 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Практичне заняття №4**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему: «Перевантаження операцій»

Варіант №18

**Виконав:**

студент гр. БС-51

Солярик А.С.

**Перевірив:**

ст. викладач каф. БМК

Сердаковський В.С.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

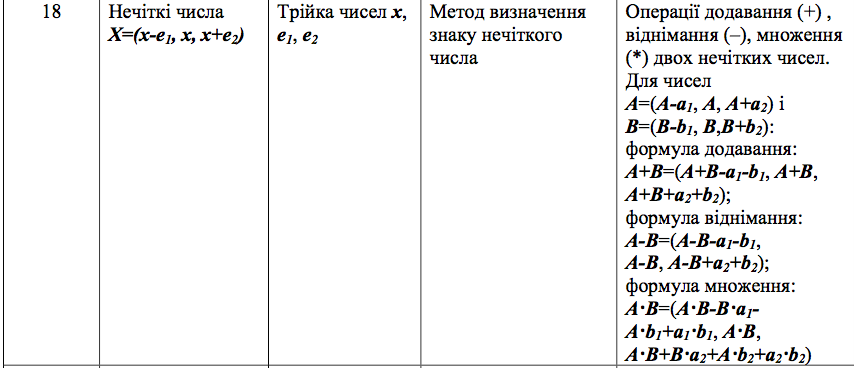
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2017

**Завдання:**

1. Вивчити теоретичні основи визначення класів, що використовують спадкування властивостей, та побудови ієрархії класів в мові С++.
2. Розробити діаграму класів в нотації UML та програмний додаток з використанням успадкування, що реалізує ієрархію класів відповідно до свого варіанту (визначення класів та їх реалізації мають бути розташовані у файлах \*.h та \*.cpp відповідно; для ініціалізаціїї даних класів використовувати введення з клавіатури):



1. Скласти і захистити звіт.

**🞏 Практичне заняття без зауважень**

**🞏 Практичне заняття має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до UML діаграми:**

**🞏 діаграма класу не відповідає коду**

**🞏 виконані не за стандартом:**

**🞏 атрибути**

**🞏 відношення**

**🞏 багатостатність**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 хибно задані специфікатори доступу**

**🞏 помилки у визначенні конструкторів / деструкторів**

**🞏 використано глобальні змінні**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Блок-схема**

****

**Лістинг програми**

**weirdNumbers.hpp**

//

//  weirdNumbers.hpp

//  lab4

//

//  Created by Anton Solyarik on 11.04.17.

//  Copyright © 2017 Anton Solyarik. All rights reserved.

//

#ifndef weirdNumbers\_hpp

#define weirdNumbers\_hpp

using namespace std;

#include <ostream>

#include <stdio.h>

class weirdNumbers

{

public:

    weirdNumbers(double x, double e1, double e2)

    {

        this->x = x;

        this->e1 = e1;

        this->e2 = e2;

    }

    weirdNumbers()

    {

    }

    ~weirdNumbers()

    {

    }

    const bool isLowerthanzero();

    friend weirdNumbers operator-(weirdNumbers& left, weirdNumbers& right);

    friend weirdNumbers operator\*(weirdNumbers& left, weirdNumbers& right);

    friend ostream& operator <<(std::ostream& os,  weirdNumbers& num);

    friend weirdNumbers operator+(weirdNumbers& left, weirdNumbers& right);

private:

    double x;

    double e1;

    double e2;

};

#endif /\* weirdNumbers\_hpp \*/

**weirdNumbers.cpp**

//

//  weirdNumbers.cpp

//  lab4

//

//  Created by Anton Solyarik on 11.04.17.

//  Copyright © 2017 Anton Solyarik. All rights reserved.

//

#include "weirdNumbers.hpp"

const bool weirdNumbers::isLowerthanzero()

{

    return x <= 0;

}

**main.cpp**

//

//  main.cpp

//  lab4

//

//  Created by Anton Solyarik on 11.04.17.

//  Copyright © 2017 Anton Solyarik. All rights reserved.

//

#include <iostream>

#include "weirdnumbers.hpp"

std::ostream& operator <<(std::ostream& os, char value) {

    return os << (value ? "True" : "False");

}

std::ostream& operator <<(std::ostream& os,  weirdNumbers& num) {

    return os <<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<std::endl<< "x: " << num.x << " e1: " << num.e1 << " e2: " << num.e2 << std::endl << "X=("<<num.x-num.e1<<","<<num.x<<","<<num.x+num.e2<<")"<<std::endl<<"IS IT LOWER THAN 0?: "<<num.isLowerthanzero()<<std::endl;

}

weirdNumbers operator+(weirdNumbers& left, weirdNumbers& right)

{

    weirdNumbers result;

    result.x = left.x+right.x;

    result.e1 = (left.e1)+(right.e1);

    result.e2 = (left.e2)+(right.e2);

    return result;

}

weirdNumbers operator -(weirdNumbers& left, weirdNumbers& right)

{

    weirdNumbers result;

    result.x = left.x-right.x;

    result.e1 = (left.e1)+(right.e1);

    result.e2 = (left.e2)+(right.e2);

    return result;

}

weirdNumbers operator \*(weirdNumbers& left, weirdNumbers& right)

{

    weirdNumbers result;

    result.x = left.x\*right.x;

    result.e1 = (left.e1 \* right.x)-(right.e1 \* left.x) + (left.e1 \* right.e1);

    result.e2 = (left.e1 \* right.x)+(right.e1 \* left.x) + (left.e1 \* right.e1);

    return result;

}

int main(int argc, const char \* argv[])

{

    weirdNumbers w1(5.2,1.666,2.13);

    weirdNumbers w2(6.34,3.56,4.88);

    weirdNumbers w3 = w1 + w2;

    std::cout << w3;

    w3 = w1 - w2;

    std::cout<<w3;

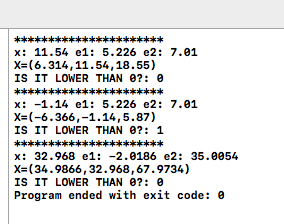
    w3 = w1\*w2;

    std::cout<<w3;

    return 0;

}

**Результат програми**



**Контрольні запитання**