# **Звіт про виконання лабораторної роботи № 3.6**

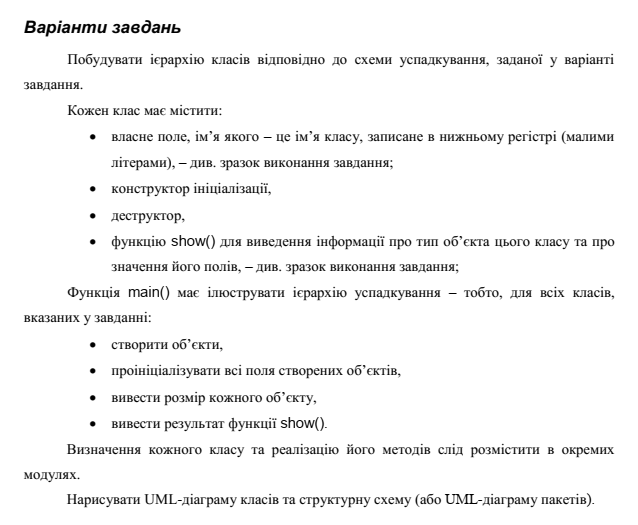
## «Множинне успадкування» з дисципліни «Об’єктно-орієнтоване програмування»

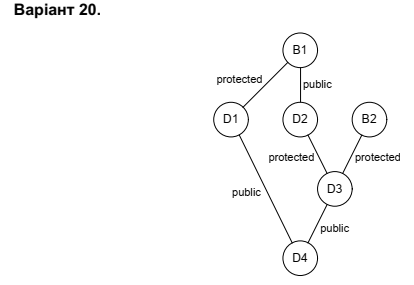
## Студента групи «**ІТ-12**» - **Степанчука Сергія**

## Мета роботи

Освоїти використання множинного успадкування.

