

**ЗВІТ**

**Про виконання лабораторної роботи № 1.3**

**«Об’єкти – параметри методів (дії над кількома об’єктами)»**

**З дисципліни**

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

**Студентки групи ІТ-12**

**Шкринди Володимира Володимировича**

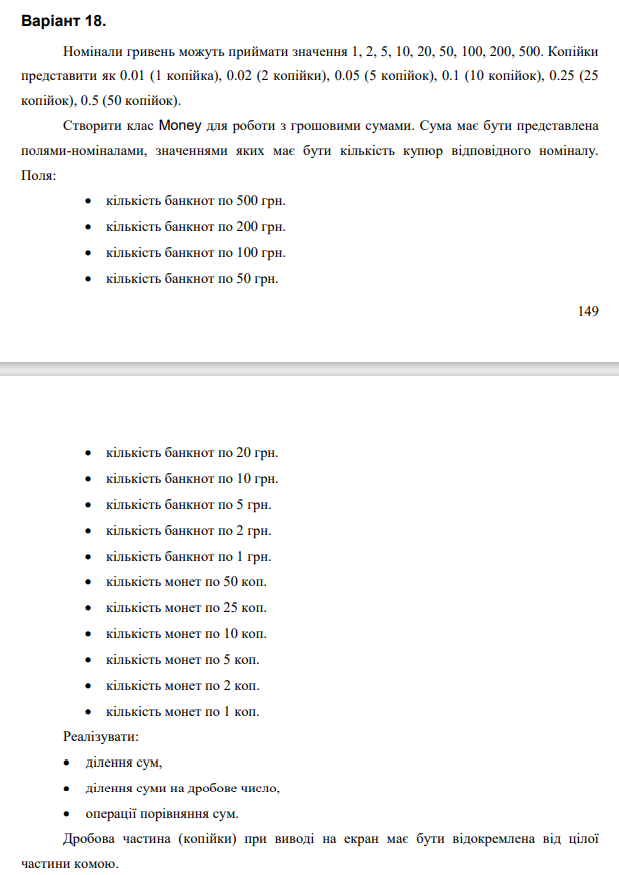
**Прийняв викладач**

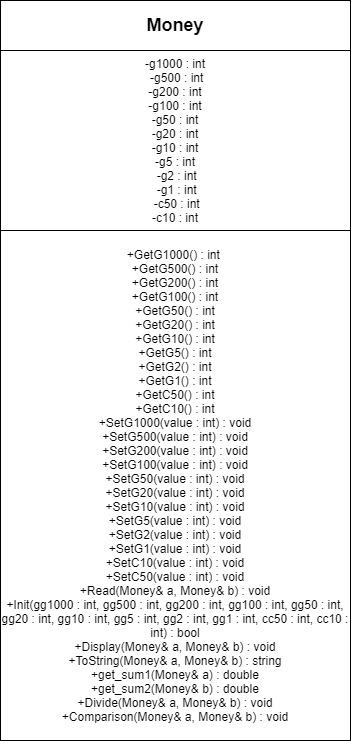
**Григорович В. Г.**

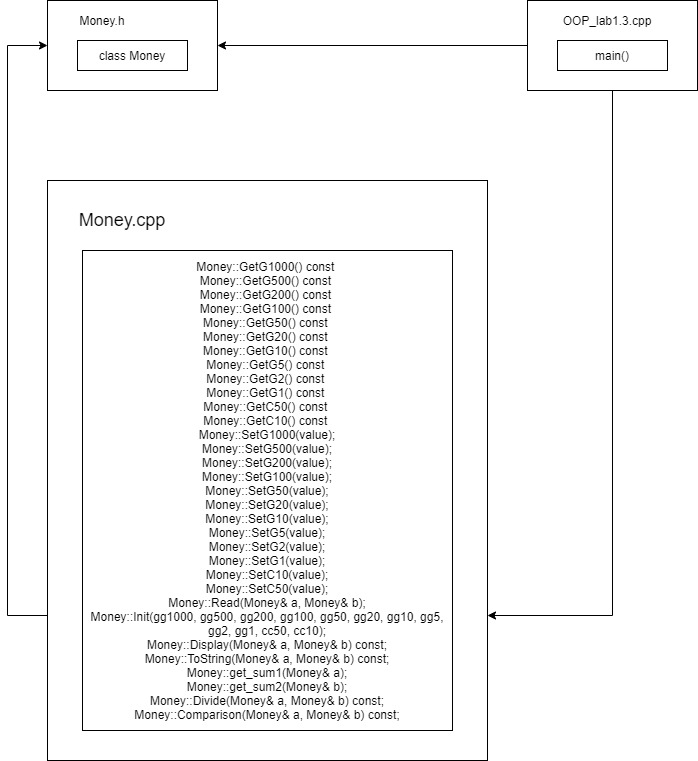
**2020**

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

**Умова завдання:**



**UML-діаграма класів:** 

**Структурна схема програми:** 

**Текст програми:**

**1.3.cpp**

#include <iostream>

#include <iomanip>

#include "Money.h"

using namespace std;

int main()

{

Money x, a, b;

x.Read(a, b);

x.Display(a, b);

cout << " SUMMA = " << x.ToString(a, b) << endl;

x.Divide(a, b);

x.Comparison(a, b);

return 0;

}  
**Money.h**

#pragma once

#include <string>

using namespace std;

class Money

{

private:

int g1000, g500, g200, g100, g50, g20, g10, g5, g2, g1, c50, c10;

public:

int GetG1000() const { return g1000; }

int GetG500() const { return g500; }

int GetG200() const { return g200; }

int GetG100() const { return g100; }

int GetG50() const { return g50; }

int GetG20() const { return g20; }

int GetG10() const { return g10; }

int GetG5() const { return g5; }

int GetG2() const { return g2; }

int GetG1() const { return g1; }

int GetC50() const { return c50; }

int GetC10() const { return c10; }

void SetG1000(int value);

void SetG500(int value);

void SetG200(int value);

