Министерство высшего образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (ПНИПУ)

Электротехнический факультет

Дисциплина: основы алгоритмизации и программирования, 2 семестр

**ОТЧЁТ**

Тема: «Шаблоны классов»

Выполнил

Студент РИС-22-2б

Зубов Р.А.

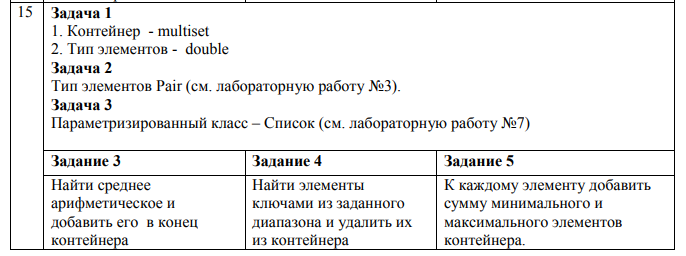
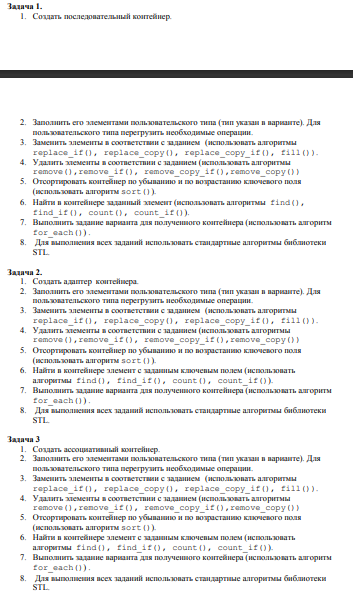
Проверил

Доцент кафедры ИТАС

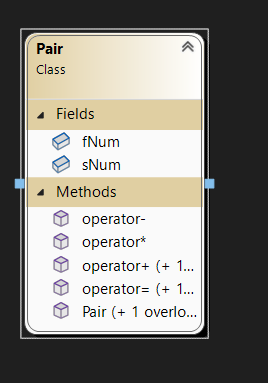
Полякова О. А.

Пермь 2023

**Постановка задачи**

****

**UML**

****

**Код программы**

**Pair.h:** #pragma once

#include <iostream>

class Pair

{

public:

int fNum = 0;

double sNum = 0;

Pair() {}

Pair(int first, double second) {

fNum = first;

sNum = second;

}

Pair& operator=(const Pair& t);

Pair& operator=(const int& t);

Pair operator\*(Pair& t);

friend std::ostream& operator<<(std::ostream& out, const Pair& t);

friend std::istream& operator>>(std::istream& in, const Pair& t);

};

**Pair.cpp:**

#include "Pair.h"

Pair& Pair::operator=(const Pair& pair) {

this->fNum = pair.fNum;

this->sNum = pair.sNum;

return \*this;

}

Pair& Pair::operator=(const int& t) {

this->fNum = t;

this->sNum = t;

return \*this;

}

Pair Pair::operator\*(Pair& pair) {

Pair new\_pair;

new\_pair.fNum = this->fNum \* pair.fNum;

new\_pair.sNum = this->sNum \* pair.sNum;

return new\_pair;

}

std::ostream& operator<<(std::ostream& out, const Pair& pair) {

out << pair.fNum << " : " << pair.sNum;

return out;

}

std::istream& operator>>(std::istream& in, Pair& pair) {

in >> pair.fNum >> pair.sNum;

return in;

}

**Source.cpp:**

#include <iostream>

#include <algorithm>

#include <list>

#include <queue>

#include <map>

#include "Pair.h"

using namespace std;

template<typename T>

void print(list<T> S)

{

for (auto it = S.begin(); it != S.end(); ++it) {

cout << \*it << " | ";

}

cout << endl;

}

void print(priority\_queue<Pair> pq)

{

while (!pq.empty()) {

cout << pq.top() << " | ";

pq.pop();

