# **Звіт про виконання лабораторної роботи № 1.6**

## «Вкладені класи» з дисципліни «Об’єктно-орієнтоване програмування»

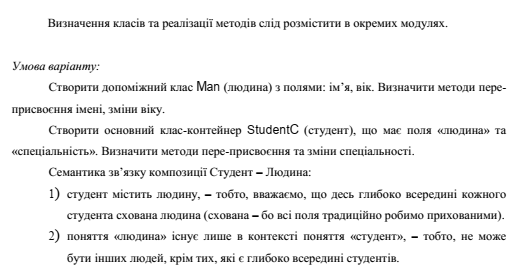
## Студента групи «**ІТ-12**» - **Степанчука Сергія**

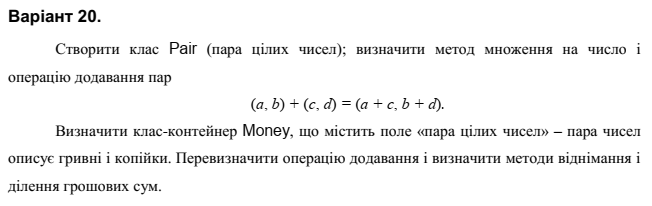
## Мета роботи

Освоїти використання вкладених та дружніх класів.

## Умова завдання



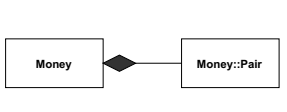




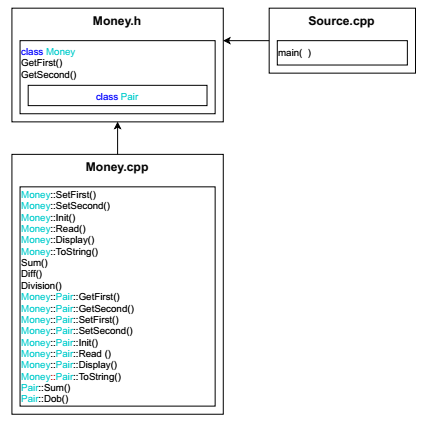
## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/OOP_Lab_1.6>

## UML-діаграма класів



## Структурна схема



## Текст програми

### // Money.h

#pragma once

#include <iostream>

#include <string>

#include <sstream>

using namespace std;

class Money

{

public:

class Pair

{

private:

int first;

int second;

public:

int GetFirst() const;

int GetSecond() const;

void SetFirst(int value);

void SetSecond(int value);

void Read();

void Init(int F, int S);

void Display() const;

string ToString() const;

friend Pair Sum(const Pair& p1, const Pair& p2);

friend Pair Dob(const Pair& p, int n);

};

int GetFirst() const { return pair.GetFirst(); }

int GetSecond() const { return pair.GetSecond(); }

void SetFirst(int value);

void SetSecond(int value);

void Init(int F, int S);

void Read();

void Display() const;

string ToString() const;

friend Money Sum(const Money& p1, const Money& p2);

friend Money Diff(const Money& p1, const Money& p2);

friend double Division(const Money& p1, const Money& p2);

private:

Pair pair;

};

### // Money.cpp

#include "Money.h"

#include <iostream>

#include <math.h>

using namespace std;

void Money::SetFirst(int value)

{

pair.SetFirst(value);

}

void Money::SetSecond(int value)

{

pair.SetSecond(value);

}

void Money::Init(int F, int S)

{

SetFirst(F);

SetSecond(S);

}

void Money::Read()

{

int x, y;

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

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

Init(x, y);

cout << endl;

}

void Money::Display() const

{

cout << GetFirst() << "." << GetSecond() << endl;

}

Money Sum(const Money& p1, const Money& p2)

{

int P\_F = p1.pair.GetFirst() + p2.pair.GetFirst();

int P\_S = p1.pair.GetSecond() + p2.pair.GetSecond();

if (P\_S > 99)

{

P\_S = P\_S - 100;

P\_F = P\_F + 1;

}

Money tmp;

tmp.Init(P\_F, P\_S);

return tmp;

}

Money Diff(const Money& p1, const Money& p2)

{

int P\_F = p1.pair.GetFirst() - p2.pair.GetFirst();

int P\_S = p1.pair.GetSecond() - p2.pair.GetSecond();

if (P\_F < 0)

{

if (P\_S > 0)

{

P\_S = abs(P\_S - 100);

P\_F = P\_F + 1;

}

else

P\_S = abs(P\_S);

}

if (P\_F >= 1 && P\_S < 0)

{

P\_S = P\_S + 100;

P\_F = P\_F - 1;

}

Money tmp;

tmp.Init(P\_F, P\_S);

return tmp;

}

double Division(const Money& p1, const Money& p2)

{

return (p1.GetFirst() + p1.GetSecond() / 100.) /

(p2.GetFirst() + p2.GetSecond() / 100.);

}

string Money::ToString() const

{

stringstream sstr;

sstr << GetFirst() << "." << GetSecond() << endl;

return sstr.str();