void SetG100(int value);

void SetG50(int value);

void SetG20(int value);

void SetG10(int value);

void SetG5(int value);

void SetG2(int value);

void SetG1(int value);

void SetC10(int value);

void SetC50(int value);

void Read(Money& a, Money& b);

bool Init(int gg1000, int gg500, int gg200, int gg100, int gg50, int gg20, int gg10, int gg5, int gg2, int gg1, int cc50, int cc10);

void Display(Money& a, Money& b) const;

string ToString(Money& a, Money& b) const;

double get\_sum1(Money& a);

double get\_sum2(Money& b);

void Divide(Money& a, Money& b) const;

void Comparison(Money& a, Money& b) const;

};

**Money.cpp**

#include "Money.h"

#include <iostream>

#include <iomanip>

#include <sstream>

void Money::SetG1000(int value)

{

if (value > 0)

g1000 = value;

else

g1000 = 0;

}

void Money::SetG500(int value)

{

if (value > 0)

g500 = value;

else

g500 = 0;

}

void Money::SetG200(int value)

{

if (value > 0)

g200 = value;

else

g200 = 0;

}

void Money::SetG100(int value)

{

if (value > 0)

g100 = value;

else

g100 = 0;

}

void Money::SetG50(int value)

{

if (value > 0)

g50 = value;

else

g50 = 0;

}

void Money::SetG20(int value)

{

if (value > 0)

g20 = value;

else

g20 = 0;

}

void Money::SetG10(int value)

{

if (value > 0)

g10 = value;

else

g10 = 0;

}

void Money::SetG5(int value)

{

if (value > 0)

g5 = value;

else

g5 = 0;

}

void Money::SetG2(int value)

{

if (value > 0)

g2 = value;

else

g2 = 0;

}

void Money::SetG1(int value)

{

if (value > 0)

g1 = value;

else

g1 = 0;

