_s());

//удалили элементы, начиная с i и до конца вектора

v.erase(i, v.end());

print\_vector(v);

}Задача 2

// 18.13(2).cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

// 18.11(4).cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#include "Time.h"

#include "stack"

#include "vector"

#include <iostream>

#include <algorithm>

using namespace std;

typedef stack<Time> St;

typedef vector<Time> Vec;

Time s;

//формирование вектора

Vec make\_vector(int n)

{

Time a;

Vec v;

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

{

cin >> a;

v.push\_back(a);

}

return v;

}

//печать вектора

void print\_vector(Vec v)

{

for (int i = 0; i < v.size(); i++)cout << v[i] << endl;

cout << endl;

}

struct Greater\_s //больше, чем s

{

bool operator()(Time t)

{

if (t > s) return true; else return false;

}

};

Time srednee(Vec v)

{

Time s = v[0];

//перебор вектора

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

s = s + v[i];

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

return s / n;

}

struct Comp\_less // для сортировки по убыванию

{

bool operator()(Time t1, Time t2)

{

if (t1 > t2)return true;

else return false;

}

};

struct Equal\_s

{

bool operator()(Time t)

{

if (t == s == 0) {

return true;

}

else {

return false;

}

}

};

St make\_stack(int n)

{

St s;

Time t;

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

{

cin >> t;//ввод переменной

s.push(t);//добавление ее в стек

}

return s;//вернуть стек как результат функции

}

Vec copy\_stack\_to\_vector(St s)

{

Vec v;

while (!s.empty())//пока стек не пустой

{

//добавить в вектор элемент из вершиы стека

v.push\_back(s.top());

s.pop();

}

return v; //вернуть вектор как результат функции

}

//копирует вектор в стек

St copy\_vector\_to\_stack(Vec v)

{

St s;

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

{

s.push(v[i]);//добавить в стек элемент вектора

}

return s; //вернуть стек как результат функции

}

void print\_stack(St s) {

St b;

Vec v;

v = copy\_stack\_to\_vector(s);

b = copy\_vector\_to\_stack(v);

while (!b.empty())

{

cout << b.top() << " ";

b.pop();

}

}

Time Max(St s)

{

Time m = s.top();//переменной m присваиваем значение из вершины стека

Vec v = copy\_stack\_to\_vector(s);//копируем стек в вектор

while (!s.empty())//пока стк не пустой

{

if (s.top() > m)m = s.top();//срвниваем m и элемнт в вершине стека

s.pop();//удаляем элемент из стека

}

s = copy\_vector\_to\_stack(v);//копиуем вектор в стек

return m; //возвращаем m

}

int main()

{

setlocale(LC\_ALL, "rus");

int n;

cout << "N?";

cin >> n;

Vec v;

v = make\_vector(n);

print\_vector(v);

Vec::iterator i;

//поставили итератор i на максимальный элемент

i = max\_element(v.begin(), v.end());

cout << "max=" << \*(i) << endl;

Time m = \*(i);

s = m;//нашли среднее арифметическое вектора

/\*cout << "sred=" << s << endl;

//замена с использованием предиката

replace\_if(v.begin(), v.end(), Greater\_s(), m);

cout << "ZAMENA" << endl;

print\_vector(v);

\*/

cout << "ZAMENA max na new :" << endl;

Time r;

cin >> r;

replace\_if(v.begin(), v.end(), Equal\_s(), r);

St st;

st = copy\_vector\_to\_stack( v);

print\_stack(st);

cout << endl<< "Delete min : ";

i = min\_element(v.begin(), v.end());

s = \*i;

v = copy\_stack\_to\_vector(st);

cout << endl;

//переместили элементы совпадающие с min в конец вектора

i = remove\_if(v.begin(), v.end(), Equal\_s());

//удалили элементы, начиная с i и до конца вектора

v.erase(i, v.end());

st = copy\_vector\_to\_stack(v);

print\_stack(st);