}

int Money::Pair::GetFirst() const

{

return first;

}

int Money::Pair::GetSecond() const

{

return second;

}

void Money::Pair::SetFirst(int value)

{

first = value;

};

void Money::Pair::SetSecond(int value)

{

second = value;

};

void Money::Pair::Init(int F, int S)

{

SetFirst(F);

SetSecond(S);

}

void Money::Pair::Read()

{

int x, y;

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

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

Init(x, y);

cout << endl;

}

void Money::Pair::Display() const

{

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

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

cout << "(" << first << ", " << second << ")";

cout << endl;

}

Money::Pair Sum(const Money::Pair& p1, const Money::Pair& p2)

{

Money::Pair tmp;

tmp.first = p1.first + p2.first;

tmp.second = p1.second + p2.second;

return tmp;

}

Money::Pair Dob(const Money::Pair& p, int n)

{

Money::Pair tmp;

tmp.first = p.first \* n;

tmp.second = p.second \* n;

return tmp;

}

string Money::Pair::ToString() const

{

stringstream sstr;

sstr << "(" << first << ", " << second << ")";

return sstr.str();

}

### // main.cpp

#include <iostream>

#include "Money.h"

using namespace std;

int main()

{

//////////////////////////пара//////////////////////////

Money::Pair p1, p2;

cout << "Input first pair" << endl;

p1.Read();

cout << "Input second pair" << endl;

p2.Read();

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

p1.Display();

cout << endl;

cout << "Second pair a is c, b is d" << endl;

p2.Display();

cout << endl;

//додавання пар

cout << "Sum of pairs:" << Sum(p1, p2).ToString() << endl << endl;

//множення на число

int number;

cout << "(number, you want to multiply on) n = ";

cin >> number; cout << endl;

cout << "First pair multyplied on n : " << Dob(p1, number).ToString() << endl;

cout << "Second pair multiplyed on n: " << Dob(p2, number).ToString() << endl;

cout << "--------------------------------------------" << endl << endl;

//////////////////////////гроші//////////////////////////

Money m1, m2;

cout << "Input money 1: " << endl;

m1.Read();

cout << "Input money 2: " << endl;

m2.Read();

cout << "Money 1: ";

m1.Display();

cout << "Money 2: ";

m2.Display();

cout << endl;

//додавання пар

cout << "Sum of money: " << Sum(m1, m2).ToString() << endl;

//віднімання пар

cout << "Difference of money: " << Diff(m1, m2).ToString() << endl;

//ділення пар

cout << "Division of money: " << Division(m1, m2) << endl;

return 0;

}

## UnitTest

### Код

#include "pch.h"

#include "CppUnitTest.h"

#include "../Program/Money.h"

#include "../Program/Money.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest15

{

TEST\_CLASS(UnitTest15)

{

public:

TEST\_METHOD(TestMethod1)

{

Money m1, m2;

m1.Init(6, 0);

m2.Init(2, 0);

double D = Division(m1, m2);

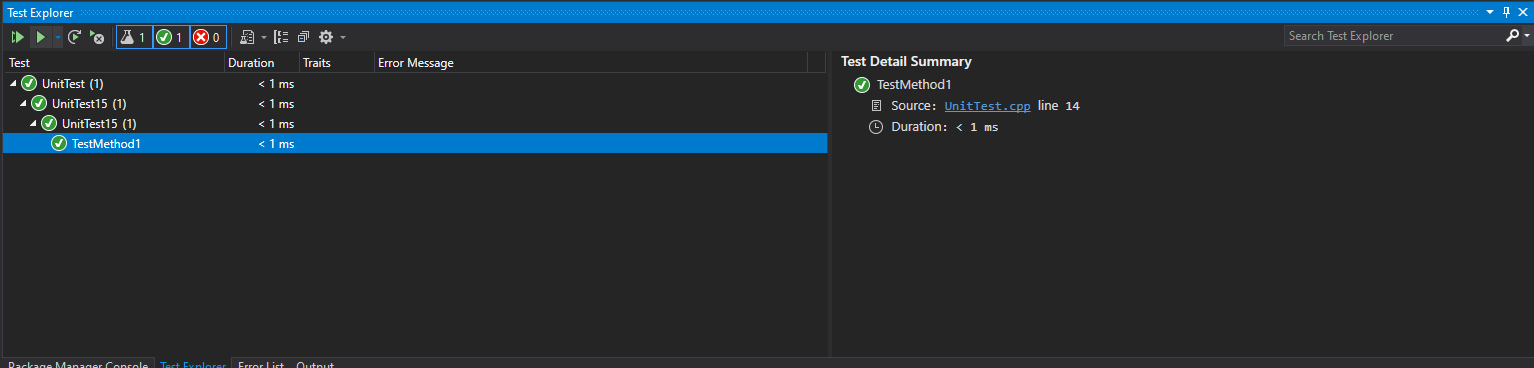
Assert::AreEqual(D, 3.);

}

};

}

### Результат



## Висновок

Навчився використання вкладених та дружніх класів.