}

void Money::SetC50(int value)

{

if (value > 0)

c50 = value;

else

c50 = 0;

}

void Money::SetC10(int value)

{

if (value > 0)

c10 = value;

else

c10 = 0;

}

bool Money::Init(int gg1000, int gg500, int gg200, int gg100, int gg50, int gg20, int gg10, int gg5, int gg2, int gg1, int cc50, int cc10)

{

if (gg1000 > 0)

{

SetG1000(gg1000);

return true;

}

if (gg500 > 0)

{

SetG500(gg500);

return true;

}

if (gg200 > 0)

{

SetG200(gg200);

return true;

}

if (gg100 > 0)

{

SetG100(gg100);

return true;

}

if (gg50 > 0)

{

SetG50(gg50);

return true;

}

if (gg20 > 0)

{

SetG20(gg20);

return true;

}

if (gg10 > 0)

{

SetG10(gg10);

return true;

}

if (gg5 > 0)

{

SetG5(gg5);

return true;

}

if (gg2 > 0)

{

SetG2(gg2);

return true;

}

if (gg1 > 0)

{

SetG1(gg1);

return true;

}

if (cc50 > 0)

{

SetC50(cc50);

return true;

}

if (cc10 > 0)

{

SetC10(cc10);

return true;

}

else

return false;

}

double Money::get\_sum1(Money& a) {

return (a.g1000 \* 1000. + a.g500 \* 500. + a.g200 \* 200. + a.g100 \* 100. + a.g50 \* 50. +

a.g20 \* 20. + a.g10 \* 10. + a.g5 \* 5. + a.g2 \* 2. + a.g1 \* 1. + a.c50 \* 0.50 + a.c10 \* 0.1);

}

double Money::get\_sum2(Money& b) {

return (b.g1000 \* 1000. + b.g500 \* 500. + b.g200 \* 200. + b.g100 \* 100. + b.g50 \* 50. +

b.g20 \* 20. + b.g10 \* 10. + b.g5 \* 5. + b.g2 \* 2. + b.g1 \* 1. + b.c50 \* 0.50 + b.c10 \* 0.1);

}

void Money::Read(Money& a, Money& b)

{

do {

cout << " -------- FIRST SUMM -------- " << endl;

cout << " 1000 - ? "; cin >> a.g1000;

cout << " 500 - ? "; cin >> a.g500;

cout << " 200 - ? "; cin >> a.g200;

cout << " 100 - ? "; cin >> a.g100;

cout << " 50 - ? "; cin >> a.g50;

cout << " 20 - ? "; cin >> a.g20;

cout << " 10 - ? "; cin >> a.g10;

cout << " 5 - ? "; cin >> a.g5;

cout << " 2 - ? "; cin >> a.g2;

cout << " 1 - ? "; cin >> a.g1;

cout << " 0.50 - ? "; cin >> a.c50;

cout << " 0.10 - ? "; cin >> a.c10;

cout << endl;

} while (!Init(a.g1000, a.g500, a.g200, a.g100, a.g50, a.g20, a.g10, a.g5, a.g2, a.g1, a.c50, a.c10));

do {

cout << " -------- SECOND SUMM -------- " << endl;

cout << " 1000 - ? "; cin >> b.g1000;

cout << " 500 - ? "; cin >> b.g500;

cout << " 200 - ? "; cin >> b.g200;

cout << " 100 - ? "; cin >> b.g100;

cout << " 50 - ? "; cin >> b.g50;

cout << " 20 - ? "; cin >> b.g20;

cout << " 10 - ? "; cin >> b.g10;

cout << " 5 - ? "; cin >> b.g5;

cout << " 2 - ? "; cin >> b.g2;

cout << " 1 - ? "; cin >> b.g1;

cout << " 0.50 - ? "; cin >> b.c50;

cout << " 0.10 - ? "; cin >> b.c10;

cout << endl << endl;

} while (!Init(b.g1000, b.g500, b.g200, b.g100, b.g50, b.g20, b.g10, b.g5, b.g2, b.g1, b.c50, b.c10));

