Міністерство освіти і науки України

Національний технічний університет України „КПІ”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки

інформації та управління

**ЗВІТ**

з комп’ютерного практикуму № 2

на тему :

„ НАСЛІДУВАННЯ с++ ”

Варіант №19

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виконав студент** |  | Цицилюк Анна Валеріївна |  |  |
|  |  | (№ групи, прізвище, ім’я, по батькові ) |  |  |
|  |  |  |  |  |
| **Прийняв** |  |  |  |  |
|  |  | (посада, прізвище, ім’я, по батькові ) |  |  |

Київ 2017

Зміст

[1 ЦІЛЬ РОБОТИ 3](#_Toc497850691)

[2 ПОСТАНОВКА ЗАДАЧІ 4](#_Toc497850692)

[3 UML ДІАГРАМА 5](#_Toc497850693)

[4 ПРОГРАМНИЙ КОД 6](#_Toc497850694)

[Main.cpp 6](#_Toc497850695)

[Intercept.cpp 6](#_Toc497850696)

[Intercept.hpp 6](#_Toc497850697)

[TriangleA.cpp 6](#_Toc497850698)

[TriangleA.hpp 7](#_Toc497850699)

[TriangleB.cpp 7](#_Toc497850700)

[TriangleB.hpp 7](#_Toc497850701)

[CroppedPiramid.cpp 8](#_Toc497850702)

[CroppedPiramid.hpp 8](#_Toc497850703)

[5 РЕЗУЛЬТАТ РОБОТИ 10](#_Toc497850704)

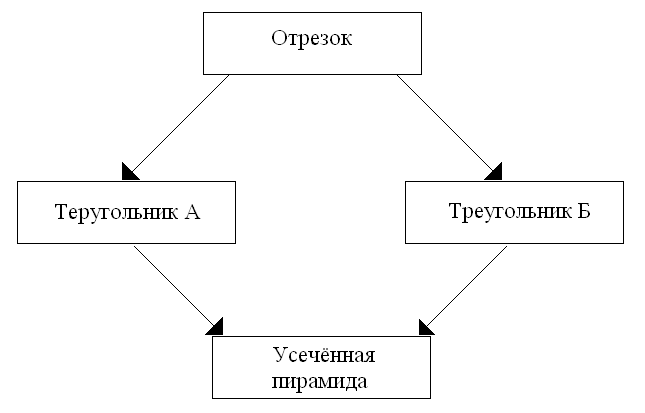
[6 ВИСНОВОК 11](#_Toc497850705)

# ЦІЛЬ РОБОТИ

Мета роботи - вивчити основні концепції об'єктно-орієнтованого програмування.Вивчити особливості успадкування і множинного наслідування.

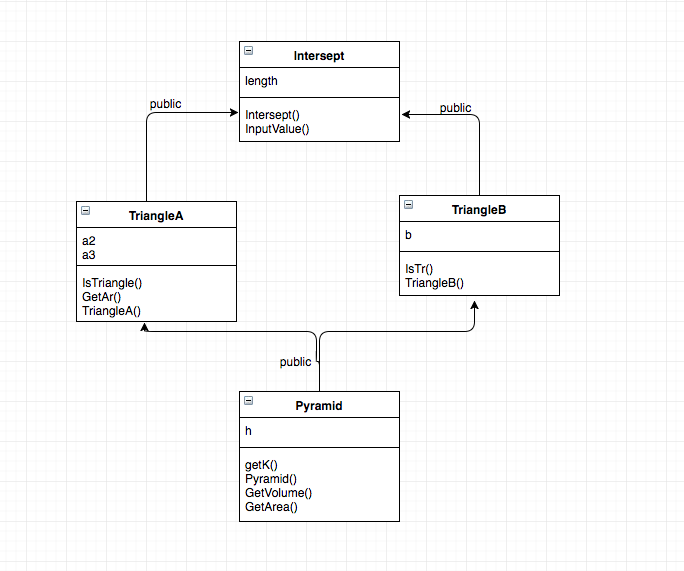
# ПОСТАНОВКА ЗАДАЧІ

Визнасити ієрархію класів



Визначити площу кожної фігури(для зрізаної піраміди бокову площу і об'єм).Ініціалізація даних проводиться користувачем.