// по убыванию

cout << endl << "Sortirovka po ubivaniu:" << endl;

v = copy\_stack\_to\_vector(st);

sort(v.begin(), v.end(), Comp\_less());

st = copy\_vector\_to\_stack(v);

print\_stack(st);

//по возрастанию

cout << endl<< "Sortirovka po vozrasaniu:" << endl;

v = copy\_stack\_to\_vector(st);

sort(v.begin(), v.end());

st = copy\_vector\_to\_stack(v);

print\_stack(st);

i = find\_if(v.begin(), v.end(), Equal\_s());

if (i != v.end())//если нет конца вектора

cout << \*(i) << endl;

else

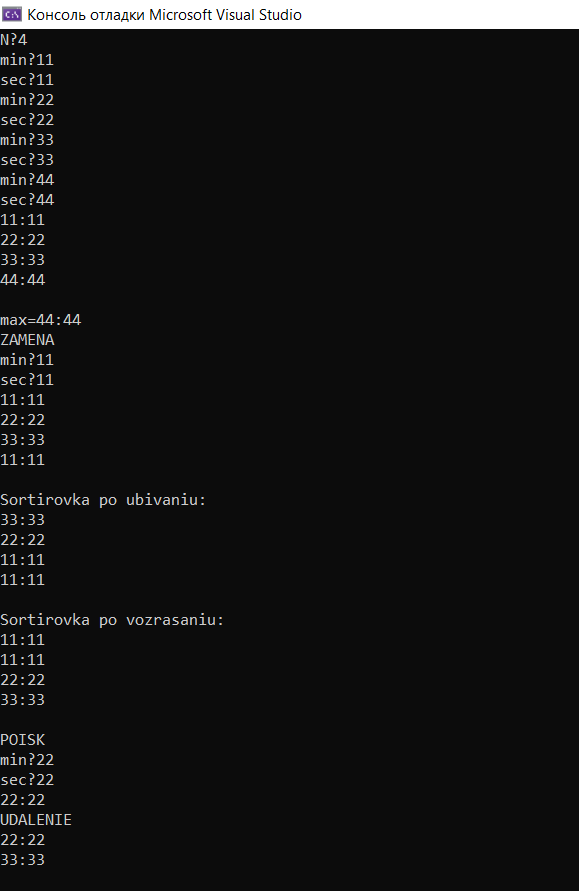
cout << "Not such element!" << endl;

}

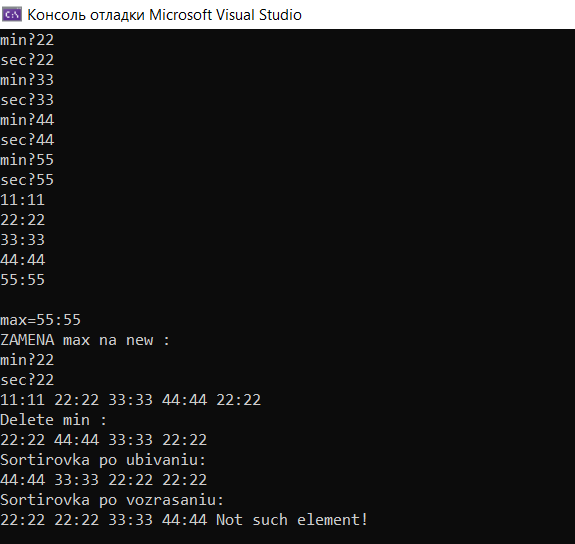
Задача 3

1. Демонстрация работы программы

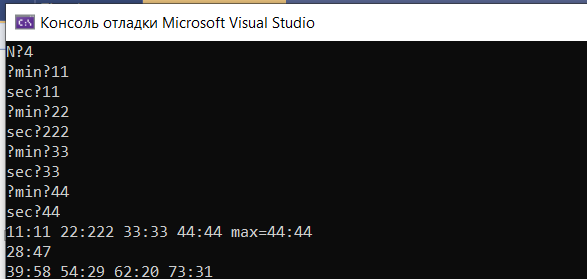
Задача 1



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