}

void Money::Display(Money& a, Money& b) const

{

Money x;

cout << " -------- FIRST SUMM -------- " << endl;

cout << " 1000 - " << a.g1000 \* 1000. << endl;

cout << " 500 - " << a.g500 \* 500. << endl;

cout << " 200 - " << a.g200 \* 200. << endl;

cout << " 100 - " << a.g100 \* 100. << endl;

cout << " 50 - " << a.g50 \* 50. << endl;

cout << " 20 - " << a.g20 \* 20. << endl;

cout << " 10 - " << a.g10 \* 10. << endl;

cout << " 5 - " << a.g5 \* 5. << endl;

cout << " 2 - " << a.g2 \* 2. << endl;

cout << " 1 - " << a.g1 \* 1. << endl;

cout << " 0.50 - " << a.c50 \* 0.5 << endl;

cout << " 0.10 - " << a.c10 \* 0.1 << endl;

cout << " TOTAL - " << x.get\_sum1(a) << endl;

cout << endl;

cout << " -------- SECOND SUMM -------- " << endl;

cout << " 1000 - " << b.g1000 \* 1000. << endl;

cout << " 500 - " << b.g500 \* 500. << endl;

cout << " 200 - " << b.g200 \* 200. << endl;

cout << " 100 - " << b.g100 \* 100. << endl;

cout << " 50 - " << b.g50 \* 50. << endl;

cout << " 20 - " << b.g20 \* 20. << endl;

cout << " 10 - " << b.g10 \* 10. << endl;

cout << " 5 - " << b.g5 \* 5. << endl;

cout << " 2 - " << b.g2 \* 2. << endl;

cout << " 1 - " << b.g1 \* 1. << endl;

cout << " 0.50 - " << b.c50 \* 0.5 << endl;

cout << " 0.10 - " << b.c10 \* 0.1 << endl;

cout << " TOTAL - " << x.get\_sum2(b) << endl;

cout << endl << endl;

}

void Money::Divide(Money& a, Money& b) const

{

Money x;

if (x.get\_sum1(a) > x.get\_sum2(b))

{

cout << " DIVIDE = " << setprecision(2) << x.get\_sum1(a) / x.get\_sum2(b) << endl;

}

if (x.get\_sum1(a) < x.get\_sum2(b))

{

cout << " DIVIDE = " << setprecision(2) << x.get\_sum2(a) / x.get\_sum1(b) << endl;

}

}

void Money::Comparison(Money& a, Money& b) const

{

Money x;

if (x.get\_sum1(a) > x.get\_sum2(b)) { cout << " SUMM\_1 > SUMM\_2 " << endl; }

if (x.get\_sum1(a) < x.get\_sum2(b)) { cout << " SUMM\_1 < SUMM\_2 " << endl; }

if (x.get\_sum1(a) == x.get\_sum2(b)) { cout << " SUMM\_1 = SUMM\_2 " << endl; }

}

string Money::ToString(Money& a, Money& b) const

{

Money x;

double s = x.get\_sum1(a) + x.get\_sum2(b);

stringstream sstr;

sstr << s << " UAH.";

return sstr.str();

}

**Результати Unit тесту:**

#include "pch.h"

#include "CppUnitTest.h"

#include "C:\Users\admin\source\repos\OOP\_lab1.3\Money.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace OOPunit13

{

TEST\_CLASS(OOPunit13)

{

public:

TEST\_METHOD(TestMethod1)

{

Money money;

bool t;

