ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА

ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



ЗВІТ

про виконання лабораторної роботи № 1,5

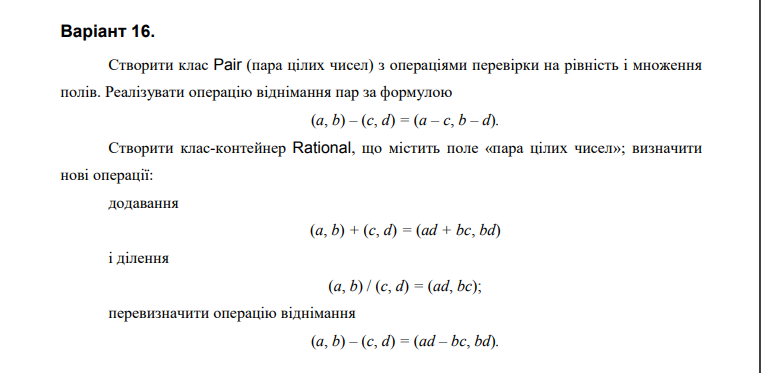
«Композиція класів та об’єктів»

з дисципліни «Об’єктно-орієнтоване програмування»

студентки групи ІТ-12

Прендкович Ірини Віталіївни

Умова завдання



Відповідь

Pair.h

////////////////////////////////////////////////////////////////////////////////////////

#pragma once

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

class Pair

{

private:

int first;

int second;

public:

double GetA() const { return first; }

double GetB() const { return second; }

void SetA(int value);

void SetB(int value);

bool Init(int first, int second);

void Display() const;

void Read();

bool equal(Pair a, Pair b);

Pair minus(Pair a, Pair b);

Pair multiply(Pair a, Pair b);

};

Pair.cpp

////////////////////////////////////////////////////////////////////////////////////////

#include "Pair.h"

#include <iostream>

using namespace std;

void Pair::SetA(int value)

{

first = value;

}

void Pair::SetB(int value)

{

second = value;

}

bool Pair::Init(int first1, int second1)

{

if ((-100 <= first <= 100) && (-100 <= second <= 100))

{

first = first1;

second = second1;

return true;

}

else

return false;

}

void Pair::Read()

{

int a, b;

do

{

cout << " enter a = "; cin >> a;

cout << "enter b = "; cin >> b;

} while (!(Init(a, b)));

}

Pair Pair::minus(Pair a, Pair b)

{

Pair tmp;

tmp.first = a.first - b.first;

tmp.second = a.second - b.second;

cout << "Minus = " << endl;

return tmp;

}

Pair Pair::multiply(Pair a, Pair b)

{

Pair tmp;

tmp.first = a.first \* b.first;

tmp.second = a.second \* b.second;

cout << "Multiply = " << endl;

return tmp;

}

void Pair::Display() const

{

cout << "a = " << first << endl;

cout << "b = " << second << endl;

}

bool Pair::equal(Pair a, Pair b)

{

if (a.first == b.first && a.second == b.second )

return true;

else

return false;

}

Rational.h

////////////////////////////////////////////////////////////////////////////////////////

#pragma once

#include "Pair.h"

class Rational

{

private:

Pair pair;

public:

double GetA() const { return pair.GetA(); }

double GetB() const { return pair.GetB(); }

void SetA(int value);

void SetB(int value);

void Init(int a, int b);

void Display() const;

void Read();

Rational minus(Rational a, Rational b);

Rational plus(Rational a, Rational b);

Rational div(Rational a, Rational b);

};

Rational.cpp

////////////////////////////////////////////////////////////////////////////////////////

#include "Rational.h"

#include <iostream>

#include <math.h>

using namespace std;

void Rational::SetA(int value)

{

if (value > 0)

pair.SetA(value);

else

pair.SetA(0);

}

void Rational::SetB(int value)

{

if (value > 0)

pair.SetB(value);

else

pair.SetB(0);

}

void Rational::Init(int a, int b)

{

SetA(a);

SetB(b);

}

void Rational::Read()

{

double A, B;

cout << "a = "; cin >> A;

cout << "b = "; cin >> B;

}

void Rational::Display() const

{

pair.Display();

}

Rational Rational::minus(Rational a, Rational b)

{

pair.SetA(a.GetA() \* b.GetB() - a.GetB() \* b.GetA());

pair.SetB(a.GetB() \* b.GetB());

cout << "minus = " << endl;

return \*this;

}

Rational Rational::plus(Rational a, Rational b)