## Умова завдання

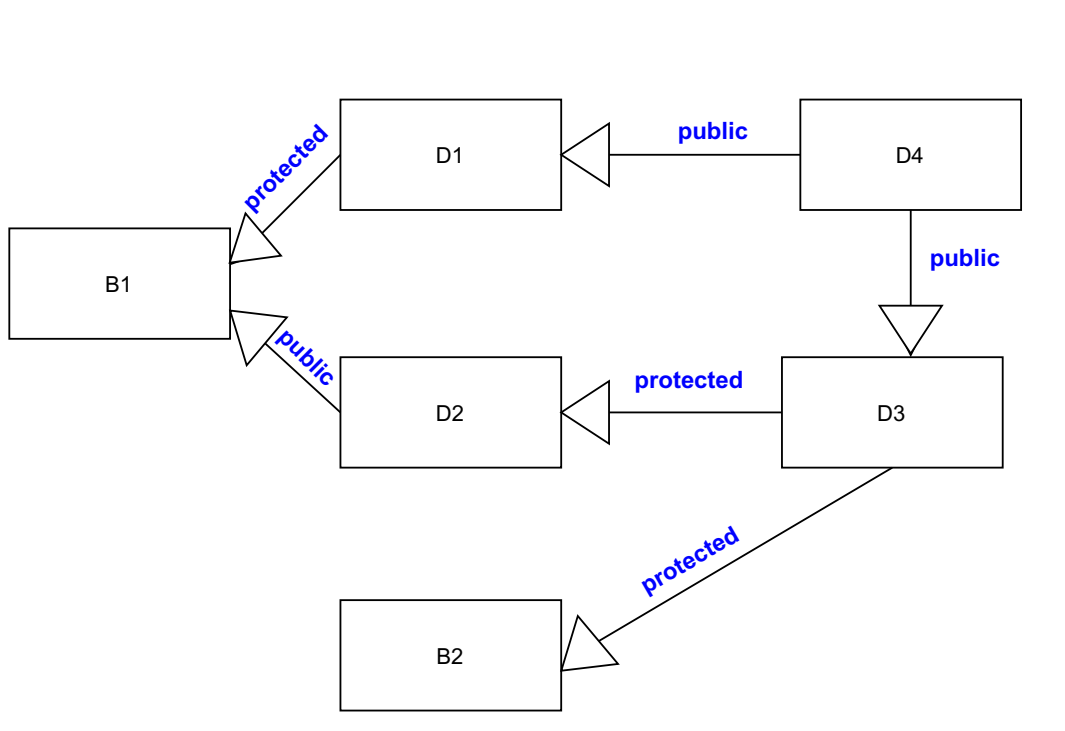




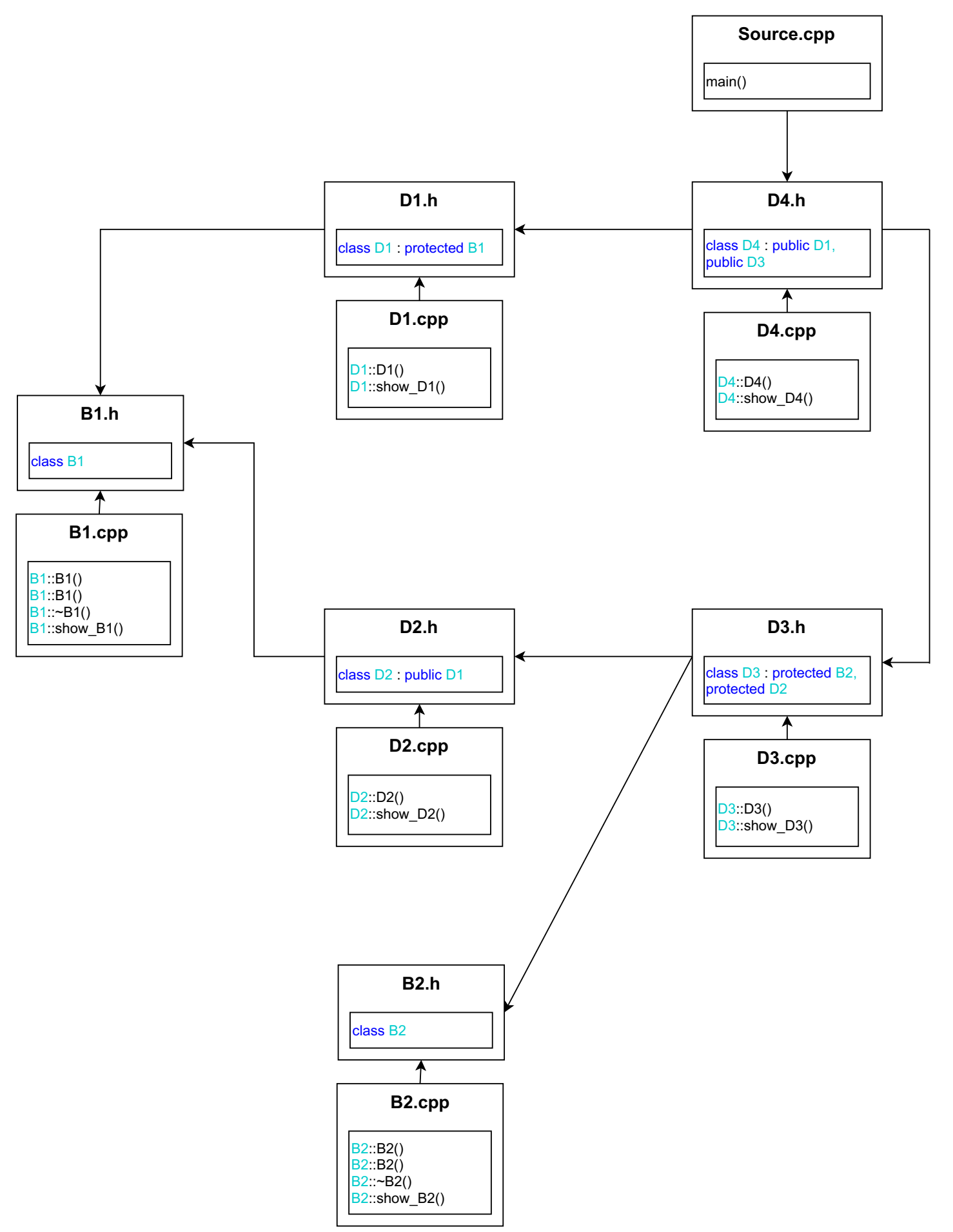
## Посилання на git-репозиторій з проектом:

<https://github.com/SergiyStepanchuk/OOP_Lab_3.6>

## UML-діаграма класів



## Структурна схема



## Текст програми

//B1.h

#pragma once

#include <iostream>

#include <Windows.h>

using namespace std;

class B1

{

private:

int b1;

public:

B1();

B1(int b1);

~B1();

void show\_B1();

};

//B1.cpp

#include "B1.h"

B1::B1()

: b1(0)

{ }

B1::B1(int x)

{

b1 = x;

}

B1::~B1()

{}

void B1::show\_B1()

{

cout << "class B1:" << endl;

cout << "show\_B1()" << endl

<< "B1::b1 = " << b1 << endl << endl;

