

**ЗВІТ**

**Про виконання лабораторної роботи № 1.5**

**«Композиція класів та об’єктів»**

**З дисципліни**

**«Об’єктно-орієнтоване програмування»**

**Студентки групи ІТ-12**

**Шкринда Володимир Володимирович**

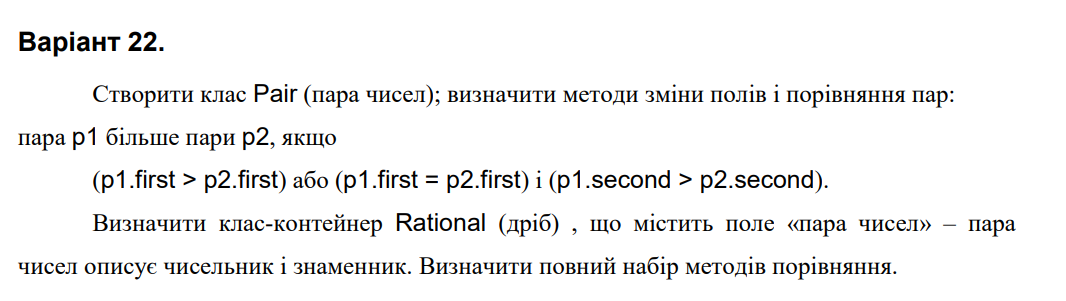
**Прийняв викладач**

**Майба І.В.**

**2021**

**Мета роботи:** Освоїти використання композитних класів та об’єктів.

**Умова завдання:**



**Текст програми:**

**Source.cpp**

#include <iostream>

#include "Rational.h"

#include "Pair.h"

int main()

{

Pair a, b, c, d;

Rational f;

int result;

cout<< endl;

cout << "Pair a " << endl;

cout << "(x/y)" << endl;

a.Init(1, 2);

a.Display();

cout << endl;

cout << "Pair b " << endl;

cout << "(x/y)" << endl;

b.Init(2, 1);

b.Display();

cout << endl;

cout << "Pair c" << endl;

cout << "(x/y)" << endl;

c.Init(1, 2);

c.Display();

cout << endl;

cout << "a and b" << endl;

result = f.ComparePairs(a, b);

f.ComparePairsResult(result);

cout << endl;

cout << "a and c" << endl;

result = f.ComparePairs(a, c);

f.ComparePairsResult(result);

cout << endl;

cout << "b and c" << endl;

result = f.ComparePairs(b, c);

f.ComparePairsResult(result);

return 0;

}

**Pair.h**

#pragma once

#include <iostream>

#include <sstream>

#include <string>

using namespace std;

class Pair

{

double first, second;

public:

void SetFirst(double value);

void SetSecond(double value);

double GetFirst() const;

double GetSecond() const;

void Init(double x, double y);

void Read();

void Display();

string toString() const;

};

**Pair.cpp**

#include "Pair.h"

#include <iostream>

using namespace std;

void Pair::SetFirst(double value) { first = value; }

void Pair::SetSecond(double value) { second = value; }

double Pair::GetFirst() const { return first; }

double Pair::GetSecond() const { return second; }

void Pair::Init(double x, double y)

{

first = x;

second = y;

}

void Pair::Read()

{

cout << "First = "; cin >> first;

cout << "Second = "; cin >> second;

}

void Pair::Display()

{

cout << endl;

cout << "First = " << GetFirst() << endl;

cout << "Second = " << GetSecond() << endl;

}

string Pair::toString() const

{

stringstream sout;

cout << endl;

sout << "First = " << GetFirst() << endl;

sout << "Second = " << GetSecond() << endl;

return sout.str();

}

**Rational.h**

#pragma once

#include "Pair.h"

class Rational

{

Pair p;

public:

void SetP(double first, double second);

Pair GetP() const;

void Display();

void Read();

string toString() const;

double ComparePairs(Pair p1, Pair p2) const;

void ComparePairsResult(int result);

};

**Rational.cpp**

#include "Rational.h"

#include "Pair.h"

#include <iostream>

void Rational::SetP(double first, double second)

{

p.Init(first, second);

}

Pair Rational::GetP() const { return p; }

void Rational::Display()

{

cout << endl;

cout << "First = " << p.GetFirst() << endl;

cout << "Second = " << p.GetSecond() << endl;

}

void Rational::Read()

{

double x, y;

cout << "First = "; cin >> x; p.SetFirst(x);

cout << "Second = "; cin >> y; p.SetSecond(y);