{

pair.SetA(a.GetA() \* b.GetB() + a.GetB() \* b.GetA());

pair.SetB(a.GetB() \* b.GetB());

cout << "plus = " << endl;

return \*this;

}

Rational Rational::div(Rational a, Rational b)

{

pair.SetA(a.GetA() \* b.GetB());

pair.SetB(a.GetB() \* b.GetA());

cout << "div = " << endl;

return \*this;

}

////////////////////////////////////////////////////////////////////////////////////////

Lab 1.5

/////////////////////////////////////////////////////////////////////////////////////////

#include "Rational.h"

#include "Pair.h"

#include <iostream>

#include <Windows.h>

#include <iomanip>

using namespace std;

int main()

{

Pair pair;

Pair a, b, c, d, m;

pair.Read();

a.Read();

b.Read();

pair.Display();

c = d.minus(a, b);

c.Display();

m = d.multiply(a, b);

m.Display();

if (pair.equal(a, b))

cout << "values are equal" << endl;

else

cout << "values are not equal" << endl;

Rational x, y, R, P, D, M;

x.SetA(2);

y.SetB(7);

x.SetB(3);

y.SetA(5);

P = R.plus(x, y);

P.Display();

D = R.div(x, y);

D.Display();

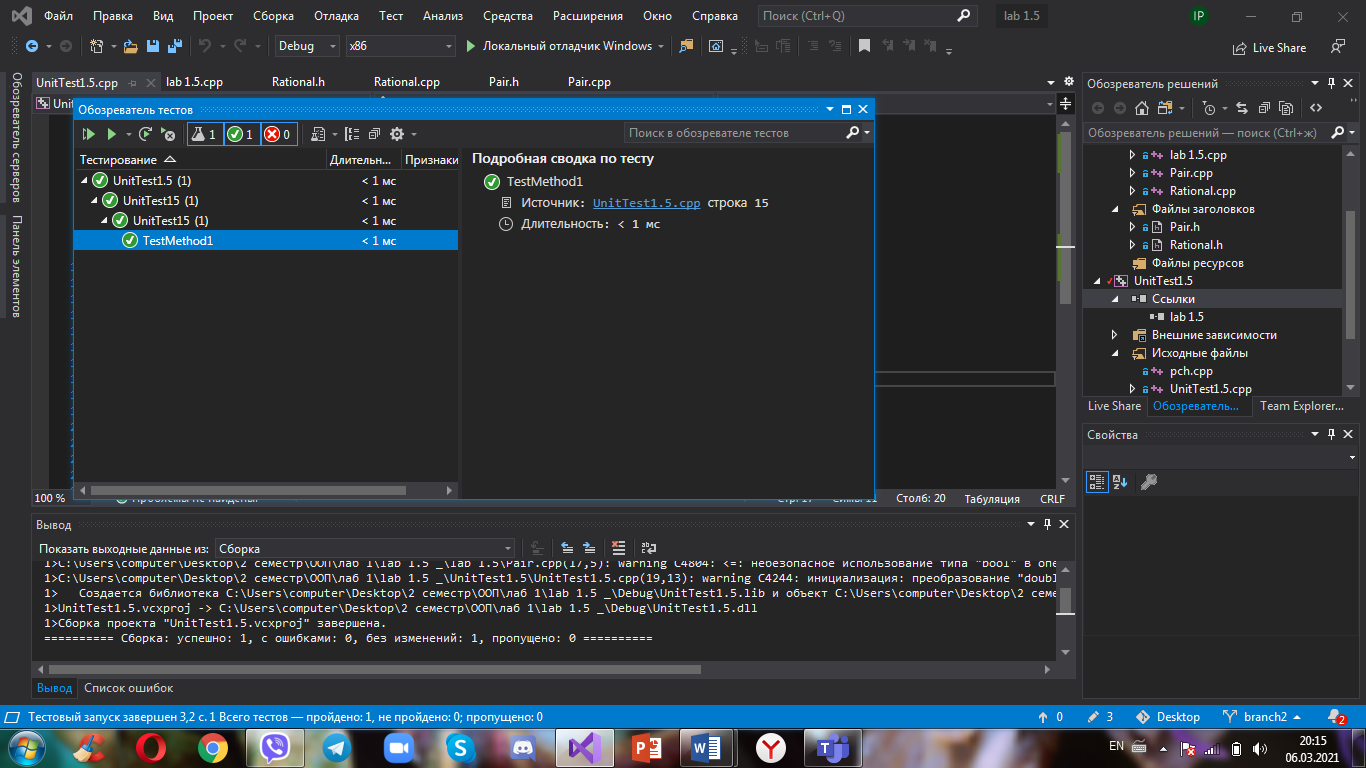
M = R.minus(x, y);