}

cout << endl;

}

void print(map<int, Pair> M)

{

for (const auto& [key, value] : M) {

cout << key << " - " << value << " | ";

}

cout << endl;

}

//Task 1

void task1()

{

list<Pair> L;

L.push\_back(Pair(1, 1.1));

L.push\_back(Pair(2, 2.2));

L.push\_back(Pair(3, 3.3));

L.push\_back(Pair(4, 4.4));

print(L);

//find avg

Pair sum(0, 0.0);

for (auto it = L.begin(); it != L.end(); ++it)

sum = sum + \*it;

sum.fNum /= L.size(); sum.sNum /= L.size();

L.push\_back(sum);

print(L);

////find key

Pair key1, key2;

cout << "Enter keys to delete: "; cin >> key1; cin >> key2;

L.erase(remove\_if(L.begin(), L.end(), [&key1, &key2](const Pair& p) {

return (p >= key1 && p <= key2);

}));

print(L);

//sum max min

cout << "Adding sum to elems: \n";

Pair MMsum = \*min\_element(L.begin(), L.end())

+ \*max\_element(L.begin(), L.end());

for\_each(L.begin(), L.end(), [&MMsum](Pair& p) {p = p + MMsum; });

print(L);

}

//Task 2

void task2() {

priority\_queue<Pair> Q;

Q.push(Pair(1, 1.1));

Q.push(Pair(2, 2.2));

Q.push(Pair(3, 3.3));

Q.push(Pair(4, 4.4));

print(Q);

priority\_queue<Pair> Qbuf;

Pair sum(0, 0.0);

while (!Q.empty())

{

sum = sum + Q.top();

Qbuf.push(Q.top());

Q.pop();

}

sum.fNum /= Qbuf.size(); sum.sNum /= Qbuf.size();

Qbuf.push(sum);

print(Qbuf);

Pair key1, key2;

cout << "Enter keys to delete: "; cin >> key1; cin >> key2;

while (!Qbuf.empty())

{

Pair check = Qbuf.top();

Qbuf.pop();

if (check > key1 || check < key2)

Q.push(check);

}

print(Q);

cout << "Adding sum to elems: \n";

Pair min = Qbuf.top();

Pair max = Qbuf.top();

while (!Qbuf.empty())

{

if (max < Qbuf.top()) max = Qbuf.top();

if (Qbuf.top() < min) min = Qbuf.top();

Q.push(Qbuf.top());

Qbuf.pop();

}

while (!Q.empty())

{

Qbuf.push(Q.top() + min + max);

Q.pop();

}

print(Qbuf);

}

//Task 3

void task3() {

map <int, Pair> M = { {1, Pair(1, 1.1)}, {2, Pair(2, 2.2)} ,

{3, Pair(3, 3.3)} , {4, Pair(4, 4.4)} , {5, Pair(5, 5.5)} };

print(M);

Pair sum(0, 0.0);

for (auto& it : M)

sum = sum + it.second;

sum.fNum /= M.size();

M[6] = (sum.fNum /= M.size(), sum.sNum /= M.size());

print(M);

int key1, key2;

cout << "Enter keys to delete: "; cin >> key1; cin >> key2;

M.erase(key1); M.erase(key2);

print(M);

cout << "Adding sum of max min\n";

map<int, Pair>::iterator max = max\_element(M.begin(), M.end(),

[](const auto& a, const auto& b) { return a.second < b.second; });

map<int, Pair>::iterator min = min\_element(M.begin(), M.end(),

[](const auto& a, const auto& b) { return a.second < b.second; });

Pair SumP = max->second + min->second;

for (auto& it : M) {

it.second += SumP;

}

print(M);

}

int main()

{

srand(time(0));

cout << "Results: \n";

cout << "===================\n\n";

cout << "------TASK1------\n";

task1();

cout << "------TASK2------\n";

task2();

cout << "------TASK3------\n";

task3();

return 0;

}

**Результаты программ**