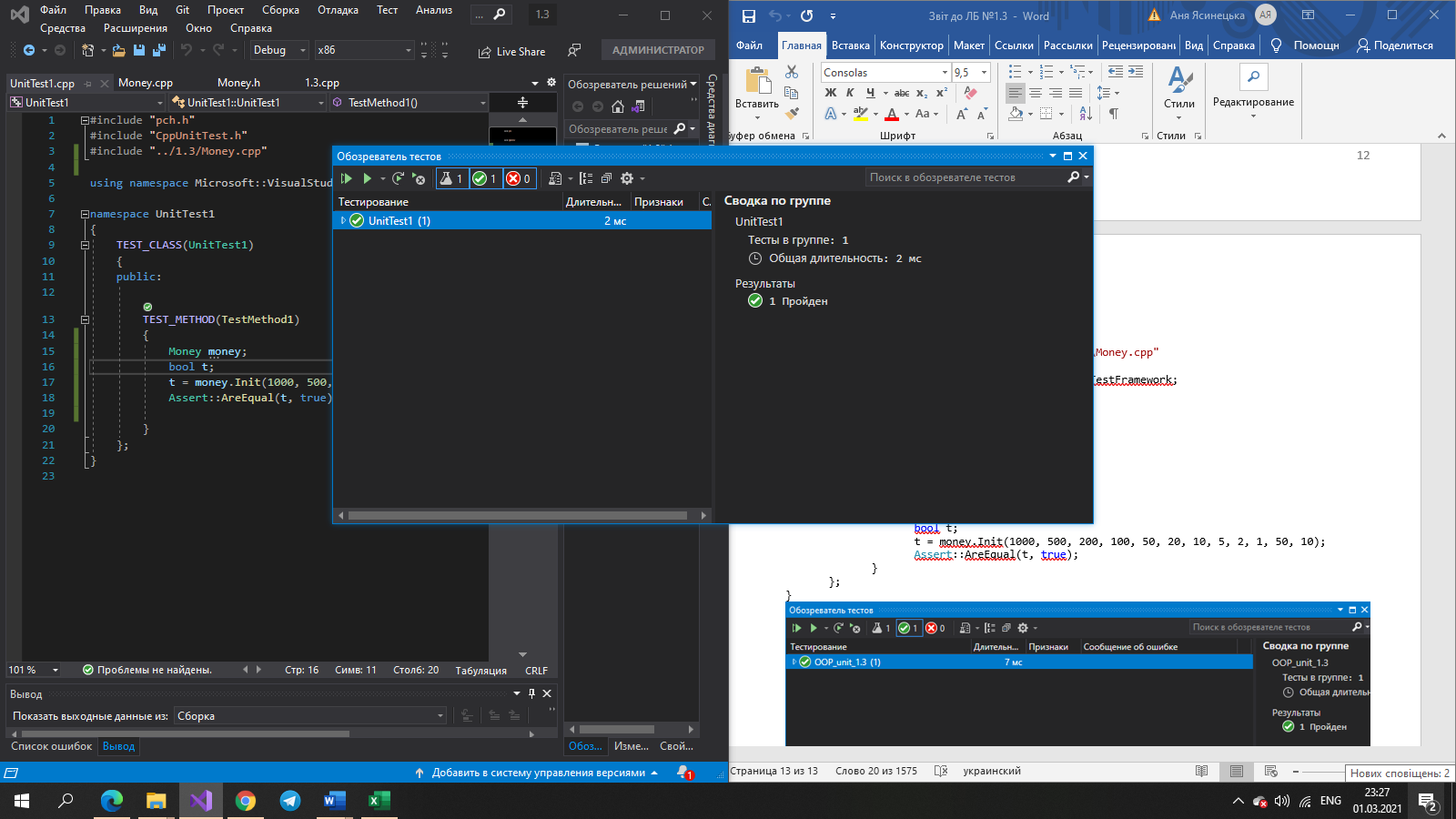
t = money.Init(1000, 500, 200, 100, 50, 20, 10, 5, 2, 1, 50, 10);

Assert::AreEqual(t, true);

}

};

}



**Посилання на git-репозиторій з проектом:** [**https://github.com/Ysntska/1.3\_oop.git**](https://github.com/Ysntska/1.3_oop.git)

**Висновок:** У ході лабораторної роботи я навчвся використовувати класи та об’єкти.