M.Display();

cin.get();

return 0;

}



#include "pch.h"

#include "CppUnitTest.h"

#include "../lab 1.5/Pair.h"

#include "../lab 1.5/Pair.cpp"

#define TESTING

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest15

{

TEST\_CLASS(UnitTest15)

{

public:

TEST\_METHOD(TestMethod1)

{

Pair c;

c.SetA(2);

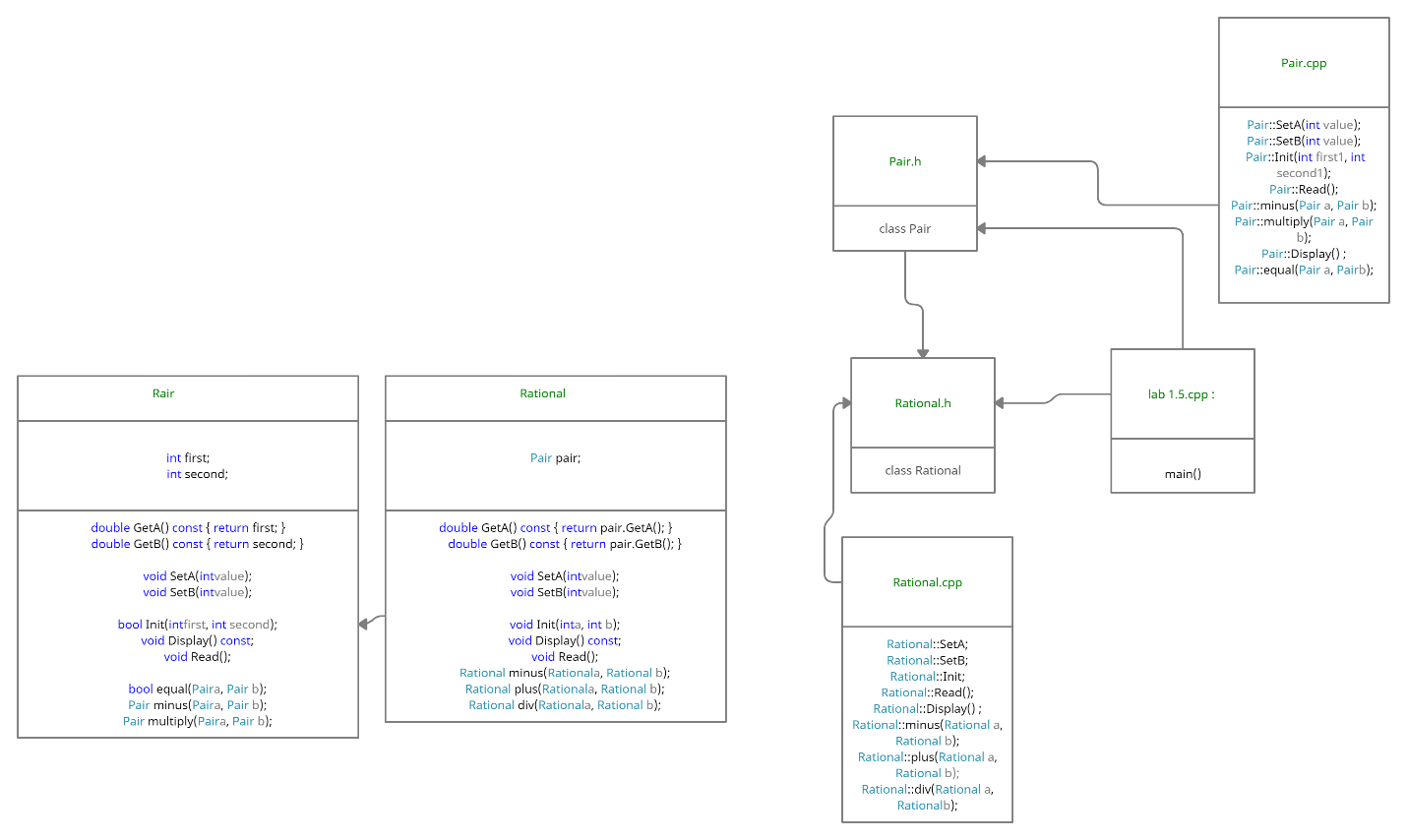
int test = c.GetA();

Assert::AreEqual(test, 2);

}

};

}



Висновок

На цій лабораторній роботі я освоїла використання композитних класів та об’єктів.