}

string Rational::toString() const

{

stringstream sout;

cout << endl;

sout << "First = " << p.GetFirst() << endl;

sout << "Second = " << p.GetSecond() << endl;

return sout.str();

}

double Rational::ComparePairs(Pair p1, Pair p2) const

{

if (p1.GetFirst() > p2.GetFirst() ||

(p1.GetFirst() == p2.GetFirst() && p1.GetSecond() > p2.GetSecond()))

return 1;

if (p1.GetFirst() < p2.GetFirst() ||

(p1.GetFirst() == p2.GetFirst() && p1.GetSecond() < p2.GetSecond()))

return 2;

if (p1.GetFirst() == p2.GetFirst() &&

p1.GetSecond() == p2.GetSecond())

return 3;

}

void Rational::ComparePairsResult(int result)

{

switch (result)

{

case 1:

cout << endl;

cout << "P1 > P2" << endl;

break;

case 2:

cout << endl;

cout << "P1 < P2" << endl;

break;

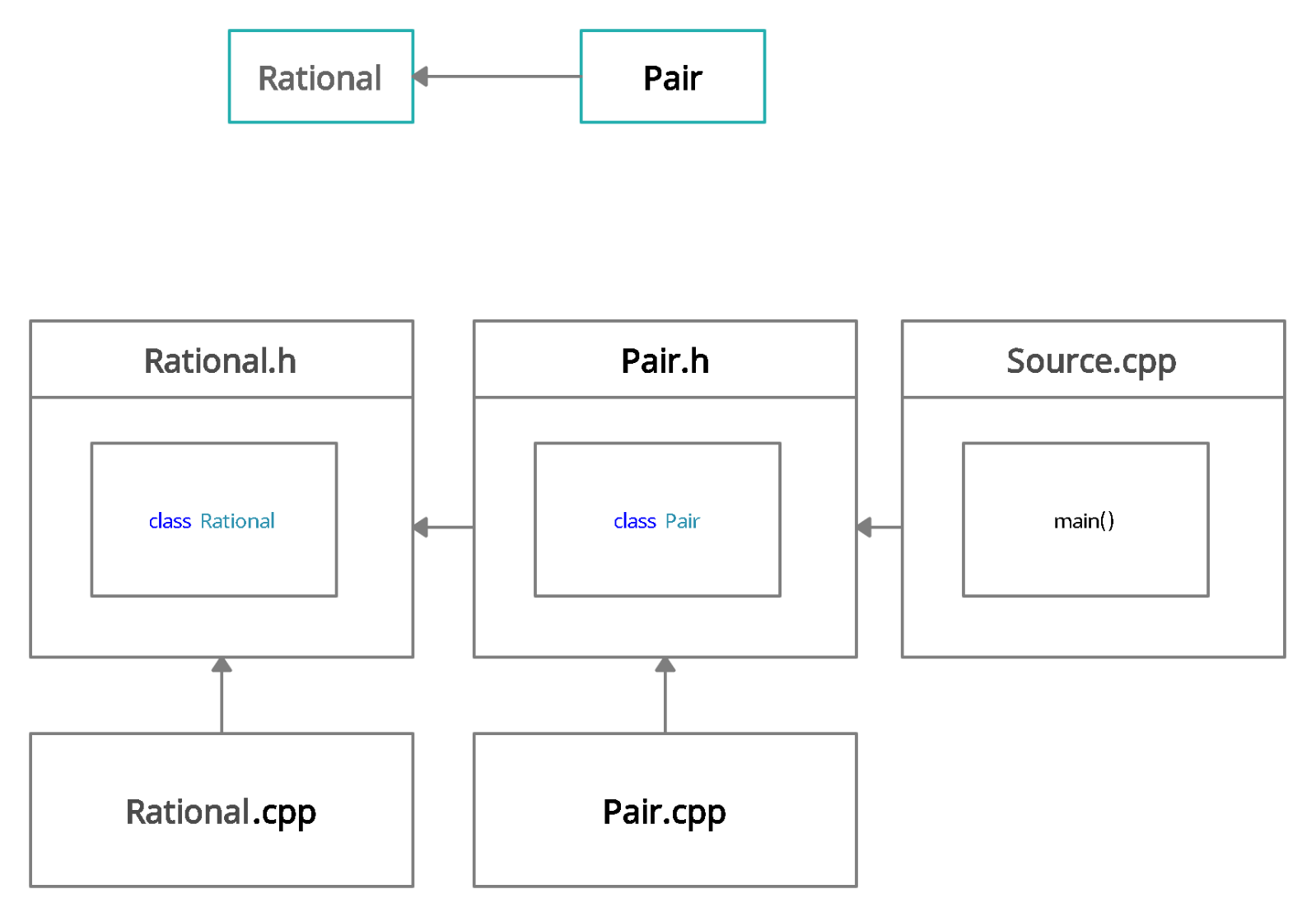
case 3:

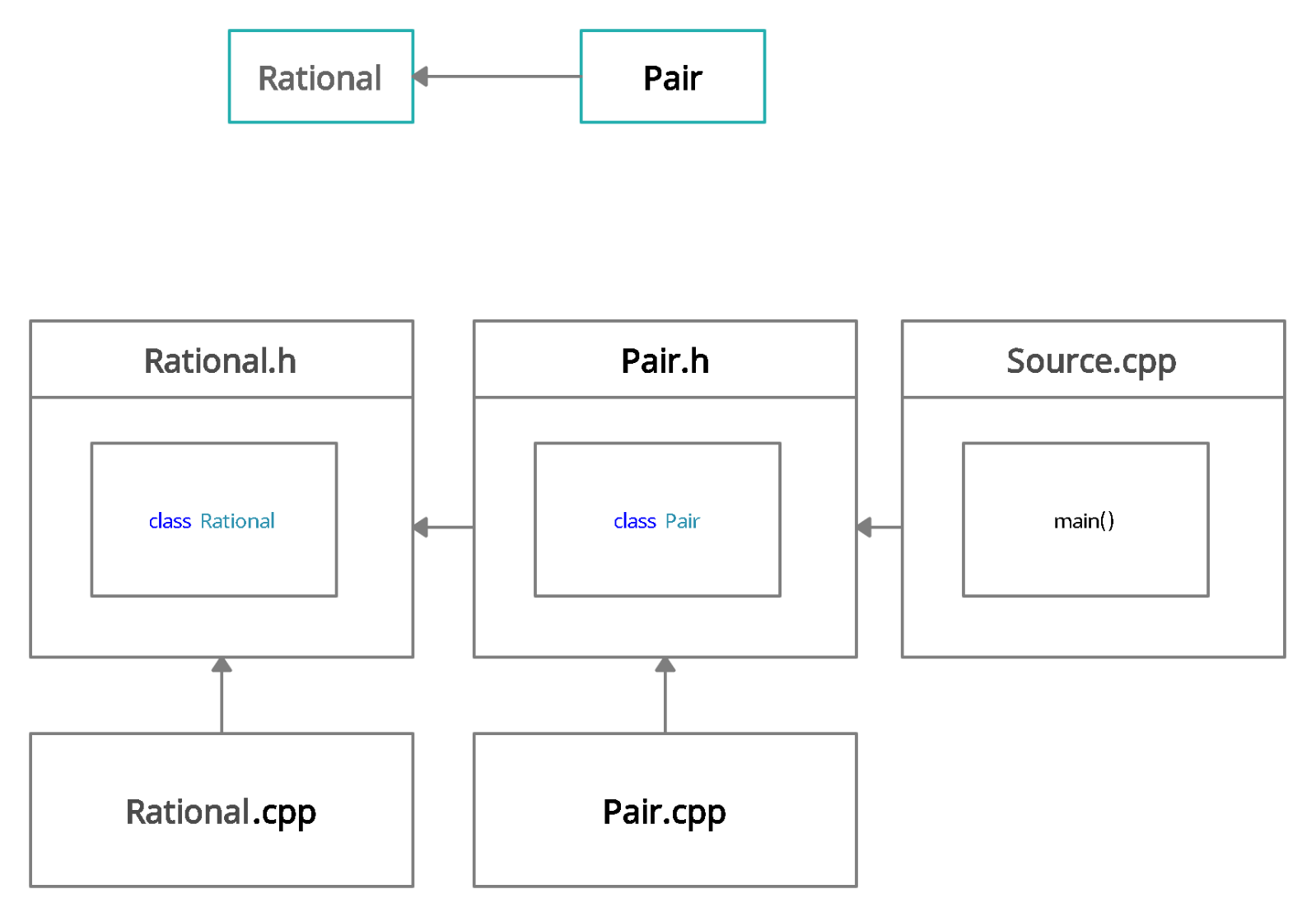
cout << endl;

cout << "P1 = P2" << endl;

break;

}

}**UML-діаграма класів: **

**Структурна схема програми:** 

**Результати Unit тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "../Lab1\_5/Rational.cpp"

#include "../Lab1\_5/Pair.cpp"

#include "../Lab1\_5/Pair.h"

#include "../Lab1\_5/Rational.h"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest1