}

//B2.h

#pragma once

#include <iostream>

#include <Windows.h>

using namespace std;

class B2

{

private:

int b2;

public:

B2();

B2(int b1);

~B2();

void show\_B2();

};

//B2.cpp

#include "B2.h"

B2::B2()

: b2(0)

{ }

B2::B2(int x)

{

b2 = x;

}

B2::~B2()

{}

void B2::show\_B2()

{

cout << "class B2:" << endl;

cout << "show\_B2()" << endl

<< "B2::b2 = " << b2 << endl << endl;

}

//D1.h

#pragma once

#include "B1.h"

#include <iostream>

#include <Windows.h>

using namespace std;

class D1 : protected B1

{

private:

int d1;

public:

D1(int d1, int b1);

void show\_D1();

};

//D1.cpp

#include "D1.h"

D1::D1(int x, int i)

: B1(i)

{

d1 = x;

}

void D1::show\_D1()

{

cout << "class D1:" << endl;

show\_B1();

cout << "show\_D1()" << endl

<< "D1::d1 = " << d1 << endl << endl;

}

//D2.h

#pragma once

#include "B1.h"

#include <iostream>

#include <Windows.h>

using namespace std;

class D2 : public B1

{

private:

int d2;

public:

D2(int d2, int b1);

void show\_D2();

};

//D2.cpp

#include "D2.h"

D2::D2(int x, int i)

: B1(i)

{

d2 = x;

}

void D2::show\_D2()

{

cout << "class D2:" << endl;

show\_B1();

cout << "show\_D2()" << endl

<< "D2::d2 = " << d2 << endl << endl;

}

//D3.h

#pragma once

#include "B2.h"

#include "D2.h"

class D3 : protected B2, protected D2

{

private:

int d3;

public:

D3(int d3, int b2, int d2, int b1);

void show\_D3();

};

//D3.cpp

#include "D3.h"

D3::D3(int x, int y, int z, int m)

: B2(y), D2(z, m)

{

d3 = x;

}

void D3::show\_D3()

{

cout << "class D3:" << endl;

show\_D2();

show\_B2();

cout << "show\_D3()" << endl

<< "D3::d3 = " << d3 << endl << endl;

}

//D4.h

#pragma once

#include "D1.h"

#include "D3.h"

class D4 : public D1, public D3

{

private:

int d4;

public:

D4(int d1, int d2, int d3, int d4, int b1, int b2);

void show\_D4();

};

// D4.cpp

#include "D4.h"

D4::D4(int x, int y, int z, int m, int i, int j)

: D1(i, x), D3(j, y, z, x)

{

d4 = m;

}

void D4::show\_D4()

{

cout << "class D4:" << endl;

show\_D1();

show\_D3();

cout << "show\_D4()" << endl

<< "D4::d4 = " << d4 << endl << endl;

}

//Source.cpp

#include "D4.h"

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

B1 o1(777);

cout << "B1 o1(777);" << endl;

cout << "sizeof(B1) = " << sizeof(B1) << endl;

cout << endl << "Ієрархія класу B1: " << endl;

o1.show\_B1();

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

B2 o2(888);

cout << "B2 o2(888);" << endl;

cout << "sizeof(B2) = " << sizeof(B2) << endl;

cout << endl << "Ієрархія класу B2: " << endl;

o2.show\_B2();

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

D1 n1(1000, 2000);

cout << "D1 n1(1000, 2000);" << endl;

cout << "sizeof(D1) = " << sizeof(D1) << endl;

cout << endl << "Ієрархія класу D1: " << endl;

n1.show\_D1();

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

D2 n2(10, 20);

cout << "D2 n2(10, 20);" << endl;

cout << "sizeof(D2) = " << sizeof(D2) << endl;

cout << endl << "Ієрархія класу D2: " << endl;

n2.show\_D2();

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

D3 n3(100, 200, 300, 400);

cout << "D3 n3(100, 200, 300, 400);" << endl;

cout << "sizeof(D3) = " << sizeof(D3) << endl;

cout << endl << "Ієрархія класу D3: " << endl;

n3.show\_D3();

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

D4 n4(1, 2, 3, 4, 5, 6);

cout << "D4 n4(1, 2, 3, 4, 5, 6);" << endl;

cout << "sizeof(D4) = " << sizeof(D4) << endl;

cout << endl << "Ієрархія класу D4: " << endl;

n4.show\_D4();

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

}

## UnitTest

### Код

#include "pch.h"

#include "CppUnitTest.h"

#include "../Lab\_3.3-G/Bill.h"