# UML ДІАГРАМА



# ПРОГРАМНИЙ КОД

## Main.cpp

#include <iostream>

#include "CroppedPyramid.hpp"

using namespace std;

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

    cout<<"Цицилюк Анна"<<endl<<"Варіант 19 рівень В"<<endl<<"Спроектувати діаграму класів-відрізок-трикутники А і В та зрізана піпаміда"<<endl;

    int yes=1;

    do

    { Pyramid MyPir;

    if(MyPir.IsTiangle())

    {cout<<MyPir.GetArea()<<endl;

   cout<< MyPir.GetVolume()<<endl;

    }

    else cout<<"Введеного трикутника основи не існує"<<endl;

        cout<<"введіть 0 для виходу";cin>>yes;

    }

    while (yes);

    return 0;

}

## Intercept.cpp

#include "Intercept.hpp"

Intercept::Intercept()

{

    do InputValue();

    while (length<0);

}

void Intercept::InputValue(){

    std::cout<<"Введіть довжину відрізка основи "<<std::endl;

    std::cin>>length;

}

## Intercept.hpp

#ifndef Intercept\_hpp

#define Intercept\_hpp

#include <stdio.h>

#include <iostream>

class Intercept

{

protected:

    float length;

    public :

    Intercept();

    void InputValue();

};

#endif /\* Intercept\_hpp \*/

## TriangleA.cpp

#include "TriangleA.hpp"

#include "math.h"

void TriangleA::InputValue()

{

    do{

    std::cout<<"Введіть значення сторін 1 трикутника"<<std::endl;

        std::cin>>a2>>a3;}

    while (a2<0||a3<0);

}

TriangleA::TriangleA()

{InputValue();};

bool TriangleA:: IsTiangle()

{

    if(length+a2>a3&&a2+a3>length&&length+a3>a2)

        return 1;

    else return 0;

}

float TriangleA::GetAr()

{

    float p=(length+a2+a3)/2;

    float s=sqrt(p\*(p-length)\*(p-a2)\*(p-a3));

    return s;

}

## TriangleA.hpp

#ifndef TriangleA\_hpp

#define TriangleA\_hpp

#include "Intercept.hpp"

#include <stdio.h>

//#include <iostream>

class TriangleA:virtual public Intercept

{

protected:

    float a2,a3;

public:

    bool IsTiangle();

    TriangleA();

    void InputValue();

     float GetAr();

};

#endif /\* TriangleA\_hpp \*/

## TriangleB.cpp

#include "TriangleB.hpp"

TriangleB::TriangleB(){};

void TriangleB::InputValue()

{

    do

    {std::cout<<"Введіть значення сторін 2 трикутника"<<std::endl;

        std::cin>>b;}

    while (b<0);

}

bool TriangleB::IsTr()

{

    if(b+b>length)

        return 1;

    else return 0;

}

## TriangleB.hpp

#ifndef TriangleB\_hpp

#define TriangleB\_hpp

#include "Intercept.hpp"

#include <stdio.h>

class TriangleB:virtual public Intercept

{

protected:

    float b;

public:

    TriangleB();

    void InputValue();

    bool IsTr();

 //s    float GetVolume();

};

#endif /\* TriangleB\_hpp \*/

## CroppedPiramid.cpp

#include "CroppedPyramid.hpp"

#include "math.h"

using namespace std;

    Pyramid::Pyramid():TriangleA(),TriangleB(){

        do

        {std::cout<<"Введіть значення сторін 2 трикутника"<<std::endl;

            std::cin>>b;}

        while (b<0);

        do{cout<<"H"<<endl;

            cin>>h;}

        while(h<0);

}

float Pyramid::getK(){

    float k;

    float l=length/2;

    float H=sqrt(b\*b-l\*l);

    k=H/h;

    return k;

}

float Pyramid::GetArea(){

    float k=getK();

    float S;

    float H=sqrt(pow((k-1),2)+pow(h,2));

      //      cout<<a1<<endl<<a2<<endl<<a3<<endl;

    S=((k+1)/2)\*H\*(length+a2+a3);

    return S;

}

float Pyramid::GetVolume()

{

    float k=getK();

    float S1=GetAr();

   //cout<<"S1 "<<S1<<endl;

    float S2=S1\*k\*k;

  //  cout<<"S2"<<S2<<endl;

    float V=(h/3)\*(S1+S2+sqrt(S1\*S2));

    return V;

}

## CroppedPiramid.hpp

#ifndef CroppedPyramid\_hpp

#define CroppedPyramid\_hpp

#include "TriangleA.hpp"

#include "TriangleB.hpp"

#include <stdio.h>

class Pyramid:public TriangleA, TriangleB

{

private:

    float getK();

    float h;

    float k;

public:

    Pyramid();

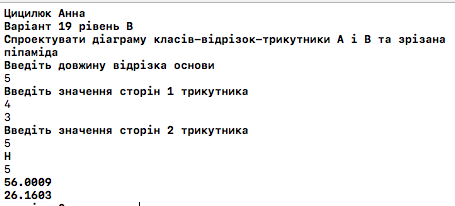
   float GetArea();

   float GetVolume();

};

#endif /\* CroppedPyramid\_hpp \*/

# РЕЗУЛЬТАТ РОБОТИ



# ВИСНОВОК

Я вивчила основні концепції об'єктно-орієнтованого програмування.Також засвоїла знання про особливості успадкування і множинного наслідування.