{

TEST\_CLASS(UnitTest1)

{

public:

TEST\_METHOD(TestMethod1)

{

Pair a, b;

Rational f;

a.Init(4, 3);

b.Init(2, 1);

double res1, res2;

res1 = f.ComparePairs(a, b);

res2 = f.ComparePairs(a, b);

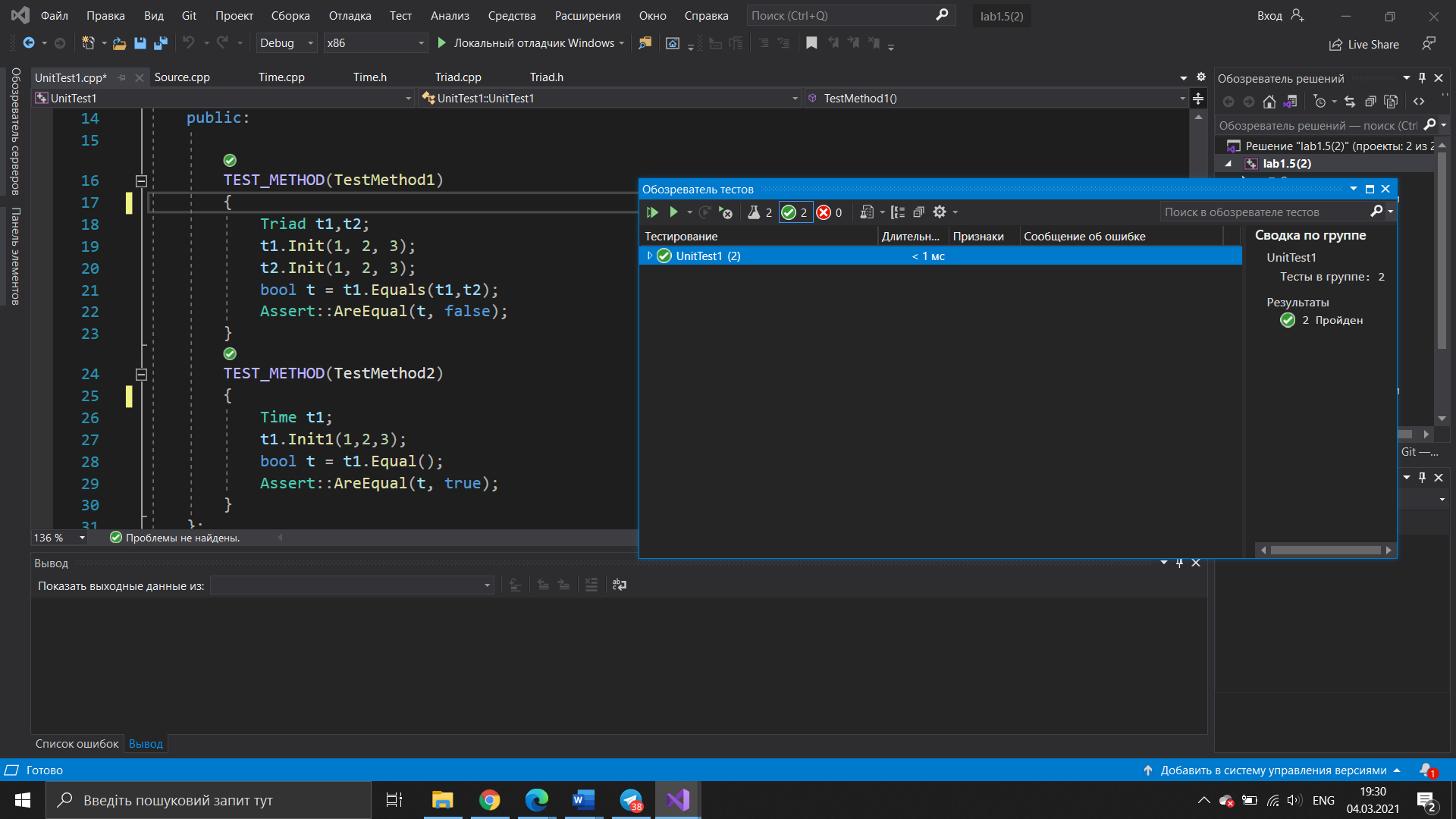
Assert::AreEqual(res1, 1.0);

Assert::AreEqual(res2, 1.0);

}

};

}



**Посилання на git-репозиторій з проектом:** [**https://github.com/Ysntska/1.5\_oop.git**](https://github.com/Ysntska/1.5_oop.git)

**Висновок: Я о**своїа використання композитних класів та об’єктів