#include "../Lab\_3.3-G/Bill.cpp"

#include "../Lab\_3.3-G/Time.h"

#include "../Lab\_3.3-G/Time.cpp"

#include "../Lab\_3.3-G/Object.h"

#include "../Lab\_3.3-G/Object.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTest33G

{

TEST\_CLASS(UnitTest33G)

{

public:

TEST\_METHOD(TestMethod1)

{

Bill B1;

B1.SetSurname("Stepanchuk");

string surname = "Stepanchuk ";

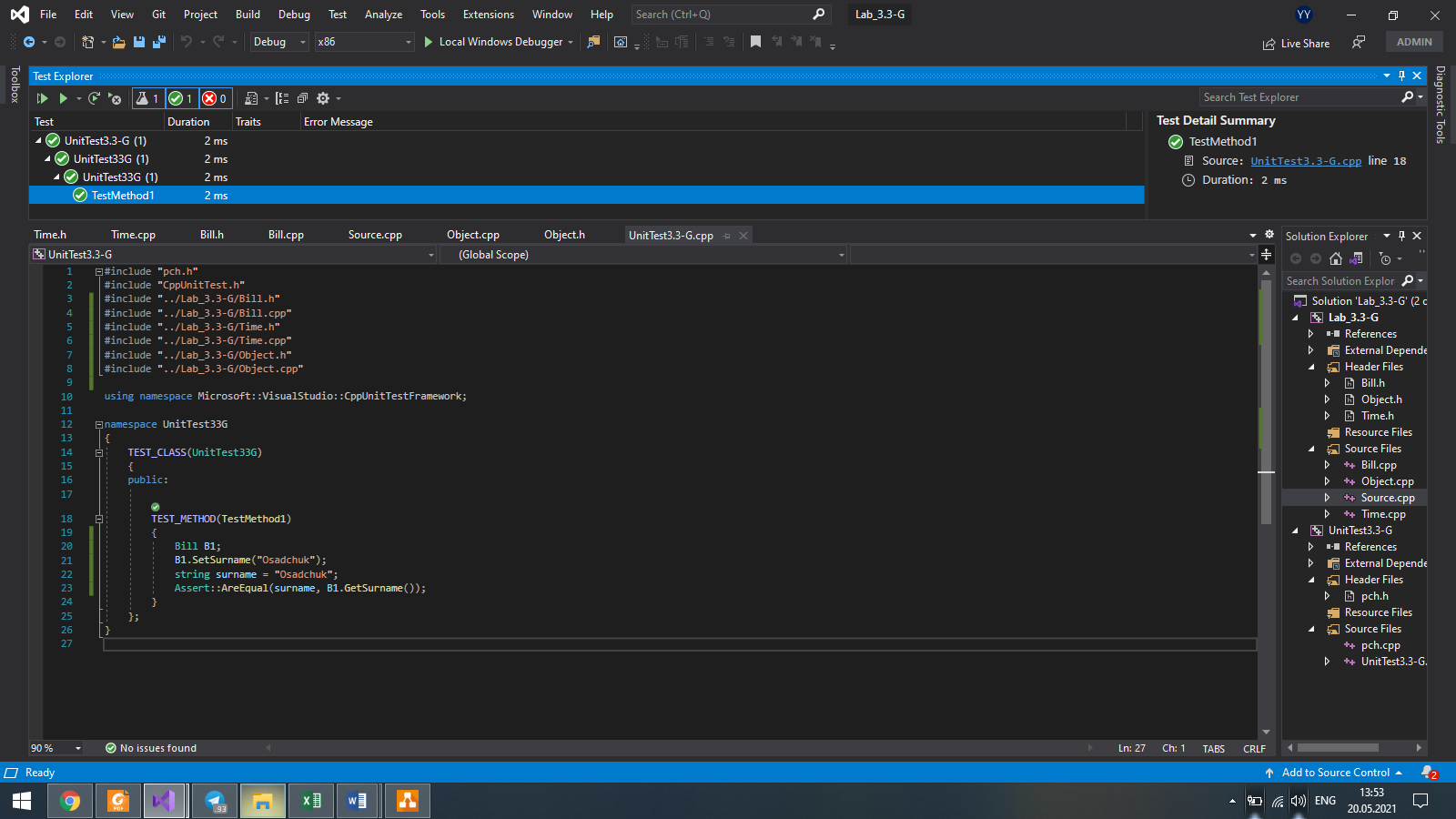
Assert::AreEqual(surname, B1.GetSurname());

}

};

}

### Результат



## Висновок

Освоїв використання множинного успадкування.