Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**Пермский национальный исследовательский политехнический университет**

Факультет Электротехнический Кафедра ИТАС

Специальность Промышленная Робототехника

**ОТЧЁТ**

**о лабораторной работе №11**

Перегрузка операций

|  |  |
| --- | --- |
|  | Выполнил:  Студент группы ПРТ-21-1Б  Торган Г.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Проверил:  Доцент кафедры ИТАС Полякова О.А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Пермь 2022**

**Постановка задачи:**

Задача 1

1. Контейнер - вектор

2. Тип элементов - float

Задача 2

Тип элементов Pair (см. лабораторную работу No3).

Задача 3

Параметризированный класс – Список (см. лабораторную работу No7)

Задача 4

Адаптер контейнера – очередь.

Задача 5

Параметризированный класс – Список

Адаптер контейнера – очередь.

Задание 3 Задание 4 Задание 5

14

Найти минимальный

элемент и добавить его на

заданную позицию

контейнера

Найти меньше среднего

арифметического и

удалить их из контейнера

Каждый элемент разделить на

максимальный элемент

контейнера.

**Текст программы:**

**Lab11.cpp**

#include <iostream>

#include <vector>

#include <cstdlib>

using namespace std;

typedef vector<int>Vec;

Vec make\_vector(int n)

{

Vec v;

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

{

int a = rand() % 100 - 50;

v.push\_back(a);

}

return v;

}

void print\_vector(Vec v)

{

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

cout << v[i] << " ";

cout << endl;

}

int srednee(Vec v)

{

int s = 0;

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

s += v[i];

int n = v.size();

return s / n;

}

void add\_vector(Vec& v, int el, int pos)

{

v.insert(v.begin() + pos, el);

}

int max(Vec v)

{

int m = v[0],

n = 0;

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

if (m < v[i])

{

m = v[i];

n = i;

}

return n;

}

void del\_vector(Vec& v, int pos)

{

v.erase(v.begin() + pos);

}

int min(Vec v)

{

int m = v[0],

n = 0;

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

if (m > v[i])

{

m = v[i];

n = i;

}

return n;

}

void delenie(Vec& v)

{

int m = min(v);

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

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

}

int main()

{

try

{

vector<int> v;

vector<int>::iterator vi = v.begin();

int n;

cout << "N?"; cin >> n;

v = make\_vector(n);

print\_vector(v);

int el = srednee(v);

cout << "pos?";

int pos;

cin >> pos;

if (pos > v.size())throw 1;

add\_vector(v, el, pos);

print\_vector(v);

int n\_max = max(v);

del\_vector(v, n\_max);

print\_vector(v);

delenie(v);//

print\_vector(v);

}

catch (int)

{

cout << "error!";

}

return 0;

}

**Ответ для варианта №14**

